PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

SEDE AMBATO

TECNOLOGIA EN INFORMATICA

TRABAJO PRACTICO DE GRADO

TEMA:

SISTEMA INTEGRADO DE ROL DE PAGOS DE LA EMPRESA ELECTRICA AMBATO S.A. REGIONAL CENTRO NORTE

AUTORES:

HUGO ORLANDO BARRERA BARRAGAN

FREDDY FRANCISCO VILLEGAS MORA

MERCEDES ISABEL YANEZ BAYAS

DIRECTOR:

TLGO. VICTOR CHUNCHA

ASESOR:

LCDO. RAUL PARRA

TOMO II

AMBATO - ECUADOR

1995

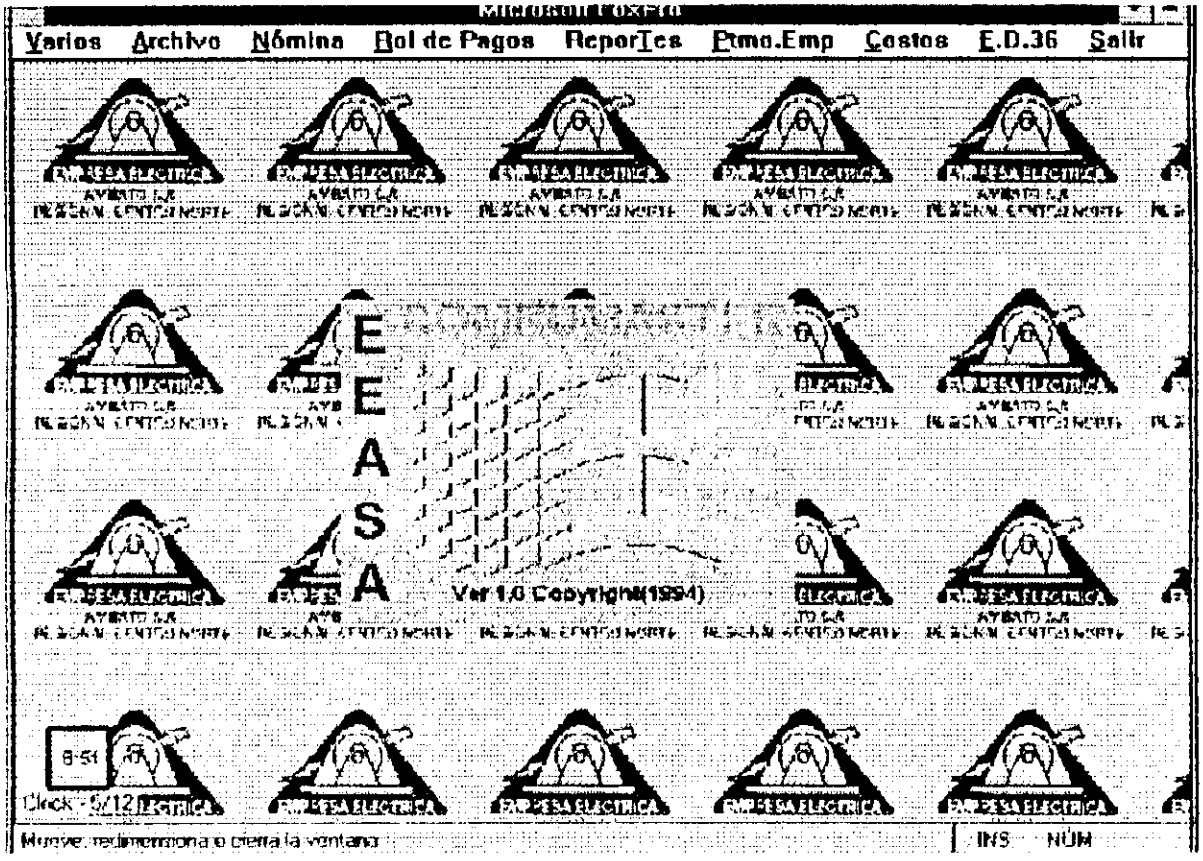
6. DOCUMENTACION DEL SISTEMA

6.1. LISTADOS Y CORRIDOS DEL SISTEMA

```
     1:
*:*************************************************************************
*************
*
     2:  *:
     3:  *: Procedure file: C:\DOC\SET.PRG
     4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
     5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
     6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
     7:  *:  Last modified: 18/04/1995 at 21:35:28
     8:  *:
     9:  *:  Procs & Fncts: ENTRAR
    10:  *:                : MCLAVE
    11:  *:
    12:  *:          Set by: CTRLERR              (procedure in
GRAFICOS.PRG)
    13:  *:
    14:  *:           Calls: EMPRESA.MPR
    15:  *:
    16:  *:           Uses: SET.DBF                Alias: SET
    17:  *:                : PROGRAMA.DBF          Alias: CLV
    18:  *:
    19:  *:       CDX files: SET.CDX
    20:  *:                : PROGRAMA.CDX
    21:  *:
    22:  *:       Documented 19:14:07
FoxDoc
version 3.00a
    23:
*:*************************************************************************
*************
*
    24:      SET TALK OFF
    25:      SET STATUS OFF
    26:      SET SAFETY OFF
    27:      SET DELETE ON
    28:      SET UNIQUE OFF
    29:      SET NEAR ON
    30:      SET EXACT ON
    31:      SET DECIMALS TO 0
    32:      SET SAFETY OFF
    33:      SET EXACT ON
    34:      SET ECHO OFF
    35:      SET SCORE OFF
    36:      SET ESCAPE OFF
    37:      SET MOUSE ON
    38:      SET CENTURY ON
    39:      SET EXCLUSIVE ON
    40:      SET DELETED ON
    41:      SET MULTILOCKS ON
    42:      SET HOURS TO 24
```

```
 45:      SET DATE TO BRITISH
 46:      SET DISPLAY TO VGA50
 47:      SET STATUS BAR ON
 48:      ON KEY LABE f4 ACTI WIND calculator
 49:      CLEAR
 50:    ┌─IF !USED('SET')
 51:    │    USE SET ORDER clave ALIAS SET
 52:    └─ENDIF
 53:    ┌─IF !USED('CLV')
 54:    │    USE programa ORDER codigo ALIAS clv IN 0
 55:    └─ENDIF
 56:      SELE SET
 57:      SEEK 'BITM'
 58:      SEEK 'MDIR'
 59:      STOR directorio TO mdirectorio
 60:      SET DEFA TO &mdirectorio
 61:
 62:      SET PATH TO srol, spte, sdmo, BITM, MENU,;
 63:         srol\dbfs, srol\mnsw, srol\mnsd, srol\pntd,;
 64:         srol\pntw, srol\prgs, srol\rptd, srol\rptw,;
 65:         spte\dbfs, spte\mnsd, spte\mnsw, spte\pntw,;
 66:         spte\pntd, spte\prgs, spte\rptd, spte\rptw,;
 67:         sdmo\dbfs, sdmo\pntd, sdmo\pntw, sdmo\rptd
 68:
 69:      *, SDMO\RPTW, SDMO\MNSD, SDMO\MNSW, SDMO\PRGS
 70:
 71:
 72:      DEFINE WIND MOVER FROM 0,0 TO 10,10
 73:      DEFINE WIND SET FROM 0,0 TO 52,92
 74:      ACTI WIND SET
 75:      RELE WIND SET
 76:      DO empresa.mpr
 77: <═══RETURN
 78:      ****************
 79:
*!*************************************************************************
*************
*
 80:  *!
 81:  *!        Procedure: ENTRAR
 82:  *!
 83:
*!*************************************************************************
*************
*
 84:      PROCEDURE entrar
 85:          ****************
 86:          PARAMETER texto,mfil,MCOL
 87:          ACTI WIND MOVER
 88:          @1,1 SAY texto FONT "R",25 STYLE "BI"
 89:        ┌─FOR i = 1 TO mfil
 90:        │    MOVE WIND MOVER TO mfil,MCOL
 91:        └─ENDFOR
 92:        ╔═FOR i = 1 TO MCOL
```

```
 93:          ||     MOVE WIND MOVER TO mfil,MCOL
 94:          ||ENDFOR
 95:  <=======RETURN
 96:           ****************
 97:
*!*****************************************************************************
*************
*
 98:  *!
 99:  *!       Procedure: MCLAVE
100:  *!
101:
*!*****************************************************************************
*************
*
102:      PROCEDURE mclave
103:           ****************
104:           validar = validar + 1
105:         ┌─IF validar = 4
106:  <======┼═══RETURN .T.
107:         └─ENDIF
108:           SELE clv
109:           SEEK mclave
110:  <=======RETURN FOUND()
111:  *:  EOF: SET.act
```


Varios Archivo Nómina Bol de Pagos Reportes Prmo.Emp Costos E.D.36 Salir
E
E
A
S
A
Ver 1.0 Copyright(1994)
8:54
Mueve, redimensiona o cierra la ventana
INS NÚM

```
     1:
*:**********************************************************************
************
*
     2:  *:
     3:  *: Procedure file: C:\DOC\EMPRESA.MPR
     4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
     5:  *:          Author: EL GRUPO FREDY - HUGO - MERCEDES
     6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
     7:  *:  Last modified: 12/05/1996 at 10:15:00
     8:  *:
     9:  *:  Procs & Fncts: _RADOLYWTB
    10:  *:                : _RADOLYXCU
    11:  *:                : _RADOLYXDX
    12:  *:
    13:  *:          Set by: SET.PRG
    14:  *:
    15:  *:           Calls: LOGO                (procedure in
GRAFICOS.PRG)
    16:  *:                : FOX2.PRG
    17:  *:                : _RADOLYWTB          (procedure in
EMPRESA.MPR)
    18:  *:                : ACERCA              (procedure in
GRAFICOS.PRG)
    19:  *:                : CTRL_RES.SPR
    20:  *:                : CTRL_RCT.PRG
    21:  *:                : VREG_PTE.SPR
    22:  *:                : VREG_ENG.SPR
    23:  *:                : VREG_AST.SPR
    24:  *:                : VREG_SBT.SPR
    25:  *:                : VREG_CAH.SPR
    26:  *:                : VCTR_RIR.SPR
    27:  *:                : AACT_SST.SPR
    28:  *:                : AACT_DPT.SPR
    29:  *:                : AACT_SEC.SPR
    30:  *:                : DGP_CARG.SPR
    31:  *:                : AP_PRMCT.SPR
    32:  *:                : AP_PRMCC.SPR
    33:  *:                : AP_PRMVR.SPR
    34:  *:                : AT_ESCLS.SPR
    35:  *:                : AT_IMPRT.SPR
    36:  *:                : IT_IMPTB.SPR
    37:  *:                : PERSONAL.PRG
    38:  *:                : CTRL_OIG.SPR
    39:  *:                : AC_IMPRT.SPR
    40:  *:                : VF_IMPRTA.SPR
    41:  *:                : CTRL_ROL.SPR
    42:  *:                : CTRL_DVB.SPR
    43:  *:                : CTRL_PRO.SPR
    44:  *:                : CTRL_GTO.SPR
    45:  *:                : CTRL_CIE.SPR
    46:  *:                : CTRL_VNG.SPR
    47:  *:                : CTRL_CRL.SPR
```

```
   48:  *:                 : CTRL_PTE.SPR
   49:  *:                 : _RADOLYXCU          (procedure in
EMPRESA.MPR)
   50:  *:                 : GNR_ACPE.SPR
   51:  *:                 : _RADOLYXDX          (procedure in
EMPRESA.MPR)
   52:  *:                 : CTRL_DMO.SPR
   53:  *:                 : DSC_GRTR.SPR
   54:  *:                 : DSC_CTAS.SPR
   55:  *:                 : PROC_DMO.SPR
   56:  *:                 : FPD26.PRG
   57:  *:                 : EXP_APTM.SPR
   58:  *:                 : EXP_ADMO.SPS
   59:  *:
   60:  *:       Documented 19:14:08
FoxDoc
version 3.00a
   61:
*:**************************************************************************
**********
*
   62:      *
************************************************************
   63:      *        *
   64:      *        * 12/05/1996              EMPRESA.MPR
10:15:00
   65:      *        *
   66:      *
************************************************************
   67:      *        *
   68:      *        * Nombre del autor
   69:      *        *
   70:      *        * Copyright (c) 1996 Nombre de la empresa
   71:      *        * Direcci≤n
   72:      *        * Ciudad,      C.P.
   73:      *        * Pa¢s
   74:      *        *
   75:      *        * Description:
   76:      *        * Este programa lo ha generado autom▀ticamente
GENMENU.
   77:      *        *
   78:      *
************************************************************
   79:
   80:
   81:      *
************************************************************
   82:      *        *
   83:      *        *                    C≤digo de configuraci≤n
   84:      *        *
   85:      *
************************************************************
   86:      *
   87:
```

```
   88:      **********
   89:      #SECTION 1
   90:      **********
   91:      PUBLIC W,llave,
fsdesc,fscampc,fscampn,fsbotones,fsboton1,pathbmps,
salirg
   92:      STOR '' TO W,llave,
fsdesc,fscampc,fscampn,fsbotones,fsboton1,pathbmps
   93:      STOR .T. TO salirg
   94:      SET TALK OFF
   95:      SET STATUS OFF
   96:      SET SAFETY OFF
   97:      SET DELETE ON
   98:      SET UNIQUE OFF
   99:      SET NEAR ON
  100:      SET EXACT ON
  101:      SET DECIMALS TO 0
  102:      SET SYSMENU TO
  103:      SET SAFETY OFF
  104:      SET EXACT ON
  105:      SET ECHO OFF
  106:      SET SCORE OFF
  107:      SET ESCAPE OFF
  108:      SET MOUSE ON
  109:      SET CENTURY ON
  110:      SET DISPLAY TO VGA50
  111:      SET EXCLUSIVE ON
  112:      SET DELETED ON
  113:      SET MULTILOCKS ON
  114:      SET HOURS TO 24
  115:      SET MESSAGE TO 24 CENTER
  116:      SET HELP OFF
  117:      SET DATE TO BRITISH
  118:      SET STATUS BAR ON
  119:      SET PATH TO srol, spte, sdmo, BITM, MENU,;
  120:         srol\dbfs, srol\mnsw, srol\mnsd, srol\pntd,;
  121:         srol\pntw, srol\prgs, srol\rptd, srol\rptw,;
  122:         spte\dbfs, spte\mnsd, spte\mnsw, spte\pntw,;
  123:         spte\pntd, spte\prgs, spte\rptd, spte\rptw,;
  124:         sdmo\dbfs, sdmo\pntd, sdmo\pntw, sdmo\rptd
  125:
  126:      SET PROCEDURE TO graficos
  127:      ON KEY LABE f4 ACTI WIND calculator
  128:      ***********
  129:      #SECTION 2
  130:      ***********
  131:      DO LOGO
  132:
  133:
  134:      *
**************************************************************
  135:      *        *
  136:      *        *                          Definici≤n de men·
  137:      *        *
```

```
  138:      *
************************************************************
  139:      *
  140:
  141:      SET SYSMENU TO
  142:
  143:      SET SYSMENU AUTOMATIC
  144:
  145:      DEFINE PAD _rad0lywoe OF _msysmenu PROMPT "\<Varios"
COLOR SCHEME 3
;
  146:         KEY f1, "Ayuda"
  147:      DEFINE PAD _rad0lywpp OF _msysmenu PROMPT "\<Archivo"
COLOR SCHEME
3
  148:      DEFINE PAD _rad0lywpw OF _msysmenu PROMPT "\<N≤mina"
COLOR SCHEME 3
;
  149:         MESSAGE "Nominas de Empleados "
  150:      DEFINE PAD _rad0lywq3 OF _msysmenu PROMPT "\<Rol de
Pagos" COLOR
SCHEME 3
  151:      DEFINE PAD _rad0lywqb OF _msysmenu PROMPT "Repor\<Tes"
COLOR SCHEME
3
  152:      DEFINE PAD _rad0lywqh OF _msysmenu PROMPT "\<Ptmo.Emp"
COLOR SCHEME
3
  153:      DEFINE PAD _rad0lywqn OF _msysmenu PROMPT "\<Costos"
COLOR SCHEME 3
  154:      DEFINE PAD _rad0lywrf OF _msysmenu PROMPT "\<E.D.36"
COLOR SCHEME 3
  155:      DEFINE PAD _rad0lywrl OF _msysmenu PROMPT "\<Salir"
COLOR SCHEME 3
  156:      ON PAD _rad0lywoe OF _msysmenu ACTIVATE POPUP varios
  157:      ON PAD _rad0lywpp OF _msysmenu ACTIVATE POPUP archivo
  158:      ON PAD _rad0lywpw OF _msysmenu ACTIVATE POPUP n≤mina
  159:      ON PAD _rad0lywq3 OF _msysmenu ACTIVATE POPUP
roldepagos
  160:      ON SELECTION PAD _rad0lywqb OF _msysmenu DO fox2
  161:      ON PAD _rad0lywqh OF _msysmenu ACTIVATE POPUP ptmoemp
  162:      ON PAD _rad0lywqn OF _msysmenu ACTIVATE POPUP costos
  163:      ON PAD _rad0lywrf OF _msysmenu ACTIVATE POPUP ed36
  164:      ON SELECTION PAD _rad0lywrl OF _msysmenu ;
  165:         DO _rad0lywtb ;
  166:         IN LOCFILE("\SIRP\MENU\EMPRESA" ,"MPX;MPR¦FXP;PRG"
,"¬ D≤nde estβ
EMPRESA?")
  167:
  168:      DEFINE POPUP varios MARGIN RELATIVE SHADOW COLOR
SCHEME 4
  169:      DEFINE BAR 1 OF varios PROMPT "\<Acerca" ;
  170:         MESSAGE 'Acerca del Sistema'
  171:      DEFINE BAR 2 OF varios PROMPT "\<Respaldo" ;
```

```
  172:        MESSAGE 'Respaldo de Informaci≤n'
  173:     DEFINE BAR 3 OF varios PROMPT "Recons\<Truir" ;
  174:        MESSAGE 'Reconstrucci≤n de Tablas de Indices'
  175:     DEFINE BAR 4 OF varios PROMPT "Re\<Gistro Autom./
Datos/Dscto. en
Rol" ;
  176:        MESSAGE 'REGISTRA DATOS PARA DESCUENTOS EN ROL
AUTOMATICAMENTE'
  177:     DEFINE BAR 5 OF varios PROMPT "Preparar
Arc\<Hivo/Reg.Inf." ;
  178:        MESSAGE 'GENERA EL ARCHIVO NECESARIO PARA REISTRAR
INFORMACION
DE ROLES DE PAGO'
  179:     DEFINE BAR 6 OF varios PROMPT "\<Calculadora"
  180:     ON SELECTION BAR 1 OF varios DO acerca
  181:     ON SELECTION BAR 2 OF varios DO ctrl_res.spr
  182:     ON SELECTION BAR 3 OF varios DO ctrl_rct
  183:     ON BAR 4 OF varios ACTIVATE POPUP registroau
  184:     ON BAR 5 OF varios ACTIVATE POPUP prepararar
  185:     ON SELECTION BAR 6 OF varios ACTI WIND calculator
  186:
  187:     DEFINE POPUP registroau MARGIN RELATIVE SHADOW COLOR
SCHEME 4
  188:     DEFINE BAR 1 OF registroau PROMPT "\<Ptmo. Empresa"
  189:     DEFINE BAR 2 OF registroau PROMPT "\<Energ¢a"
  190:     DEFINE BAR 3 OF registroau PROMPT "\<Asistencia"
  191:     DEFINE BAR 4 OF registroau PROMPT "\<Sobretiempos"
  192:     DEFINE BAR 5 OF registroau PROMPT "\<Cooperativa de
Ahorros"
  193:     ON SELECTION BAR 1 OF registroau DO vreg_pte.spr
  194:     ON SELECTION BAR 2 OF registroau DO vreg_eng.spr
  195:     ON SELECTION BAR 3 OF registroau DO vreg_ast.spr
  196:     ON SELECTION BAR 4 OF registroau DO vreg_sbt.spr
  197:     ON SELECTION BAR 5 OF registroau DO vreg_cah.spr
  198:
  199:     DEFINE POPUP prepararar MARGIN RELATIVE SHADOW COLOR
SCHEME 4
  200:     DEFINE BAR 1 OF prepararar PROMPT "\<Roles de Pago"
  201:     ON SELECTION BAR 1 OF prepararar DO vctr_rir.spr
  202:
  203:     DEFINE POPUP archivo MARGIN RELATIVE SHADOW COLOR
SCHEME 4
  204:     DEFINE BAR 1 OF archivo PROMPT "\ACTUALIZAR"
  205:     DEFINE BAR 2 OF archivo PROMPT "\-"
  206:     DEFINE BAR 3 OF archivo PROMPT "\<Sistemas" ;
  207:        MESSAGE 'CREA, MODIFICA, ELIMINA SISTEMAS DE ROLES
DE PAGO'
  208:     DEFINE BAR 4 OF archivo PROMPT "\<Departamentos" ;
  209:        MESSAGE 'ACTUALIZACION DE DEPARTAMENTOS'
  210:     DEFINE BAR 5 OF archivo PROMPT "Se\<Cciones" ;
  211:        MESSAGE 'ACTUALIZACION DE SECCIONES Y GRUPOS DE
TRABAJO'
  212:     DEFINE BAR 6 OF archivo PROMPT "Car\<Gos" ;
  213:        MESSAGE 'ACTUALIZACION DE CARGOS'
```

```
  214:      DEFINE BAR 7 OF archivo PROMPT "\-"
  215:      DEFINE BAR 8 OF archivo PROMPT "\PARAMETROS"
  216:      DEFINE BAR 9 OF archivo PROMPT "\-"
  217:      DEFINE BAR 10 OF archivo PROMPT "C≤digo de \<Trabajo"
;
  218:         MESSAGE 'ACTUALIZACION DE LEYES DEL CODIGO DE
TRABAJO'
  219:      DEFINE BAR 11 OF archivo PROMPT "Contrato Co\<Lectivo"
;
  220:         MESSAGE 'ACTUALIZACION DE ARTICULOS DEL CONTRATO
COLECTIVO'
  221:      DEFINE BAR 12 OF archivo PROMPT "\<Varios" ;
  222:         MESSAGE 'ACTUALIZACION DE VARIOS PARAMETROS '
  223:      DEFINE BAR 13 OF archivo PROMPT "\-"
  224:      DEFINE BAR 14 OF archivo PROMPT "\TABLAS"
  225:      DEFINE BAR 15 OF archivo PROMPT "\-"
  226:      DEFINE BAR 16 OF archivo PROMPT "\<Escalas" ;
  227:         MESSAGE 'ACTUALIZA VALORES DE SUELDOS, GASTOS Y
COMISARIATO EN
BASE A ESCALAS'
  228:      DEFINE BAR 17 OF archivo PROMPT "Impuesto a la
\<Renta" ;
  229:         MESSAGE 'ACTUALIZACION DE LA TABLA PARA EL CALCULO
DEL IMPUESTO
A LA RENTA'
  230:      DEFINE BAR 18 OF archivo PROMPT "\<Impresi≤n de
Tablas" ;
  231:         MESSAGE 'IMPRIME TABLAS DE SUELDOS, GASTOS,
COMISARIATO E
IMPUESTO A LA RENTA'
  232:      ON SELECTION BAR 3 OF archivo DO aact_sst.spr
  233:      ON SELECTION BAR 4 OF archivo DO aact_dpt.spr
  234:      ON SELECTION BAR 5 OF archivo DO aact_sec.spr
  235:      ON SELECTION BAR 6 OF archivo DO dgp_carg.spr
  236:      ON SELECTION BAR 10 OF archivo DO ap_prmct.spr
  237:      ON SELECTION BAR 11 OF archivo DO ap_prmcc.spr
  238:      ON SELECTION BAR 12 OF archivo DO ap_prmvr.spr
  239:      ON SELECTION BAR 16 OF archivo DO at_escls.spr
  240:      ON SELECTION BAR 17 OF archivo DO at_imprt.spr
  241:      ON SELECTION BAR 18 OF archivo DO it_imptb.spr
  242:
  243:      DEFINE POPUP n≤mina MARGIN RELATIVE SHADOW COLOR
SCHEME 4
  244:      DEFINE BAR 1 OF n≤mina PROMPT "\FICHA"
  245:      DEFINE BAR 2 OF n≤mina PROMPT "\-"
  246:      DEFINE BAR 3 OF n≤mina PROMPT "\<Datos Personales" ;
  247:         MESSAGE 'ACTUALIZA DATOS GENERALES DE PERSONAL'
  248:      DEFINE BAR 4 OF n≤mina PROMPT "\-"
  249:      DEFINE BAR 5 OF n≤mina PROMPT "\DATOS PARA IMPUESTO A
LA RENTA" ;
  250:         MESSAGE 'ACTUALIZACION DE DATOS PARA IMPUESTO A LA
RENTA'
  251:      DEFINE BAR 6 OF n≤mina PROMPT "\-"
  252:      DEFINE BAR 7 OF n≤mina PROMPT "\<Otros Ingresos
```

```
Gravables" ;
  253:        MESSAGE 'REGISTRO DE DATOS QUE NO SON CONSIDERADOS
EN EL ROL DE
PAGOS'
  254:     DEFINE BAR 8 OF nómina PROMPT "\<Valores Acumulados" ;
  255:        MESSAGE 'VALORES DE DATOS QUE SE ACUMULAN MEDIANTE
LOS PROCESOS
DE ROL'
  256:     DEFINE BAR 9 OF nómina PROMPT "Valor \<Fijo a Ser
Descontado
Mensualmente" ;
  257:        MESSAGE 'REGISTRAR VALORES FIJOS QUE SERAN
DESCONTADOS
MENSUALMENTE COMO IMPUESTO A LA RENTA'
  258:     ON SELECTION BAR 3 OF nómina DO personal
  259:     ON SELECTION BAR 7 OF nómina DO ctrl_oig.spr
  260:     ON SELECTION BAR 8 OF nómina DO ac_imprt.spr
  261:     ON SELECTION BAR 9 OF nómina DO vf_imprta.spr
  262:
  263:     DEFINE POPUP roldepagos MARGIN RELATIVE SHADOW COLOR
SCHEME 4
  264:     DEFINE BAR 1 OF roldepagos PROMPT "\REGISTRO DE DATOS"
;
  265:        MESSAGE "Datos del Rol"
  266:     DEFINE BAR 2 OF roldepagos PROMPT "\-"
  267:     DEFINE BAR 3 OF roldepagos PROMPT "\<Otros Ingresos y
Asistencia" ;
  268:        MESSAGE 'REGISTRO DE DATOS PARA OTROS INGRESOS Y
ASISTENCIA'
  269:     DEFINE BAR 4 OF roldepagos PROMPT "D\<Escuentos en Rol
de Pagos" ;
  270:        MESSAGE 'REGISTRO DE DATOS QUE SERAN DESCONTADOS A
TRAVES DEL
ROL DE PAGOS'
  271:     DEFINE BAR 5 OF roldepagos PROMPT "De\<Scuentos
Varios" ;
  272:        MESSAGE 'REGISTRA DATOS PARA SER DESCONTADOS EN
BONIFICACIONES O
CUALQUIER BENEFICIO SOCIAL O POR CONTRATO COLECTIVO '
  273:     DEFINE BAR 6 OF roldepagos PROMPT "\-"
  274:     DEFINE BAR 7 OF roldepagos PROMPT "\PROCESOS"
  275:     DEFINE BAR 8 OF roldepagos PROMPT "\-"
  276:     DEFINE BAR 9 OF roldepagos PROMPT "\<Procesar Rol de
Pagos" ;
  277:        MESSAGE 'REALIZA PROCESAMIENTO DE  ROL DE PAGOS'
  278:     DEFINE BAR 10 OF roldepagos PROMPT "Procesos \<Varios"
;
  279:        MESSAGE 'PROCESA BONIFICACIONES, O CUALQUIER OTRO
BENEFICIO
SOCIAL A PAGARSE '
  280:     DEFINE BAR 11 OF roldepagos PROMPT "Procesar/Acumular
\<Gastos"
  281:     DEFINE BAR 12 OF roldepagos PROMPT "\<Cierre Mensual"
;
```

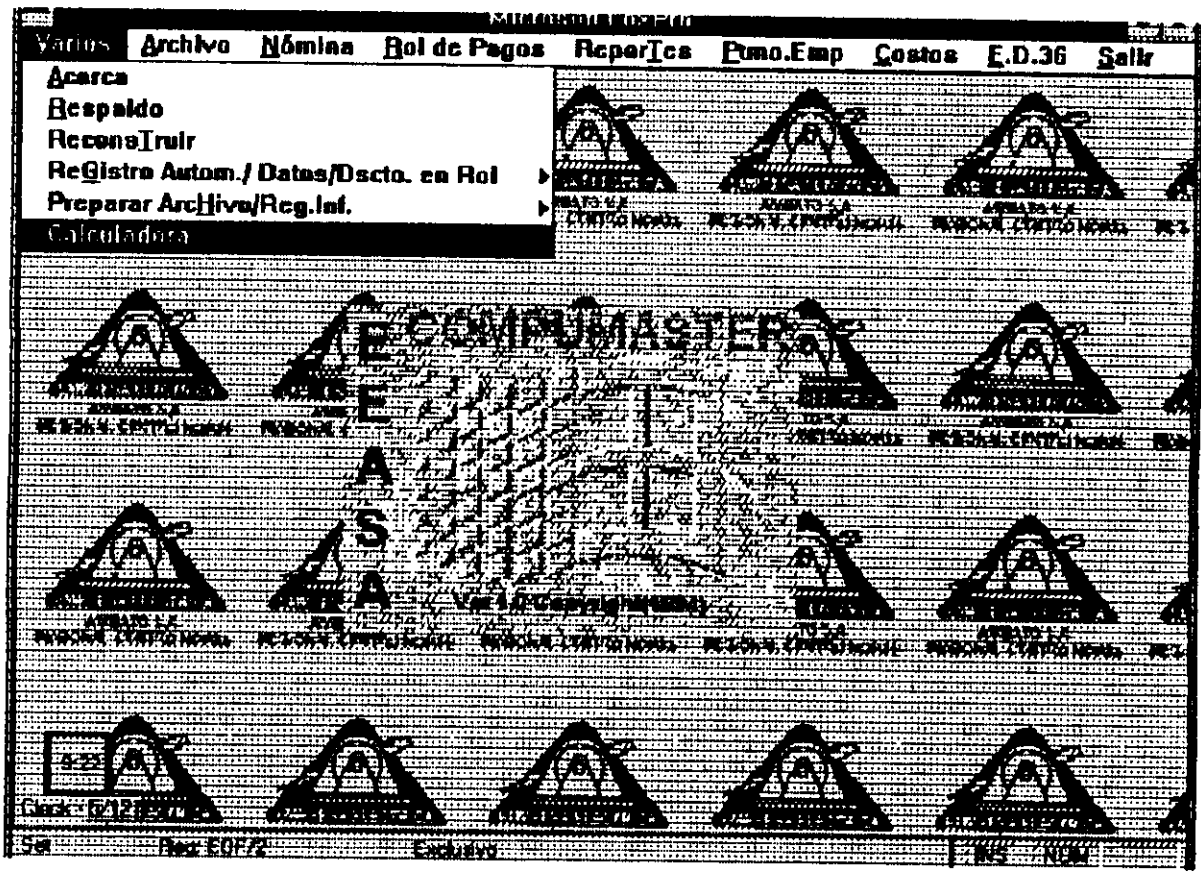
```
  282:        MESSAGE 'REALIZA EL CIERRE Y ACUMULACION MENSUAL DE
LA
INFORMACION DE ROLES DE PAGO'
  283:      DEFINE BAR 13 OF roldepagos PROMPT "\-"
  284:      DEFINE BAR 14 OF roldepagos PROMPT "\CONSULTAS"
  285:      DEFINE BAR 15 OF roldepagos PROMPT "\-"
  286:      DEFINE BAR 16 OF roldepagos PROMPT "Valores
\<Negativos"
  287:      DEFINE BAR 17 OF roldepagos PROMPT "Consulta Rol
\<Individual"
  288:      ON SELECTION BAR 3 OF roldepagos DO ctrl_rol.spr WITH
2
  289:      ON SELECTION BAR 4 OF roldepagos DO ctrl_rol.spr WITH
3
  290:      ON SELECTION BAR 5 OF roldepagos DO ctrl_dvb.spr WITH
'Do Pr_dvbon'
  291:      ON SELECTION BAR 9 OF roldepagos DO ctrl_pro.spr
  292:      ON SELECTION BAR 10 OF roldepagos DO ctrl_dvb.spr WITH
'pr_dvbon'
  293:      ON SELECTION BAR 11 OF roldepagos DO ctrl_gto.spr
  294:      ON SELECTION BAR 12 OF roldepagos DO ctrl_cie.spr
  295:      ON SELECTION BAR 16 OF roldepagos DO ctrl_vng.spr
  296:      ON SELECTION BAR 17 OF roldepagos DO ctrl_crl.spr
  297:
  298:      DEFINE POPUP ptmoemp MARGIN RELATIVE SHADOW COLOR
SCHEME 4
  299:      DEFINE BAR 1 OF ptmoemp PROMPT "\ACTUALIZACION"
  300:      DEFINE BAR 2 OF ptmoemp PROMPT "\-"
  301:      DEFINE BAR 3 OF ptmoemp PROMPT "\<Datos de Prθstamos"
;
  302:        MESSAGE 'ACTUALIZACION DE DATOS DE PRESTAMOS
EMPRESA'
  303:      DEFINE BAR 4 OF ptmoemp PROMPT "\-"
  304:      DEFINE BAR 5 OF ptmoemp PROMPT "\VARIOS"
  305:      DEFINE BAR 6 OF ptmoemp PROMPT "\-"
  306:      DEFINE BAR 7 OF ptmoemp PROMPT "\<Procesar
Informaci≤n" ;
  307:        MESSAGE 'PROCESA LOS VALORES DE PRESTAMOS A
DESCONTAR EN LOS ROL
DE PAGOS'
  308:      DEFINE BAR 8 OF ptmoemp PROMPT "\<Generar Asientos
Contables" ;
  309:        MESSAGE 'CREA ASIENTOS CONTABLES DE LOS PRESTAMOS
DE EMPRESA'
  310:      DEFINE BAR 9 OF ptmoemp PROMPT "\-"
  311:      DEFINE BAR 10 OF ptmoemp PROMPT "\REPORTES"
  312:      DEFINE BAR 11 OF ptmoemp PROMPT "\-"
  313:      DEFINE BAR 12 OF ptmoemp PROMPT "\<Reportes Generales"
;
  314:        MESSAGE 'VARIOS INFORMES DE PRESTAMOS EMPRESA QUE
SE IMPRIMIRAN
BAJO PLANTILLA DE FOXPRO PARA D O S'
  315:      ON SELECTION BAR 3 OF ptmoemp DO ctrl_pte.spr
  316:      ON SELECTION BAR 7 OF ptmoemp ;
```

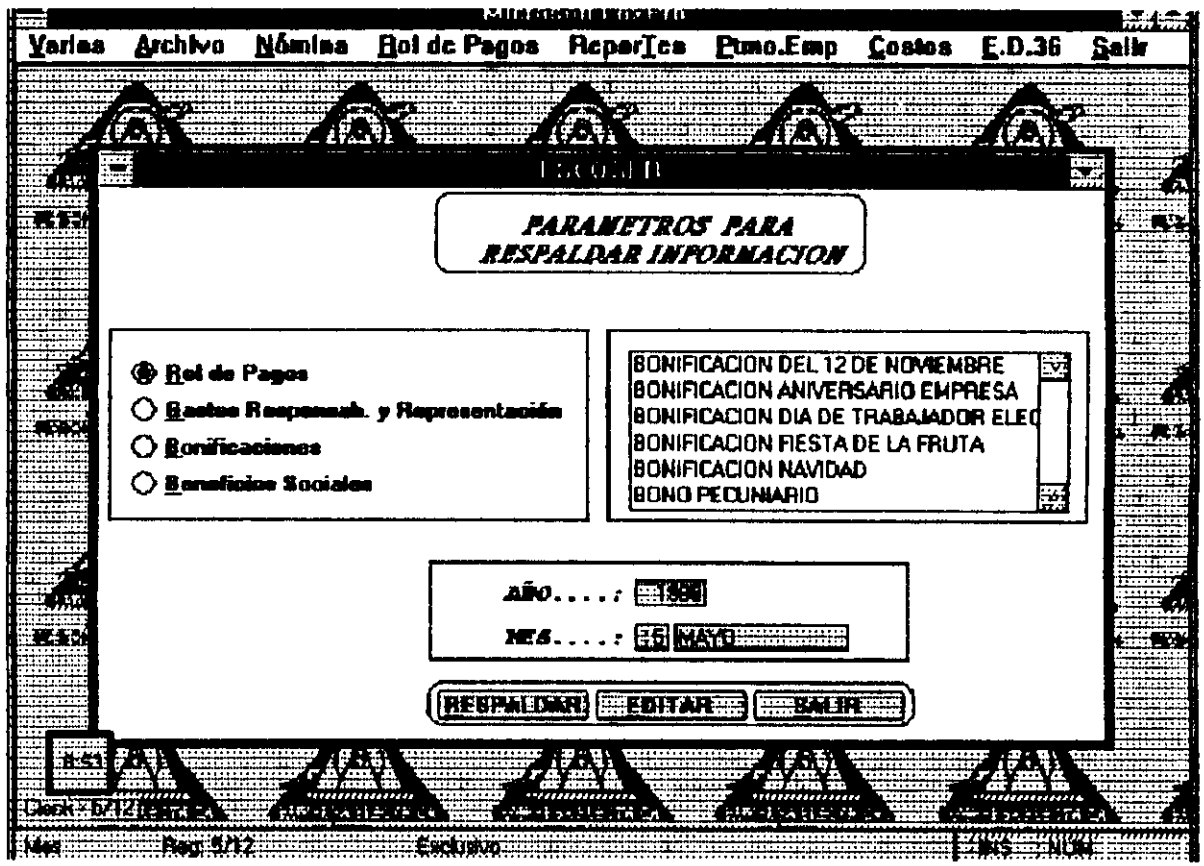
```
   317:          DO _rad0lyxcu ;
   318:          IN LOCFILE("\SIRP\MENU\EMPRESA" ,"MPX;MPR|FXP;PRG"
,"¿Dónde está
EMPRESA?")
   319:       ON SELECTION BAR 8 OF ptmoemp DO gnr_acpe.spr
   320:       ON SELECTION BAR 12 OF ptmoemp ;
   321:          DO _rad0lyxdx ;
   322:          IN LOCFILE("\SIRP\MENU\EMPRESA" ,"MPX;MPR|FXP;PRG"
,"¿Dónde está
EMPRESA?")
   323:
   324:       DEFINE POPUP costos MARGIN RELATIVE SHADOW COLOR
SCHEME 4
   325:       DEFINE BAR 1 OF costos PROMPT "\ACTUALIZACION"
   326:       DEFINE BAR 2 OF costos PROMPT "\-"
   327:       DEFINE BAR 3 OF costos PROMPT "Partes Diarios de
\<Labor" ;
   328:          MESSAGE 'REGISTRO DE PARTES DIARIOS DE LABOR'
   329:       DEFINE BAR 4 OF costos PROMPT "\<Grupos de Trabajo" ;
   330:          MESSAGE 'ACTUALIZA LA TABLA DE GRUPOS DE TRABAJO'
   331:       DEFINE BAR 5 OF costos PROMPT "\<Cuentas Contables"
   332:       DEFINE BAR 6 OF costos PROMPT "\-"
   333:       DEFINE BAR 7 OF costos PROMPT "\PROCESOS"
   334:       DEFINE BAR 8 OF costos PROMPT "\-"
   335:       DEFINE BAR 9 OF costos PROMPT "\<Procesar Informaci≤n"
;
   336:          MESSAGE 'PROCESA LA INFORMACION DE MANO DE OBRA'
   337:       DEFINE BAR 10 OF costos PROMPT "\-"
   338:       DEFINE BAR 11 OF costos PROMPT "\REPORTES GENERALES"
   339:       DEFINE BAR 12 OF costos PROMPT "\-"
   340:       DEFINE BAR 13 OF costos PROMPT "Rol Integrado con
\<Beneficios
Sociales" ;
   341:          MESSAGE 'SE OBTIENEN REPORTES REFERENTES AL ROL CON
BENEFICIOS
SOCIALES'
   342:       DEFINE BAR 14 OF costos PROMPT "\<Distribuci≤n de Mano
de Obra" ;
   343:          MESSAGE 'RESULTADOS DEL PROCESO DE DISTRIBUCION DE
MANO DE OBRA'
   344:       ON SELECTION BAR 3 OF costos DO ctrl_dmo.spr
   345:       ON SELECTION BAR 4 OF costos DO dsc_grtr.spr
   346:       ON SELECTION BAR 5 OF costos DO dsc_ctas.spr
   347:       ON SELECTION BAR 9 OF costos DO proc_dmo.spr
   348:       ON SELECTION BAR 13 OF costos DO fpd26 WITH
'in_rolbs.spr'
   349:       ON SELECTION BAR 14 OF costos DO fpd26 WITH
'in_prdmo.spr'
   350:
   351:       DEFINE POPUP ed36 MARGIN RELATIVE SHADOW COLOR SCHEME
4
   352:       DEFINE BAR 1 OF ed36 PROMPT "\ASIENTOS CONTABLES"
   353:       DEFINE BAR 2 OF ed36 PROMPT "\-"
   354:       DEFINE BAR 3 OF ed36 PROMPT "De \<Prθstamos Empresa"
```

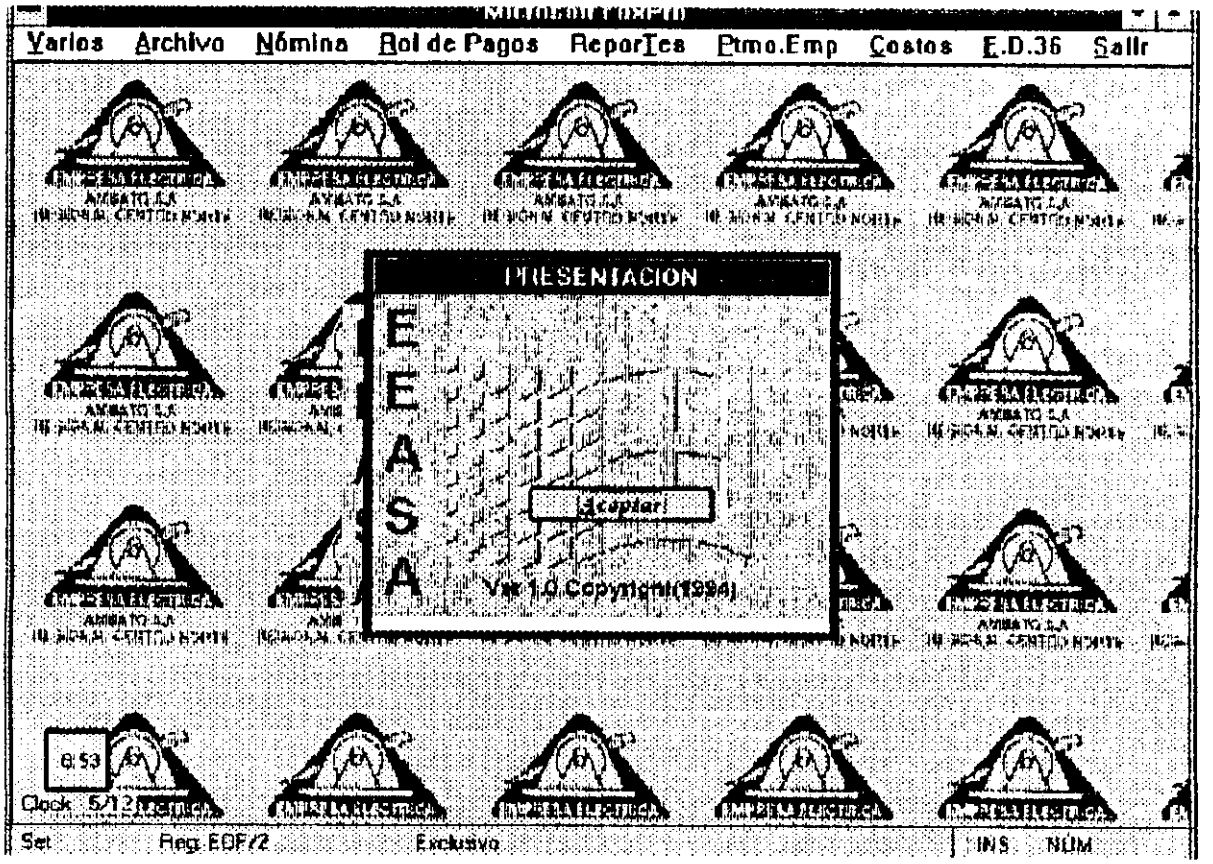
```
  355:      DEFINE BAR 4 OF ed36 PROMPT "De \<Distribuci≤n de Mano
de Obra"
  356:      ON SELECTION BAR 3 OF ed36 DO exp_aptm.spr
  357:      ON SELECTION BAR 4 OF ed36 DO exp_admo.sps
  358:
  359:
  360:      *
************************************************************
  361:      *        *
  362:      *        *            C≤digo y procedimientos de
limpieza
  363:      *        *
  364:      *
************************************************************
  365:      *
  366:
  367:    ╔═DO WHILE salirg
  368:    ╠    ACTIVATE MENU _msysmenu
  369:    ╚═ENDDO
  370: <═══DEACT MENU _msysmenu
  371:      RELE WIND ALL
  372:      CLOSE ALL
  373:      CLEAR ALL
  374:      CLEAR
  375:      SET SYSMENU TO DEFA
  376:      *quit
  377:
  378:      *
************************************************************
  379:      *        *
  380:      *        * _RADOLYWTB  ON SELECTION PAD
  381:      *        *
  382:      *        * Procedure Origin:
  383:      *        *
  384:      *        * From Menu:  EMPRESA.MPR,              Record:
106
  385:      *        * Called By:  ON SELECTION PAD
  386:      *        * Prompt:     Salir
  387:      *        * Snippet:    1
  388:      *        *
  389:      *
************************************************************
  390:      *
  391:
*!*************************************************************************
*************
*
  392:  *!
  393:  *!       Procedure: _RADOLYWTB
  394:  *!
  395:  *!       Called by: EMPRESA.MPR
  396:  *!
  397:
*!*************************************************************************
```

```
**************
*
  398:      PROCEDURE _rad0lywtb
  399:          SET HELP ON
  400:          STOR .F. TO salirg
  401: <=======RETURN
  402:
  403:
  404:
  405:          *
************************************************************
  406:          *       *
  407:          *       * _RADOLYXCU  ON SELECTION BAR 7 OF POPUP
ptmoemp
  408:          *       *
  409:          *       * Procedure Origin:
  410:          *       *
  411:          *       * From Menu:  EMPRESA.MPR,
Record:   78
  412:          *       * Called By:  ON SELECTION BAR 7 OF POPUP
ptmoemp
  413:          *       * Prompt:     Procesar Informaci≤n
  414:          *       * Snippet:    2
  415:          *       *
  416:          *
************************************************************
  417:          *
  418:
*!*************************************************************************
**************
*
  419:  *!
  420:  *!       Procedure: _RADOLYXCU
  421:  *!
  422:  *!       Called by: EMPRESA.MPR
  423:  *!
  424:
*!*************************************************************************
**************
*
  425:      PROCEDURE _rad0lyxcu
  426:          RUN C:\sisaplic\fpd26\fox.exe -t C:\sirp\set_ptmo
1
  427:          SET DISP TO VGA50
  428: <=======RETURN
  429:
  430:
  431:          *
************************************************************
  432:          *       *
  433:          *       * _RADOLYXDX  ON SELECTION BAR 12 OF POPUP
ptmoemp
  434:          *       *
  435:          *       * Procedure Origin:
```

```
  436:          *       *
  437:          *       * From Menu:  EMPRESA.MPR,
Record:   83
  438:          *       * Called By:  ON SELECTION BAR 12 OF POPUP
ptmoemp
  439:          *       * Prompt:     Reportes Generales
  440:          *       * Snippet:    3
  441:          *       *
  442:          *
**************************************************************
  443:          *
  444:
*!*************************************************************************
************
*
  445:  *!
  446:  *!       Procedure: _RADOLYXDX
  447:  *!
  448:  *!       Called by: EMPRESA.MPR
  449:  *!
  450:
*!*************************************************************************
************
*
  451:      PROCEDURE _rad0lyxdx
  452:          RUN C:\sisaplic\fpd26\fox.exe -t C:\sirp\set_ptmo
2
  453:          SET DISP TO VGA50
  454: <=======RETURN
  455:  *:  EOF: EMPRESA.ac2
```


Varios
Archivo
Nómina
Rol de Pagos
ReporTes
Pmo.Emp
Costos
E.D.36
Salir
Acerca
Respaldo
ReconsTruir
ReGistro Autom./ Datos/Dscto. en Rol
Preparar ArcHivo/Reg.Inf.
Calculadora
COMPUMASTER
E
A
S
A


Varios Archivo Nómina Rol de Pagos ReporTes Ptmo.Emp Costos E.D.36 Salir
PARAMETROS PARA
RESPALDAR INFORMACION
Rol de Pagos
Gastos Responsab. y Representación
Bonificaciones
Beneficios Sociales
BONIFICACION DEL 12 DE NOVIEMBRE
BONIFICACION ANIVERSARIO EMPRESA
BONIFICACION DIA DE TRABAJADOR ELEC
BONIFICACION FIESTA DE LA FRUTA
BONIFICACION NAVIDAD
BONO PECUNIARIO
AÑO.....: 1998
MES.....: 06 MAYO
RESPALDAR
EDITAR
SALIR


Varios Archivo Nómina Rol de Pagos ReporTes Ptmo.Emp Costos E.D.36 Salir
PRESENTACION
E
E
A
S
A
Aceptar
Ver 1.0 Copyright(1994)
8:53
Clock 5/12
Set
Reg EOF/2
Exclusivo
INS
NUM

```
     1:
*:*************************************************************************
*************
*
     2:  *:
     3:  *: Procedure file: C:\DOC\CTRL_RES.SPR
     4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
     5:  *:          Author: EL GRUPO FREDY - HUGO - MERCEDES
     6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
     7:  *:  Last modified: 11/05/1996 at 16:20:26
     8:  *:
     9:  *:  Procs & Fncts: _RACOZOUEG()
    10:  *:                : _RACOZOUJJ()
    11:  *:                : _RACOZOUMB()
    12:  *:                : _RACOZOUR3()
    13:  *:                : RESPALDAR
    14:  *:                : EDITAR
    15:  *:                : SALIR
    16:  *:
    17:  *:          Set by: EMPRESA.MPR
    18:  *:
    19:  *:           Calls: _RACOZOUEG()       (function in
CTRL_RES.SPR)
    20:  *:                : _RACOZOUJJ()       (function in
CTRL_RES.SPR)
    21:  *:                : _RACOZOUMB()       (function in
CTRL_RES.SPR)
    22:  *:                : _RACOZOUR3()       (function in
CTRL_RES.SPR)
    23:  *:
    24:  *:       Documented 19:14:12
FoxDoc
version 3.00a
    25:
*:*************************************************************************
*************
*
    26:      *
************************************************************
    27:      *       *
    28:      *       * 11/05/1996            CTRL_RES.SPR
 16:20:24
    29:      *       *
    30:      *
************************************************************
    31:      *       *
    32:      *       * Nombre del autor
    33:      *       *
    34:      *       * Copyright (c) 1996 Nombre de la empresa
    35:      *       * Direcci≤n
    36:      *       * Ciudad,      C.P.
    37:      *       * Pa¢s
    38:      *       *
```

```
    39:      *        * Description:
    40:      *        * Este programa lo ha generado autom▀ticamente
GENSCRN.
    41:      *        *
    42:      *
**************************************************************
    43:
    44:
    45:      *
**************************************************************
    46:      *        *
    47:      *        *  CTRL_RES/Windows C≤digo de configuraci≤n -
SECCION 1
    48:      *        *
    49:      *
**************************************************************
    50:      *
    51:
    52:      #REGION 1
    53:      **********
    54:      REGIONAL opc_scr, opc_rol, mmes, vmes, manio, mrol,
mcod_ciu,
mfiltro, mclave, opc_resp
    55:      STOR ' ' TO opc_scr, mmes,opc_rol, mfiltro, mclave
    56:      STOR 1 TO mrol,mcod_ciu, opc_resp
    57:      STOR YEAR(DATE()) TO manio
    58:      STOR MONTH(DATE()) TO vmes
    59:
    60:      **********
    61:
    62:      #REGION 0
    63:      REGIONAL m.currarea, m.talkstat, m.compstat
    64:
    65:    ┌─IF SET("TALK") = "ON"
    66:    │    SET TALK OFF
    67:    │    m.talkstat = "ON"
    68:    ├─ELSE
    69:    │    m.talkstat = "OFF"
    70:    └─ENDIF
    71:      m.compstat = SET("COMPATIBLE")
    72:      SET COMPATIBLE FOXPLUS
    73:
    74:      m.rborder = SET("READBORDER")
    75:      SET readborder ON
    76:
    77:      m.currarea = SELECT()
    78:
    79:
    80:      *
**************************************************************
    81:      *        *
    82:      *        *  CTRL_RES/Windows Bases de datos, ╒ndices,
relaciones
    83:      *        *
```

```
*************************************************************
*
*
IF USED("acr")
    SELECT acr
    SET ORDER TO TAG "cod_per"
ELSE
    SELECT 0
    USE (LOCFILE("srol\dbfs\ac_imprt.dbf","DBF","Buscar
ac_imprt?"));
        AGAIN ALIAS acr ;
        ORDER TAG "cod_per"
ENDIF

IF USED("mes")
    SELECT mes
    SET ORDER TO 0
ELSE
    SELECT 0
    USE (LOCFILE("srol\dbfs\meses.org","DBF","Buscar
meses?"));
        AGAIN ALIAS mes ;
        ORDER 0
ENDIF

IF USED("bnf")
    SELECT bnf
    SET ORDER TO TAG "codigo"
ELSE
    SELECT 0
    USE (LOCFILE("srol\dbfs\bonifica.dbf","DBF","Buscar
bonifica?"));
        AGAIN ALIAS bnf ;
        ORDER TAG "codigo"
ENDIF

IF USED("var")
    SELECT VAR
    SET ORDER TO TAG "bonific"
ELSE
    SELECT 0
    USE (LOCFILE("srol\dbfs\varios.dbf","DBF","Buscar
varios?"));
        AGAIN ALIAS VAR ;
        ORDER TAG "bonific"
ENDIF

IF USED("sst")
    SELECT sst
    SET ORDER TO TAG "todos"
ELSE
    SELECT 0
    USE (LOCFILE("srol\dbfs\sistema.dbf","DBF","Buscar
```

```
sistema?")):
  133:     |        AGAIN ALIAS sst ;
  134:     |        ORDER TAG "todos"
  135:     └─ENDIF
  136:
  137:     ┌─IF USED("ctr")
  138:     |     SELECT ctr
  139:     |     SET ORDER TO 0
  140:     ├─ELSE
  141:     |     SELECT 0
  142:     |     USE (LOCFILE("srol\dbfs\ctrl_rol.dbf","DBF","Buscar
ctrl_rol?"));
  143:     |        AGAIN ALIAS ctr ;
  144:     |        ORDER 0
  145:     └─ENDIF
  146:
  147:       SELECT acr
  148:
  149:       SELECT bnf
  150:       SELECT bnf
  151:
  152:       SET RELATION OFF INTO acr
  153:       SET RELATION TO codigo INTO acr ADDITIVE
  154:
  155:       SELECT acr
  156:
  157:       *
************************************************************
  158:       *        *
  159:       *        *              Windows Definiciones de ventana
  160:       *        *
  161:       *
************************************************************
  162:       *
  163:
  164:     ┌─IF NOT WEXIST("ctrl_par") ;
  165:     |        OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.PJX" ;
  166:     |        OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.SCX" ;
  167:     |        OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.MNX" ;
  168:     |        OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.PRG" ;
  169:     |        OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.FRX" ;
  170:     |        OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.QPR"
  171:     |     DEFINE WINDOW ctrl_par ;
  172:     |        AT  0.000, 0.000  ;
  173:     |        SIZE 24.077,108.000 ;
  174:     |        TITLE "ESCOGER" ;
  175:     |        FONT "MS Sans Serif", 8 ;
  176:     |        FLOAT ;
  177:     |        NOCLOSE ;
  178:     |        MINIMIZE ;
  179:     |        SYSTEM
  180:     |     MOVE WINDOW ctrl_par CENTER
  181:     └─ENDIF
  182:
```

```
183:
184:     *
*************************************************************
185:     *       *
186:     *       *  CTRL_RES/Windows C≤digo de configuraci≤n -
SECCION 2
187:     *       *
188:     *
*************************************************************
189:     *
190:
191:     #REGION 1
192:     **********
193:     SELE mes
194:     LOCATE FOR mes = vmes
195:     STOR nom_mes TO mmes
196:
197:     #REGION 1
198:     DEFINE POPUP _rac0z0u5g ;
199:        PROMPT FIELD var.descrip ;
200:        SCROLL
201:
202:     *
*************************************************************
203:     *       *
204:     *       *        CTRL_RES/Windows Distribuci≤n de
pantalla
205:     *       *
206:     *
*************************************************************
207:     *
208:
209:     #REGION 1
210:     IF WVISIBLE("ctrl_par")
211:        ACTIVATE WINDOW ctrl_par SAME
212:     ELSE
213:        ACTIVATE WINDOW ctrl_par NOSHOW
214:     ENDIF
215:     @ 19.077,43.800 SAY "MES....:" ;
216:        FONT "Courier New", 10 ;
217:        STYLE "BIT"
218:     @ 17.154,43.800 SAY "A±O....:" ;
219:        FONT "Courier New", 10 ;
220:        STYLE "BIT"
221:     @ 21.615,35.400 TO 23.538,87.800 ;
222:        PEN 1, 8 ;
223:        STYLE "16"
224:     @ 1.000,39.000 SAY "PARAMETROS  PARA " + CHR(13) + ;
225:        "RESPALDAR INFORMACION " ;
226:        SIZE 2.000,30.429, 0.000 ;
227:        PICTURE "@I" ;
228:        FONT "MS Serif", 10 ;
229:        STYLE "BIT"
230:     @ 0.231,36.200 TO 3.693,82.600 ;
```

```
231:          PEN 1, 8 ;
232:          STYLE "16"
233:       @ 6.154,1.200 TO 14.616,52.800 ;
234:          PEN 1, 8
235:       @ 17.231,58.200 GET manio ;
236:          SIZE 1.000,6.600 ;
237:          DEFAULT " " ;
238:          FONT "MS Sans Serif", 8 ;
239:          PICTURE "9999" ;
240:          VALID manio > 1990 AND manio < 2999 ;
241:          ERROR "AÑO INVALIDO "
242:       @ 19.154,58.200 GET vmes ;
243:          SIZE 1.000,2.600 ;
244:          DEFAULT " " ;
245:          FONT "MS Sans Serif", 8 ;
246:          PICTURE "99" ;
247:          VALID _rac0z0ueg() ;
248:          ERROR "MES INVALIDO (1..12)"
249:       @ 19.154,62.400 GET mes.nom_mes ;
250:          SIZE 1.000,17.800 ;
251:          DEFAULT " " ;
252:          FONT "MS Sans Serif", 8 ;
253:          PICTURE "@K" ;
254:          WHEN .F.
255:       @ 22.000,36.600 GET opc_scr ;
256:          PICTURE "@*HN \<RESPALDAR;\<EDITAR;\<SALIR" ;
257:          SIZE 1.308,13.500,0.667 ;
258:          DEFAULT 1 ;
259:          FONT "MS Sans Serif", 8 ;
260:          STYLE "B" ;
261:          VALID _rac0z0ujj()
262:       @ 7.462,3.600 GET opc_resp ;
263:          PICTURE "@*RVN \<Rol de Pagos;\<Gastos Responsab. y
Representaci¤n;\<Bonificaciones;\<Beneficios Sociales" ;
264:          SIZE 1.308,39.667,0.308 ;
265:          DEFAULT 1 ;
266:          FONT "MS Sans Serif", 8 ;
267:          STYLE "BT" ;
268:          VALID _rac0z0umb()
269:       @ 16.385,35.600 TO 20.693,87.200 ;
270:          PEN 1, 8
271:       @ 7.077,57.200 GET opc_bon ;
272:          PICTURE "@&N" ;
273:          POPUP _rac0z0u5g ;
274:          SIZE 6.923,46.800 ;
275:          DEFAULT " " ;
276:          FONT "MS Sans Serif", 8 ;
277:          STYLE "" ;
278:          VALID _rac0z0ur3()
279:       @ 6.154,54.800 TO 14.616,106.400 ;
280:          PEN 1, 8
281:
282:    IF NOT WVISIBLE("ctrl_par")
283:       ACTIVATE WINDOW ctrl_par
```

```
284:    ENDIF
285:
286:    READ CYCLE ;
287:       VALID .F.
288:
289:    RELEASE WINDOW ctrl_par
290:
291:    *
************************************************************
292:    *       *
293:    *       *              Windows Cerrando bases de datos
294:    *       *
295:    *
************************************************************
296:    *
297:
298:    IF USED("acr")
299:       SELECT acr
300:       USE
301:    ENDIF
302:
303:    IF USED("mes")
304:       SELECT mes
305:       USE
306:    ENDIF
307:
308:    IF USED("bnf")
309:       SELECT bnf
310:       USE
311:    ENDIF
312:
313:    IF USED("var")
314:       SELECT VAR
315:       USE
316:    ENDIF
317:
318:    IF USED("sst")
319:       SELECT sst
320:       USE
321:    ENDIF
322:
323:    IF USED("ctr")
324:       SELECT ctr
325:       USE
326:    ENDIF
327:
328:    SELECT (m.currarea)
329:
330:    RELEASE POPUPS _rac0z0u5g
331:
332:    #REGION 0
333:
334:    SET readborder &rborder
335:
```

```
  336:     IF m.talkstat = "ON"
  337:        SET TALK ON
  338:     ENDIF
  339:     IF m.compstat = "ON"
  340:        SET COMPATIBLE ON
  341:     ENDIF
  342:
  343:
  344:     *
**************************************************************
  345:     *        *
  346:     *        *            CTRL_RES/Windows Código de
limpieza
  347:     *        *
  348:     *
**************************************************************
  349:     *
  350:
  351:     #REGION 1
  352:     ********************
  353:
  354:
  355:     *
**************************************************************
  356:     *        *
  357:     *        * CTRL_RES/Windows Procedimientos y funciones
de soporte
  358:     *        *
  359:     *
**************************************************************
  360:     *
  361:
  362:     #REGION 1
  363:
*!*************************************************************************
************
*
  364:  *!
  365:  *!       Procedure: RESPALDAR
  366:  *!
  367:  *!           Calls: CTRLERR             (procedure in
GRAFICOS.PRG)
  368:  *!                : CALCULO.PRG
  369:  *!                : GRAFICOS.PRG
  370:  *!
  371:  *!            Uses: PERSONAL.DBF            Alias: DGP
  372:  *!                : MENSUAL.DBF             Alias: MEN
  373:  *!                : ROL.DBF                 Alias: ROL
  374:  *!                : ING_Q1.DBF              Alias: INQ1
  375:  *!                : EGR_Q1.DBF              Alias: EGQ1
  376:  *!                : LEY_LABO.DBF            Alias: LEYL
  377:  *!                : LEY_CONT.DBF            Alias: LCC
  378:  *!                : AC_IMPRTA.DBF           Alias: ACR
  379:  *!                : LISTBANC.DBF            Alias: BNC
```

```
*!                  : ESCALA.DBF              Alias: ESCL
*!                  : ING_Q2.DBF              Alias: INQ2
*!                  : EGR_Q2.DBF              Alias: EGQ2
*!                  : ING_MEN.DBF             Alias: INMN
*!                  : EGR_MEN.DBF             Alias: EGMN
*!                  : IN_EG_JU.DBF
*!                  : O_ING_GRAV.DBF
*!                  : PTMO_EMP.DBF
*!                  : GASTOS.DBF
*!                  : BONIFICA.DBF
*!
*!        CDX files: PERSONAL.CDX
*!                  : MENSUAL.CDX
*!                  : ROL.CDX
*!                  : ING_Q1.CDX
*!                  : EGR_Q1.CDX
*!                  : LEY_CONT.CDX
*!                  : AC_IMPRTA.CDX
*!                  : LISTBANC.CDX
*!                  : ESCALA.CDX
*!                  : ING_Q2.CDX
*!                  : EGR_Q2.CDX
*!                  : ING_MEN.CDX
*!                  : EGR_MEN.CDX
*!                  : IN_EG_JU.CDX
*!                  : O_ING_GRAV.CDX
*!                  : PTMO_EMP.CDX
*!                  : GASTOS.CDX
*!                  : BONIFICA.CDX
*!
*!*****************************************************************************
**************
*
      PROCEDURE respaldar
          ********************
          ON ERROR DO ctrlerr WITH ;
             ERROR(), MESSAGE(), MESSAGE(1), PROGRAM(),
LINENO()

          DO CASE
          CASE opc_resp = 1
               STOR 0 TO tipo
               IF mclave = 'SMQ1'
                  q = 'TIPO = 1'
               ELSE
                  q = 'TIPO # 1'
               ENDIF
               USE personal
               COPY TO A:personal WITH CDX;
                  FOR num_ciu = mcod_ciu

               USE mensual
```

```
430:            COPY TO A:mensual WITH CDX;
431:               FOR ciudad = mcod_ciu AND ;
432:               anio = manio AND mes = vmes
433:
434:            USE rol
435:            COPY TO A:rol WITH CDX;
436:               FOR ciudad = mcod_ciu AND;
437:               VAL(anio) = manio AND;
438:               VAL(mes)  = vmes AND &q
439:
440:            USE ing_q1
441:            COPY TO A:ing_q1 WITH CDX
442:
443:            USE egr_q1
444:            COPY TO A:egr_q1 WITH CDX
445:
446:
447:            USE ley_labo
448:            COPY TO A:ley_labo WITH CDX
449:
450:            USE ley_cont
451:            COPY TO A:ley_cont WITH CDX
452:
453:            USE ac_imprta
454:            COPY TO A:ac_imprta WITH CDX
455:
456:            USE listbanc
457:            COPY TO A:listbanc WITH CDX;
458:               FOR LEFT(clave,2) = 'SM' AND;
459:               anio = manio AND mes = vmes
460:
461:            USE escala
462:            COPY TO A:escala WITH CDX
463:            WAIT 'INSERTE EL DISKETTE NUMERO 2'
464:
465:            USE ing_q2
466:            COPY TO A:ing_q2 WITH CDX
467:
468:            USE egr_q2
469:            COPY TO A:egr_q2 WITH CDX
470:
471:            USE ing_men
472:            COPY TO A:ing_men WITH CDX;
473:               FOR VAL(SUBS(cod_rol,2,2)) = vmes;
474:               AND SUBS(cod_rol,4,2) =
RIGHT(STR(manio),2)
475:
476:            USE egr_men
477:            COPY TO A:egr_men WITH CDX;
478:               FOR VAL(SUBS(cod_rol,2,2)) = vmes;
479:               AND SUBS(cod_rol,4,2) =
RIGHT(STR(manio),2)
480:
481:            USE in_eg_ju
```

```
   482:                COPY TO A:in_eg_ju WITH CDX;
   483:                   FOR VAL(SUBS(cod_rol,2,2)) = vmes;
   484:                   AND SUBS(cod_rol,4,2) =
RIGHT(STR(manio),2)
   485:
   486:                USE o_ing_grav
   487:                COPY TO A:o_ing_grav WITH CDX ;
   488:                   FOR anio = manio AND mes = vmes
   489:
   490:                COPY FILE prgs\calculo.prg TO A:calculo.prg
   491:                COPY FILE prgs\graficos.prg TO
A:graficos.prg
   492:
   493:             ===CASE opr = 2
   494:
   495:                USE ptmo_emp
   496:                COPY TO A:ptmo_emp WITH CDX;
   497:                   FOR anio = manio AND mes = vmes
   498:
   499:                USE listbanc
   500:                COPY TO A:listbanc WITH CDX;
   501:                   FOR LEFT(clave,2) = 'PT' AND;
   502:                   anio = manio AND mes = vmes
   503:
   504:                USE gastos
   505:                COPY TO A:gastos WITH CDX;
   506:                   FOR anio = manio AND;
   507:                   mes = vmes
   508:
   509:             ===CASE opr = 4
   510:                USE personal
   511:                COPY TO A:personal WITH CDX
   512:
   513:                USE bonifica
   514:                COPY TO A:bonifica WITH CDX
   515:
   516:                USE listbanc
   517:                COPY TO A:listbanc WITH CDX;
   518:                   FOR LEFT(clave,1) = 'B'
   519:
   520:             ===CASE opr = 5
   521:                USE personal
   522:                COPY TO A:personal WITH CDX
   523:
   524:                USE bonifica
   525:                COPY TO A:bonifica WITH CDX;
   526:                   FOR LEFT(clave,1) = 'X'
   527:
   528:
   529:                USE listbanc
   530:                COPY TO A:listbanc WITH CDX;
   531:                   FOR LEFT(clave,1) = 'X'
   532:             ==ENDCASE
   533:               USE
```

```
  534: <=======RETURN
  535:
  536:          ****************
  537:
*!*************************************************************************
*************
*
  538:  *!
  539:  *!       Procedure: EDITAR
  540:  *!
  541:  *!       Called by: CREAR              (procedure in
AACT_SST.SPR)
  542:  *!
  543:  *!           Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
  544:  *!                : CABEZA             (procedure in
PERSONAL.PRG)
  545:  *!                : INGRESO            (procedure in
ACT_PTMO.SPR)
  546:  *!                : CREA_TMP           (procedure in
MO_PDLAB.SPR)
  547:  *!                : MBROW              (procedure in
MO_PDLAB.SPR)
  548:  *!                : REF_SCR            (procedure in
MO_PDLAB.SPR)
  549:  *!                : MENSAJES           (procedure in
MO_PDLAB.SPR)
  550:  *!
  551:
*!*************************************************************************
*************
*
  552:      PROCEDURE editar
  553:          ****************
  554:          _CUROBJ= OBJNUM(opc_rol)
  555: <=======RETURN
  556:
  557:          ***************
  558:
*!*************************************************************************
*************
*
  559:  *!
  560:  *!       Procedure: SALIR
  561:  *!
  562:  *!       Called by: _RAA1732TG()       (function in
AT_ESCLS.SPR)
  563:  *!                : DA_ASISG.PRG
  564:  *!                : DE_EGRQ1.PRG
  565:  *!                : DE_EGRQ2.PRG
  566:  *!                : DE_EGRMN.PRG
  567:  *!                : DE_IEJUB.PRG
  568:  *!
  569:  *!           Calls: DEAC_KEYS          (procedure in
```

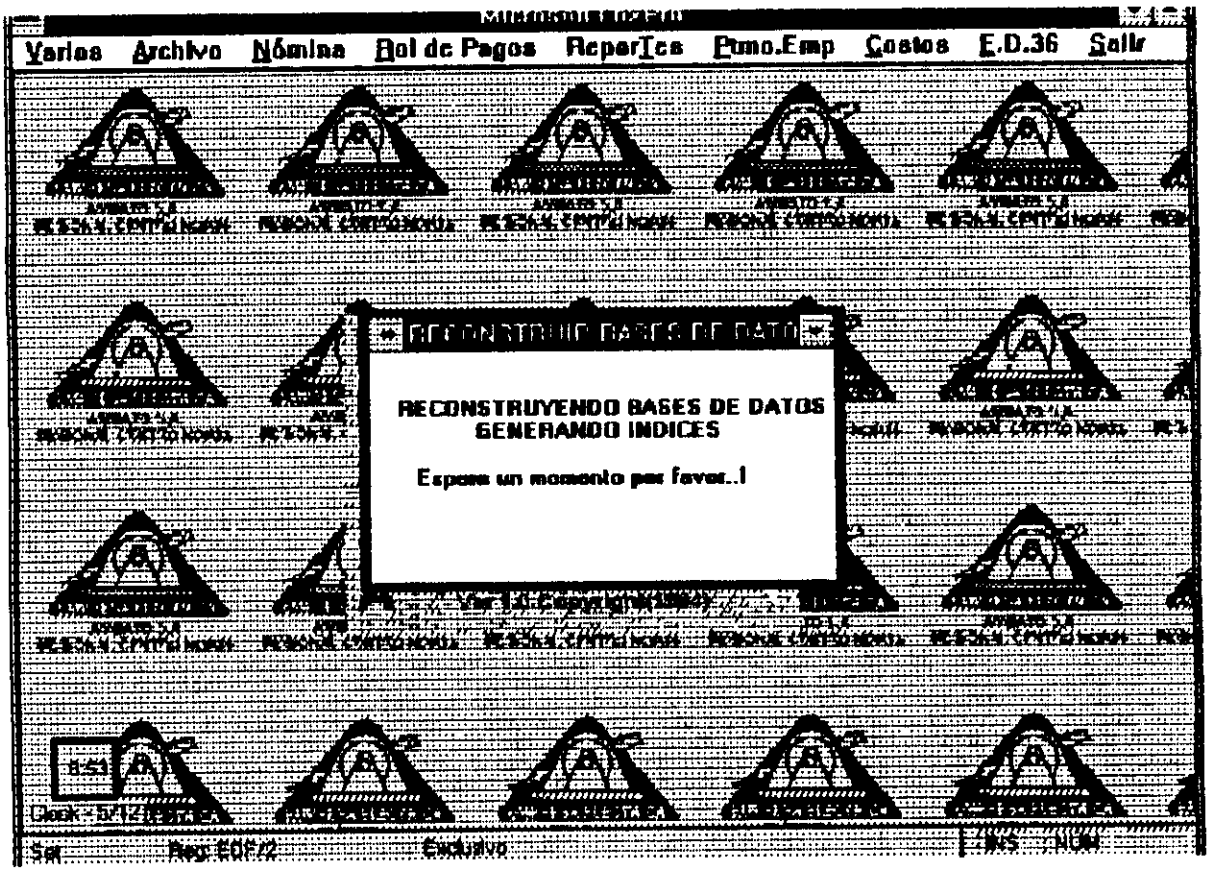
```
AACT_SST.SPR)
  570:  *!
  571:
*!*************************************************************************
************
*
  572:      PROCEDURE salir
  573:          ***************
  574:          CLEAR READ
  575: <=======RETURN
  576:          ****************
  577:
*!*************************************************************************
************
*
  578:  *!
  579:  *!       Procedure: CTRLERR
  580:  *!
  581:  *!       Called by: RESPALDAR          (procedure in
CTRL_RES.SPR)
  582:  *!
  583:  *!           Calls: OFF.PRG
  584:  *!                : SET.PRG
  585:  *!
  586:
*!*************************************************************************
************
*
  587:      PROCEDURE ctrlerr
  588:          ****************
  589:
*!*************************************************************************
************
*
  590:  *!
  591:  *!       Procedure: CTRLERR
  592:  *!
  593:  *!       Called by: RESPALDAR          (procedure in
CTRL_RES.SPR)
  594:  *!
  595:  *!           Calls: OFF.PRG
  596:  *!                : SET.PRG
  597:  *!
  598:
*!*************************************************************************
************
*
  599:      PROCEDURE ctrlerr
  600:          PARAMETER merror, mens, mens1, mprog, mn·mlφn
  601:          ? ´N·mero de error: ´ + LTRIM(STR(merror))
  602:          ? ´Mensaje de error: ´ + mens
  603:          ? ´Iφnea de cΣdigo con error: ´ + mens1
  604:          ? ´N·mero de lφnea de error: ´ +
LTRIM(STR(mn·mlφn))
```

```
  605:          ? 'Programa con error: ' + mprog
  606:        IF merror = 56
  607:           WAIT WIND 'INSERTE OTRO DISKETTE'
  608:           RETRY
  609:        ENDIF
  610: <======RETURN
  611:
  612:
  613:
  614:
  615:
  616:
  617:          *
*************************************************************
  618:          *       *
  619:          *       * _RACOZOUEG            VMES VALID
  620:          *       *
  621:          *       * Function Origin:
  622:          *       *
  623:          *       * From Platform:        Windows
  624:          *       * From Screen:          CTRL_RES,
Record Number:
16
  625:          *       * Variable:             VMES
  626:          *       * Called By:            VALID Clause
  627:          *       * Object Type:          Field
  628:          *       * Snippet Number:       1
  629:          *       *
  630:          *
*************************************************************
  631:          *
  632:
*!*************************************************************************
**************
*
  633:  *!
  634:  *!        Function: _RACOZOUEG
  635:  *!
  636:  *!      Called by: CTRL_RES.SPR
  637:  *!
  638:
*!*************************************************************************
**************
*
  639:      FUNCTION _racOzOueg     &&  VMES VALID
  640:          #REGION 1
  641:        IF vmes > 0 AND vmes < 13
  642:           STOR .T. TO vret
  643:           SELE mes
  644:           LOCATE FOR vmes = mes
  645:           STOR nom_mes TO mmes
  646:           SHOW GET mes.nom_mes
  647:        ELSE
  648:           STOR .F. TO vret
```

```
  649:         ENDIF
  650:  ========RETURN vret
  651:
  652:          *
************************************************************
  653:          *        *
  654:          *        * _RACOZOUJJ              OPC_SCR VALID
  655:          *        *
  656:          *        * Function Origin:
  657:          *        *
  658:          *        * From Platform:          Windows
  659:          *        * From Screen:            CTRL_RES,
Record Number:
18
  660:          *        * Variable:               OPC_SCR
  661:          *        * Called By:              VALID Clause
  662:          *        * Object Type:            Push Button
  663:          *        * Snippet Number:         2
  664:          *        *
  665:          *
************************************************************
  666:          *
  667:
*!*************************************************************************
**************
*
  668:  *!
  669:  *!        Function: _RACOZOUJJ
  670:  *!
  671:  *!      Called by: CTRL_RES.SPR
  672:  *!
  673:  *!           Calls: &OPC_SCR.PRG
  674:  *!
  675:
*!*************************************************************************
**************
*
  676:      FUNCTION _racOzOujj      &&  OPC_SCR VALID
  677:          #REGION 1
  678:          DO &opc_scr
  679:
  680:          *
************************************************************
  681:          *        *
  682:          *        * _RACOZOUMB              opc_resp VALID
  683:          *        *
  684:          *        * Function Origin:
  685:          *        *
  686:          *        * From Platform:          Windows
  687:          *        * From Screen:            CTRL_RES,
Record Number:
19
  688:          *        * Variable:               opc_resp
  689:          *        * Called By:              VALID Clause
```

```
  690:          *       * Object Type:          Radio Button
  691:          *       * Snippet Number:       3
  692:          *       *
  693:          *
************************************************************
  694:          *
  695:
*!*************************************************************************
*************
*
  696:  *!
  697:  *!        Function: _RACOZOUMB
  698:  *!
  699:  *!       Called by: CTRL_RES.SPR
  700:  *!
  701:
*!*************************************************************************
*************
*
  702:      FUNCTION _racOzOumb      &&  opc_resp VALID
  703:          #REGION 1
  704:          SELECT VAR
  705:          DO CASE
  706:          CASE opc_resp = 1
  707:              SET ORDER TO sueldos
  708:          CASE opc_resp = 2
  709:              SET ORDER TO gastos
  710:          CASE opc_resp = 3
  711:              SET ORDER TO bonifica
  712:          CASE opc_resp = 4
  713:              SET ORDER TO decimos
  714:          ENDCASE
  715:          SHOW GET opc_bon
  716:
  717:
  718:
  719:
  720:          *
************************************************************
  721:          *       *
  722:          *       * _RACOZOUR3            opc_BON VALID
  723:          *       *
  724:          *       * Function Origin:
  725:          *       *
  726:          *       * From Platform:        Windows
  727:          *       * From Screen:          CTRL_RES,
Record Number:
21
  728:          *       * Variable:             opc_BON
  729:          *       * Called By:            VALID Clause
  730:          *       * Object Type:          List
  731:          *       * Snippet Number:       4
  732:          *       *
  733:          *
```

```
*************************************************************
  734:          *
  735:
*!*************************************************************************
**********
*
  736:  *!
  737:  *!        Function: _RACOZOUR3
  738:  *!
  739:  *!       Called by: CTRL_RES.SPR
  740:  *!
  741:
*!*************************************************************************
**********
*
  742:      FUNCTION _rac0z0ur3      &&  opc_BON VALID
  743:          #REGION 1
  744:          STOR var.tipo TO mclave
  745:  *:  EOF: CTRL_RES.ac1
```


Varios Archivo Nómina Bol de Pagos Reportes Pmo.Emp Costos E.D.36 Salir
RECONSTRUIR BASES DE DATOS
RECONSTRUYENDO BASES DE DATOS
GENERANDO INDICES
Espere un momento por favor..!
Exclusivo

```
    1: 6
*:*************************************************************
*************
*
    2:  *:
    3:  *:          Program: C:\DOC\CTRL_RCT.PRG
    4:  *:           System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:           Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:        Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 12/05/1996 at 10:18:20
    8:  *:
    9:  *:        Called by: EMPRESA.MPR
   10:  *:
   11:  *:             Uses: SISTEMA.DBF              Alias: SST
   12:  *:                 : MESES.ORG                Alias: MES
   13:  *:                 : DEPARTAM.DBF             Alias: DPTO
   14:  *:                 : SECCION.DBF              Alias: SEC
   15:  *:                 : PUESTOS.DBF              Alias: PUES
   16:  *:                 : ESCALA.DBF               Alias: ESCL
   17:  *:                 : PERSONAL.DBF             Alias: DGP
   18:  *:                 : MENSUAL.DBF              Alias: MEN
   19:  *:                 : ROL.DBF                  Alias: ROL
   20:  *:                 : DSCTOSVA.DBF             Alias: DVAR
   21:  *:                 : ING_Q1.DBF               Alias: INQ1
   22:  *:                 : ING_Q2.DBF               Alias: INQ2
   23:  *:                 : EGR_Q1.DBF               Alias: EGQ1
   24:  *:                 : EGR_Q2.DBF               Alias: EGQ2
   25:  *:                 : ING_MEN.DBF              Alias: INMN
   26:  *:                 : EGR_MEN.DBF              Alias: EGMN
   27:  *:                 : IN_EG_JUB.DBF            Alias: IEJU
   28:  *:                 : TB_PPVAR.DBF             Alias: TBVAR
   29:  *:                 : VARIOS.DBF               Alias: VAR
   30:  *:
   31:  *:          Indexes: TIPO_JUB                 (tag in
SISTEMA.CDX)
   32:  *:                 : SIST_COD                 (tag in
SISTEMA.CDX)
   33:  *:                 : TODOS_ACT                (tag in
SISTEMA.CDX)
   34:  *:                 : TIPO_ACT                 (tag in
SISTEMA.CDX)
   35:  *:                 : TODOS                    (tag in
SISTEMA.CDX)
   36:  *:                 : NOGENER                  (tag in
MESES.CDX)
   37:  *:                 : MESES                    (tag in
MESES.CDX)
   38:  *:                 : DPTO_COD                 (tag in
DEPARTAM.CDX)
   39:  *:                 : DPTO_NOM                 (tag in
DEPARTAM.CDX)
   40:  *:                 : SISTEMA                  (tag in
DEPARTAM.CDX)
```

```
   41:  *:                    : SECC_COD                (tag in
SECCION.CDX)
   42:  *:                    : SECC_NOM                (tag in
SECCION.CDX)
   43:  *:                    : PART_PRE                (tag in
SECCION.CDX)
   44:  *:                    : SEC_REP                 (tag in
SECCION.CDX)
   45:  *:                    : COD_DPTO                (tag in
SECCION.CDX)
   46:  *:                    : PUES_COD                (tag in
PUESTOS.CDX)
   47:  *:                    : PUES_DES                (tag in
PUESTOS.CDX)
   48:  *:                    : CODIGO                  (tag in
PUESTOS.CDX)
   49:  *:                    : DENOMINACI              (tag in
PUESTOS.CDX)
   50:  *:                    : ESCALA                  (tag in
ESCALA.CDX)
   51:  *:                    : COD_PER                 (tag in
PERSONAL.CDX)
   52:  *:                    : NOMBRE                  (tag in
PERSONAL.CDX)
   53:  *:                    : CEDULA                  (tag in
PERSONAL.CDX)
   54:  *:                    : NDXCTA                  (tag in
PERSONAL.CDX)
   55:  *:                    : PER_AMB                 (tag in
PERSONAL.CDX)
   56:  *:                    : PER_PAS                 (tag in
PERSONAL.CDX)
   57:  *:                    : PER_PAL                 (tag in
PERSONAL.CDX)
   58:  *:                    : PER_JUB                 (tag in
PERSONAL.CDX)
   59:  *:                    : COD_ACT                 (tag in
PERSONAL.CDX)
   60:  *:                    : JEFATURA                (tag in
PERSONAL.CDX)
   61:  *:                    : GRU_CTA                 (tag in
PERSONAL.CDX)
   62:  *:                    : CATEGORIA               (tag in
PERSONAL.CDX)
   63:  *:                    : NUM_CIU                 (tag in
PERSONAL.CDX)
   64:  *:                    : NUM_SEC                 (tag in
PERSONAL.CDX)
   65:  *:                    : MENS_COD                (tag in
MENSUAL.CDX)
   66:  *:                    : GRUPO                   (tag in
ROL.CDX)
   67:  *:                    : COD_MES                 (tag in
ROL.CDX)
```

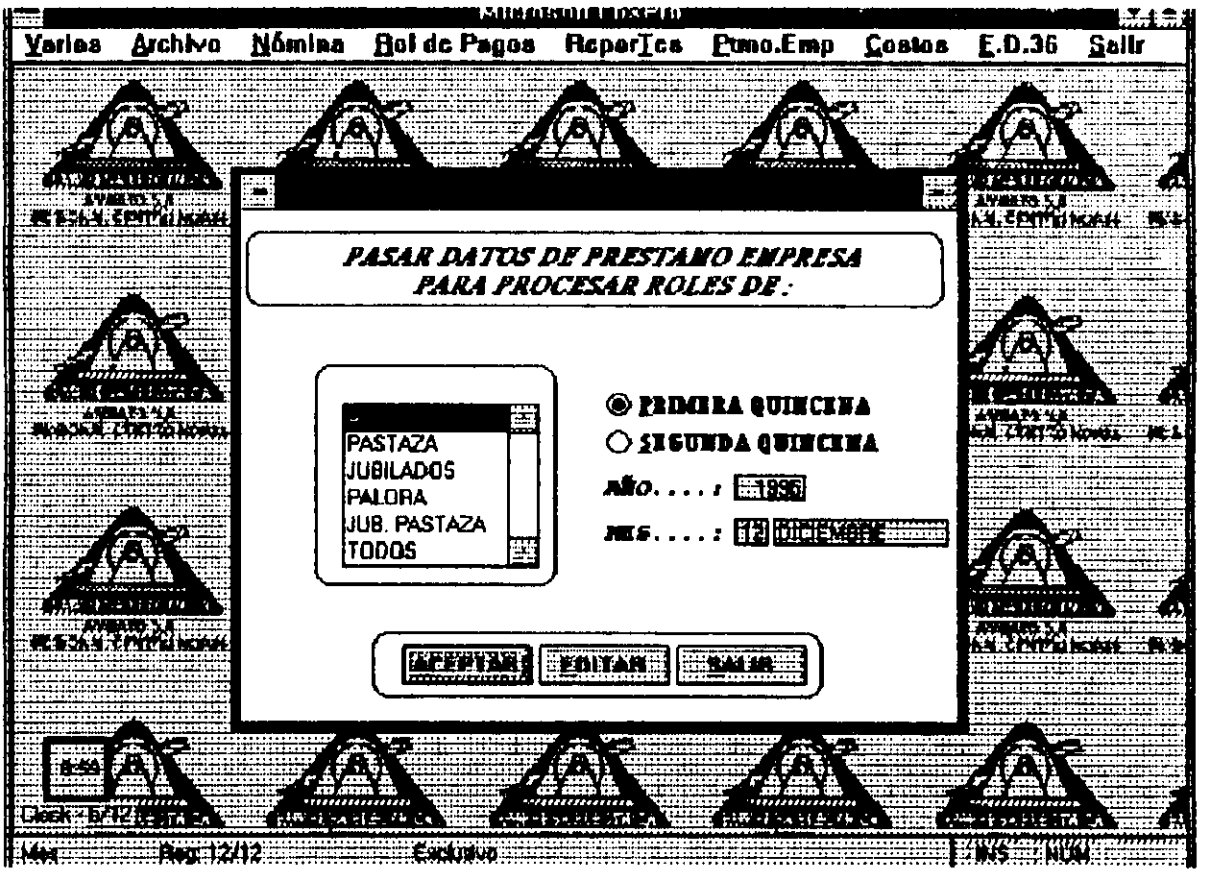
```
   68:  *:                  : AAMMCG                  (tag in
ROL.CDX)
   69:  *:                  : COD_ROL                 (tag in
ROL.CDX)
   70:  *:                  : BANE                    (tag in
DSCTOSVA.CDX)
   71:  *:                  : BFFF                    (tag in
DSCTOSVA.CDX)
   72:  *:                  : BDTE                    (tag in
DSCTOSVA.CDX)
   73:  *:                  : B12N                    (tag in
DSCTOSVA.CDX)
   74:  *:                  : BNAV                    (tag in
DSCTOSVA.CDX)
   75:  *:                  : BPEC                    (tag in
DSCTOSVA.CDX)
   76:  *:                  : VARI_COD                (tag in
DSCTOSVA.CDX)
   77:  *:                  : BON_VAR                 (tag in
DSCTOSVA.CDX)
   78:  *:                  : VAR_DSC                 (tag in
DSCTOSVA.CDX)
   79:  *:                  : CLABOT_E1               (tag in
TB_PPVAR.CDX)
   80:  *:                  : CLABOT_AS               (tag in
TB_PPVAR.CDX)
   81:  *:                  : CLABOT_E2               (tag in
TB_PPVAR.CDX)
   82:  *:                  : CLABOT_EM               (tag in
TB_PPVAR.CDX)
   83:  *:                  : CLABOT_EJ               (tag in
TB_PPVAR.CDX)
   84:  *:                  : PD_EGRQ1                (tag in
TB_PPVAR.CDX)
   85:  *:                  : PD_ASIST                (tag in
TB_PPVAR.CDX)
   86:  *:                  : PD_EGRQ2                (tag in
TB_PPVAR.CDX)
   87:  *:                  : PD_EGRMN                (tag in
TB_PPVAR.CDX)
   88:  *:                  : PD_INEGJ                (tag in
TB_PPVAR.CDX)
   89:  *:                  : BONIFIC                 (tag in
VARIOS.CDX)
   90:  *:                  : DECIMOS                 (tag in
VARIOS.CDX)
   91:  *:                  : SUELDOS                 (tag in
VARIOS.CDX)
   92:  *:                  : GASTOS                  (tag in
VARIOS.CDX)
   93:  *:                  : PRESTAMOS               (tag in
VARIOS.CDX)
   94:  *:
   95:  *:        CDX files: SISTEMA.CDX
```

```
  96:  *:                  : MESES.CDX
  97:  *:                  : DEPARTAM.CDX
  98:  *:                  : SECCION.CDX
  99:  *:                  : PUESTOS.CDX
 100:  *:                  : ESCALA.CDX
 101:  *:                  : PERSONAL.CDX
 102:  *:                  : MENSUAL.CDX
 103:  *:                  : ROL.CDX
 104:  *:                  : DSCTOSVA.CDX
 105:  *:                  : ING_Q1.CDX
 106:  *:                  : ING_Q2.CDX
 107:  *:                  : EGR_Q1.CDX
 108:  *:                  : EGR_Q2.CDX
 109:  *:                  : ING_MEN.CDX
 110:  *:                  : EGR_MEN.CDX
 111:  *:                  : IN_EG_JUB.CDX
 112:  *:                  : TB_PPVAR.CDX
 113:  *:                  : VARIOS.CDX
 114:  *:
 115:  *:       Documented 19:14:14
FoxDoc
version 3.00a
 116:
*:**************************************************************************
*************
*
 117:     IF !WEXIST('indices')
 118:        DEFINE WINDOW indices;
 119:           AT  0, 0  ;
 120:           SIZE 10,50 ;
 121:           TITLE "RECONSTRUIR BASES DE DATOS" ;
 122:           FONT "MS Sans Serif", 8 ;
 123:           FLOAT ;
 124:           NOCLOSE ;
 125:           MINIMIZE ;
 126:           SYSTEM
 127:     ENDIF
 128:      MOVE WINDOW indices CENTER
 129:      ACTI WIND indices
 130:      @ 2,3 SAY 'RECONSTRUYENDO BASES DE DATOS'          FONT
"MS Sans
Serif", 8 STYLE 'b'
 131:      @ 3,5 SAY '         GENERANDO INDICES '            FONT
"MS Sans
Serif", 8 STYLE 'b'
 132:      @ 5,5 SAY 'Espere un momento por favor..!'         FONT
"MS Sans
Serif", 8 STYLE 'b'
 133:      READ
 134:      SET TALK ON
 135:      CLOSE DATA
 136:      USE sistema ALIAS sst
 137:      INDEX ON cod_ciu TAG tipo_jub FOR tipo=2
 138:      INDEX ON cod_ciu TAG sist_cod FOR tipo#9
```

```
  139:      INDEX ON cod_ciu TAG todos_act FOR tipo#2
  140:      INDEX ON cod_ciu TAG tipo_act FOR tipo#2 .AND. tipo#9
  141:      INDEX ON cod_ciu TAG todos
  142:      USE meses.org ALIAS mes
  143:      INDEX ON mes TAG nogener FOR NOT generado
  144:      INDEX ON mes TAG meses
  145:      USE departam ALIAS dpt
  146:      INDEX ON cod_dep TAG dpto_cod
  147:      INDEX ON nombre TAG dpto_nom
  148:      INDEX ON cod_ciu TAG sistema
  149:      USE seccion ALIAS sec
  150:      INDEX ON cod_sec TAG secc_cod
  151:      INDEX ON nombre TAG secc_nom
  152:      INDEX ON cod_dpto+cod_sec+SUBSTR(cuent_cont,1,10) TAG
part_pre
  153:      INDEX ON cod_dpto+cod_sec+SUBSTR(cuent_cont,1,10) TAG
sec_rep
  154:      INDEX ON cod_dpto TAG cod_dpto
  155:      USE puestos ALIAS pto
  156:      INDEX ON cod_pues TAG pues_cod
  157:      INDEX ON descrip  TAG pues_des
  158:      INDEX ON cod_pues TAG codigo
  159:      INDEX ON descrip TAG denominaci
  160:      USE escala ALIAS esc
  161:      INDEX ON escala TAG escala
  162:      USE personal ALIAS dgp
  163:      INDEX ON cod_per TAG cod_per
  164:      INDEX ON nombre TAG nombre
  165:      INDEX ON cedula TAG cedula
  166:      INDEX ON
num_dep+num_sec+STR(categor-18)+SUBSTR(cuent_cont,1,10)
TAG part_pre
  167:      INDEX ON  STR(tipo_cts)+num_cts TAG ndxcta
  168:      INDEX ON cod_per  TAG per_amb  FOR num_ciu=1
  169:      INDEX ON cod_per  TAG per_pas FOR num_ciu=2
  170:      INDEX ON cod_per  TAG per_pal FOR num_ciu=4
  171:      INDEX ON cod_per  TAG per_jub FOR num_ciu=3 .OR.
num_ciu=5
  172:      INDEX ON cod_per  TAG cod_act FOR estado#2
  173:      INDEX ON cod_per  TAG jefatura FOR gast_rep+gast_res>0
  174:      INDEX ON
num_dep+num_sec+STR(categor-18)+SUBSTR(cuent_cont,1,10)
TAG gru_cta
  175:      INDEX ON STR(categor)+STR(rel_dep)+cargo TAG categoria
  176:      INDEX ON num_ciu TAG num_ciu
  177:      INDEX ON num_sec TAG num_sec
  178:      USE mensual ALIAS men
  179:      INDEX ON codigo TAG mens_cod
  180:      USE rol ALIAS rol
  181:      INDEX ON grupo TAG grupo
  182:      INDEX ON cod_per+SUBSTR(cod_rol,2,2) TAG cod_mes
UNIQUE
  183:      INDEX ON anio+mes+grupo TAG aammcg
  184:      INDEX ON
```

```
cod_dpto+num_sec+STR(jerarquia)+SUBSTR(cta_ctble,1,10) TAG
part_pre
  185:      INDEX ON cod_per TAG cod_per
  186:      INDEX ON nombres TAG nombre
  187:      INDEX ON cod_rol TAG cod_rol UNIQUE
  188:      INDEX ON
cod_dpto+num_sec+STR(jerarquia)+SUBSTR(cta_ctble,1,10) TAG
gru_cta
  189:     USE  dsctosva ALIAS dvar
  190:      INDEX ON sistema TAG sistema
  191:      INDEX ON codigo TAG bane FOR clave="BANE"
  192:      INDEX ON codigo TAG bfff FOR clave="BFFF"
  193:      INDEX ON codigo TAG bdte FOR clave="BDTE"
  194:      INDEX ON codigo TAG b12n FOR clave="B12N"
  195:      INDEX ON codigo TAG bnav FOR clave="BNAV"
  196:      INDEX ON codigo TAG bpec FOR clave="BPEC"
  197:      INDEX ON codigo TAG vari_cod
  198:      INDEX ON cod_dep+cod_sec+STR(jerarquia)+cta_ctble TAG
bon_var
  199:      INDEX ON codigo TAG var_dsc
  200:      INDEX ON codigo TAG cod_per
  201:     USE ing_q1 ALIAS inq1
  202:      INDEX ON codigo TAG codigo FOR LEN(ALLT(codigo)) = 6
  203:      INDEX ON nombres TAG nombre FOR LEN(ALLT(codigo)) = 6
  204:     USE ing_q2 ALIAS inq2
  205:      INDEX ON codigo TAG codigo FOR LEN(ALLT(codigo)) = 6
  206:      INDEX ON nombres TAG nombre FOR LEN(ALLT(codigo)) = 6
  207:     USE egr_q1 ALIAS egq1
  208:      INDEX ON codigo TAG codigo FOR LEN(ALLT(codigo)) = 6
  209:      INDEX ON nombres TAG nombre FOR LEN(ALLT(codigo)) = 6
  210:     USE egr_q2 ALIAS egq2
  211:      INDEX ON codigo TAG codigo FOR LEN(ALLT(codigo)) = 6
  212:      INDEX ON nombres TAG nombre FOR LEN(ALLT(codigo)) = 6
  213:     USE ing_men ALIAS inmn
  214:      INDEX ON codigo TAG codigo FOR LEN(ALLT(codigo)) = 6
  215:      INDEX ON nombres TAG nombre FOR LEN(ALLT(codigo)) = 6
  216:     USE egr_men ALIAS egmn
  217:      INDEX ON codigo TAG codigo FOR LEN(ALLT(codigo)) = 6
  218:      INDEX ON nombres TAG nombre FOR LEN(ALLT(codigo)) = 6
  219:     USE in_eg_jub ALIAS ieju
  220:      INDEX ON codigo TAG codigo FOR LEN(ALLT(codigo)) = 6
  221:      INDEX ON nombres TAG nombre FOR LEN(ALLT(codigo)) = 6
  222:     USE tb_ppvar ALIAS tbvar
  223:     SET UNIQUE ON
  224:      INDEX ON clave+boton TAG clabot_e1 FOR clave = 'EQ1'
UNIQUE
  225:      INDEX ON clave+boton TAG clabot_as FOR clave = 'AST'
UNIQUE
  226:      INDEX ON clave+boton TAG clabot_e2 FOR clave = 'EQ2'
UNIQUE
  227:      INDEX ON clave+boton TAG clabot_em FOR clave = 'EMN'
UNIQUE
  228:      INDEX ON clave+boton TAG clabot_ej FOR clave = 'EJU'
UNIQUE
```

```
  229:      SET UNIQUE OFF
  230:      INDEX ON clave+boton TAG pd_egrq1  FOR clave = 'EQ1'
  231:      INDEX ON clave+boton TAG pd_asist  FOR clave = 'AST'
  232:      INDEX ON clave+boton TAG pd_egrq2  FOR clave = 'EQ2'
  233:      INDEX ON clave+boton TAG pd_egrmn  FOR clave = 'EMN'
  234:      INDEX ON clave+boton TAG pd_inegj  FOR clave = 'EJU'
  235:
  236:      USE varios ALIAS VAR
  237:      INDEX ON tipo TAG bonific FOR  SUBSTR(tipo,1,1)="B"
  238:      INDEX ON tipo TAG decimos FOR  SUBSTR(tipo,1,1)="X"
  239:      INDEX ON tipo TAG sueldos FOR  SUBSTR(tipo,1,1)="S"
  240:      INDEX ON tipo TAG gastos  FOR  SUBSTR(tipo,1,1)="G"
  241:      INDEX ON tipo TAG prestamos  FOR LEFT(tipo,3)="PTM"
  242:
  243:      CLOSE DATABASES
  244:      CLEAR
  245:      ?CHR(7)
  246:      ?CHR(7)
  247:      @2,10 SAY 'ROCESO  CONCLUIDO' FONT "MS Sans Serif", 8
STYLE 'b'
  248:      STOR '' TO mcont
  249:      @ 5,15 GET mcont;
  250:         PICTURE "@* |<Aceptar" ;
  251:         SIZE 1.2,20 FONT 'R',10 STYLE "BI"
  252:      READ MODAL
  253:      RELE WINDO indices
  254:      SET DISPLAY TO VGA50
  255: <===RETURN
  256:  *:  EOF: CTRL_RCT.act
```


Varios Archivo Nómina Rol de Pagos Reportes Pmo.Emp Costos E.D.36 Salir
PASAR DATOS DE PRESTAMO EMPRESA
PARA PROCESAR ROLES DE :
PASTAZA
JUBILADOS
PALORA
JUB. PASTAZA
TODOS
PRIMERA QUINCENA
SEGUNDA QUINCENA
Año. . . . : 1996
Mes. . . . : 12 DICIEMBRE
ACEPTAR
EDITAR
SALIR
Mes
Reg 12/12
Exclusivo
INS
NUM

```
    1:
*:*************************************************************
*************
*
    2:  *:
    3:  *: Procedure file: C:\DOC\VREG_PTE.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 03/10/1995 at 20:28:02
    8:  *:
    9:  *:  Procs & Fncts: _R4517V8Y1()
   10:  *:                : _R4517V91Y()
   11:  *:                : _R4517V96Q()
   12:  *:                : ACEPTAR
   13:  *:                : PREPARAR
   14:  *:                : VAL_PRO()
   15:  *:
   16:  *:          Set by: EMPRESA.MPR
   17:  *:
   18:  *:           Calls: _R4517V8Y1()        (function in
VREG_PTE.SPR)
   19:  *:                : _R4517V91Y()        (function in
VREG_PTE.SPR)
   20:  *:                : _R4517V96Q()        (function in
VREG_PTE.SPR)
   21:  *:
   22:  *:       Documented 19:14:15
FoxDoc
version 3.00a
   23:
*:*************************************************************
*************
*
   24:      *
************************************************************
   25:      *        *
   26:      *        * 10/03/1995    .         PR_PDPE.SPR
 20:28:00
   27:      *        *
   28:      *
************************************************************
   29:      *        *
   30:      *        * Author's Name
   31:      *        *
   32:      *        * Copyright (c) 1995 Company Name
   33:      *        * Address
   34:      *        * City,     Zip
   35:      *        *
   36:      *        * Description:
   37:      *        * This program was automatically generated by
GENSCRN.
   38:      *        *
```

```
   39:      *
************************************************************
   40:
   41:
   42:      *
************************************************************
   43:      *        *
   44:      *        *          PR_PDPE/Windows Setup Code - SECTION
1
   45:      *        *
   46:      *
************************************************************
   47:      *
   48:
   49:      #REGION 1
   50:      **********
   51:      REGIONAL opc_scr, opc_rol, opc_qui, mmes, vmes, manio,
mquincena,
mrol,;
   52:         mopc_qui,mfquin,mq
   53:      STOR ' ' TO opc_scr, mmes,opc_rol, fquin
   54:      STOR 0 TO opc_qui,mopc_qui
   55:      STOR 1 TO mrol,mq
   56:      STOR .F. TO mquincena
   57:      STOR YEAR(DATE()) TO manio
   58:      STOR MONTH(DATE()) TO vmes
   59:
   60:      **********
   61:
   62:      #REGION 0
   63:      REGIONAL m.currarea, m.talkstat, m.compstat
   64:
   65:    ┌─IF SET("TALK") = "ON"
   66:    │    SET TALK OFF
   67:    │    m.talkstat = "ON"
   68:    ├─ELSE
   69:    │    m.talkstat = "OFF"
   70:    └─ENDIF
   71:      m.compstat = SET("COMPATIBLE")
   72:      SET COMPATIBLE FOXPLUS
   73:
   74:      m.rborder = SET("READBORDER")
   75:      SET readborder ON
   76:
   77:      m.currarea = SELECT()
   78:
   79:
   80:      *
************************************************************
   81:      *        *
   82:      *        *       PR_PDPE/Windows Databases, Indexes,
Relations
   83:      *        *
   84:      *
```

```
**********************************************************
   85:      *
   86:
   87:     IF USED("mes")
   88:        SELECT mes
   89:        SET ORDER TO 0
   90:     ELSE
   91:        SELECT 0
   92:        USE (LOCFILE("dbfs\meses.org","DBF","Where is
meses?"));
   93:           AGAIN ALIAS mes ;
   94:           ORDER 0
   95:     ENDIF
   96:
   97:     IF USED("sst")
   98:        SELECT sst
   99:        SET ORDER TO TAG "todos"
  100:     ELSE
  101:        SELECT 0
  102:        USE (LOCFILE("dbfs\sistema.dbf","DBF","Where is
sistema?"));
  103:           AGAIN ALIAS sst ;
  104:           ORDER TAG "todos"
  105:     ENDIF
  106:
  107:     IF USED("pte")
  108:        SELECT pte
  109:        SET ORDER TO TAG "cod_per"
  110:     ELSE
  111:        SELECT 0
  112:        USE (LOCFILE("dbfs\ptmo_emp.dbf","DBF","Where is
ptmo_emp?"));
  113:           AGAIN ALIAS pte ;
  114:           ORDER TAG "cod_per"
  115:     ENDIF
  116:
  117:     IF USED("men")
  118:        SELECT men
  119:        SET ORDER TO TAG "mens_cod"
  120:     ELSE
  121:        SELECT 0
  122:        USE (LOCFILE("dbfs\mensual.dbf","DBF","Where is
mensual?"));
  123:           AGAIN ALIAS men ;
  124:           ORDER TAG "mens_cod"
  125:     ENDIF
  126:
  127:     SELECT men
  128:
  129:     SET RELATION OFF INTO pte
  130:     SET RELATION TO men.codigo INTO pte ADDITIVE
  131:
  132:     SELECT men
  133:
```

```
*
*************************************************************
*        *
*        *                    Windows Window definitions
*        *
*
*************************************************************
*

IF NOT WEXIST("pampdr") ;
      OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.PJX" ;
      OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.SCX" ;
      OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.MNX" ;
      OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.PRG" ;
      OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.FRX" ;
      OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.QPR"
   DEFINE WINDOW pampdr ;
      AT  0.000, 0.000  ;
      SIZE 21.923,77.000 ;
      FONT "MS Sans Serif", 8 ;
      FLOAT ;
      NOCLOSE ;
      MINIMIZE ;
      SYSTEM
   MOVE WINDOW pampdr CENTER
ENDIF


*
*************************************************************
*        *
*        *          PR_PDPE/Windows Setup Code - SECTION
2
*        *
*
*************************************************************
*

#REGION 1
**********
SELE mes
LOCATE FOR mes = vmes
STOR nom_mes TO mmes
SELE sst
STOR sst.ciudad TO opc_rol

#REGION 1
DEFINE POPUP _r4517v8pl ;
   PROMPT FIELD sst.ciudad ;
   SCROLL

*
*************************************************************
*        *
```

```
182:      *        *
183:      *        *
184:      *
**********************************************************
185:      *
186:
187:      #REGION 1
188:      IF WVISIBLE("pampdr")
189:         ACTIVATE WINDOW pampdr SAME
190:      ELSE
191:         ACTIVATE WINDOW pampdr NOSHOW
192:      ENDIF
193:      @ 13.231,39.200 SAY "MES....:"  ;
194:         FONT "Courier New", 10 ;
195:         STYLE "BIT"
196:      @ 11.308,39.200 SAY "ADO....:"  ;
197:         FONT "Courier New", 10 ;
198:         STYLE "BIT"
199:      @ 6.692,8.000 TO 16.154,34.000 ;
200:         PEN 1, 8 ;
201:         STYLE "16"
202:      @ 18.154,14.200 TO 21.000,62.600 ;
203:         PEN 1, 8 ;
204:         STYLE "16"
205:      @ 7.692,39.400 GET opc_qui ;
206:         PICTURE "@*RVN |<PRIMERA QUINCENA;|<SEGUNDA
QUINCENA" ;
207:         SIZE 1.308,25.667,0.308 ;
208:         DEFAULT 1 ;
209:         FONT "MS Serif", 8 ;
210:         STYLE "BT" ;
211:         WHEN mquincena
212:      @ 11.385,53.600 GET manio ;
213:         SIZE 1.000,6.600 ;
214:         DEFAULT " " ;
215:         FONT "MS Sans Serif", 8 ;
216:         PICTURE "9999" ;
217:         VALID manio > 1990 AND manio < 2999 ;
218:         ERROR 'ADO INVALIDO ' ;
219:         DISABLE
220:      @ 13.308,53.600 GET vmes ;
221:         SIZE 1.000,2.600 ;
222:         DEFAULT " " ;
223:         FONT "MS Sans Serif", 8 ;
224:         PICTURE "99" ;
225:         VALID _r4517v8y1() ;
226:         ERROR 'MES INVALIDO (1..12)' ;
227:         DISABLE
228:      @ 18.692,17.400 GET opc_scr ;
229:         PICTURE "@*HN |<ACEPTAR;|<EDITAR;|<SALIR" ;
230:         SIZE 1.769,11.667,0.667 ;
231:         DEFAULT 1 ;
232:         FONT "MS Sans Serif", 8 ;
233:         STYLE "B" ;
```

```
234:         VALID _r4517v91y() ;
235:         DISABLE
236:      @ 1.385,9.800 SAY "PASAR DATOS DE PRESTAMO EMPRESA" +
CHR(13) + ;
237:         "PARA PROCESAR ROLES DE :"  ;
238:         SIZE 2.000,41.000, 0.000 ;
239:         PICTURE "@I" ;
240:         FONT "MS Serif", 10 ;
241:         STYLE "BIT"
242:      @ 0.923,0.600 TO 4.154,75.600 ;
243:         PEN 1, 8 ;
244:         STYLE "16"
245:      @ 13.308,57.800 GET mes.nom_mes ;
246:         SIZE 1.000,17.800 ;
247:         DEFAULT " " ;
248:         FONT "MS Sans Serif", 8 ;
249:         PICTURE "@K" ;
250:         WHEN .F.
251:      @ 8.308,11.000 GET opc_rol ;
252:         PICTURE "@&N" ;
253:         POPUP _r4517v8pl ;
254:         SIZE 6.923,20.800 ;
255:         DEFAULT " " ;
256:         FONT "MS Sans Serif", 8 ;
257:         STYLE "" ;
258:         VALID _r4517v96q()
259:
260:      IF NOT WVISIBLE("pampdr")
261:         ACTIVATE WINDOW pampdr
262:      ENDIF
263:
264:      READ CYCLE ;
265:         DEACTIVATE .F.
266:
267:      RELEASE WINDOW pampdr
268:
269:      *
************************************************************
270:      *        *
271:      *        *                  Windows Closing Databases
272:      *        *
273:      *
************************************************************
274:      *
275:
276:      IF USED("mes")
277:         SELECT mes
278:         USE
279:      ENDIF
280:
281:      IF USED("sst")
282:         SELECT sst
283:         USE
284:      ENDIF
```

```
285:
286:     IF USED("pte")
287:        SELECT pte
288:        USE
289:     ENDIF
290:
291:     IF USED("men")
292:        SELECT men
293:        USE
294:     ENDIF
295:
296:     SELECT (m.currarea)
297:
298:     RELEASE POPUPS _r4517v8pl
299:
300:     #REGION 0
301:
302:     SET readborder &rborder
303:
304:     IF m.talkstat = "ON"
305:        SET TALK ON
306:     ENDIF
307:     IF m.compstat = "ON"
308:        SET COMPATIBLE ON
309:     ENDIF
310:
311:
312:     *
************************************************************
313:     *        *
314:     *        *              PR_PDPE/Windows Cleanup Code
315:     *        *
316:     *
************************************************************
317:     *
318:
319:     #REGION 1
320:     *****************
321:
322:
323:     *
************************************************************
324:     *        *
325:     *        *  PR_PDPE/Windows Supporting Procedures and
Functions
326:     *        *
327:     *
************************************************************
328:     *
329:
330:     #REGION 1
331:
*!*************************************************************
**************
```

```
  332:  *!
  333:  *!       Procedure: ACEPTAR
  334:  *!
  335:  *!           Calls: VAL_PRO()          (function in
VREG_PTE.SPR)
  336:  *!                : PREPARAR           (procedure in
VREG_PTE.SPR)
  337:  *!                : DA_ASISG.PRG
  338:  *!                : DE_EGRQ1.PRG
  339:  *!                : DE_EGRQ2.PRG
  340:  *!                : DE_EGRMN.PRG
  341:  *!                : DE_IEJUB.PRG
  342:  *!                : EGR_VARI.PRG
  343:  *!                : ROL_Q1.SPR
  344:  *!                : ROL_Q2.SPR
  345:  *!                : ROL_MEN.SPR
  346:  *!                : ROL_JUB.SPR
  347:  *!                : ACT_PTMO.SPR
  348:  *!                : MO_PDLAB.SPR
  349:  *!
  350:  *!    Report Forms: VAL_NEGA.FRX
  351:  *!
  352:
*!*************************************************************************
*************
*
  353:      PROCEDURE aceptar
  354:          *****************
  355:          SHOW GETS DISABLE
  356:          SHOW GET opc_scr,2 ENABLE
  357:          SHOW GET opc_scr,3 ENABLE
  358:          SELECT sst
  359:          SET ORDER TO sist_cod
  360:          IF opc_rol = 'TODOS'
  361:              SCAN
  362:                  STOR ' ' TO fquin
  363:              DO CASE
  364:              CASE sst.tipo = 0
  365:                      STOR ' and q2 = OPC_QUI' TO fquin
  366:                      STOR 2 TO mq
  367:              CASE sst.tipo = 1
  368:                      STOR ' and q2 = OPC_QUI' TO fquin
  369:                      STOR 3 TO mq
  370:              CASE sst.tipo = 2
  371:                      STOR ' and q2 = OPC_QUI' TO fquin
  372:                      STOR 4 TO mq
  373:              ENDCASE
  374:                  IF val_pro()
  375:                      DO preparar WITH sst.cod_ciu
  376:                  ENDIF
  377:              ENDSCAN
  378:          ELSE
  379:              SEEK mrol
  380:              STOR ' ' TO fquin
```

```
  381:          ³=DO CASE
  382:          ³═══CASE sst.tipo = 0
  383:          ³      ┌─IF opc_qui = 1
  384:          ³      │     STOR ' and q1 = OPC_QUI' TO fquin
  385:          ³      │     STOR 1 TO mq
  386:          ³      ├─ELSE
  387:          ³      │     STOR ' and q2 = OPC_QUI' TO fquin
  388:          ³      │     STOR 2 TO mq
  389:          ³      └─ENDIF
  390:          ³═══CASE sst.tipo = 1
  391:          ³        STOR ' and q2 = OPC_QUI' TO fquin
  392:          ³        STOR 3 TO mq
  393:          ³═══CASE sst.tipo = 2
  394:          ³        STOR ' and q2 = OPC_QUI' TO fquin
  395:          ³        STOR 4 TO mq
  396:          ³=ENDCASE
  397:          ┌─IF val_pro()
  398:          │    DO preparar WITH sst.cod_ciu
  399:          └─ENDIF
  400:        └─ENDIF
  401:          SET ORDER TO todos
  402:          SHOW GET opc_rol
  403: <═══════RETURN
  404:
  405:          ******************
  406:
*!*************************************************************************
**************
*
  407:  *!
  408:  *!      Procedure: PREPARAR
  409:  *!
  410:  *!      Called by: ACEPTAR              (procedure in
VREG_PTE.SPR)
  411:  *!
  412:  *!           Uses: MENSUAL.DBF              Alias: MEN
  413:  *!               : MTMP.DBF                 Alias: MTMP
  414:  *!
  415:  *!        Indexes: MTMP_COD                 (tag in
MENSUAL.CDX)
  416:  *!               : MTMP.CDX
  417:  *!
  418:  *!      CDX files: MENSUAL.CDX
  419:  *!
  420:
*!*************************************************************************
**************
*
  421:      PROCEDURE preparar
  422:          ******************
  423:          PARAMETER mnum_ciu
  424:          malias = ALIAS()
  425:          SELE pte
  426:          SET FILTER TO VAL(pte.ciudad) = mnum_ciu AND
```

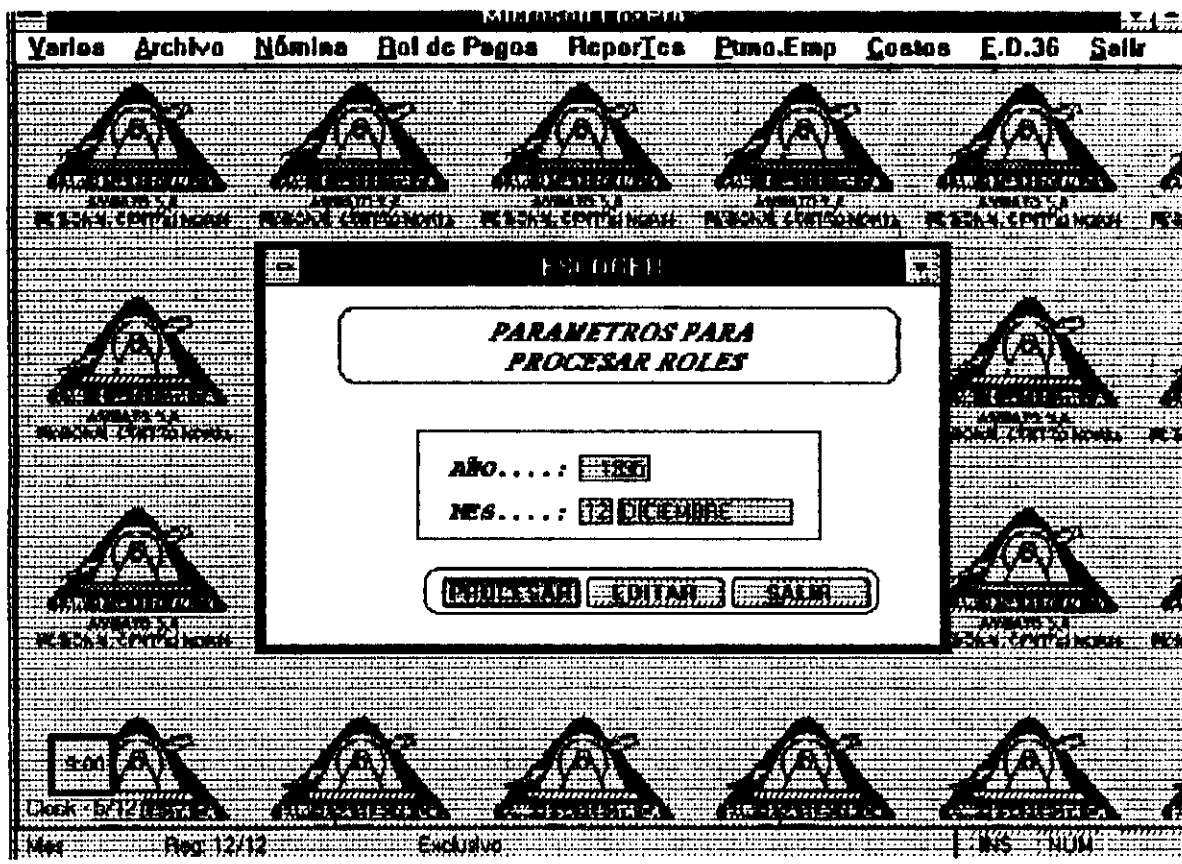
```
pte.anio = manio
AND;
  427:              pte.mes = vmes AND pte.quincena = mq
  428:           BROW
  429:           SELE men
  430:           SET FILTER TO men.ciudad = mnum_ciu AND manio =
men.anio AND
vmes = men.mes
  431:        ╔═SCAN
  432:        ┃  ┌─IF opc_qui = 1
  433:        ┃  │     REPLACE men.pre_emp1 WITH pte.v_cuotas
  434:        ┃  ├─ELSE
  435:        ┃  │     REPLACE men.pre_emp2 WITH pte.v_cuotas
  436:        ┃  └─ENDIF
  437:        ╚═ENDSCAN
  438:           SET FILTER TO
  439:           SELE &malias
  440: <═══════RETURN
  441:
  442:           ****************
  443:
*!*************************************************************************
*************
*
  444:  *!
  445:  *!        Function: VAL_PRO
  446:  *!
  447:  *!       Called by: ACEPTAR              (procedure in
VREG_PTE.SPR)
  448:  *!
  449:
*!*************************************************************************
*************
*
  450:      FUNCTION val_pro
  451:           ****************
  452:           malias = ALIAS()
  453:           SELE men
  454:           LOCATE FOR  sst.cod_ciu = men.ciudad AND sst.tipo
=
men.tip_rol;
  455:              AND manio = men.anio AND ;
  456:              (men.pre_emp1 # 0 OR men.pre_emp2 # 0) AND;
  457:              vmes = men.mes &fquin
  458:           vret = .T.
  459:        ┌─IF FOUND()
  460:        │  ┌─IF NOT WEXIST('VARIOS') && o definir en el
principal
  461:        │  │     DEFINE WINDOW varios FROM 0,0 TO 7,35 TITLE
'ALERTA ..!'
SYSTEM
  462:        │  └─ENDIF
  463:        │    ACTIVATE WINDOW varios
  464:        │    MOVE WINDOW varios CENTER
```

```
 465:          @ 1,2 SAY 'El Archivo a Procesar '+sst.ciudad
 466:          @ 2,6 SAY 'Ya ha sido generado.'
 467:          STOR 1 TO opc_ac
 468:          @ 4,5 GET opc_ac PICT '@*H |<PASAR;|<CANCELAR';
 469:             SIZE 1.308,12.500,0.308 ;
 470:             DEFAULT 1 ;
 471:             FONT "MS Serif", 8 ;
 472:             STYLE "BT"
 473:          READ MODAL
 474:          RELE WIND varios
 475:          IF opc_ac = 2
 476:             STOR .F. TO vret
 477:          ENDIF
 478:       ENDIF
 479:       SELECT &malias
 480: <=====RETURN    vret
 481:       ****************
 482:
*!*************************************************************
*************
*
 483:  *!
 484:  *!      Procedure: EDITAR
 485:  *!
 486:  *!      Called by: CREAR              (procedure in
AACT_SST.SPR)
 487:  *!
 488:  *!          Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
 489:  *!               : CABEZA             (procedure in
PERSONAL.PRG)
 490:  *!               : INGRESO            (procedure in
ACT_PTMO.SPR)
 491:  *!               : CREA_TMP           (procedure in
MO_PDLAB.SPR)
 492:  *!               : MBROW              (procedure in
MO_PDLAB.SPR)
 493:  *!               : REF_SCR            (procedure in
MO_PDLAB.SPR)
 494:  *!               : MENSAJES           (procedure in
MO_PDLAB.SPR)
 495:  *!
 496:
*!*************************************************************
*************
*
 497:     PROCEDURE editar
 498:       ****************
 499:       SHOW GETS DISABLE
 500:       SHOW GET opc_rol ENABLE
 501: <=====RETURN
 502:       ***************
 503:
*!*************************************************************
```

```
*************
*
  504:  *!
  505:  *!       Procedure: SALIR
  506:  *!
  507:  *!       Called by: _RAA1732TG()        (function in
AT_ESCLS.SPR)
  508:  *!                : DA_ASISG.PRG
  509:  *!                : DE_EGRQ1.PRG
  510:  *!                : DE_EGRQ2.PRG
  511:  *!                : DE_EGRMN.PRG
  512:  *!                : DE_IEJUB.PRG
  513:  *!
  514:  *!           Calls: DEAC_KEYS           (procedure in
AACT_SST.SPR)
  515:  *!
  516:
*!*********************************************************************
*************
*
  517:      PROCEDURE salir
  518:          ***************
  519:          SET DELE ON
  520:          CLEAR READ
  521: <=======RETURN
  522:
  523:          *
************************************************************
  524:          *        *
  525:          *        * _R4517V8Y1           VMES VALID
  526:          *        *
  527:          *        * Function Origin:
  528:          *        *
  529:          *        * From Platform:       Windows
  530:          *        * From Screen:         PR_PDPE,      Record
Number:   13
  531:          *        * Variable:            VMES
  532:          *        * Called By:           VALID Clause
  533:          *        * Object Type:         Field
  534:          *        * Snippet Number:      1
  535:          *        *
  536:          *
************************************************************
  537:          *
  538:
*!*********************************************************************
*************
*
  539:  *!
  540:  *!        Function: _R4517V8Y1
  541:  *!
  542:  *!       Called by: VREG_PTE.SPR
  543:  *!
  544:
```

```
*!*************************************************************
*************
*
  545:      FUNCTION _r4517v8y1      &&  VMES VALID
  546:          #REGION 1
  547:          IF vmes > 0 AND vmes < 13
  548:              STOR .T. TO vret
  549:              SELE mes
  550:              LOCATE FOR vmes = mes
  551:              STOR nom_mes TO mmes
  552:              SHOW GET mes.nom_mes
  553:          ELSE
  554:              STOR .F. TO vret
  555:          ENDIF
  556: <=======RETURN vret
  557:
  558:          *
************************************************************
  559:          *        *
  560:          *        * _R4517V91Y           OPC_SCR VALID
  561:          *        *
  562:          *        * Function Origin:
  563:          *        *
  564:          *        * From Platform:       Windows
  565:          *        * From Screen:         PR_PDPE,      Record
Number:   14
  566:          *        * Variable:            OPC_SCR
  567:          *        * Called By:           VALID Clause
  568:          *        * Object Type:         Push Button
  569:          *        * Snippet Number:      2
  570:          *        *
  571:          *
************************************************************
  572:          *
  573:
*!*************************************************************
*************
*
  574:  *!
  575:  *!       Function: _R4517V91Y
  576:  *!
  577:  *!      Called by: VREG_PTE.SPR
  578:  *!
  579:  *!           Calls: &OPC_SCR.PRG
  580:  *!
  581:
*!*************************************************************
*************
*
  582:      FUNCTION _r4517v91y      &&  OPC_SCR VALID
  583:          #REGION 1
  584:          DO &opc_scr
  585:
  586:          *
```

```
*************************************************************
  587:          *        *
  588:          *        * _R4517V96Q            OPC_ROL VALID
  589:          *        *
  590:          *        * Function Origin:
  591:          *        *
  592:          *        * From Platform:        Windows
  593:          *        * From Screen:          PR_PDPE,      Record
Number:   18
  594:          *        * Variable:             OPC_ROL
  595:          *        * Called By:            VALID Clause
  596:          *        * Object Type:          List
  597:          *        * Snippet Number:       3
  598:          *        *
  599:          *
*************************************************************
  600:          *
  601:
*!*********************************************************************
*************
*
  602:   *!
  603:   *!        Function: _R4517V96Q
  604:   *!
  605:   *!       Called by: VREG_PTE.SPR
  606:   *!
  607:
*!*********************************************************************
*************
*
  608:      FUNCTION _r4517v96q      &&  OPC_ROL VALID
  609:          #REGION 1
  610:          STOR sst.cod_ciu TO mrol,mnum_ciu
  611:          STOR 0 TO opc_qui
  612:          STOR .F. TO mquincena
  613:        ı=DO CASE
  614:        ã═══CASE opc_rol = 'TODOS'
  615:        │         STOR 2 TO opc_qui
  616:        │         STOR .F. TO mquincena
  617:        ã═══CASE sst.tipo = 0
  618:        │         SHOW GET opc_qui ENABL
  619:        │         STOR 1 TO opc_qui, mopc_qui
  620:        │         STOR .T. TO mquincena
  621:        E=ENDCASE
  622:          SHOW GET opc_qui
  623:          SHOW GET manio ENABLE
  624:          SHOW GET vmes ENABLE
  625:          SHOW GET opc_scr ENABLE
  626:   *:  EOF: VREG_PTE.acl
```


Varios Archivo Nóminas Bol de Pagos ReporTes Ptmo.Emp Costos E.D.36 Salir
PARAMETROS PARA
PROCESAR ROLES
Año. . . . :
Mes. . . . :
PROCESAR
EDITAR
SALIR

```
    1:
*:*************************************************************
*************
*
    2:  *:
    3:  *: Procedure file: C:\DOC\VREG_ENG.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 11/03/1996 at 20:33:34
    8:  *:
    9:  *:  Procs & Fncts: _R8N182DWV()
   10:  *:                : _R8N182E0Z()
   11:  *:                : PROCESAR
   12:  *:
   13:  *:          Set by: EMPRESA.MPR
   14:  *:
   15:  *:           Calls: _R8N182DWV()        (function in
VREG_ENG.SPR)
   16:  *:                : _R8N182E0Z()        (function in
VREG_ENG.SPR)
   17:  *:
   18:  *:       Documented 19:14:16
FoxDoc
version 3.00a
   19:
*:*************************************************************
*************
*
   20:      *
************************************************************
   21:      *        *
   22:      *        * 11/03/1996           CTRL_ENE.SPR
 20:33:33
   23:      *        *
   24:      *
************************************************************
   25:      *        *
   26:      *        * Nombre del autor
   27:      *        *
   28:      *        * Copyright (c) 1996 Nombre de la empresa
   29:      *        * Direcci#n
   30:      *        * Ciudad,     C.P.
   31:      *        * Pa#s
   32:      *        *
   33:      *        * Description:
   34:      *        * Este programa lo ha generado autom8ticamente
GENSCRN.
   35:      *        *
   36:      *
************************************************************
   37:
   38:
```

```
   39:      *
************************************************************
   40:      *        *
   41:      *        *  CTRL_ENE/Windows Código de configuración -
SECCION 1
   42:      *        *
   43:      *
************************************************************
   44:      *
   45:
   46:      #REGION 1
   47:      **********
   48:      REGIONAL opc_scr, opc_rol, opc_qui, mmes, vmes, manio,
mquincena,
mrol,;
   49:         mopc_qui,mfquin
   50:      STOR ' ' TO opc_scr, mmes,opc_rol, fquin
   51:      STOR 0 TO opc_qui,mopc_qui
   52:      STOR 1 TO mrol
   53:      STOR .F. TO mquincena
   54:      STOR YEAR(DATE()) TO manio
   55:      STOR MONTH(DATE()) TO vmes
   56:
   57:      **********
   58:
   59:      #REGION 0
   60:      REGIONAL m.currarea, m.talkstat, m.compstat
   61:
   62:    ┌─IF SET("TALK") = "ON"
   63:    │    SET TALK OFF
   64:    │    m.talkstat = "ON"
   65:    ├─ELSE
   66:    │    m.talkstat = "OFF"
   67:    └─ENDIF
   68:      m.compstat = SET("COMPATIBLE")
   69:      SET COMPATIBLE FOXPLUS
   70:
   71:      m.rborder = SET("READBORDER")
   72:      SET readborder ON
   73:
   74:      m.currarea = SELECT()
   75:
   76:
   77:      *
************************************************************
   78:      *        *
   79:      *        *  CTRL_ENE/Windows Bases de datos, índices,
relaciones
   80:      *        *
   81:      *
************************************************************
   82:      *
   83:
   84:    ┌─IF USED("ctr")
```

```
   85:     |    SELECT ctr
   86:     |    SET ORDER TO TAG "clave"
   87:     |-ELSE
   88:     |    SELECT 0
   89:     |    USE (LOCFILE("dbfs\ctrl_rol.dbf","DBF","Buscar
ctrl_rol?"));
   90:     |        AGAIN ALIAS ctr ;
   91:     |        ORDER TAG "clave"
   92:     |-ENDIF
   93:
   94:     |-IF USED("mes")
   95:     |    SELECT mes
   96:     |    SET ORDER TO 0
   97:     |-ELSE
   98:     |    SELECT 0
   99:     |    USE (LOCFILE("dbfs\meses.org","DBF","Buscar
meses?"));
  100:     |        AGAIN ALIAS mes ;
  101:     |        ORDER 0
  102:     |-ENDIF
  103:
  104:     |-IF USED("sst")
  105:     |    SELECT sst
  106:     |    SET ORDER TO TAG "sist_cod"
  107:     |-ELSE
  108:     |    SELECT 0
  109:     |    USE (LOCFILE("dbfs\sistema.dbf","DBF","Buscar
sistema?"));
  110:     |        AGAIN ALIAS sst ;
  111:     |        ORDER TAG "sist_cod"
  112:     |-ENDIF
  113:
  114:      SELECT ctr
  115:
  116:
  117:      *
************************************************************
  118:      *        *
  119:      *        *              Windows Definiciones de ventana
  120:      *        *
  121:      *
************************************************************
  122:      *
  123:
  124:     |-IF NOT WEXIST("ctrl_par") ;
  125:     |        OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.PJX" ;
  126:     |        OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.SCX" ;
  127:     |        OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.MNX" ;
  128:     |        OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.PRG" ;
  129:     |        OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.FRX" ;
  130:     |        OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.QPR"
  131:     |    DEFINE WINDOW ctrl_par ;
  132:     |        AT  0.000, 0.000  ;
  133:     |        SIZE 15.615,72.600 ;
```

```
  134:     |        TITLE "ESCOGER" ;
  135:     |        FONT "MS Sans Serif", 8 ;
  136:     |        FLOAT ;
  137:     |        NOCLOSE ;
  138:     |        MINIMIZE ;
  139:     |        SYSTEM
  140:     |     MOVE WINDOW ctrl_par CENTER
  141:     |ENDIF
  142:
  143:
  144:     *
************************************************************
  145:     *       *
  146:     *       *  CTRL_ENE/Windows Código de configuración -
SECCION 2
  147:     *       *
  148:     *
************************************************************
  149:     *
  150:
  151:     #REGION 1
  152:     **********
  153:     SELE mes
  154:     LOCATE FOR mes = vmes
  155:     STOR nom_mes TO mmes
  156:
  157:     *
************************************************************
  158:     *       *
  159:     *       *        CTRL_ENE/Windows Distribución de
pantalla
  160:     *       *
  161:     *
************************************************************
  162:     *
  163:
  164:     #REGION 1
  165:  ┌─IF WVISIBLE("ctrl_par")
  166:  │     ACTIVATE WINDOW ctrl_par SAME
  167:  ├─ELSE
  168:  │     ACTIVATE WINDOW ctrl_par NOSHOW
  169:  └─ENDIF
  170:     @ 9.231,20.000 SAY "MES....:" ;
  171:        FONT "Courier New", 10 ;
  172:        STYLE "BIT"
  173:     @ 7.308,20.000 SAY "ADO....:" ;
  174:        FONT "Courier New", 10 ;
  175:        STYLE "BIT"
  176:     @ 12.308,17.200 TO 14.385,66.200 ;
  177:        PEN 1, 8 ;
  178:        STYLE "16"
  179:     @ 7.385,34.400 GET manio ;
  180:        SIZE 1.000,6.600 ;
  181:        DEFAULT " " ;
```

```
182:        FONT "MS Sans Serif", 8 ;
183:        PICTURE "9999" ;
184:        VALID manio > 1990 AND manio < 2999 ;
185:        ERROR 'ADO INVALIDO '
186:     @ 9.308,34.400 GET vmes ;
187:        SIZE 1.000,2.600 ;
188:        DEFAULT " " ;
189:        FONT "MS Sans Serif", 8 ;
190:        PICTURE "99" ;
191:        VALID _r8n182dwv() ;
192:        ERROR 'MES INVALIDO (1..12)'
193:     @ 12.769,19.200 GET opc_scr ;
194:        PICTURE "@*HN |<PROCESAR;|<EDITAR;|<SALIR" ;
195:        SIZE 1.308,12.333,0.667 ;
196:        DEFAULT 1 ;
197:        FONT "MS Sans Serif", 8 ;
198:        STYLE "B" ;
199:        VALID _r8n182e0z()
200:     @ 1.385,22.200 SAY "PARAMETROS PARA    " + CHR(13) + ;
201:        "PROCESAR ROLES" ;
202:        SIZE 2.000,23.143, 0.000 ;
203:        PICTURE "@I" ;
204:        FONT "MS Serif", 10 ;
205:        STYLE "BIT"
206:     @ 0.923,8.000 TO 4.154,68.400 ;
207:        PEN 1, 8 ;
208:        STYLE "16"
209:     @ 9.308,38.600 GET mes.nom_mes ;
210:        SIZE 1.000,17.800 ;
211:        DEFAULT " " ;
212:        FONT "MS Sans Serif", 8 ;
213:        PICTURE "@K" ;
214:        WHEN .F.
215:     @ 6.231,16.400 TO 11.077,60.000 ;
216:        PEN 1, 8
217:
218:     IF NOT WVISIBLE("ctrl_par")
219:        ACTIVATE WINDOW ctrl_par
220:     ENDIF
221:
222:     READ CYCLE ;
223:        VALID .F.
224:
225:     RELEASE WINDOW ctrl_par
226:
227:     *
*************************************************************
228:     *        *
229:     *        *              Windows Cerrando bases de datos
230:     *        *
231:     *
*************************************************************
232:     *
233:
```

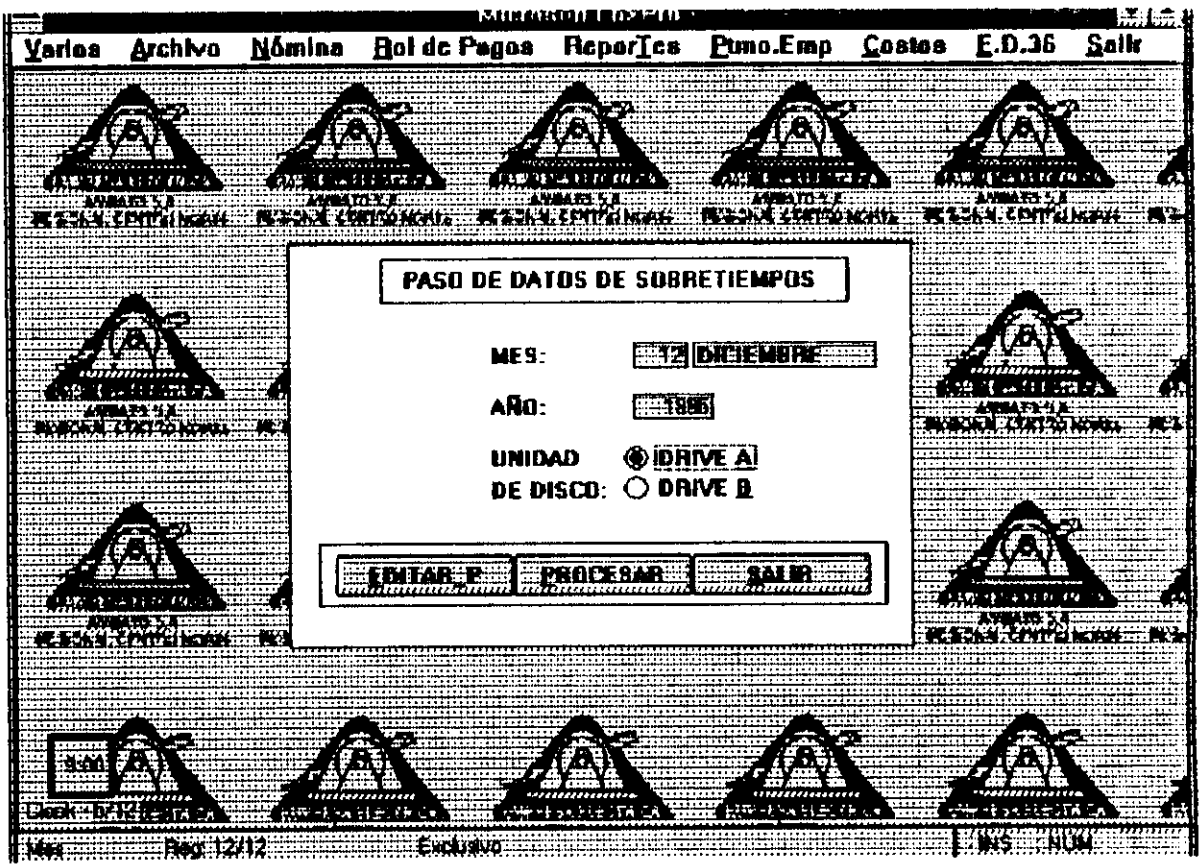
```
234:    IF USED("ctr")
235:       SELECT ctr
236:       USE
237:    ENDIF
238:
239:    IF USED("mes")
240:       SELECT mes
241:       USE
242:    ENDIF
243:
244:    IF USED("sst")
245:       SELECT sst
246:       USE
247:    ENDIF
248:
249:    SELECT (m.currarea)
250:
251:
252:    #REGION 0
253:
254:    SET readborder &rborder
255:
256:    IF m.talkstat = "ON"
257:       SET TALK ON
258:    ENDIF
259:    IF m.compstat = "ON"
260:       SET COMPATIBLE ON
261:    ENDIF
262:
263:
264:    *
************************************************************
265:    *        *
266:    *        *           CTRL_ENE/Windows Código de
limpieza
267:    *        *
268:    *
************************************************************
269:    *
270:
271:    #REGION 1
272:    *****************
273:
274:
275:    *
************************************************************
276:    *        *
277:    *        * CTRL_ENE/Windows Procedimientos y funciones
de soporte
278:    *        *
279:    *
************************************************************
280:    *
281:
```

```
  282:      #REGION 1
  283:
*!*************************************************************************
*************
*
  284:  *!
  285:  *!       Procedure: PROCESAR
  286:  *!
  287:  *!           Calls: MEDIDOR.PRG
  288:  *!                : CALCULO.PRG
  289:  *!                : VALIDAR()          (function in
CTRL_GTO.SPR)
  290:  *!                : ABRIR_ARCH         (procedure in
PROC_DMO.SPR)
  291:  *!                : GN_MPL             (procedure in
PROC_DMO.SPR)
  292:  *!                : GN_PDMO            (procedure in
PROC_DMO.SPR)
  293:  *!                : GN_CTA_RUB         (procedure in
PROC_DMO.SPR)
  294:  *!                : GN_ASI_CON         (procedure in
PROC_DMO.SPR)
  295:  *!
  296:  *!            Uses: SOBRETIE.DBF
  297:  *!                : &MARCHI                Alias: CP
  298:  *!                : DSCTOSVA.DBF           Alias: DVAR
  299:  *!                : PERSONAL.DBF           Alias: DGP
  300:  *!                : LISTBANC.DBF           Alias: BNC
  301:  *!                : TOT_ROL.DBF            Alias: TROL
  302:  *!                : PTMO_EMP.DBF
  303:  *!                : CTAS_CPE.DBF
  304:  *!                : TMP_ACPE.DBF           Alias: TMPPE
  305:  *!                : TMP_PE2.DBF
  306:  *!                : ROL_IBS.DBF
  307:  *!                : ROL_IBST.DBF
  308:  *!                : DBF.DBF
  309:  *!
  310:  *!         Indexes: CODIGO                 (tag in
PUESTOS.CDX)
  311:  *!                : CODIGO.CDX
  312:  *!                : &TMP_MPL..CDX
  313:  *!
  314:  *!       CDX files: DSCTOSVA.CDX
  315:  *!                : PERSONAL.CDX
  316:  *!                : LISTBANC.CDX
  317:  *!                : PTMO_EMP.CDX
  318:  *!                : ROL_IBS.CDX
  319:  *!
  320:
*!*************************************************************************
*************
*
  321:      PROCEDURE procesar
  322:          *****************
```

```
  323:          DO medidor WITH manio, vmes
  324: <=======RETURN
  325:          ****************
  326:
*!*****************************************************************
*************
*
  327:  *!
  328:  *!       Procedure: EDITAR
  329:  *!
  330:  *!       Called by: CREAR              (procedure in
AACT_SST.SPR)
  331:  *!
  332:  *!           Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
  333:  *!                : CABEZA             (procedure in
PERSONAL.PRG)
  334:  *!                : INGRESO            (procedure in
ACT_PTMO.SPR)
  335:  *!                : CREA_TMP           (procedure in
MO_PDLAB.SPR)
  336:  *!                : MBROW              (procedure in
MO_PDLAB.SPR)
  337:  *!                : REF_SCR            (procedure in
MO_PDLAB.SPR)
  338:  *!                : MENSAJES           (procedure in
MO_PDLAB.SPR)
  339:  *!
  340:
*!*****************************************************************
*************
*
  341:      PROCEDURE editar
  342:          ****************
  343:          *SHOW GETS DISABLE
  344:          _CUROBJ= OBJNUM(manio)
  345: <=======RETURN
  346:          ***************
  347:
*!*****************************************************************
*************
*
  348:  *!
  349:  *!       Procedure: SALIR
  350:  *!
  351:  *!       Called by: _RAA1732TG()       (function in
AT_ESCLS.SPR)
  352:  *!                : DA_ASISG.PRG
  353:  *!                : DE_EGRQ1.PRG
  354:  *!                : DE_EGRQ2.PRG
  355:  *!                : DE_EGRMN.PRG
  356:  *!                : DE_IEJUB.PRG
  357:  *!
  358:  *!           Calls: DEAC_KEYS          (procedure in
```

```
AACT_SST.SPR)
  359:   *!
  360:
*!*************************************************************
*************
*
  361:      PROCEDURE salir
  362:          ***************
  363:          CLEAR READ
  364: <=======RETURN
  365:
  366:          *
************************************************************
  367:          *         *
  368:          *         * _R8N182DWV            VMES VALID
  369:          *         *
  370:          *         * Function Origin:
  371:          *         *
  372:          *         * From Platform:        Windows
  373:          *         * From Screen:          CTRL_ENE,
Record Number:
9
  374:          *         * Variable:             VMES
  375:          *         * Called By:            VALID Clause
  376:          *         * Object Type:          Field
  377:          *         * Snippet Number:       1
  378:          *         *
  379:          *
************************************************************
  380:          *
  381:
*!*************************************************************
*************
*
  382:   *!
  383:   *!        Function: _R8N182DWV
  384:   *!
  385:   *!       Called by: VREG_ENG.SPR
  386:   *!
  387:
*!*************************************************************
*************
*
  388:      FUNCTION _r8n182dwv      &&  VMES VALID
  389:          #REGION 1
  390:         IF vmes > 0 AND vmes < 13
  391:             STOR .T. TO vret
  392:             SELE mes
  393:             LOCATE FOR vmes = mes
  394:             STOR nom_mes TO mmes
  395:             SHOW GET mes.nom_mes
  396:         ELSE
  397:             STOR .F. TO vret
  398:         ENDIF
```

```
  399: <=======RETURN vret
  400:
  401:          *
*************************************************************
  402:          *        *
  403:          *        * _R8N182E0Z            OPC_SCR VALID
  404:          *        *
  405:          *        * Function Origin:
  406:          *        *
  407:          *        * From Platform:        Windows
  408:          *        * From Screen:          CTRL_ENE,
Record Number:
10
  409:          *        * Variable:             OPC_SCR
  410:          *        * Called By:            VALID Clause
  411:          *        * Object Type:          Push Button
  412:          *        * Snippet Number:       2
  413:          *        *
  414:          *
*************************************************************
  415:          *
  416:
*!*****************************************************************
*************
*
  417:  *!
  418:  *!        Function: _R8N182E0Z
  419:  *!
  420:  *!      Called by: VREG_ENG.SPR
  421:  *!
  422:  *!           Calls: &OPC_SCR.PRG
  423:  *!
  424:
*!*****************************************************************
*************
*
  425:      FUNCTION _r8n182e0z      &&  OPC_SCR VALID
  426:          #REGION 1
  427:          DO &opc_scr
  428:  *:  EOF: VREG_ENG.acl
```


Varios Archivo Nómina Bol de Pagos Reportes Pmo.Emp Costos E.D.36 Salir
PASO DE DATOS DE SOBRETIEMPOS
MES: 12 DICIEMBRE
AÑO: 1995
UNIDAD DE DISCO: DRIVE A DRIVE B
EDITAR P PROCESAR SALIR
Mes Reg 12/12 Exclusivo INS NUM

```
    1:
*:*************************************************************************
*************
*
    2:  *:
    3:  *: Procedure file: C:\DOC\VREG_AST.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 23/01/1996 at  3:01:12
    8:  *:
    9:  *:  Procs & Fncts: _R7B06GZCE()
   10:  *:                : _R7B06GZIE()
   11:  *:                : _R7B06GZLT()
   12:  *:                : _R7B06H0GP()
   13:  *:                : _R7B06H0JZ()
   14:  *:                : EDITAR_P
   15:  *:
   16:  *:          Set by: EMPRESA.MPR
   17:  *:
   18:  *:           Calls: _R7B06GZCE()        (function in
VREG_AST.SPR)
   19:  *:                : _R7B06GZIE()        (function in
VREG_AST.SPR)
   20:  *:                : _R7B06GZLT()        (function in
VREG_AST.SPR)
   21:  *:                : _R7B06H0GP()        (function in
VREG_AST.SPR)
   22:  *:                : _R7B06H0JZ()        (function in
VREG_AST.SPR)
   23:  *:
   24:  *:       Documented 19:14:17
FoxDoc
version 3.00a
   25:
*:*************************************************************************
*************
*
   26:      *
************************************************************
   27:      *        *
   28:      *        * 23/01/1996           PASASIST.SPR
 03:01:09
   29:      *        *
   30:      *
************************************************************
   31:      *        *
   32:      *        * Nombre del autor
   33:      *        *
   34:      *        * Copyright (c) 1996 Nombre de la empresa
   35:      *        * Direcci#n
   36:      *        * Ciudad,     C.P.
   37:      *        * Pa#s
```

```
   38:      *        *
   39:      *        * Description:
   40:      *        * Este programa lo ha generado automßticamente
GENSCRN.
   41:      *        *
   42:      *
************************************************************
   43:
   44:    ı=DO CASE
   45:    ã═══CASE _WINDOWS OR _MAC   && no MAC records in screen
   46:    │
   47:    │
   48:    │        *
************************************************************
   49:    │        *        *
   50:    │        *        *  PASASIST/Windows C‡digo de
configuraci‡n - SECCION
1
   51:    │        *        *
   52:    │        *
************************************************************
   53:    │        *
   54:    │
   55:    │        #REGION 1
   56:    │        **********
   57:    │        REGIONAL manio, mmes, mdrive, opc_scr, marchi
   58:    │        SET TALK OFF
   59:    │        SET SAFETY ON
   60:    │        STOR MONTH(DATE()) TO mmes
   61:    │        STOR YEAR(DATE()) TO manio
   62:    │        STOR 1 TO mdrive
   63:    │        STOR ' ' TO opc_scr,marchi
   64:    │        **********
   65:    │
   66:    │        #REGION 0
   67:    │        REGIONAL m.currarea, m.talkstat, m.compstat
   68:    │
   69:    │       ┌─IF SET("TALK") = "ON"
   70:    │       │     SET TALK OFF
   71:    │       │     m.talkstat = "ON"
   72:    │       ├─ELSE
   73:    │       │     m.talkstat = "OFF"
   74:    │       └─ENDIF
   75:    │        m.compstat = SET("COMPATIBLE")
   76:    │        SET COMPATIBLE FOXPLUS
   77:    │
   78:    │        m.rborder = SET("READBORDER")
   79:    │        SET readborder ON
   80:    │
   81:    │        m.currarea = SELECT()
   82:    │
   83:    │
   84:    │        *
************************************************************
```

```
   85:          *        *
   86:          *        *  PASASIST/Windows Bases de datos,
Ýndices,
relaciones
   87:          *        *
   88:          *
*********************************************************
   89:          *
   90:
   91:          IF USED("men")
   92:             SELECT men
   93:             SET ORDER TO TAG "mens_cod"
   94:          ELSE
   95:             SELECT 0
   96:             USE (LOCFILE("dbfs\mensual.dbf","DBF","Buscar
mensual?"));
   97:                AGAIN ALIAS men ;
   98:                ORDER TAG "mens_cod"
   99:          ENDIF
  100:
  101:          IF USED("mes")
  102:             SELECT mes
  103:             SET ORDER TO 0
  104:          ELSE
  105:             SELECT 0
  106:             USE (LOCFILE("dbfs\meses.org","DBF","Buscar
meses?"));
  107:                AGAIN ALIAS mes ;
  108:                ORDER 0
  109:          ENDIF
  110:
  111:          SELECT men
  112:
  113:
  114:          *
*********************************************************
  115:          *        *
  116:          *        *              Windows Definiciones de
ventana
  117:          *        *
  118:          *
*********************************************************
  119:          *
  120:
  121:          IF NOT WEXIST("sobretie") ;
  122:                OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.PJX" ;
  123:                OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.SCX" ;
  124:                OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.MNX" ;
  125:                OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.PRG" ;
  126:                OR UPPER(WTITLE("SOBRETIE")) ==
```

```
"SOBRETIE.FRX" ;
  127:     |       |       OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.QPR"
  128:     |       |    DEFINE WINDOW sobretie ;
  129:     |       |       AT  0.000, 0.000  ;
  130:     |       |       SIZE 17.385,55.500 ;
  131:     |       |       FONT "MS Sans Serif", 8 ;
  132:     |       |       STYLE "B" ;
  133:     |       |       NOFLOAT ;
  134:     |       |       NOCLOSE ;
  135:     |       |       SHADOW ;
  136:     |       |       NOMINIMIZE
  137:     |       |    MOVE WINDOW sobretie CENTER
  138:     |       ENDIF
  139:     |
  140:     |
  141:     |       *
************************************************************
  142:     |       *       *
  143:     |       *       *  PASASIST/Windows Código de
configuración - SECCION
2
  144:     |       *       *
  145:     |       *
************************************************************
  146:     |       *
  147:
  148:     |       #REGION 1
  149:     |       **********
  150:     |       DEFINE WIND asist FROM 1,1 TO 6,40
  151:     |       SELE mes
  152:     |       LOCATE FOR mes = mmes
  153:     |
  154:     |       *
************************************************************
  155:     |       *       *
  156:     |       *       *        PASASIST/Windows Distribución
de pantalla
  157:     |       *       *
  158:     |       *
************************************************************
  159:     |       *
  160:
  161:     |       #REGION 1
  162:     |       IF WVISIBLE("sobretie")
  163:     |          ACTIVATE WINDOW sobretie SAME
  164:     |       ELSE
  165:     |          ACTIVATE WINDOW sobretie NOSHOW
  166:     |       ENDIF
  167:     |       @ 1.077,10.167 SAY "PASO DE DATOS DE
SOBRETIEMPOS" ;
  168:     |          FONT "MS Sans Serif", 8 ;
  169:     |          STYLE "B"
  170:     |       @ 0.615,8.167 TO 2.615,50.167 ;
```

```
171:            PEN 1, 8
172:         @ 4.385,18.000 SAY "MES:" ;
173:            FONT "MS Sans Serif", 8 ;
174:            STYLE "B"
175:         @ 4.385,31.167 GET mmes ;
176:            SIZE 1.000,4.800 ;
177:            RANGE 1, 12 ;
178:            DEFAULT " " ;
179:            FONT "MS Sans Serif", 8 ;
180:            PICTURE "99" ;
181:            VALID _r7b06gzce() ;
182:            ERROR "MES ERRONEO [1..12]"
183:         @ 6.615,18.000 SAY "ADO:" ;
184:            FONT "MS Sans Serif", 8 ;
185:            STYLE "B"
186:         @ 6.615,31.167 GET manio ;
187:            SIZE 1.000,7.600 ;
188:            DEFAULT " " ;
189:            FONT "MS Sans Serif", 8 ;
190:            PICTURE "9999" ;
191:            VALID manio > 1990 AND manio < 2999 ;
192:            ERROR "AÑO INVALIDO"
193:         @ 8.846,18.000 SAY "UNIDAD" ;
194:            FONT "MS Sans Serif", 8 ;
195:            STYLE "B"
196:         @ 10.231,18.000 SAY "DE DISCO:" ;
197:            FONT "MS Sans Serif", 8 ;
198:            STYLE "B"
199:         @ 8.692,30.000 GET mdrive ;
200:            PICTURE "@*RVN DRIVE |<A;DRIVE |<B" ;
201:            SIZE 1.308,12.833,0.000 ;
202:            DEFAULT 1 ;
203:            FONT "MS Sans Serif", 8 ;
204:            STYLE "B" ;
205:            VALID _r7b06gzie()
206:         @ 13.538,4.000 GET opc_scr ;
207:            PICTURE "@*HN |<EDITAR_P;|<PROCESAR;|<SALIR"
;
208:            SIZE 1.769,16.000,0.000 ;
209:            DEFAULT 1 ;
210:            FONT "MS Sans Serif", 8 ;
211:            STYLE "B" ;
212:            VALID _r7b06gzlt() ;
213:            DISABLE
214:         @ 13.000,2.500 TO 15.769,53.500 ;
215:            PEN 1, 8
216:         @ 4.385,36.500 GET mes.nom_mes ;
217:            SIZE 1.000,15.667 ;
218:            DEFAULT " " ;
219:            FONT "MS Sans Serif", 8 ;
220:            STYLE "B" ;
221:            PICTURE "@K" ;
222:            WHEN .F.
223:
```

```
224:          IF NOT WVISIBLE("sobretie")
225:             ACTIVATE WINDOW sobretie
226:          ENDIF
227:
228:          READ CYCLE ;
229:             DEACTIVATE .F.
230:
231:          RELEASE WINDOW sobretie
232:
233:          *
************************************************************
234:          *        *
235:          *        *              Windows Cerrando bases de
datos
236:          *        *
237:          *
************************************************************
238:          *
239:
240:          IF USED("men")
241:             SELECT men
242:             USE
243:          ENDIF
244:
245:          IF USED("mes")
246:             SELECT mes
247:             USE
248:          ENDIF
249:
250:          SELECT (m.currarea)
251:
252:
253:          #REGION 0
254:
255:          SET readborder &rborder
256:
257:          IF m.talkstat = "ON"
258:             SET TALK ON
259:          ENDIF
260:          IF m.compstat = "ON"
261:             SET COMPATIBLE ON
262:          ENDIF
263:
264:
265:          *
************************************************************
266:          *        *
267:          *        *           PASASIST/Windows Código de
limpieza
268:          *        *
269:          *
************************************************************
270:          *
271:
```

```
  317:
  318:          IF USED("sbt")
  319:              SELECT sbt
  320:              SET ORDER TO 0
  321:          ELSE
  322:              SELECT 0
  323:              USE
(LOCFILE("sobretie\sobretie.dbf","DBF","Buscar
sobretie?"));
  324:                  AGAIN ALIAS sbt ;
  325:                  ORDER 0
  326:          ENDIF
  327:
  328:
  329:      *
************************************************************
  330:      *       *
  331:      *       *                MS-DOS Definiciones de
ventana
  332:      *       *
  333:      *
************************************************************
  334:      *
  335:
  336:          IF NOT WEXIST("sobretie") ;
  337:                  OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.PJX" ;
  338:                  OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.SCX" ;
  339:                  OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.MNX" ;
  340:                  OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.PRG" ;
  341:                  OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.FRX" ;
  342:                  OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.QPR"
  343:              DEFINE WINDOW sobretie ;
  344:                  FROM INT((SROW()-16)/2),INT((SCOL()-55)/2)
;
  345:                  TO
INT((SROW()-16)/2)+15,INT((SCOL()-55)/2)+54 ;
  346:                  NOFLOAT ;
  347:                  NOCLOSE ;
  348:                  SHADOW ;
  349:                  NOMINIMIZE ;
  350:                  COLOR SCHEME 1
  351:          ENDIF
  352:
  353:
  354:      *
************************************************************
  355:      *       *
  356:      *       *   PASASIST/MS-DOS Código de
```

```
  405:          DEFAULT 1 ;
  406:          VALID _r7b06h0gp()
  407:       @ 12,3 GET opc_scr ;
  408:          PICTURE "@*HN |<EDITAR_P;|<PROCESAR;|<SALIR"
;
  409:          SIZE 1,16,0 ;
  410:          DEFAULT 1 ;
  411:          VALID _r7b06h0jz() ;
  412:          DISABLE
  413:       @ 11,1 TO 13,52
  414:
  415:       IF NOT WVISIBLE("sobretie")
  416:          ACTIVATE WINDOW sobretie
  417:       ENDIF
  418:
  419:       READ CYCLE ;
  420:          DEACTIVATE .F.
  421:
  422:       RELEASE WINDOW sobretie
  423:
  424:       *
************************************************************
  425:       *        *
  426:       *        *              MS-DOS Cerrando bases de
datos
  427:       *        *
  428:       *
************************************************************
  429:       *
  430:
  431:       IF USED("sbt")
  432:          SELECT sbt
  433:          USE
  434:       ENDIF
  435:
  436:       SELECT (m.currarea)
  437:
  438:
  439:       #REGION 0
  440:       IF m.talkstat = "ON"
  441:          SET TALK ON
  442:       ENDIF
  443:       IF m.compstat = "ON"
  444:          SET COMPATIBLE ON
  445:       ENDIF
  446:
  447:
  448:       *
************************************************************
  449:       *        *
  450:       *        *           PASASIST/MS-DOS Código de
limpieza
  451:       *        *
  452:       *
```

```
*************************************************************
  453:          *
  454:
  455:          #REGION 1
  456:          ******************
  457:
  458:    ENDCASE
  459:
  460:
  461:      *
*************************************************************
  462:      *        *
  463:      *        * _R7B06GZCE            MMES VALID
  464:      *        *
  465:      *        * Function Origin:
  466:      *        *
  467:      *        * From Platform:        Windows
  468:      *        * From Screen:          PASASIST,      Record
Number:     7
  469:      *        * Variable:             MMES
  470:      *        * Called By:            VALID Clause
  471:      *        * Object Type:          Field
  472:      *        * Snippet Number:       1
  473:      *        *
  474:      *
*************************************************************
  475:      *
  476:
*!*************************************************************************
*************
*
  477:   *!
  478:   *!        Function: _R7B06GZCE
  479:   *!
  480:   *!       Called by: VREG_AST.SPR
  481:   *!
  482:
*!*************************************************************************
*************
*
  483:      FUNCTION _r7b06gzce     &&  MMES VALID
  484:          #REGION 1
  485:          SELE mes
  486:          LOCATE FOR mes = mmes
  487:          SHOW GET mes.nom_mes
  488:
  489:          *
*************************************************************
  490:          *        *
  491:          *        * _R7B06GZIE            MDRIVE VALID
  492:          *        *
  493:          *        * Function Origin:
  494:          *        *
  495:          *        * From Platform:        Windows
```

```
  496:          *        * From Screen:          PASASIST,
Record Number:
12
  497:          *        * Variable:             MDRIVE
  498:          *        * Called By:            VALID Clause
  499:          *        * Object Type:          Radio Button
  500:          *        * Snippet Number:       2
  501:          *        *
  502:          *
************************************************************
  503:          *
  504:
*!*************************************************************************
*************
*
  505:   *!
  506:   *!        Function: _R7B06GZIE
  507:   *!
  508:   *!       Called by: VREG_AST.SPR
  509:   *!
  510:
*!*************************************************************************
*************
*
  511:      FUNCTION _r7b06gzie     &&  MDRIVE VALID
  512:          #REGION 1
  513:          marchi = IIF(mdrive = 1,'A:\DASIST','B:\DASIST')
  514:          SHOW GET opc_scr ENABLE
  515:          ACTI WIND asist
  516:          ?CHR(7)
  517:          @2,2 SAY ' Inserte el disco en el Drive '+ IIF(
mdrive =
1,'A:','B:')
  518:          READ
  519:          DEAC WIND asist
  520:
  521:
  522:          *
************************************************************
  523:          *        *
  524:          *        * _R7B06GZLT            OPC_SCR VALID
  525:          *        *
  526:          *        * Function Origin:
  527:          *        *
  528:          *        * From Platform:        Windows
  529:          *        * From Screen:          PASASIST,
Record Number:
13
  530:          *        * Variable:             OPC_SCR
  531:          *        * Called By:            VALID Clause
  532:          *        * Object Type:          Push Button
  533:          *        * Snippet Number:       3
  534:          *        *
  535:          *
```

```
**********************************************************
  536:          *
  537:
*!*************************************************************
*************
*
  538:  *!
  539:  *!       Function: _R7B06GZLT
  540:  *!
  541:  *!      Called by: VREG_AST.SPR
  542:  *!
  543:  *!           Calls: &OPC_SCR.PRG
  544:  *!
  545:
*!*************************************************************
*************
*
  546:      FUNCTION _r7b06gzlt      &&  OPC_SCR VALID
  547:          #REGION 1
  548:          DO &opc_scr
  549:
  550:          *
 **********************************************************
  551:          *       *
  552:          *       * _R7B06H0GP              MDRIVE VALID
  553:          *       *
  554:          *       * Function Origin:
  555:          *       *
  556:          *       * From Platform:          MS-DOS
  557:          *       * From Screen:            PASASIST,
 Record Number:
 28
  558:          *       * Variable:               MDRIVE
  559:          *       * Called By:              VALID Clause
  560:          *       * Object Type:            Radio Button
  561:          *       * Snippet Number:         4
  562:          *       *
  563:          *
 **********************************************************
  564:          *
  565:
*!*************************************************************
*************
*
  566:  *!
  567:  *!       Function: _R7B06H0GP
  568:  *!
  569:  *!      Called by: VREG_AST.SPR
  570:  *!
  571:
*!*************************************************************
*************
*
  572:      FUNCTION _r7b06h0gp      &&  MDRIVE VALID
```

```
  573:          #REGION 1
  574:          marchi = IIF(mdrive = 1,'A:\SOBRET','B:\SOBRET')
  575:          SHOW GET opc_scr ENABLE
  576:
  577:          *
**************************************************************
  578:          *        *
  579:          *        * _R7B06H0JZ            OPC_SCR VALID
  580:          *        *
  581:          *        * Function Origin:
  582:          *        *
  583:          *        * From Platform:        MS-DOS
  584:          *        * From Screen:          PASASIST,
Record Number:
29
  585:          *        * Variable:             OPC_SCR
  586:          *        * Called By:            VALID Clause
  587:          *        * Object Type:          Push Button
  588:          *        * Snippet Number:       5
  589:          *        *
  590:          *
**************************************************************
  591:          *
  592:
*!*****************************************************************
*************
*
  593:   *!
  594:   *!       Function: _R7B06H0JZ
  595:   *!
  596:   *!      Called by: VREG_AST.SPR
  597:   *!
  598:   *!           Calls: &OPC_SCR.PRG
  599:   *!
  600:
*!*****************************************************************
*************
*
  601:      FUNCTION _r7b06h0jz     &&  OPC_SCR VALID
  602:          #REGION 1
  603:          DO &opc_scr
  604:
  605:
  606:          *
**************************************************************
  607:          *        *
  608:          *        * PASASIST/MS-DOS Procedimientos y
funciones de soporte
  609:          *        *
  610:          *
**************************************************************
  611:          *
  612:
  613:          #REGION 1
```

```
  614:
  615:          *
*************************************************************
  616:          *        *
  617:          *        *                  PASASIST Procedure
EDITAR_P
  618:          *        *
  619:          *
*************************************************************
  620:          *
  621:
  622:
  623:
*!*************************************************************************
*************
*
  624:  *!
  625:  *!       Procedure: EDITAR_P
  626:  *!
  627:
*!*************************************************************************
*************
*
  628:      PROCEDURE editar_p
  629:        1=DO CASE
  630:        ã=====CASE _DOS
  631:              |         ******************
  632:              |         _CUROBJ = OBJNUM(mmes)
  633:              |         SHOW GETS ENABLE
  634:              |         SHOW GET opc_scr DISABLE
  635: <============         RETURN
  636:              |         ******************
  637:        ã=====CASE _WINDOWS
  638:              |         ******************
  639:              |         _CUROBJ = OBJNUM(mmes)
  640:              |         SHOW GETS ENABLE
  641:              |         SHOW GET opc_scr DISABLE
  642: <============         RETURN
  643:              |         ******************
  644:        E=ENDCASE
  645:
  646:          *
*************************************************************
  647:          *        *
  648:          *        *                  PASASIST Procedure
PROCESAR
  649:          *        *
  650:          *
*************************************************************
  651:          *
  652:
  653:
  654:
*!*************************************************************************
```

```
*************
*
  655:  *!
  656:  *!       Procedure: PROCESAR
  657:  *!
  658:  *!           Calls: MEDIDOR.PRG
  659:  *!                : CALCULO.PRG
  660:  *!                : VALIDAR()              (function in
CTRL_GTO.SPR)
  661:  *!                : ABRIR_ARCH             (procedure in
PROC_DMO.SPR)
  662:  *!                : GN_MPL                 (procedure in
PROC_DMO.SPR)
  663:  *!                : GN_PDMO                (procedure in
PROC_DMO.SPR)
  664:  *!                : GN_CTA_RUB             (procedure in
PROC_DMO.SPR)
  665:  *!                : GN_ASI_CON             (procedure in
PROC_DMO.SPR)
  666:  *!
  667:  *!           Uses: SOBRETIE.DBF
  668:  *!                : &MARCHI                     Alias: CP
  669:  *!                : DSCTOSVA.DBF                Alias: DVAR
  670:  *!                : PERSONAL.DBF                Alias: DGP
  671:  *!                : LISTBANC.DBF                Alias: BNC
  672:  *!                : TOT_ROL.DBF                 Alias: TROL
  673:  *!                : PTMO_EMP.DBF
  674:  *!                : CTAS_CPE.DBF
  675:  *!                : TMP_ACPE.DBF                Alias: TMPPE
  676:  *!                : TMP_PE2.DBF
  677:  *!                : ROL_IBS.DBF
  678:  *!                : ROL_IBST.DBF
  679:  *!                : DBF.DBF
  680:  *!
  681:  *!        Indexes: CODIGO                    (tag in
PUESTOS.CDX)
  682:  *!                : CODIGO.CDX
  683:  *!                : &TMP_MPL..CDX
  684:  *!
  685:  *!      CDX files: DSCTOSVA.CDX
  686:  *!                : PERSONAL.CDX
  687:  *!                : LISTBANC.CDX
  688:  *!                : PTMO_EMP.CDX
  689:  *!                : ROL_IBS.CDX
  690:  *!
  691:
*!*************************************************************************
*************
*
  692:      PROCEDURE procesar
  693:        i=DO CASE
  694:        ã═══CASE _DOS
  695:          |       ******************
  696:          |       SELE sbt
```

```
   697:          |          GO TOP
   698:          |          LOCATE FOR mmes = MONTH(sbt.fecha) AND manio
=
YEAR(sbt.fecha)
   699:          |       ┌─IF !FOUND()
   700:          |       │     ACTI WIND sobret
   701:          |       │     ?CHR(7)
   702:          |       │     @2,5 SAY 'No Existen Datos para Proceso'
   703:          |       │     READ
   704:          |       │     DEAC WIND sobret
   705:          |       ├─ELSE
   706:          |       │     ACTI WIND sobret
   707:          |       │     ?CHR(7)
   708:          |       │     @2,2 SAY ' Inserte el disco en el Drive
'+ IIF( mdrive
= 1,'A:','B:')
   709:          |       │     READ
   710:          |       │     DEAC WIND sobret
   711:          |       │     SELE sbt.fecha AS
fecha,ALLT(STR(sbt.codigo)) AS
codigo, SUM(sbt.horaext_m) AS hext_40, SUM(sbt.horasup_l) ;
   712:          |       │          AS hext_50, SUM(sbt.horanoc_k) AS
hext_100 FROM
C:\aplic\sobretie\sobretie sbt INTO DBF &marchi;
   713:          |       │          ORDER BY codigo WHERE
MONTH(sbt.fecha)= mmes AND
YEAR(sbt.fecha) = manio;
   714:          |       │          GROUP BY codigo
   715:          |       │     SELE sobret
   716:          |       │     INDEX ON codigo TAG  codigo
   717:          |       │     ACTI WIND sobret
   718:          |       │     ?CHR(7)
   719:          |       │     @2,15 SAY 'Proceso Terminado...!'
   720:          |       │     READ
   721:          |       │     DEAC WIND sobret
   722:          |       └─ENDIF
   723: <════════|═══════RETURN
   724:          |          ***************
   725:          à═══CASE _WINDOWS
   726:          |          ******************
   727:          |          USE &marchi ORDER codigo ALIAS ast IN 0
   728:          |          SELE ast
   729:          |          GO TOP
   730:          |          LOCATE FOR mmes = VAL(ast.mes) AND
SUBS(ALLT(STR(manio)),3,2) = ast.anio
   731:          |       ┌─IF !FOUND()
   732:          |       │     ACTI WIND asist
   733:          |       │     ?CHR(7)
   734:          |       │     @2,5 SAY 'No Existen Datos para Proceso'
   735:          |       │     READ
   736:          |       │     DEAC WIND asist
   737:          |       ├─ELSE
   738:          |       │     ACTI WIND asist
   739:          |       │     ?CHR(7)
```

```
   740:                    @2,2 SAY ' Procesando Informaci¥n '
   741:                    SELE men
   742:                    SET RELATION TO men.codigo INTO ast ADDI
   743:                  ╔═SCAN FOR men.anio = manio AND men.mes =
mmes
   744:                  ╟─┌─IF !EOF('AST')
   745:                  ╟─│   REPLACE men.faltasi WITH
ast.faltasi
   746:                  ╟─│   REPLACE men.faltasj WITH
ast.faltasj
   747:                  ╟─└─ENDIF
   748:                  ╚═ENDSCAN
   749:                    CLEAR
   750:                    ?CHR(7)
   751:                    @2,15 SAY 'Proceso Terminado...!'
   752:                    READ
   753:                    DEAC WIND asist
   754:                  ENDIF
   755:                  SELE ast
   756:                  USE
   757: <════════════════RETURN
   758:                  ***************
   759:          ENDCASE
   760:
   761:          *
*************************************************************
   762:          *        *
   763:          *        *                    PASASIST Procedure SALIR
   764:          *        *
   765:          *
*************************************************************
   766:          *
   767:
   768:
   769:
*!*************************************************************************
*************
*
   770:  *!
   771:  *!      Procedure: SALIR
   772:  *!
   773:  *!      Called by: _RAA1732TG()        (function in
AT_ESCLS.SPR)
   774:  *!               : DA_ASISG.PRG
   775:  *!               : DE_EGRQ1.PRG
   776:  *!               : DE_EGRQ2.PRG
   777:  *!               : DE_EGRMN.PRG
   778:  *!               : DE_IEJUB.PRG
   779:  *!
   780:  *!          Calls: DEAC_KEYS           (procedure in
AACT_SST.SPR)
   781:  *!
   782:
*!*************************************************************************
```

```
**************
*
  783:      PROCEDURE salir
  784:        DO CASE
  785:        ÂÍÍÍÍCASE _DOS
  786:        ³       ***************
  787:        ³     ÚÄIF USED('sobret')
  788:        ³     ³   SELE sobret
  789:        ³     ³   USE
  790:        ³     ÀÄENDIF
  791:        ³       RELE WIND sobret
  792:        ³       CLEAR READ
  793: <ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍRETURN
  794:        ÂÍÍÍÍCASE _WINDOWS
  795:        ³       ***************
  796:        ³     ÚÄIF USED('sobret')
  797:        ³     ³   SELE sobret
  798:        ³     ³   USE
  799:        ³     ÀÄENDIF
  800:        ³       RELE WIND sobret
  801:        ³       CLEAR READ
  802: <ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍRETURN
  803:        ÀÍENDCASE
  804:  *:  EOF: VREG_AST.ac1
```


Varios Archivo Nómina Rol de Pagos Reportes Func.Emp Costos E.D.36 Salir
PREPARAR ARCHIVO PARA REGISTRO DE DATOS
PARA PROCESAR ROLES DE :
PASTAZA
JUBILADOS
PALORA
JUB. PASTAZA
TODOS
PRIMERA QUINCENA
SEGUNDA QUINCENA
Año. . . . : 1996
Mes. . . . : 12 DICIEMBRE
ACEPTAR
EDITAR
SALIR
Mes
Reg: 12/12
Exclusivo
NUM

```
    1:
*:*************************************************************
*************
*
    2:  *:
    3:  *: Procedure file: C:\DOC\VREG_SBT.SPR
    4:  *:         System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:         Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:      Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 15/10/1995 at 10:15:34
    8:  *:
    9:  *:  Procs & Fncts: _R4H0LZL0A()
   10:  *:                : _R4H0LZL5X()
   11:  *:                : _R4H0LZL9B()
   12:  *:                : _R4H0LZM3T()
   13:  *:                : _R4H0LZM6I()
   14:  *:
   15:  *:         Set by: EMPRESA.MPR
   16:  *:
   17:  *:          Calls: _R4H0LZL0A()       (function in
VREG_SBT.SPR)
   18:  *:                 : _R4H0LZL5X()     (function in
VREG_SBT.SPR)
   19:  *:                 : _R4H0LZL9B()     (function in
VREG_SBT.SPR)
   20:  *:                 : _R4H0LZM3T()     (function in
VREG_SBT.SPR)
   21:  *:                 : _R4H0LZM6I()     (function in
VREG_SBT.SPR)
   22:  *:
   23:  *:      Documented 19:14:19
FoxDoc
version 3.00a
   24:
*:*************************************************************
*************
*
   25:      *
************************************************************
   26:      *       *
   27:      *       * 10/15/1995           PASOBRET.SPR
 10:15:31
   28:      *       *
   29:      *
************************************************************
   30:      *       *
   31:      *       * Author's Name
   32:      *       *
   33:      *       * Copyright (c) 1995 Company Name
   34:      *       * Address
   35:      *       * City,     Zip
   36:      *       *
   37:      *       * Description:
```

```
   38:      *        * This program was automatically generated by
GENSCRN.
   39:      *        *
   40:      *
************************************************************
   41:
   42:    1=DO CASE
   43:    â═══CASE _WINDOWS
   44:    │
   45:    │
   46:    │       *
************************************************************
   47:    │       *        *
   48:    │       *        *          PASOBRET/Windows Setup Code -
SECTION 1
   49:    │       *        *
   50:    │       *
************************************************************
   51:    │       *
   52:    │
   53:    │       #REGION 1
   54:    │       **********
   55:    │       REGIONAL manio, mmes, mdrive, opc_scr, marchi
   56:    │       SET TALK OFF
   57:    │       SET SAFETY ON
   58:    │       STOR MONTH(DATE()) TO mmes
   59:    │       STOR YEAR(DATE()) TO manio
   60:    │       STOR 1 TO mdrive
   61:    │       STOR ' ' TO opc_scr,marchi
   62:    │       **********
   63:    │
   64:    │       #REGION 0
   65:    │       REGIONAL m.currarea, m.talkstat, m.compstat
   66:    │
   67:    │      ┌─IF SET("TALK") = "ON"
   68:    │      │     SET TALK OFF
   69:    │      │     m.talkstat = "ON"
   70:    │      ├─ELSE
   71:    │      │     m.talkstat = "OFF"
   72:    │      └─ENDIF
   73:    │       m.compstat = SET("COMPATIBLE")
   74:    │       SET COMPATIBLE FOXPLUS
   75:    │
   76:    │       m.rborder = SET("READBORDER")
   77:    │       SET readborder ON
   78:    │
   79:    │       m.currarea = SELECT()
   80:    │
   81:    │
   82:    │       *
************************************************************
   83:    │       *        *
   84:    │       *        *      PASOBRET/Windows Databases,
Indexes, Relations
```

```
   85:          *       *
   86:          *
*************************************************************
   87:          *
   88:
   89:          IF USED("men")
   90:              SELECT men
   91:              SET ORDER TO TAG "mens_cod"
   92:          ELSE
   93:              SELECT 0
   94:              USE (LOCFILE("dbfs\mensual.dbf","DBF","Where
is
mensual?"));
   95:                  AGAIN ALIAS men ;
   96:                  ORDER TAG "mens_cod"
   97:          ENDIF
   98:
   99:          IF USED("mes")
  100:              SELECT mes
  101:              SET ORDER TO 0
  102:          ELSE
  103:              SELECT 0
  104:              USE (LOCFILE("dbfs\meses.org","DBF","Where is
meses?"));
  105:                  AGAIN ALIAS mes ;
  106:                  ORDER 0
  107:          ENDIF
  108:
  109:          SELECT men
  110:
  111:
  112:          *
*************************************************************
  113:          *       *
  114:          *       *                Windows Window
definitions
  115:          *       *
  116:          *
*************************************************************
  117:          *
  118:
  119:          IF NOT WEXIST("sobretie") ;
  120:                OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.PJX" ;
  121:                OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.SCX" ;
  122:                OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.MNX" ;
  123:                OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.PRG" ;
  124:                OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.FRX" ;
  125:                OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.QPR"
```

```
  126:          DEFINE WINDOW sobretie ;
  127:             AT  0.000, 0.000  ;
  128:             SIZE 17.385,55.500 ;
  129:             FONT "MS Sans Serif", 8 ;
  130:             STYLE "B" ;
  131:             NOFLOAT ;
  132:             NOCLOSE ;
  133:             SHADOW ;
  134:             NOMINIMIZE
  135:          MOVE WINDOW sobretie CENTER
  136:       ENDIF
  137:
  138:
  139:       *
************************************************************
  140:       *        *
  141:       *        *          PASOBRET/Windows Setup Code -
SECTION 2
  142:       *        *
  143:       *
************************************************************
  144:       *
  145:
  146:       #REGION 1
  147:       **********
  148:       DEFINE WIND sobret FROM 1,1 TO 6,40
  149:       SELE mes
  150:       LOCATE FOR mes = mmes
  151:
  152:
  153:       *
************************************************************
  154:       *        *
  155:       *        *              PASOBRET/Windows Screen
Layout
  156:       *        *
  157:       *
************************************************************
  158:       *
  159:
  160:       #REGION 1
  161:       IF WVISIBLE("sobretie")
  162:          ACTIVATE WINDOW sobretie SAME
  163:       ELSE
  164:          ACTIVATE WINDOW sobretie NOSHOW
  165:       ENDIF
  166:       @ 1.077,10.167 SAY "PASO DE DATOS DE
SOBRETIEMPOS"  ;
  167:             FONT "MS Sans Serif", 8 ;
  168:             STYLE "B"
  169:       @ 0.615,8.167 TO 2.615,50.167 ;
  170:             PEN 1, 8
  171:       @ 4.385,18.000 SAY "MES:"  ;
  172:             FONT "MS Sans Serif", 8 ;
```

```
173:            STYLE "B"
174:         @ 4.385,31.167 GET mmes ;
175:            SIZE 1.000,4.800 ;
176:            RANGE 1, 12 ;
177:            DEFAULT " " ;
178:            FONT "MS Sans Serif", 8 ;
179:            PICTURE "99" ;
180:            VALID _r4h0lzl0a() ;
181:            ERROR "MES ERRONEO [1..12]"
182:         @ 6.615,18.000 SAY "ADO:"  ;
183:            FONT "MS Sans Serif", 8 ;
184:            STYLE "B"
185:         @ 6.615,31.167 GET manio ;
186:            SIZE 1.000,7.600 ;
187:            DEFAULT " " ;
188:            FONT "MS Sans Serif", 8 ;
189:            PICTURE "9999" ;
190:            VALID manio > 1990 AND manio < 2999 ;
191:            ERROR "AÑO INVALIDO"
192:         @ 8.846,18.000 SAY "UNIDAD"  ;
193:            FONT "MS Sans Serif", 8 ;
194:            STYLE "B"
195:         @ 10.231,18.000 SAY "DE DISCO:"  ;
196:            FONT "MS Sans Serif", 8 ;
197:            STYLE "B"
198:         @ 8.692,30.000 GET mdrive ;
199:            PICTURE "@*RVN DRIVE |<A;DRIVE |<B" ;
200:            SIZE 1.308,12.833,0.000 ;
201:            DEFAULT 1 ;
202:            FONT "MS Sans Serif", 8 ;
203:            STYLE "B" ;
204:            VALID _r4h0lzl5x()
205:         @ 13.538,4.000 GET opc_scr ;
206:            PICTURE "@*HN |<EDITAR_P;|<PROCESAR;|<SALIR"
;
207:            SIZE 1.769,16.000,0.000 ;
208:            DEFAULT 1 ;
209:            FONT "MS Sans Serif", 8 ;
210:            STYLE "B" ;
211:            VALID _r4h0lzl9b() ;
212:            DISABLE
213:         @ 13.000,2.500 TO 15.769,53.500 ;
214:            PEN 1, 8
215:         @ 4.385,36.500 GET mes.nom_mes ;
216:            SIZE 1.000,15.833 ;
217:            DEFAULT " " ;
218:            FONT "MS Sans Serif", 8 ;
219:            STYLE "B" ;
220:            PICTURE "@K" ;
221:            WHEN .F.
222:
223:         IF NOT WVISIBLE("sobretie")
224:            ACTIVATE WINDOW sobretie
225:         ENDIF
```

```
  226:
  227:          READ CYCLE ;
  228:             DEACTIVATE .F.
  229:
  230:          RELEASE WINDOW sobretie
  231:
  232:          *
**************************************************************
  233:          *        *
  234:          *        *                    Windows Closing
Databases
  235:          *        *
  236:          *
**************************************************************
  237:          *
  238:
  239:          IF USED("men")
  240:             SELECT men
  241:             USE
  242:          ENDIF
  243:
  244:          IF USED("mes")
  245:             SELECT mes
  246:             USE
  247:          ENDIF
  248:
  249:          SELECT (m.currarea)
  250:
  251:
  252:          #REGION 0
  253:
  254:          SET readborder &rborder
  255:
  256:          IF m.talkstat = "ON"
  257:             SET TALK ON
  258:          ENDIF
  259:          IF m.compstat = "ON"
  260:             SET COMPATIBLE ON
  261:          ENDIF
  262:
  263:
  264:          *
**************************************************************
  265:          *        *
  266:          *        *                PASOBRET/Windows Cleanup
Code
  267:          *        *
  268:          *
**************************************************************
  269:          *
  270:
  271:          #REGION 1
  272:          ******************
  273:
```

```
  274:    ã═══CASE _DOS
  275:    │
  276:    │
  277:    │       *
**********************************************************
  278:    │       *       *
  279:    │       *       *          PASOBRET/MS-DOS Setup Code -
SECTION 1
  280:    │       *       *
  281:    │       *
**********************************************************
  282:    │       *
  283:    │
  284:    │       #REGION 1
  285:    │       **********
  286:    │       REGIONAL manio, mmes, mdrive, opc_scr, marchi
  287:    │       SET TALK OFF
  288:    │       SET SAFETY ON
  289:    │       STOR MONTH(DATE()) TO mmes
  290:    │       STOR YEAR(DATE()) TO manio
  291:    │       STOR 1 TO mdrive
  292:    │       STOR ' ' TO opc_scr,marchi
  293:    │       **********
  294:    │
  295:    │       #REGION 0
  296:    │       REGIONAL m.currarea, m.talkstat, m.compstat
  297:    │
  298:    │      ┌─IF SET("TALK") = "ON"
  299:    │      │     SET TALK OFF
  300:    │      │     m.talkstat = "ON"
  301:    │      ├─ELSE
  302:    │      │     m.talkstat = "OFF"
  303:    │      └─ENDIF
  304:    │       m.compstat = SET("COMPATIBLE")
  305:    │       SET COMPATIBLE FOXPLUS
  306:    │
  307:    │       m.currarea = SELECT()
  308:    │
  309:    │
  310:    │       *
**********************************************************
  311:    │       *       *
  312:    │       *       *      PASOBRET/MS-DOS Databases,
Indexes, Relations
  313:    │       *       *
  314:    │       *
**********************************************************
  315:    │       *
  316:    │
  317:    │      ┌─IF USED("sbt")
  318:    │      │     SELECT sbt
  319:    │      │     SET ORDER TO 0
  320:    │      ├─ELSE
  321:    │      │     SELECT 0
```

```
  322:     |     |     USE
(LOCFILE("sobretie|sobretie.dbf","DBF","Where is
sobretie?"));
  323:     |     |         AGAIN ALIAS sbt ;
  324:     |     |         ORDER 0
  325:     |     ENDIF
  326:     |
  327:     |
  328:     |     *
************************************************************
  329:     |     *        *
  330:     |     *        *                    MS-DOS Window
definitions
  331:     |     *        *
  332:     |     *
************************************************************
  333:     |     *
  334:     |
  335:     |     IF NOT WEXIST("sobretie") ;
  336:     |     |         OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.PJX" ;
  337:     |     |         OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.SCX" ;
  338:     |     |         OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.MNX" ;
  339:     |     |         OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.PRG" ;
  340:     |     |         OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.FRX" ;
  341:     |     |         OR UPPER(WTITLE("SOBRETIE")) ==
"SOBRETIE.QPR"
  342:     |     |     DEFINE WINDOW sobretie ;
  343:     |     |         FROM INT((SROW()-16)/2),INT((SCOL()-55)/2)
;
  344:     |     |         TO
INT((SROW()-16)/2)+15,INT((SCOL()-55)/2)+54 ;
  345:     |     |         NOFLOAT ;
  346:     |     |         NOCLOSE ;
  347:     |     |         SHADOW ;
  348:     |     |         NOMINIMIZE ;
  349:     |     |         COLOR SCHEME 1
  350:     |     ENDIF
  351:     |
  352:     |
  353:     |     *
************************************************************
  354:     |     *        *
  355:     |     *        *          PASOBRET/MS-DOS Setup Code -
SECTION 2
  356:     |     *        *
  357:     |     *
************************************************************
  358:     |     *
  359:     |
```

```
360:          #REGION 1
361:          **********
362:          DEFINE WIND sobret FROM 1,1 TO 6,40
363:
364:
365:          *
*************************************************************
366:          *        *
367:          *        *                    PASOBRET/MS-DOS Screen
Layout
368:          *        *
369:          *
*************************************************************
370:          *
371:
372:          #REGION 1
373:          IF WVISIBLE("sobretie")
374:             ACTIVATE WINDOW sobretie SAME
375:          ELSE
376:             ACTIVATE WINDOW sobretie NOSHOW
377:          ENDIF
378:          @ 1,13 SAY "PASO DE DATOS DE SOBRETIEMPOS" ;
379:             SIZE 1,29, 0
380:          @ 0,11 TO 2,43
381:          @ 4,18 SAY "MES:" ;
382:             SIZE 1,4, 0
383:          @ 4,29 GET mmes ;
384:             SIZE 1,2 ;
385:             DEFAULT " " ;
386:             PICTURE "99" ;
387:             VALID mmes>0 AND mmes <12 ;
388:             ERROR "MES ERRONEO [1..12]"
389:          @ 6,18 SAY "AÑO:" ;
390:             SIZE 1,4, 0
391:          @ 6,29 GET manio ;
392:             SIZE 1,4 ;
393:             DEFAULT " " ;
394:             PICTURE "9999" ;
395:             VALID manio > 1990 AND manio < 2999 ;
396:             ERROR "AÑO INVALIDO"
397:          @ 8,18 SAY "UNIDAD" ;
398:             SIZE 1,6, 0
399:          @ 9,18 SAY "DE DISCO:" ;
400:             SIZE 1,9, 0
401:          @ 8,28 GET mdrive ;
402:             PICTURE "@*RVN DRIVE |<A;DRIVE |<B" ;
403:             SIZE 1,11,0 ;
404:             DEFAULT 1 ;
405:             VALID _r4h0lzm3t()
406:          @ 12,3 GET opc_scr ;
407:             PICTURE "@*HN |<EDITAR_P;|<PROCESAR;|<SALIR"
;
408:             SIZE 1,16,0 ;
409:             DEFAULT 1 ;
```

```
410:            VALID _r4h01zm61() ;
411:            DISABLE
412:          @ 11,1 TO 13,52
413:
414:          IF NOT WVISIBLE("sobretie")
415:             ACTIVATE WINDOW sobretie
416:          ENDIF
417:
418:          READ CYCLE ;
419:             DEACTIVATE .F.
420:
421:          RELEASE WINDOW sobretie
422:
423:          *
************************************************************
424:          *        *
425:          *        *                    MS-DOS Closing
Databases
426:          *        *
427:          *
************************************************************
428:          *
429:
430:          IF USED("sbt")
431:             SELECT sbt
432:             USE
433:          ENDIF
434:
435:          SELECT (m.currarea)
436:
437:
438:          #REGION 0
439:          IF m.talkstat = "ON"
440:             SET TALK ON
441:          ENDIF
442:          IF m.compstat = "ON"
443:             SET COMPATIBLE ON
444:          ENDIF
445:
446:
447:          *
************************************************************
448:          *        *
449:          *        *                PASOBRET/MS-DOS Cleanup
Code
450:          *        *
451:          *
************************************************************
452:          *
453:
454:          #REGION 1
455:          ********************
456:
457:       ENDCASE
```

```
  458:
  459:
  460:      *
************************************************************
  461:      *        *
  462:      *        * _R4H0LZL0A            MMES VALID
  463:      *        *
  464:      *        * Function Origin:
  465:      *        *
  466:      *        * From Platform:        Windows
  467:      *        * From Screen:          PASOBRET,      Record
Number:     7
  468:      *        * Variable:             MMES
  469:      *        * Called By:            VALID Clause
  470:      *        * Object Type:          Field
  471:      *        * Snippet Number:       1
  472:      *        *
  473:      *
************************************************************
  474:      *
  475:
*!*************************************************************************
*************
*
  476:   *!
  477:   *!        Function: _R4H0LZL0A
  478:   *!
  479:   *!       Called by: VREG_SBT.SPR
  480:   *!
  481:
*!*************************************************************************
*************
*
  482:      FUNCTION _r4h0lzl0a      &&  MMES VALID
  483:          #REGION 1
  484:          SELE mes
  485:          LOCATE FOR mes = mmes
  486:          SHOW GET mes.nom_mes
  487:
  488:          *
************************************************************
  489:          *        *
  490:          *        * _R4H0LZL5X            MDRIVE VALID
  491:          *        *
  492:          *        * Function Origin:
  493:          *        *
  494:          *        * From Platform:        Windows
  495:          *        * From Screen:          PASOBRET,
Record Number:
12
  496:          *        * Variable:             MDRIVE
  497:          *        * Called By:            VALID Clause
  498:          *        * Object Type:          Radio Button
  499:          *        * Snippet Number:       2
```

```
  500:          *       *
  501:          *
************************************************************
  502:          *
  503:
*!*************************************************************************
*************
*
  504:  *!
  505:  *!       Function: _R4H0LZL5X
  506:  *!
  507:  *!      Called by: VREG_SBT.SPR
  508:  *!
  509:
*!*************************************************************************
*************
*
  510:      FUNCTION _r4h0lzl5x      &&  MDRIVE VALID
  511:          #REGION 1
  512:          marchi = IIF(mdrive = 1,'A:\SOBRET','B:\SOBRET')
  513:          SHOW GET opc_scr ENABLE
  514:          ACTI WIND sobret
  515:          ?CHR(7)
  516:          @2,2 SAY ' Inserte el disco en el Drive '+ IIF(
mdrive =
1,'A:','B:')
  517:          READ
  518:          DEAC WIND sobret
  519:
  520:
  521:          *
************************************************************
  522:          *       *
  523:          *       * _R4H0LZL9B            OPC_SCR VALID
  524:          *       *
  525:          *       * Function Origin:
  526:          *       *
  527:          *       * From Platform:        Windows
  528:          *       * From Screen:          PASOBRET,
Record Number:
13
  529:          *       * Variable:             OPC_SCR
  530:          *       * Called By:            VALID Clause
  531:          *       * Object Type:          Push Button
  532:          *       * Snippet Number:       3
  533:          *       *
  534:          *
************************************************************
  535:          *
  536:
*!*************************************************************************
*************
*
  537:  *!
```

```
538:  *!        Function: _R4H0LZL9B
539:  *!
540:  *!       Called by: VREG_SBT.SPR
541:  *!
542:  *!           Calls: &OPC_SCR.PRG
543:  *!
544:
*!*************************************************************************
*************
*
545:      FUNCTION _r4h0lzl9b      &&  OPC_SCR VALID
546:          #REGION 1
547:          DO &opc_scr
548:
549:          *
************************************************************
550:          *        *
551:          *        * _R4H0LZM3T              MDRIVE VALID
552:          *        *
553:          *        * Function Origin:
554:          *        *
555:          *        * From Platform:          MS-DOS
556:          *        * From Screen:            PASOBRET,
Record Number:
28
557:          *        * Variable:               MDRIVE
558:          *        * Called By:              VALID Clause
559:          *        * Object Type:            Radio Button
560:          *        * Snippet Number:         4
561:          *        *
562:          *
************************************************************
563:          *
564:
*!*************************************************************************
*************
*
565:  *!
566:  *!        Function: _R4H0LZM3T
567:  *!
568:  *!       Called by: VREG_SBT.SPR
569:  *!
570:
*!*************************************************************************
*************
*
571:      FUNCTION _r4h0lzm3t      &&  MDRIVE VALID
572:          #REGION 1
573:          marchi = IIF(mdrive = 1,'A:\SOBRET','B:\SOBRET')
574:          SHOW GET opc_scr ENABLE
575:
576:          *
************************************************************
577:          *        *
```

```
  578:          *        * _R4H0LZM6I            OPC_SCR VALID
  579:          *        *
  580:          *        * Function Origin:
  581:          *        *
  582:          *        * From Platform:        MS-DOS
  583:          *        * From Screen:          PASOBRET,
Record Number:
29
  584:          *        * Variable:             OPC_SCR
  585:          *        * Called By:            VALID Clause
  586:          *        * Object Type:          Push Button
  587:          *        * Snippet Number:       5
  588:          *        *
  589:          *
************************************************************
  590:          *
  591:
*!*************************************************************************
*************
*
  592:  *!
  593:  *!       Function: _R4H0LZM6I
  594:  *!
  595:  *!      Called by: VREG_SBT.SPR
  596:  *!
  597:  *!          Calls: &OPC_SCR.PRG
  598:  *!
  599:
*!*************************************************************************
*************
*
  600:      FUNCTION _r4h0lzm6i     &&  OPC_SCR VALID
  601:          #REGION 1
  602:          DO &opc_scr
  603:
  604:
  605:          *
************************************************************
  606:          *        *
  607:          *        *  PASOBRET/MS-DOS Supporting Procedures
and Functions
  608:          *        *
  609:          *
************************************************************
  610:          *
  611:
  612:          #REGION 1
  613:
  614:          *
************************************************************
  615:          *        *
  616:          *        *                PASOBRET Procedure
EDITAR_P
  617:          *        *
```

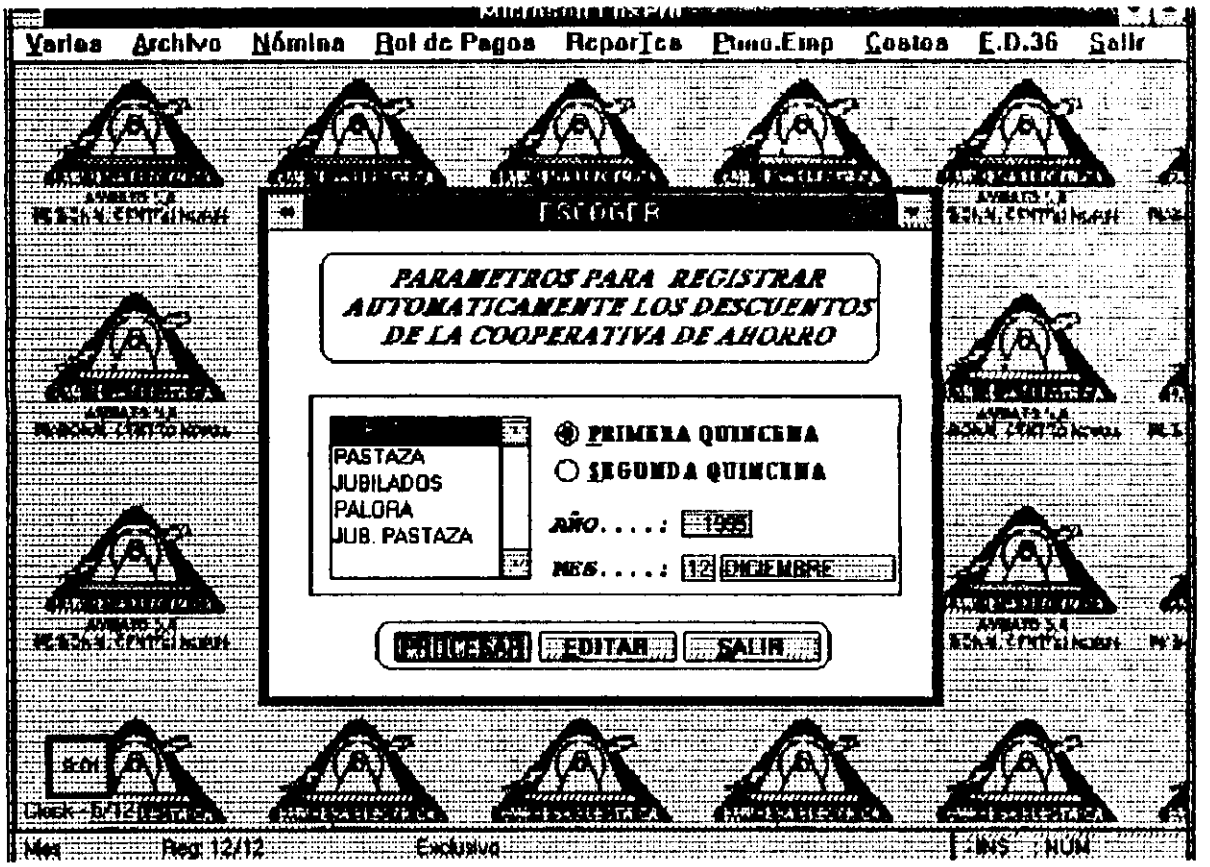
```
  618:          *
*************************************************************
  619:          *
  620:
  621:
  622:
*!*************************************************************************
*************
*
  623:  *!
  624:  *!       Procedure: EDITAR_P
  625:  *!
  626:
*!*************************************************************************
*************
*
  627:      PROCEDURE editar_p
  628:        1=DO CASE
  629:        à====CASE _DOS
  630:           |        ******************
  631:           |        _CUROBJ = OBJNUM(mmes)
  632:           |        SHOW GETS ENABLE
  633:           |        SHOW GET opc_scr DISABLE
  634: <=========|=======RETURN
  635:           |        ******************
  636:        à==|===CASE _WINDOWS
  637:           |        ******************
  638:           |        _CUROBJ = OBJNUM(mmes)
  639:           |        SHOW GETS ENABLE
  640:           |        SHOW GET opc_scr DISABLE
  641: <=========|=======RETURN
  642:           |        ******************
  643:        ê==|=ENDCASE
  644:
  645:          *
*************************************************************
  646:          *        *
  647:          *        *                  PASOBRET Procedure
PROCESAR
  648:          *        *
  649:          *
*************************************************************
  650:          *
  651:
  652:
  653:
*!*************************************************************************
*************
*
  654:  *!
  655:  *!       Procedure: PROCESAR
  656:  *!
  657:  *!            Calls: MEDIDOR.PRG
  658:  *!                 : CALCULO.PRG
```

```
   659:  *!                 : VALIDAR()          (function in
CTRL_GTO.SPR)
   660:  *!                 : ABRIR_ARCH         (procedure in
PROC_DMO.SPR)
   661:  *!                 : GN_MPL             (procedure in
PROC_DMO.SPR)
   662:  *!                 : GN_PDMO            (procedure in
PROC_DMO.SPR)
   663:  *!                 : GN_CTA_RUB         (procedure in
PROC_DMO.SPR)
   664:  *!                 : GN_ASI_CON         (procedure in
PROC_DMO.SPR)
   665:  *!
   666:  *!           Uses: SOBRETIE.DBF
   667:  *!                 : &MARCHI                 Alias: CP
   668:  *!                 : DSCTOSVA.DBF            Alias: DVAR
   669:  *!                 : PERSONAL.DBF            Alias: DGP
   670:  *!                 : LISTBANC.DBF            Alias: BNC
   671:  *!                 : TOT_ROL.DBF             Alias: TROL
   672:  *!                 : PTMO_EMP.DBF
   673:  *!                 : CTAS_CPE.DBF
   674:  *!                 : TMP_ACPE.DBF            Alias: TMPPE
   675:  *!                 : TMP_PE2.DBF
   676:  *!                 : ROL_IBS.DBF
   677:  *!                 : ROL_IBST.DBF
   678:  *!                 : DBF.DBF
   679:  *!
   680:  *!        Indexes: CODIGO               (tag in
PUESTOS.CDX)
   681:  *!                 : CODIGO.CDX
   682:  *!                 : &TMP_MPL..CDX
   683:  *!
   684:  *!      CDX files: DSCTOSVA.CDX
   685:  *!                 : PERSONAL.CDX
   686:  *!                 : LISTBANC.CDX
   687:  *!                 : PTMO_EMP.CDX
   688:  *!                 : ROL_IBS.CDX
   689:  *!
   690:
*!*****************************************************************
*************
*
   691:      PROCEDURE procesar
   692:        ı=DO CASE
   693:        ã═══CASE _DOS
   694:        │       ******************
   695:        │       SELE sbt
   696:        │       GO TOP
   697:        │       LOCATE FOR mmes = MONTH(sbt.fecha) AND manio
=
YEAR(sbt.fecha)
   698:        │       ┌─IF !FOUND()
   699:        │       │   ACTI WIND sobret
   700:        │       │   ?CHR(7)
```

```
   701:          |        |     @2,5 SAY 'No Existen Datos para Proceso'
   702:          |        |     READ
   703:          |        |     DEAC WIND sobret
   704:          |        |  ELSE
   705:          |        |     ACTI WIND sobret
   706:          |        |     ?CHR(7)
   707:          |        |     @2,2 SAY ' Inserte el disco en el Drive
'+ IIF( mdrive
= 1,'A:','B:')
   708:          |        |     READ
   709:          |        |     DEAC WIND sobret
   710:          |        |     SELE sbt.fecha AS
fecha,ALLT(STR(sbt.codigo)) AS
codigo, SUM(sbt.horaext_m) AS hext_40, SUM(sbt.horasup_l) ;
   711:          |        |           AS hext_50, SUM(sbt.horanoc_k) AS
hext_100 FROM
C:\aplic\sobretie\sobretie sbt INTO DBF &marchi;
   712:          |        |           ORDER BY codigo WHERE
MONTH(sbt.fecha)= mmes AND
YEAR(sbt.fecha) = manio;
   713:          |        |           GROUP BY codigo
   714:          |        |     SELE sobret
   715:          |        |     INDEX ON codigo TAG  codigo
   716:          |        |     ACTI WIND sobret
   717:          |        |     ?CHR(7)
   718:          |        |     @2,15 SAY 'Proceso Terminado...!'
   719:          |        |     READ
   720:          |        |     DEAC WIND sobret
   721:          |        |  ENDIF
   722: <========|========RETURN
   723:          |        |  ***************
   724:          |     <==CASE _WINDOWS
   725:          |        |  ******************
   726:          |        |  USE &marchi ORDER codigo ALIAS sbt IN 0
   727:          |        |  SELE sbt
   728:          |        |  GO TOP
   729:          |        |  *LOCATE FOR MMES = MONTH(SBT.FECHA) AND
MANIO =
YEAR(SBT.FECHA)
   730:          |        |  IF EOF()
   731:          |        |     ACTI WIND sobret
   732:          |        |     ?CHR(7)
   733:          |        |     @2,5 SAY 'No Existen Datos para Proceso'
   734:          |        |     READ
   735:          |        |     DEAC WIND sobret
   736:          |        |  ELSE
   737:          |        |     ACTI WIND sobret
   738:          |        |     ?CHR(7)
   739:          |        |     @2,2 SAY ' Procesando Informaci¥n '
   740:          |        |     SELE men
   741:          |        |     SET RELATION TO men.codigo INTO sbt ADDI
   742:          |        |  ==SCAN
   743:          |        |     *for men.anio = manio and men.mes =
mmes
```

```
  744:         |      |  ||  IF !EOF('sbt')
  745:         |      |  ||  *                replace men.hex1
with
sbt.hext_40
  746:         |      |  ||  *                replace men.hex2
with
sbt.hext_50
  747:         |      |  ||  *                replace men.hex3
with
sbt.hext_100
  748:         |      |  ||     REPLACE men.hex1 WITH sbt.hext_100
  749:         |      |  ||     REPLACE men.hex2 WITH sbt.hext_50
  750:         |      |  ||     REPLACE men.hex3 WITH sbt.hext_40
  751:         |      |  ||  ENDIF
  752:         |      |  ENDSCAN
  753:         |      |  CLEAR
  754:         |      |  ?CHR(7)
  755:         |      |  @2,15 SAY 'Proceso Terminado...!'
  756:         |      |  READ
  757:         |      |  DEAC WIND sobret
  758:         |      ENDIF
  759:         |      SELE sbt
  760:         |      USE
  761: <=======|======RETURN
  762:         |      ***************
  763:       ENDCASE
  764:
  765:       *
************************************************************
  766:       *        *
  767:       *        *                    PASOBRET Procedure SALIR
  768:       *        *
  769:       *
************************************************************
  770:       *
  771:
  772:
  773:
*!*************************************************************************
*************
*
  774:  *!
  775:  *!      Procedure: SALIR
  776:  *!
  777:  *!      Called by: _RAA1732TG()       (function in
AT_ESCLS.SPR)
  778:  *!               : DA_ASISG.PRG
  779:  *!               : DE_EGRQ1.PRG
  780:  *!               : DE_EGRQ2.PRG
  781:  *!               : DE_EGRMN.PRG
  782:  *!               : DE_IEJUB.PRG
  783:  *!
  784:  *!          Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
```

```
  785:  *!
  786:
*!*************************************************************************
*************
*
  787:      PROCEDURE salir
  788:        DO CASE
  789:        CASE _DOS
  790:             ***************
  791:          IF USED('sobret')
  792:             SELE sobret
  793:             USE
  794:          ENDIF
  795:          RELE WIND sobret
  796:          CLEAR READ
  797:          RETURN
  798:        CASE _WINDOWS
  799:             ***************
  800:          IF USED('sobret')
  801:             SELE sobret
  802:             USE
  803:          ENDIF
  804:          RELE WIND sobret
  805:          CLEAR READ
  806:          RETURN
  807:        ENDCASE
  808:  *:  EOF: VREG_SBT.ac1
```


Varios Archivo Nómina Rol de Pagos Reportes Pmo.Emp Costos E.D.36 Salir
ESCOGER
PARAMETROS PARA REGISTRAR AUTOMATICAMENTE LOS DESCUENTOS DE LA COOPERATIVA DE AHORRO
PASTAZA
JUBILADOS
PALORA
JUB. PASTAZA
PRIMERA QUINCENA
SEGUNDA QUINCENA
AÑO. . . . : 1995
MES. . . . : 12 DICIEMBRE
PROCESAR
EDITAR
SALIR
Reg 12/12
Exclusiva
NUM

```
    1:
*:*************************************************************
*************
*
    2:  *:
    3:  *: Procedure file: C:\DOC\VREG_CAH.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 24/03/1996 at 12:10:06
    8:  *:
    9:  *:  Procs & Fncts: _R900Q2XQ5()
   10:  *:                : _R900Q2XWC()
   11:  *:                : _R900Q2Y0R()
   12:  *:
   13:  *:          Set by: EMPRESA.MPR
   14:  *:
   15:  *:           Calls: _R900Q2XQ5()        (function in
VREG_CAH.SPR)
   16:  *:                : _R900Q2XWC()        (function in
VREG_CAH.SPR)
   17:  *:                : _R900Q2Y0R()        (function in
VREG_CAH.SPR)
   18:  *:
   19:  *:       Documented 19:14:21
FoxDoc
version 3.00a
   20:
*:*************************************************************
*************
*
   21:      *
************************************************************
   22:      *        *
   23:      *        * 24/03/1996           CTRL_CAH.SPR
 12:10:04
   24:      *        *
   25:      *
************************************************************
   26:      *        *
   27:      *        * Nombre del autor
   28:      *        *
   29:      *        * Copyright (c) 1996 Nombre de la empresa
   30:      *        * Direcci¥n
   31:      *        * Ciudad,     C.P.
   32:      *        * Pa¥s
   33:      *        *
   34:      *        * Description:
   35:      *        * Este programa lo ha generado autom▀ticamente
GENSCRN.
   36:      *        *
   37:      *
************************************************************
```

```
   38:
   39:
   40:      *
************************************************************
   41:      *        *
   42:      *        *  CTRL_CAH/Windows Código de configuración -
SECCION 1
   43:      *        *
   44:      *
************************************************************
   45:      *
   46:
   47:      #REGION 1
   48:      **********
   49:      REGIONAL opc_scr, opc_rol, opc_qui, mmes, vmes, manio,
mquincena,
mrol,;
   50:         mopc_qui,mfquin,vciu,vtip
   51:      STOR ' ' TO opc_scr, mmes,opc_rol, fquin
   52:      STOR 0 TO opc_qui,mopc_qui,vtip
   53:      STOR 1 TO mrol,vciu
   54:      STOR .F. TO mquincena
   55:      STOR YEAR(DATE()) TO manio
   56:      STOR MONTH(DATE()) TO vmes
   57:
   58:      **********
   59:
   60:      #REGION 0
   61:      REGIONAL m.currarea, m.talkstat, m.compstat
   62:
   63:    ┌─IF SET("TALK") = "ON"
   64:    │    SET TALK OFF
   65:    │    m.talkstat = "ON"
   66:    ├─ELSE
   67:    │    m.talkstat = "OFF"
   68:    └─ENDIF
   69:      m.compstat = SET("COMPATIBLE")
   70:      SET COMPATIBLE FOXPLUS
   71:
   72:      m.rborder = SET("READBORDER")
   73:      SET readborder ON
   74:
   75:      m.currarea = SELECT()
   76:
   77:
   78:      *
************************************************************
   79:      *        *
   80:      *        *  CTRL_CAH/Windows Bases de datos, índices,
relaciones
   81:      *        *
   82:      *
************************************************************
   83:      *
```

```
   84:
   85:     IF USED("rol")
   86:        SELECT rol
   87:        SET ORDER TO TAG "cod_per"
   88:     ELSE
   89:        SELECT 0
   90:        USE (LOCFILE("dbfs\rol.dbf","DBF","Buscar  rol?"));
   91:           AGAIN ALIAS rol ;
   92:           ORDER TAG "cod_per"
   93:     ENDIF
   94:
   95:     IF USED("mes")
   96:        SELECT mes
   97:        SET ORDER TO 0
   98:     ELSE
   99:        SELECT 0
  100:        USE (LOCFILE("dbfs\meses.org","DBF","Buscar
meses?"));
  101:           AGAIN ALIAS mes ;
  102:           ORDER 0
  103:     ENDIF
  104:
  105:     IF USED("sst")
  106:        SELECT sst
  107:        SET ORDER TO TAG "sist_cod"
  108:     ELSE
  109:        SELECT 0
  110:        USE (LOCFILE("dbfs\sistema.dbf","DBF","Buscar
sistema?"));
  111:           AGAIN ALIAS sst ;
  112:           ORDER TAG "sist_cod"
  113:     ENDIF
  114:
  115:     IF USED("dgp")
  116:        SELECT dgp
  117:        SET ORDER TO TAG "cod_per"
  118:     ELSE
  119:        SELECT 0
  120:        USE (LOCFILE("dbfs\personal.dbf","DBF","Buscar
personal?"));
  121:           AGAIN ALIAS dgp ;
  122:           ORDER TAG "cod_per"
  123:     ENDIF
  124:
  125:     IF USED("men")
  126:        SELECT men
  127:        SET ORDER TO 0
  128:     ELSE
  129:        SELECT 0
  130:        USE (LOCFILE("dbfs\mensual.dbf","DBF","Buscar
mensual?"));
  131:           AGAIN ALIAS men ;
  132:           ORDER 0
  133:     ENDIF
```

```
  134:
  135:     IF USED("ctr")
  136:         SELECT ctr
  137:         SET ORDER TO 0
  138:     ELSE
  139:         SELECT 0
  140:         USE (LOCFILE("dbfs\ctrl_rol.dbf","DBF","Buscar
ctrl_rol?"));
  141:             AGAIN ALIAS ctr ;
  142:             ORDER 0
  143:     ENDIF
  144:
  145:     IF USED("acr")
  146:         SELECT acr
  147:         SET ORDER TO TAG "cod_per"
  148:     ELSE
  149:         SELECT 0
  150:         USE (LOCFILE("dbfs\ac_imprt.dbf","DBF","Buscar
ac_imprt?"));
  151:             AGAIN ALIAS acr ;
  152:             ORDER TAG "cod_per"
  153:     ENDIF
  154:
  155:     SELECT rol
  156:
  157:     SET RELATION OFF INTO acr
  158:     SET RELATION TO cod_per INTO acr ADDITIVE
  159:
  160:     SELECT rol
  161:
  162:     SET RELATION OFF INTO dgp
  163:     SET RELATION TO cod_per INTO dgp ADDITIVE
  164:
  165:     SELECT rol
  166:
  167:     *
************************************************************
  168:     *        *
  169:     *        *              Windows Definiciones de ventana
  170:     *        *
  171:     *
************************************************************
  172:     *
  173:
  174:     IF NOT WEXIST("ctrl_par") ;
  175:           OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.PJX" ;
  176:           OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.SCX" ;
  177:           OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.MNX" ;
  178:           OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.PRG" ;
  179:           OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.FRX" ;
  180:           OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.QPR"
  181:        DEFINE WINDOW ctrl_par ;
  182:           AT  0.000, 0.000  ;
  183:           SIZE 19.923,71.400 ;
```

```
  232:          PEN 1, 8 ;
  233:          STYLE "16"
  234:       @ 1.385,6.400 SAY "PARAMETROS PARA  REGISTRAR" +
CHR(13) + ;
  235:          "AUTOMATICAMENTE LOS DESCUENTOS" + CHR(13) + ;
  236:          "DE LA COOPERATIVA DE AHORRO" + CHR(13) + ;
  237:          "" ;
  238:          SIZE 3.000,42.571, 0.000 ;
  239:          PICTURE "@I" ;
  240:          FONT "MS Serif", 10 ;
  241:          STYLE "BIT"
  242:       @ 0.923,5.600 TO 5.615,66.000 ;
  243:          PEN 1, 8 ;
  244:          STYLE "16"
  245:       @ 7.000,4.400 TO 15.692,68.200 ;
  246:          PEN 1, 8
  247:       @ 7.923,6.600 GET opc_rol ;
  248:          PICTURE "@&N" ;
  249:          POPUP _r900q2xg4 ;
  250:          SIZE 6.923,21.400 ;
  251:          DEFAULT " " ;
  252:          FONT "MS Sans Serif", 8 ;
  253:          STYLE "" ;
  254:          VALID _r900q2xq5()
  255:       @ 8.000,31.000 GET opc_qui ;
  256:          PICTURE "@*RVN |<PRIMERA QUINCENA;|<SEGUNDA
QUINCENA" ;
  257:          SIZE 1.308,25.667,0.308 ;
  258:          DEFAULT 1 ;
  259:          FONT "MS Serif", 8 ;
  260:          STYLE "BT" ;
  261:          WHEN mquincena
  262:       @ 12.077,45.000 GET manio ;
  263:          SIZE 1.000,6.600 ;
  264:          DEFAULT " " ;
  265:          FONT "MS Sans Serif", 8 ;
  266:          PICTURE "9999" ;
  267:          VALID manio > 1990 AND manio < 2999 ;
  268:          ERROR 'ADO INVALIDO '
  269:       @ 14.000,45.000 GET vmes ;
  270:          SIZE 1.000,2.600 ;
  271:          DEFAULT " " ;
  272:          FONT "MS Sans Serif", 8 ;
  273:          PICTURE "99" ;
  274:          VALID _r900q2xwc() ;
  275:          ERROR 'MES INVALIDO (1..12)'
  276:       @ 14.000,49.200 GET mes.nom_mes ;
  277:          SIZE 1.000,17.800 ;
  278:          DEFAULT " " ;
  279:          FONT "MS Sans Serif", 8 ;
  280:          PICTURE "@K" ;
  281:          WHEN .F.
  282:       @ 17.231,13.600 GET opc_scr ;
  283:          PICTURE "@*HN |<PROCESAR;|<EDITAR;|<SALIR" ;
```

```
284:        SIZE 1.308,12.333,0.667 ;
285:        DEFAULT 1 ;
286:        FONT "MS Sans Serif", 8 ;
287:        STYLE "B" ;
288:        VALID _r900q2y0r()
289:
290:    IF NOT WVISIBLE("ctrl_par")
291:       ACTIVATE WINDOW ctrl_par
292:    ENDIF
293:
294:    READ CYCLE ;
295:       VALID .F.
296:
297:    RELEASE WINDOW ctrl_par
298:
299:    *
************************************************************
300:    *        *
301:    *        *              Windows Cerrando bases de datos
302:    *        *
303:    *
************************************************************
304:    *
305:
306:    IF USED("rol")
307:       SELECT rol
308:       USE
309:    ENDIF
310:
311:    IF USED("mes")
312:       SELECT mes
313:       USE
314:    ENDIF
315:
316:    IF USED("sst")
317:       SELECT sst
318:       USE
319:    ENDIF
320:
321:    IF USED("dgp")
322:       SELECT dgp
323:       USE
324:    ENDIF
325:
326:    IF USED("men")
327:       SELECT men
328:       USE
329:    ENDIF
330:
331:    IF USED("ctr")
332:       SELECT ctr
333:       USE
334:    ENDIF
335:
```

```
336:     IF USED("acr")
337:        SELECT acr
338:        USE
339:     ENDIF
340:
341:     SELECT (m.currarea)
342:
343:     RELEASE POPUPS _r900q2xg4
344:
345:     #REGION 0
346:
347:     SET readborder &rborder
348:
349:     IF m.talkstat = "ON"
350:        SET TALK ON
351:     ENDIF
352:     IF m.compstat = "ON"
353:        SET COMPATIBLE ON
354:     ENDIF
355:
356:
357:     *
************************************************************
358:     *        *
359:     *        *            CTRL_CAH/Windows Código de
limpieza
360:     *        *
361:     *
************************************************************
362:     *
363:
364:     #REGION 1
365:     ******************
366:
367:
368:     *
************************************************************
369:     *        *
370:     *        * CTRL_CAH/Windows Procedimientos y funciones
de soporte
371:     *        *
372:     *
************************************************************
373:     *
374:
375:     #REGION 1
376:
*!*************************************************************************
************
*
377:  *!
378:  *!      Procedure: PROCESAR
379:  *!
380:  *!           Calls: MEDIDOR.PRG
```

```
  381:  *!                : CALCULO.PRG
  382:  *!                : VALIDAR()          (function in
CTRL_GTO.SPR)
  383:  *!                : ABRIR_ARCH         (procedure in
PROC_DMO.SPR)
  384:  *!                : GN_MPL             (procedure in
PROC_DMO.SPR)
  385:  *!                : GN_PDMO            (procedure in
PROC_DMO.SPR)
  386:  *!                : GN_CTA_RUB         (procedure in
PROC_DMO.SPR)
  387:  *!                : GN_ASI_CON         (procedure in
PROC_DMO.SPR)
  388:  *!
  389:  *!           Uses: SOBRETIE.DBF
  390:  *!                : &MARCHI                 Alias: CP
  391:  *!                : DSCTOSVA.DBF            Alias: DVAR
  392:  *!                : PERSONAL.DBF            Alias: DGP
  393:  *!                : LISTBANC.DBF            Alias: BNC
  394:  *!                : TOT_ROL.DBF             Alias: TROL
  395:  *!                : PTMO_EMP.DBF
  396:  *!                : CTAS_CPE.DBF
  397:  *!                : TMP_ACPE.DBF            Alias: TMPPE
  398:  *!                : TMP_PE2.DBF
  399:  *!                : ROL_IBS.DBF
  400:  *!                : ROL_IBST.DBF
  401:  *!                : DBF.DBF
  402:  *!
  403:  *!        Indexes: CODIGO                    (tag in
PUESTOS.CDX)
  404:  *!                : CODIGO.CDX
  405:  *!                : &TMP_MPL..CDX
  406:  *!
  407:  *!      CDX files: DSCTOSVA.CDX
  408:  *!                : PERSONAL.CDX
  409:  *!                : LISTBANC.CDX
  410:  *!                : PTMO_EMP.CDX
  411:  *!                : ROL_IBS.CDX
  412:  *!
  413:
*!*************************************************************************
*************
*
  414:      PROCEDURE procesar
  415:          ******************
  416:          IF NOT WEXIST("Mproc")
  417:             DEFINE WINDOW mproc ;
  418:                FROM  0.000, 0.000  ;
  419:                TO 7,40;
  420:                PANEL;
  421:                FONT "MS Sans Serif", 8 ;
  422:                NOCLOSE
  423:             MOVE WINDOW mproc CENTER
  424:          ENDIF
```

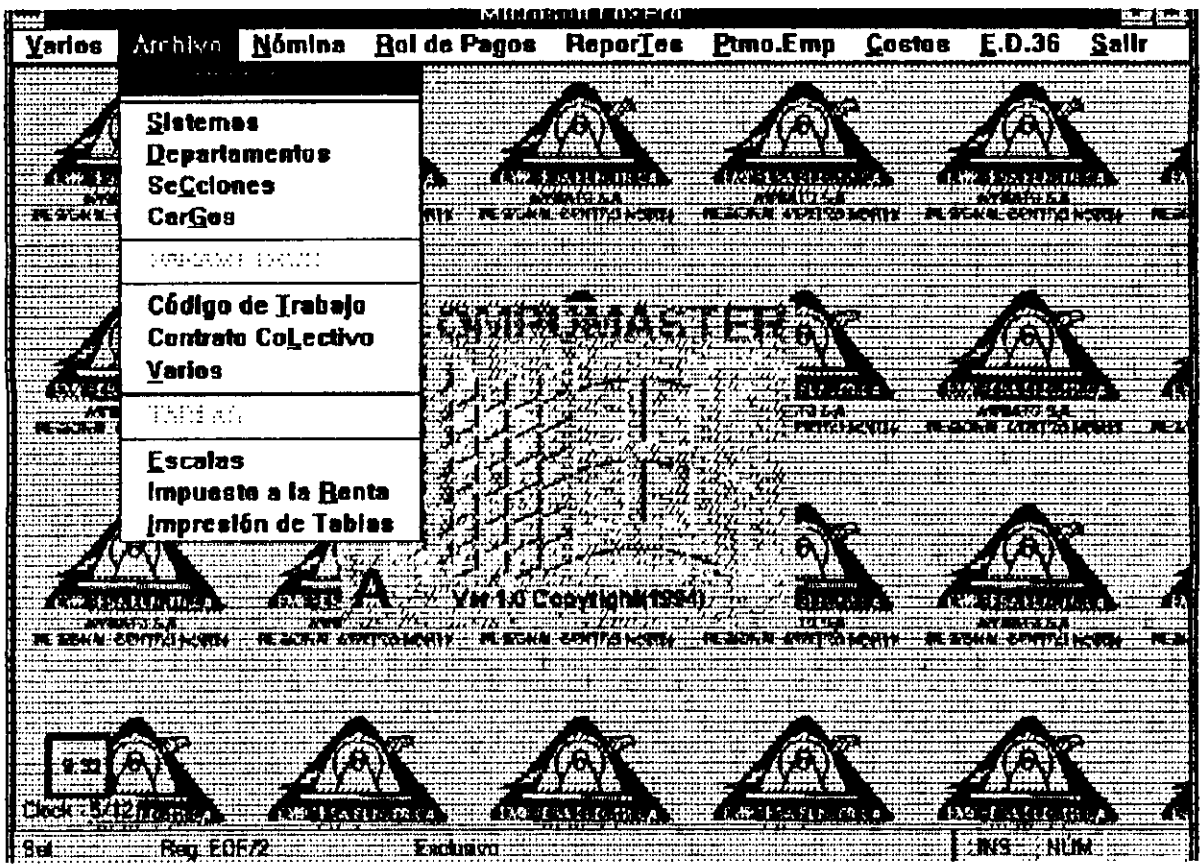
```
  425:          ACTI WIND mproc
  426:          ?CHR(7)
  427:          @1,15 SAY " Realizando Proceso ... "
  428:          SELE men
  429:          SET FILTER TO anio = manio AND mes = vmes
  430:          STOR 'a:\salro'+ALLT(STR(vmes))+ALLT(STR(opc_qui))
TO marchi
  431:          USE &marchi ALIAS cp IN 0
  432:          SELE cp
  433:          INDEX ON ALLT(STR(codigo)) TO codigo
  434:          SELE men
  435:          SET RELATIO TO codigo INTO cp
  436:        ╔═SCAN FOR anio = manio AND mes = vmes AND ciudad =
vciu
  437:        ║   ┌─IF opc_qui = 1
  438:        ║   │    REPLACE men.coop_ah1 WITH (cp.ahorro +
cp.prestamo +
cp.interes +;
  439:        ║   │          cp.casascom + cp.caport + cp.otrdes +
cp.pagpext + ;
  440:        ║   │          cp.cmort + cp.cingreso + cp.otrdes +
cp.pagpdec + ;
  441:        ║   │          cp.pagpbon + cp.intext + cp.intdec +
cp.intbon)
  442:        ║   ├─ELSE
  443:        ║   │    REPLACE men.coop_ah2 WITH (cp.ahorro +
cp.prestamo +
cp.interes +;
  444:        ║   │          cp.casascom + cp.caport + cp.otrdes +
cp.pagpext + ;
  445:        ║   │          cp.cmort + cp.cingreso + cp.otrdes +
cp.pagpdec + ;
  446:        ║   │          cp.pagpbon + cp.intext + cp.intdec +
cp.intbon)
  447:        ║   └─ENDIF
  448:        ╚═ENDSCAN
  449:          CLEAR
  450:          ?CHR(7)
  451:          @1,15 SAY "Proceso Terminado ... "
  452:          STOR '' TO macepta
  453:          @ 2.5,17 GET macepta ;
  454:             PICTURE "@* \<ACEPTAR" ;
  455:             DEFAULT 1
  456:          READ MODAL
  457:          RELEASE WIND mproc
  458:  <═══════RETURN
  459:
  460:          ****************
  461:
*!*************************************************************
*************
*
  462:  *!
  463:  *!       Procedure: EDITAR
```

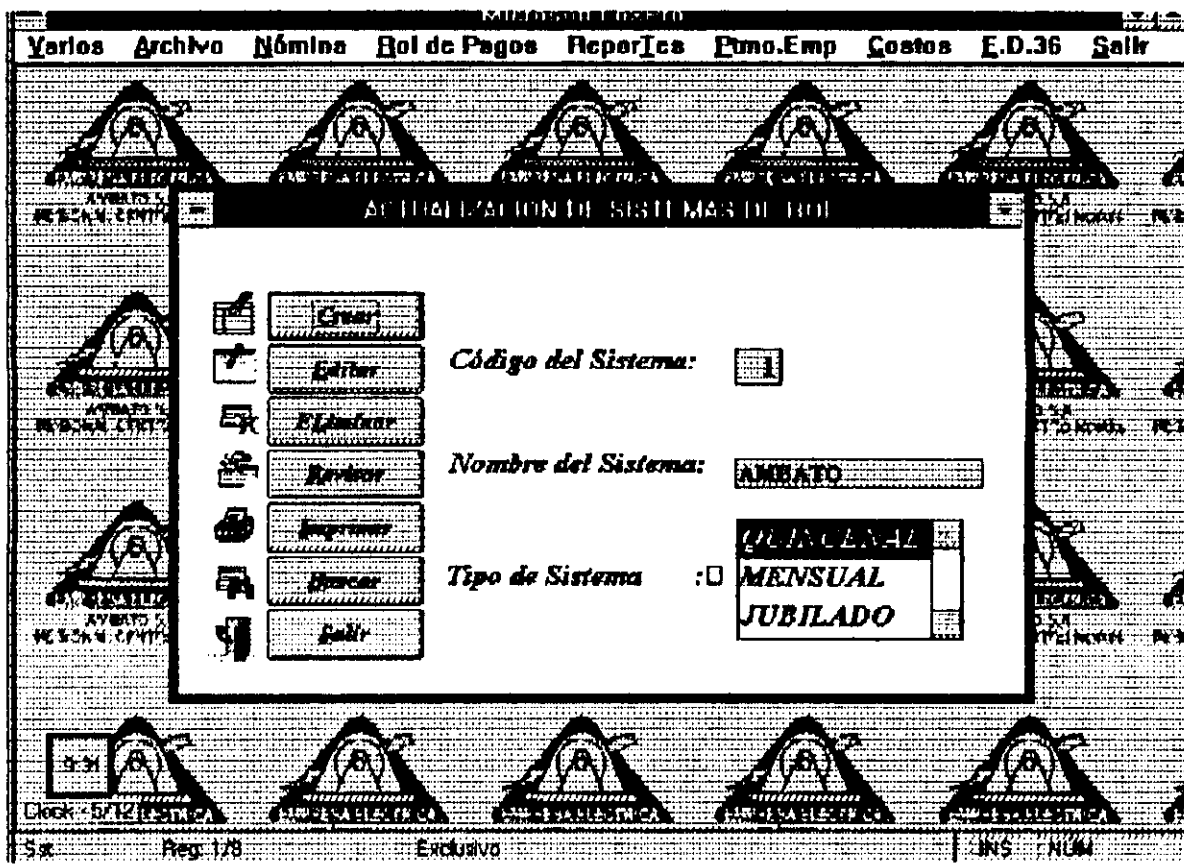
```
   464:  *!
   465:  *!        Called by: CREAR                    (procedure in
AACT_SST.SPR)
   466:  *!
   467:  *!             Calls: DEAC_KEYS               (procedure in
AACT_SST.SPR)
   468:  *!                  : CABEZA                  (procedure in
PERSONAL.PRG)
   469:  *!                  : INGRESO                 (procedure in
ACT_PTMO.SPR)
   470:  *!                  : CREA_TMP                (procedure in
MO_PDLAB.SPR)
   471:  *!                  : MBROW                   (procedure in
MO_PDLAB.SPR)
   472:  *!                  : REF_SCR                 (procedure in
MO_PDLAB.SPR)
   473:  *!                  : MENSAJES                (procedure in
MO_PDLAB.SPR)
   474:  *!
   475:
*!*************************************************************************
*************
*
   476:      PROCEDURE editar
   477:          ****************
   478:          _CUROBJ= OBJNUM(opc_rol)
   479: <=======RETURN
   480:
   481:          ****************
   482:
*!*************************************************************************
*************
*
   483:  *!
   484:  *!        Procedure: SALIR
   485:  *!
   486:  *!        Called by: _RAA1732TG()             (function in
AT_ESCLS.SPR)
   487:  *!                 : DA_ASISG.PRG
   488:  *!                 : DE_EGRQ1.PRG
   489:  *!                 : DE_EGRQ2.PRG
   490:  *!                 : DE_EGRMN.PRG
   491:  *!                 : DE_IEJUB.PRG
   492:  *!
   493:  *!             Calls: DEAC_KEYS               (procedure in
AACT_SST.SPR)
   494:  *!
   495:
*!*************************************************************************
*************
*
   496:      PROCEDURE salir
   497:          ****************
   498:          CLEAR READ
```

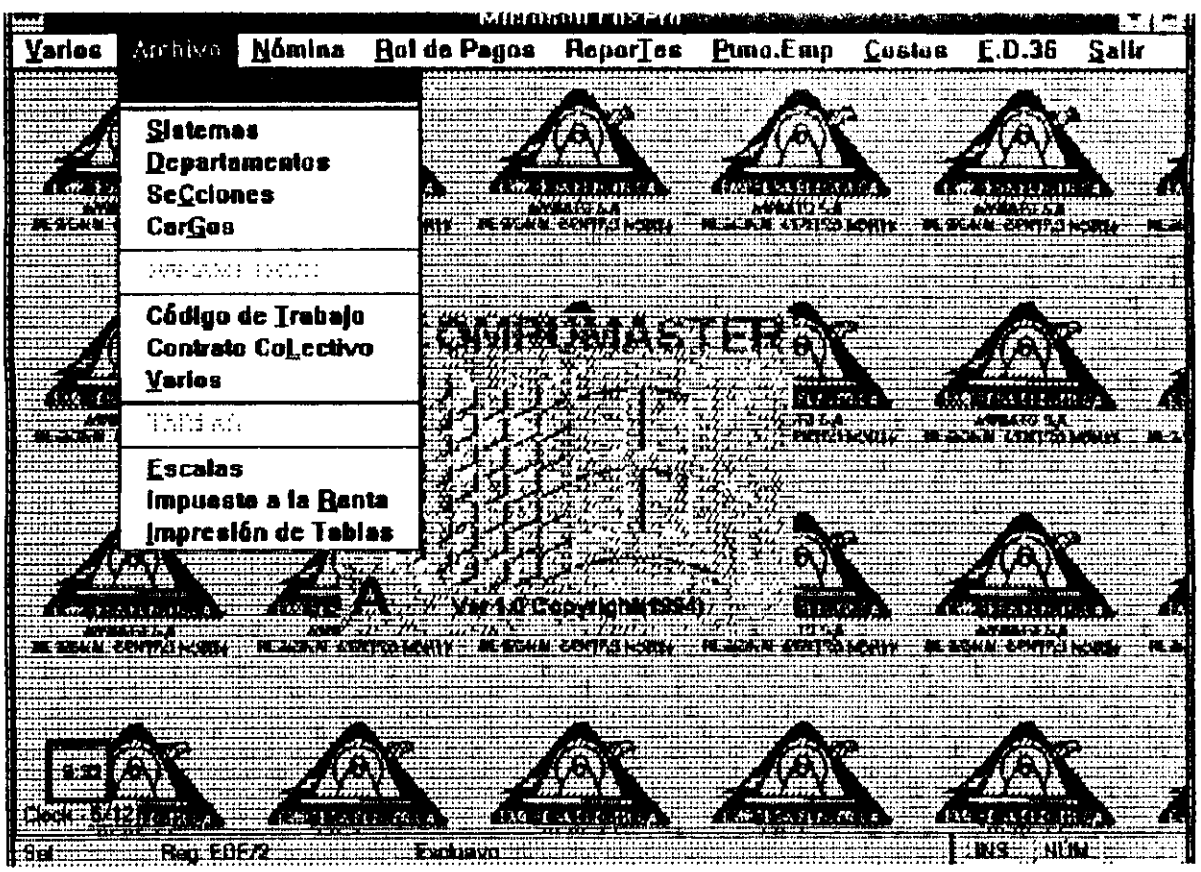
```
  499: <=======RETURN
  500:
  501:          *
**************************************************************
  502:          *        *
  503:          *        * _R900Q2XQ5            opc_rol VALID
  504:          *        *
  505:          *        * Function Origin:
  506:          *        *
  507:          *        * From Platform:        Windows
  508:          *        * From Screen:          CTRL_CAH,
Record Number:
17
  509:          *        * Variable:             opc_rol
  510:          *        * Called By:            VALID Clause
  511:          *        * Object Type:          List
  512:          *        * Snippet Number:       1
  513:          *        *
  514:          *
**************************************************************
  515:          *
  516:
*!*************************************************************************
*************
*
  517:  *!
  518:  *!       Function: _R900Q2XQ5
  519:  *!
  520:  *!      Called by: VREG_CAH.SPR
  521:  *!
  522:
*!*************************************************************************
*************
*
  523:      FUNCTION _r900q2xq5     &&  opc_rol VALID
  524:          #REGION 1
  525:          STOR sst.cod_ciu TO vciu
  526:          STOR sst.tipo TO vtip
  527:          STOR 2 TO opc_qui
  528:          STOR .F. TO mquincena
  529:          IF sst.tipo = 0
  530:             STOR 1 TO opc_qui
  531:             STOR .T. TO mquincena
  532:          ENDIF
  533:
  534:          *
**************************************************************
  535:          *        *
  536:          *        * _R900Q2XWC            VMES VALID
  537:          *        *
  538:          *        * Function Origin:
  539:          *        *
  540:          *        * From Platform:        Windows
  541:          *        * From Screen:          CTRL_CAH,
```

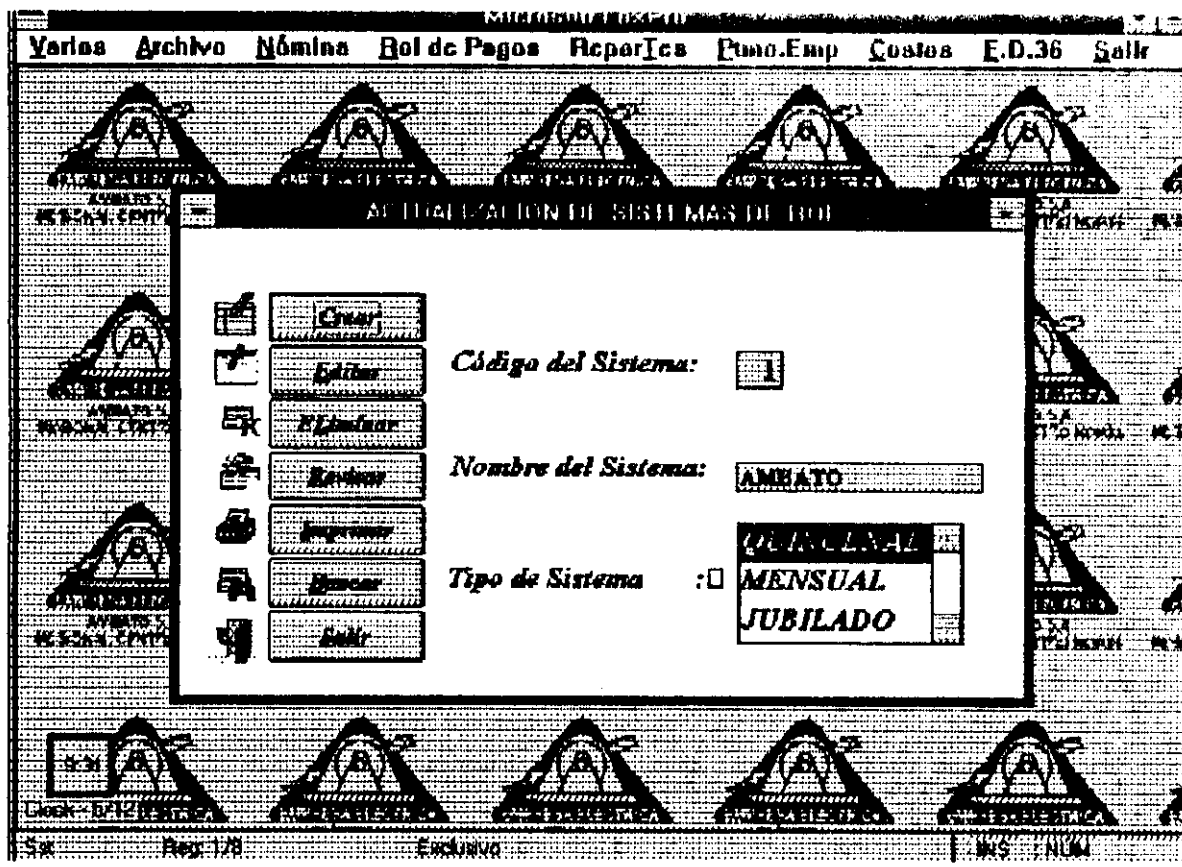
```
Record Number:
20
  542:          *       * Variable:             VMES
  543:          *       * Called By:            VALID Clause
  544:          *       * Object Type:          Field
  545:          *       * Snippet Number:       2
  546:          *       *
  547:          *
************************************************************
  548:          *
  549:
*!*************************************************************************
*************
*
  550:  *!
  551:  *!       Function: _R900Q2XWC
  552:  *!
  553:  *!      Called by: VREG_CAH.SPR
  554:  *!
  555:
*!*************************************************************************
*************
*
  556:      FUNCTION _r900q2xwc     &&  VMES VALID
  557:          #REGION 1
  558:          IF vmes > 0 AND vmes < 13
  559:             STOR .T. TO vret
  560:             SELE mes
  561:             LOCATE FOR vmes = mes
  562:             STOR nom_mes TO mmes
  563:             SHOW GET mes.nom_mes
  564:          ELSE
  565:             STOR .F. TO vret
  566:          ENDIF
  567: <=======RETURN vret
  568:
  569:          *
************************************************************
  570:          *       *
  571:          *       * _R900Q2Y0R            OPC_SCR VALID
  572:          *       *
  573:          *       * Function Origin:
  574:          *       *
  575:          *       * From Platform:        Windows
  576:          *       * From Screen:          CTRL_CAH,
Record Number:
22
  577:          *       * Variable:             OPC_SCR
  578:          *       * Called By:            VALID Clause
  579:          *       * Object Type:          Push Button
  580:          *       * Snippet Number:       3
  581:          *       *
  582:          *
************************************************************
```

```
583:          *
584:
*!*****************************************************************************
*************
*
585:   *!
586:   *!        Function: _R900Q2Y0R
587:   *!
588:   *!       Called by: VREG_CAH.SPR
589:   *!
590:   *!           Calls: &OPC_SCR.PRG
591:   *!
592:
*!*****************************************************************************
*************
*
593:      FUNCTION _r900q2y0r      &&  OPC_SCR VALID
594:          #REGION 1
595:          DO &opc_scr
596:   *:  EOF: VREG_CAH.acl
```


Varios
Archivo
Nómina
Rol de Pagos
ReporTes
Ptmo.Emp
Costos
E.D.36
Salir
Sistemas
Departamentos
SeCciones
CarGos
Código de Trabajo
Contrato CoLectivo
Varios
Escalas
Impuesto a la Renta
Impresión de Tablas
Ver 1.0 Copyright(1994)
Sel
Reg: 6/6/2
Exclusivo
INS
NUM


Varios Archivo Nómina Rol de Pagos Reportes Pmo.Emp Costos E.D.36 Salir
Crear
Editar
Eliminar
Revisar
Imprimir
Buscar
Salir
Código del Sistema:
1
Nombre del Sistema:
AMBATO
Tipo de Sistema :
QUINCENAL
MENSUAL
JUBILADO
Reg: 1/8
Exclusivo
INS
NUM


Varios
Archivo
Nómina
Rol de Pagos
ReporTes
Pmo.Emp
Costos
E.D.36
Salir
Sistemas
Departamentos
SeCciones
CarGos
Código de Trabajo
Contrato CoLectivo
Varios
Escalas
Impuesto a la Renta
Impresión de Tablas
SUPERMASTER
Ver 1.0 Copyright 1994
Reg: E0672
Exclusivo
NUM


Varios Archivo Nómina Bol de Pagos ReporTes Pmo.Emp Costos E.D.36 Salir
Crear
Editar
Eliminar
Revisar
Imprimir
Buscar
Salir
Código del Sistema:
1
Nombre del Sistema:
AMBATO
Tipo de Sistema :
QUINCENAL
MENSUAL
JUBILADO

```
    1:
*:*************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\AACT_SST.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 20/04/1996 at 19:38:42
    8:  *:
    9:  *:  Procs & Fncts: MSCATTER
   10:  *:                : ACT_KEYS
   11:  *:                : _R9R163S5N()
   12:  *:                : _R9R163SSE()
   13:  *:                : _R9R163SWQ()
   14:  *:                : CREAR
   15:  *:                : ELIMINAR
   16:  *:                : IMPRIMIR
   17:  *:                : REVISAR
   18:  *:                : BUSCAR
   19:  *:                : VAL_BUSCAR
   20:  *:                : DEAC_KEYS
   21:  *:                : CTR_POSP
   22:  *:                : GRABAR
   23:  *:                : CANCELAR
   24:  *:
   25:  *:          Set by: EMPRESA.MPR
   26:  *:
   27:  *:           Calls: MSCATTER           (procedure in
AACT_SST.SPR)
   28:  *:                : ACT_KEYS           (procedure in
AACT_SST.SPR)
   29:  *:                : _R9R163S5N()       (function in
AACT_SST.SPR)
   30:  *:                : _R9R163SSE()       (function in
AACT_SST.SPR)
   31:  *:                : _R9R163SWQ()       (function in
AACT_SST.SPR)
   32:  *:
   33:  *:      Documented 19:14:24
FoxDoc version 3.00a
   34:
*:*************************************************************
**************
   35:      *
************************************************************
   36:      *        *
   37:      *        * 20/04/1996            AACT_SST.SPR
 19:38:39
   38:      *        *
   39:      *
************************************************************
   40:      *        *
```

```
*        * Nombre del autor
*        *
*        * Copyright (c) 1996 Nombre de la empresa
*        * Direcci¥n
*        * Ciudad,     C.P.
*        * Pa¥s
*        *
*        * Description:
*        * Este programa lo ha generado automáticamente
GENSCRN.
*        *
*
************************************************************


************************************************************
*
*        *  AACT_SST/Windows C¥digo de configuraci¥n -
SECCION 1
*        *
*
************************************************************
*

#REGION 1
**********
REGIONAL opc_scr, creando, editando, reg_ant

STOR ''   TO opc_scr
STOR .F. TO creando, editando

**********

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
   SET TALK OFF
   m.talkstat = "ON"
ELSE
   m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.rborder = SET("READBORDER")
SET readborder ON

m.currarea = SELECT()


*
************************************************************
```

```
   89:      *        *
   90:      *        *  AACT_SST/Windows Bases de datos, Yndices,
relaciones
   91:      *        *
   92:      *
************************************************************
   93:      *
   94:
   95:     IF USED("sst")
   96:        SELECT sst
   97:        SET ORDER TO TAG "todos"
   98:     ELSE
   99:        SELECT 0
  100:        USE (LOCFILE("srol\dbfs\sistema.dbf","DBF","Buscar
sistema?"));
  101:           AGAIN ALIAS sst ;
  102:           ORDER TAG "todos"
  103:     ENDIF
  104:
  105:     IF USED("dpt")
  106:        SELECT dpt
  107:        SET ORDER TO TAG "sistema"
  108:     ELSE
  109:        SELECT 0
  110:        USE (LOCFILE("srol\dbfs\departam.dbf","DBF","Buscar
departam?"));
  111:           AGAIN ALIAS dpt ;
  112:           ORDER TAG "sistema"
  113:     ENDIF
  114:
  115:     IF USED("tsst")
  116:        SELECT tsst
  117:        SET ORDER TO TAG "tipo"
  118:     ELSE
  119:        SELECT 0
  120:        USE (LOCFILE("srol\dbfs\tipo_sst.dbf","DBF","Buscar
tipo_sst?"));
  121:           AGAIN ALIAS tsst ;
  122:           ORDER TAG "tipo"
  123:     ENDIF
  124:
  125:     SELECT dpt
  126:
  127:     SELECT sst
  128:     SELECT sst
  129:
  130:     SET RELATION OFF INTO tsst
  131:     SET RELATION TO tipo INTO tsst ADDITIVE
  132:
  133:     SELECT dpt
  134:
  135:     *
************************************************************
  136:     *        *
```

```
  137:      *        *              Windows Definiciones de ventana
  138:      *        *
  139:      *
************************************************************
  140:      *
  141:
  142:     IF NOT WEXIST("aact_sst") ;
  143:           OR UPPER(WTITLE("AACT_SST")) == "AACT_SST.PJX" ;
  144:           OR UPPER(WTITLE("AACT_SST")) == "AACT_SST.SCX" ;
  145:           OR UPPER(WTITLE("AACT_SST")) == "AACT_SST.MNX" ;
  146:           OR UPPER(WTITLE("AACT_SST")) == "AACT_SST.PRG" ;
  147:           OR UPPER(WTITLE("AACT_SST")) == "AACT_SST.FRX" ;
  148:           OR UPPER(WTITLE("AACT_SST")) == "AACT_SST.QPR"
  149:        DEFINE WINDOW aact_sst ;
  150:           AT  0.000, 0.000  ;
  151:           SIZE 13.100,64.857 ;
  152:           TITLE "ACTUALIZACION DE SISTEMAS DE ROL" ;
  153:           FONT "Times New Roman", 12 ;
  154:           STYLE "BI" ;
  155:           FLOAT ;
  156:           NOCLOSE ;
  157:           MINIMIZE ;
  158:           SYSTEM
  159:        MOVE WINDOW aact_sst CENTER
  160:     ENDIF
  161:
  162:
  163:      *
************************************************************
  164:      *        *
  165:      *        *  AACT_SST/Windows Código de configuración -
SECCION 2
  166:      *        *
  167:      *
************************************************************
  168:      *
  169:
  170:      #REGION 1
  171:      **********
  172:      SELE sst
  173:      SCATTER MEMVAR BLANK
  174:      DO mscatter
  175:      STOR RECNO() TO reg_ant
  176:      DO act_keys
  177:
  178:     IF NOT WEXIST('waiting')
  179:        DEFI WIND waiting FROM 0,0 TO 12,35 PANEL FONT
"Times New Roman", 9;
  180:           STYLE "BIT"
  181:        MOVE WIND waiting CENTER
  182:     ENDIF
  183:     IF NOT WEXIST('BUSCAR') && o definir en el principal
  184:        DEFINE WINDOW buscar FROM 0,0 TO 10,30 TITLE
'BUSCAR' SYSTEM ;
```

```
185:          FONT "Times New Roman", 8 STYLE "BIT"
186:       MOVE WIND buscar CENTER
187:    ENDIF
188:
189:
190:    #REGION 1
191:    DEFINE POPUP _r9r163s1c ;
192:       PROMPT FIELD tsst.detalle ;
193:       SCROLL
194:
195:    *
************************************************************
196:    *        *
197:    *        *        AACT_SST/Windows Distribuci¥n de
pantalla
198:    *        *
199:    *
************************************************************
200:    *
201:
202:    #REGION 1
203:    IF WVISIBLE("aact_sst")
204:       ACTIVATE WINDOW aact_sst SAME
205:    ELSE
206:       ACTIVATE WINDOW aact_sst NOSHOW
207:    ENDIF
208:     @ 1.800,6.857 GET opc_scr ;
209:       PICTURE "@*VN
|<Crear;|<Editar;E|<Liminar;|<Revisar;|<Imprimir;|<Buscar;|<Salir
" ;
210:       SIZE 1.647,16.800,0.118 ;
211:       DEFAULT 1 ;
212:       FONT "Times New Roman", 10 ;
213:       STYLE "BI" ;
214:       VALID _r9r163s5n()
215:    @ 3.200,2.571 SAY
(LOCFILE("bitm|program.bmp","BMP|ICO|PCT|ICN", "¬D¥nde est¤
program?" )) BITMAP ;
216:       SIZE 1.250,3.571 ;
217:       STRETCH ;
218:       STYLE "T"
219:    @ 1.800,1.714 SAY
(LOCFILE("bitm|memo.bmp","BMP|ICO|PCT|ICN", "¬D¥nde est¤ memo?"
)) BITMAP ;
220:       SIZE 1.250,5.286 ;
221:       STRETCH ;
222:       STYLE "T"
223:    @ 4.800,2.857 SAY
(LOCFILE("bitm|wzdelete.bmp","BMP|ICO|PCT|ICN", "¬D¥nde est¤
wzdelete?" )) BITMAP ;
224:       SIZE 1.250,3.571 ;
225:       STRETCH ;
226:       STYLE "T"
227:    @ 6.150,2.857 SAY
```

```
(LOCFILE("bitm\query.bmp","BMP\ICO\PCT\ICN", "┐Dónde está query?"
)) BITMAP ;
  228:          SIZE 1.250,3.571 ;
  229:          STRETCH ;
  230:          STYLE "T"
  231:       @ 7.850,2.429 SAY
(LOCFILE("bitm\wzprint.bmp","BMP\ICO\PCT\ICN", "┐Dónde está
wzprint?" )) BITMAP ;
  232:          SIZE 1.250,3.571 ;
  233:          STRETCH ;
  234:          STYLE "T"
  235:       @ 9.400,2.571 SAY
(LOCFILE("bitm\wzlocate.bmp","BMP\ICO\PCT\ICN", "┐Dónde está
wzlocate?" )) BITMAP ;
  236:          SIZE 1.250,3.571 ;
  237:          STRETCH ;
  238:          STYLE "T"
  239:       @ 11.000,2.286 SAY
(LOCFILE("bitm\close.bmp","BMP\ICO\PCT\ICN", "┐Dónde está close?"
)) BITMAP ;
  240:          SIZE 1.250,3.571 ;
  241:          STRETCH ;
  242:          STYLE "T"
  243:       @ 3.250,20.857 SAY "Código del Sistema:" ;
  244:          FONT "Times New Roman", 12 ;
  245:          STYLE "BIT"
  246:       @ 6.200,20.857 SAY "Nombre del Sistema:" ;
  247:          FONT "Times New Roman", 12 ;
  248:          STYLE "BIT"
  249:       @ 9.350,20.571 SAY "Tipo de Sistema        :" + CHR(13)
+ ;
  250:          "" ;
  251:          FONT "Times New Roman", 12 ;
  252:          STYLE "BIT"
  253:       @ 3.450,43.143 GET m.cod_ciu ;
  254:          SIZE 0.950,2.857 ;
  255:          RANGE 1, 8 ;
  256:          DEFAULT 0 ;
  257:          FONT "Times New Roman", 12 ;
  258:          STYLE "B" ;
  259:          PICTURE "@K" ;
  260:          WHEN creando ;
  261:          VALID _r9r163sse() ;
  262:          ERROR ' CODIGO DE SISTEMA YA HA SIDO REGISTRADO'
  263:       @ 6.600,43.143 GET m.ciudad ;
  264:          SIZE 0.938,21.333 ;
  265:          DEFAULT " " ;
  266:          FONT "Times New Roman", 10 ;
  267:          STYLE "B" ;
  268:          PICTURE "@K!" ;
  269:          WHEN editando OR creando ;
  270:          VALID !EMPTY(m.ciudad) ;
  271:          ERROR 'REGISTRE EL NOMBRE DEL SISTEMA'
  272:       @ 8.250,43.000 GET m.tipo ;
```

```
273:          PICTURE "@&N" ;
274:          POPUP _r9r163s1c ;
275:          SIZE 3.300,17.000 ;
276:          DEFAULT " " ;
277:          FONT "Times New Roman", 12 ;
278:          STYLE "BI" ;
279:          VALID _r9r163swq()
280:
281:       IF NOT WVISIBLE("aact_sst")
282:          ACTIVATE WINDOW aact_sst
283:       ENDIF
284:
285:       READ CYCLE ;
286:          VALID .F.
287:
288:       RELEASE WINDOW aact_sst
289:
290:       *
************************************************************
291:       *          *
292:       *          *                Windows Cerrando bases de datos
293:       *          *
294:       *
************************************************************
295:       *
296:
297:       IF USED("sst")
298:          SELECT sst
299:          USE
300:       ENDIF
301:
302:       IF USED("dpt")
303:          SELECT dpt
304:          USE
305:       ENDIF
306:
307:       IF USED("tsst")
308:          SELECT tsst
309:          USE
310:       ENDIF
311:
312:       SELECT (m.currarea)
313:
314:       RELEASE POPUPS _r9r163s1c
315:
316:       #REGION 0
317:
318:       SET readborder &rborder
319:
320:       IF m.talkstat = "ON"
321:          SET TALK ON
322:       ENDIF
323:       IF m.compstat = "ON"
324:          SET COMPATIBLE ON
```

```
  325:    ENDIF
  326:
  327:
  328:    *
************************************************************
  329:    *        *
  330:    *        *            AACT_SST/Windows Código de
limpieza
  331:    *        *
  332:    *
************************************************************
  333:    *
  334:
  335:    #REGION 1
  336:    ***************
  337:
  338:
  339:    *
************************************************************
  340:    *        *
  341:    *        * AACT_SST/Windows Procedimientos y funciones
de soporte
  342:    *        *
  343:    *
************************************************************
  344:    *
  345:
  346:    #REGION 1
  347:
*!*************************************************************************
**************
  348:  *!
  349:  *!       Procedure: CREAR
  350:  *!
  351:  *!           Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
  352:  *!                : BUSC_CLV()         (function in
DGP_CARG.SPR)
  353:  *!                : EDITAR             (procedure in
CTRL_RES.SPR)
  354:  *!                : PROCESO            (procedure in
DSC_GRTR.SPR)
  355:  *!                : INGRESO            (procedure in
ACT_PTMO.SPR)
  356:  *!                : CREA_TMP           (procedure in
MO_PDLAB.SPR)
  357:  *!                : MBROW              (procedure in
MO_PDLAB.SPR)
  358:  *!                : MENSAJES           (procedure in
MO_PDLAB.SPR)
  359:  *!                : REF_SCR            (procedure in
MO_PDLAB.SPR)
  360:  *!
  361:
```

```
*!*************************************************************
**************
  362:      PROCEDURE crear
  363:           ***************
  364:           DO deac_keys
  365:           reg_ant = RECNO()
  366:           SCATTER MEMV BLANK
  367:           STOR .T. TO creando
  368:           _CUROBJ = OBJNUM(m.cod_ciu)
  369:           SHOW GET opc_scr DISABLE
  370:           SHOW GET opc_scr,1 PROMPT '\<Grabar' ENABLE
  371:           SHOW GET opc_scr,2 PROMPT '\<Cancelar' ENABLE
  372: <=======RETURN
  373:
  374:           *****************
  375:
*!*************************************************************
**************
  376:  *!
  377:  *!       Procedure: EDITAR
  378:  *!
  379:  *!       Called by: CREAR              (procedure in
AACT_SST.SPR)
  380:  *!
  381:  *!            Calls: DEAC_KEYS         (procedure in
AACT_SST.SPR)
  382:  *!                 : CABEZA            (procedure in
PERSONAL.PRG)
  383:  *!                 : INGRESO           (procedure in
ACT_PTMO.SPR)
  384:  *!                 : CREA_TMP          (procedure in
MO_PDLAB.SPR)
  385:  *!                 : MBROW             (procedure in
MO_PDLAB.SPR)
  386:  *!                 : REF_SCR           (procedure in
MO_PDLAB.SPR)
  387:  *!                 : MENSAJES          (procedure in
MO_PDLAB.SPR)
  388:  *!
  389:
*!*************************************************************
**************
  390:      PROCEDURE editar
  391:           ***************
  392:           DO deac_keys
  393:           STOR .T. TO editando
  394:           _CUROBJ = OBJNUM(m.ciudad)
  395:           SHOW GET opc_scr DISABLE
  396:           SHOW GET opc_scr,1 PROMPT '\<Grabar' ENABLE
  397:           SHOW GET opc_scr,2 PROMPT '\<Cancelar' ENABLE
  398: <=======RETURN
  399:
  400:           ******************
  401:
```

```
*!*************************************************************
**************
  402:  *!
  403:  *!       Procedure: ELIMINAR
  404:  *!
  405:  *!           Calls: MSCATTER          (procedure in
AACT_SST.SPR)
  406:  *!                : DEAC_KEYS         (procedure in
AACT_SST.SPR)
  407:  *!                : ACT_KEYS          (procedure in
AACT_SST.SPR)
  408:  *!                : BORRAR            (procedure in
DSC_GRTR.SPR)
  409:  *!                : ELIM_MPL          (procedure in
MO_PDLAB.SPR)
  410:  *!                : CTR_POSP          (procedure in
AACT_SST.SPR)
  411:  *!
  412:
*!*************************************************************
**************
  413:     PROCEDURE eliminar
  414:         ******************
  415:         SELECT dpt
  416:         SEEK m.cod_ciu
  417:         ACTIVATE WINDOW waiting
  418:         ?CHR(7)
  419:       ┌─IF FOUND()
  420:       │    @1,1 SAY '  Lo Siento.  Pero no se puede '
  421:       │    @2,1 SAY '   Sistemas que contienen
informaci¥n'
  422:       │    STOR 1 TO mcont
  423:       │    ?CHR(7)
  424:       │    @ 4,10 GET mcont;
  425:       │       PICTURE "@*h |<Aceptar" ;
  426:       │       SIZE 1,15 FONT 'R',12 STYLE "BI"
  427:       │    READ MODAL
  428:       ├─ELSE
  429:       │    @0,0 SAY " El Sistema: " + m.ciudad +' serß
ELIMINADO ...!'
  430:       │    STOR 1 TO mcont
  431:       │    ?CHR(7)
  432:       │    @ 3,8 GET mcont;
  433:       │       PICTURE "@*h |<Cancelar;|<Eliminar" ;
  434:       │       SIZE 1,10 FONT 'R',12 STYLE "BI"
  435:       │    READ MODAL
  436:       │  ┌─IF mcont = 2
  437:       │  │    WAIT( "Borrando Sistema ..........!")WINDOW
TIME(1)
  438:       │  │    SELE sst
  439:       │  │    DELE
  440:       │  │    GO TOP
  441:       │  │    DO mscatter
  442:       │  │    reg_ant = RECNO()
```

```
  443:          |  |     SHOW GETS
  444:          |  ENDIF
  445:          ENDIF
  446:            DEACT WIND waiting
  447:            SELECT sst
  448: <=======RETURN
  449:            ******************
  450:
*!*********************************************************************
**************
  451:  *!
  452:  *!       Procedure: IMPRIMIR
  453:  *!
  454:  *!            Calls: MENSAJE2           (procedure in
ACT_PTMO.SPR)
  455:  *!
  456:  *!    Report Forms: DSC_GRTR.FRX
  457:  *!                : CTA_REP.FRX
  458:  *!
  459:
*!*********************************************************************
**************
  460:      PROCEDURE imprimir
  461:          ******************
  462:          ?CHR(7)
  463:          WAIT WIND 'PROCESO NO PREPARADO'
  464: <=======RETURN
  465:          ******************
  466:
*!*********************************************************************
**************
  467:  *!
  468:  *!       Procedure: REVISAR
  469:  *!
  470:  *!            Calls: MSCATTER           (procedure in
AACT_SST.SPR)
  471:  *!                 : DEAC_KEYS          (procedure in
AACT_SST.SPR)
  472:  *!                 : ACT_KEYS           (procedure in
AACT_SST.SPR)
  473:  *!                 : VAL_CAMPO()        (function in
AP_PRMCT.SPR)
  474:  *!                 : REFRESH()          (function in
AP_PRMCT.SPR)
  475:  *!                 : MBROW              (procedure in
MO_PDLAB.SPR)
  476:  *!
  477:
*!*********************************************************************
**************
  478:      PROCEDURE revisar
  479:          ******************
  480:          DEFINE WINDOW revisar FROM 0,0 TO 10,40 TITLE
'REVISANDO SISTEMAS DE ROL ..!' FONT "Times New Roman", 10 STYLE
```

```
"BIT"
  481:
  482:          MOVE WIND revisar TO 3.250, 20.857
  483:          ACTI WIND revisar IN WIND aact_sst
  484:          BROW FIELD sst.cod_ciu:H='CODIGO', sst.ciudad :H =
'SISTEMA', tsst.detalle:H='TIPO DE ROL';
  485:             NOEDIT NODELE NOMENU
  486:          RELE WIND revisar
  487:          reg_ant = RECNO()
  488:          DO mscatter
  489:          SHOW GETS
  490: <=======RETURN
  491:
  492:          ****************
  493:
*!*************************************************************************
**************
  494:  *!
  495:  *!       Procedure: BUSCAR
  496:  *!
  497:  *!       Called by: SECC()              (function in
PERSONAL.PRG)
  498:  *!                : _R390JW4G3()        (function in
DSC_GRTR.SPR)
  499:  *!                : _RA81B8ER0()        (function in
ACT_PTMO.SPR)
  500:  *!                : _RA81B8GQD()        (function in
ACT_PTMO.SPR)
  501:  *!                : _R6108FBLN()        (function in
MO_PDLAB.SPR)
  502:  *!                : _R5K0M9Z09()        (function in
DSC_CTA.SPR)
  503:  *!
  504:  *!           Calls: VAL_BUSCAR          (procedure in
AACT_SST.SPR)
  505:  *!                : BUSC_CLV()          (function in
DGP_CARG.SPR)
  506:  *!                : DESPLEGA            (procedure in
PERSONAL.PRG)
  507:  *!                : BUSCARP()           (function in
ACT_PTMO.SPR)
  508:  *!                : REF_SCR             (procedure in
MO_PDLAB.SPR)
  509:  *!                : MENSAJES            (procedure in
MO_PDLAB.SPR)
  510:  *!                : ASIGNAM             (procedure in
MO_PDLAB.SPR)
  511:  *!
  512:
*!*************************************************************************
**************
  513:     PROCEDURE buscar
  514:         ****************
  515:         ACTIVATE WINDOW buscar
```

```
  560:  *!                : DE_IEJUB.PRG
  561:  *!
  562:  *!          Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
  563:  *!
  564:
*!*************************************************************
**************
  565:      PROCEDURE salir
  566:          ***************
  567:          RELE WIND waiting
  568:          RELE WIND buscar
  569:          DO deac_keys
  570:          CLEAR READ
  571: <=======RETURN
  572:
  573:          ******************
  574:
*!*************************************************************
**************
  575:  *!
  576:  *!       Procedure: ACT_KEYS
  577:  *!
  578:  *!       Called by: AACT_SST.SPR
  579:  *!                : AACT_DPT.SPR
  580:  *!                : AACT_SEC.SPR
  581:  *!                : DGP_CARG.SPR
  582:  *!                : AC_IMPRT.SPR
  583:  *!                : DSC_GRTR.SPR
  584:  *!                : ELIMINAR          (procedure in
AACT_SST.SPR)
  585:  *!                : REVISAR           (procedure in
AACT_SST.SPR)
  586:  *!                : CTR_POSP          (procedure in
AACT_SST.SPR)
  587:  *!                : GRABAR            (procedure in
AACT_SST.SPR)
  588:  *!                : CANCELAR          (procedure in
AACT_SST.SPR)
  589:  *!                : ROL_Q1.SPR
  590:  *!                : ROL_Q2.SPR
  591:  *!                : ACT_PTMO.SPR
  592:  *!                : MO_PDLAB.SPR
  593:  *!                : DSC_CTA.SPR
  594:  *!                : _R6108FB17()      (function in
MO_PDLAB.SPR)
  595:  *!
  596:  *!          Calls: CTR_POSP           (procedure in
AACT_SST.SPR)
  597:  *!
  598:
*!*************************************************************
**************
  599:      PROCEDURE act_keys
```

```
  600:          ******************
  601:          ON KEY LABEL pgup DO ctr_posp WITH
'ANTERIOR','SKIP -1'
  602:          ON KEY LABEL pgdn DO ctr_posp WITH
'SIGUIENTE','SKIP'
  603:          ON KEY LABEL home DO ctr_posp WITH 'PRIMERO','GOTO
TOP'
  604:          ON KEY LABEL END DO ctr_posp WITH 'ULTIMO','GOTO
BOTTOM'
  605: <========RETURN
  606:
  607:          ********************
  608:
*!*************************************************************************
**************
  609:  *!
  610:  *!       Procedure: DEAC_KEYS
  611:  *!
  612:  *!       Called by: EDITAR              (procedure in
CTRL_RES.SPR)
  613:  *!                : SALIR               (procedure in
CTRL_RES.SPR)
  614:  *!                : CREAR               (procedure in
AACT_SST.SPR)
  615:  *!                : ELIMINAR            (procedure in
AACT_SST.SPR)
  616:  *!                : REVISAR             (procedure in
AACT_SST.SPR)
  617:  *!                : CTR_POSP            (procedure in
AACT_SST.SPR)
  618:  *!                : _R6108FBI7()        (function in
MO_PDLAB.SPR)
  619:  *!
  620:
*!*************************************************************************
**************
  621:      PROCEDURE deac_keys
  622:          ********************
  623:          ON KEY LABEL pgup
  624:          ON KEY LABEL pgdn
  625:          ON KEY LABEL home
  626:          ON KEY LABEL END
  627: <========RETURN
  628:
  629:          ******************
  630:
*!*************************************************************************
**************
  631:  *!
  632:  *!       Procedure: CTR_POSP
  633:  *!
  634:  *!       Called by: ACT_KEYS            (procedure in
AACT_SST.SPR)
  635:  *!                : ELIMINAR            (procedure in
```

```
AACT_SST.SPR)
  636:  *!
  637:  *!          Calls: MSCATTER          (procedure in
AACT_SST.SPR)
  638:  *!               : DEAC_KEYS         (procedure in
AACT_SST.SPR)
  639:  *!               : ASIGNAM           (procedure in
MO_PDLAB.SPR)
  640:  *!               : REF_SCR           (procedure in
MO_PDLAB.SPR)
  641:  *!               : ACT_KEYS          (procedure in
AACT_SST.SPR)
  642:  *!
  643:
*!*************************************************************
**************
  644:      PROCEDURE ctr_posp
  645:          ******************
  646:          PARAMETER opc_posp, comando
  647:        ı=DO CASE
  648:        ã═══CASE BOF() AND comando = 'SKIP -1'
  649:        │       GOTO TOP
  650:        ã═══CASE EOF() AND comando = 'SKIP'
  651:        │       GOTO BOTTOM
  652:        ã═══OTHERWISE
  653:        │       &comando
  654:        ê=ENDCASE
  655:        ı=DO CASE
  656:        ã═══CASE BOF()
  657:        │       GOTO TOP
  658:        │       WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
  659:        ã═══CASE EOF()
  660:        │       GOTO BOTTOM
  661:        │       WAIT WIND NOWAIT 'FIN DE ARCHIVO'
  662:        ê=ENDCASE
  663:        ı=DO CASE
  664:        ã═══CASE opc_posp = 'PRIMERO'
  665:        │       WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
  666:        ã═══CASE opc_posp = 'SIGUIENTE'
  667:        │       ┌─IF EOF()
  668:        │       │   GO BOTT
  669:        │       │   WAIT WIND NOWAIT 'FIN DE ARCHIVO'
  670:        │       └─ENDIF
  671:        ã═══CASE opc_posp = 'ANTERIOR'
  672:        │       ┌─IF BOF()
  673:        │       │   GOTO TOP
  674:        │       │   WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
  675:        │       └─ENDIF
  676:        ã═══CASE opc_posp = 'ULTIMO'
  677:        │       WAIT WIND NOWAIT 'FIN DE ARCHIVO'
  678:        ê=ENDCASE
  679:          DO mscatter
  680:          SHOW GETS
  681:  <═══════RETURN
```

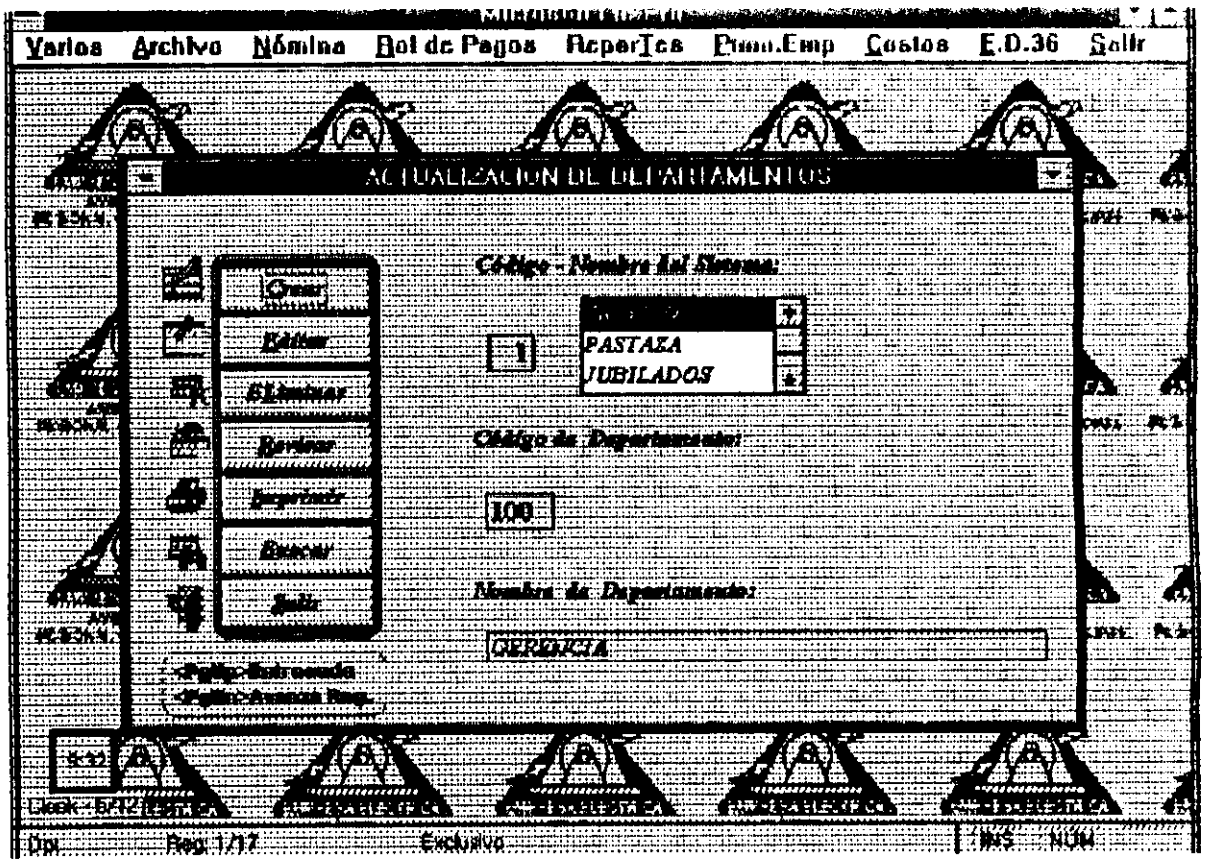
```
   682:
   683:          ****************
   684:
 *!*************************************************************************
 **************
   685:  *!
   686:  *!       Procedure: GRABAR
   687:  *!
   688:  *!       Called by: ING                  (procedure in
 PERSONAL.PRG)
   689:  *!                : MERCE()              (function in
 PERSONAL.PRG)
   690:  *!                : JUBILAR              (procedure in
 PERSONAL.PRG)
   691:  *!
   692:  *!           Calls: ACT_KEYS            (procedure in
 AACT_SST.SPR)
   693:  *!                : ELIM_MPL             (procedure in
 MO_PDLAB.SPR)
   694:  *!                : INSERTAR()           (function in
 MO_PDLAB.SPR)
   695:  *!                : BORRA_TMP            (procedure in
 MO_PDLAB.SPR)
   696:  *!                : REF_SCR              (procedure in
 MO_PDLAB.SPR)
   697:  *!
   698:  *!            Uses: TMP.DBF
   699:  *!
   700:
*!*************************************************************************
**************
   701:      PROCEDURE grabar
   702:          ****************
   703:         IF creando
   704:            APPEN BLANK
   705:         ENDIF
   706:          reg_ant = RECNO()
   707:          GATHER MEMVAR
   708:          SHOW GETS
   709:          SHOW GET opc_scr,1 PROMPT '|<Crear'
   710:          SHOW GET opc_scr,2 PROMPT '|<Editar'
   711:          SHOW GET opc_scr ENABLE
   712:          DO act_keys
   713: <========RETURN
   714:
   715:          ******************
   716:
*!*************************************************************************
**************
   717:  *!
   718:  *!       Procedure: CANCELAR
   719:  *!
   720:  *!       Called by: ING                  (procedure in
PERSONAL.PRG)
```

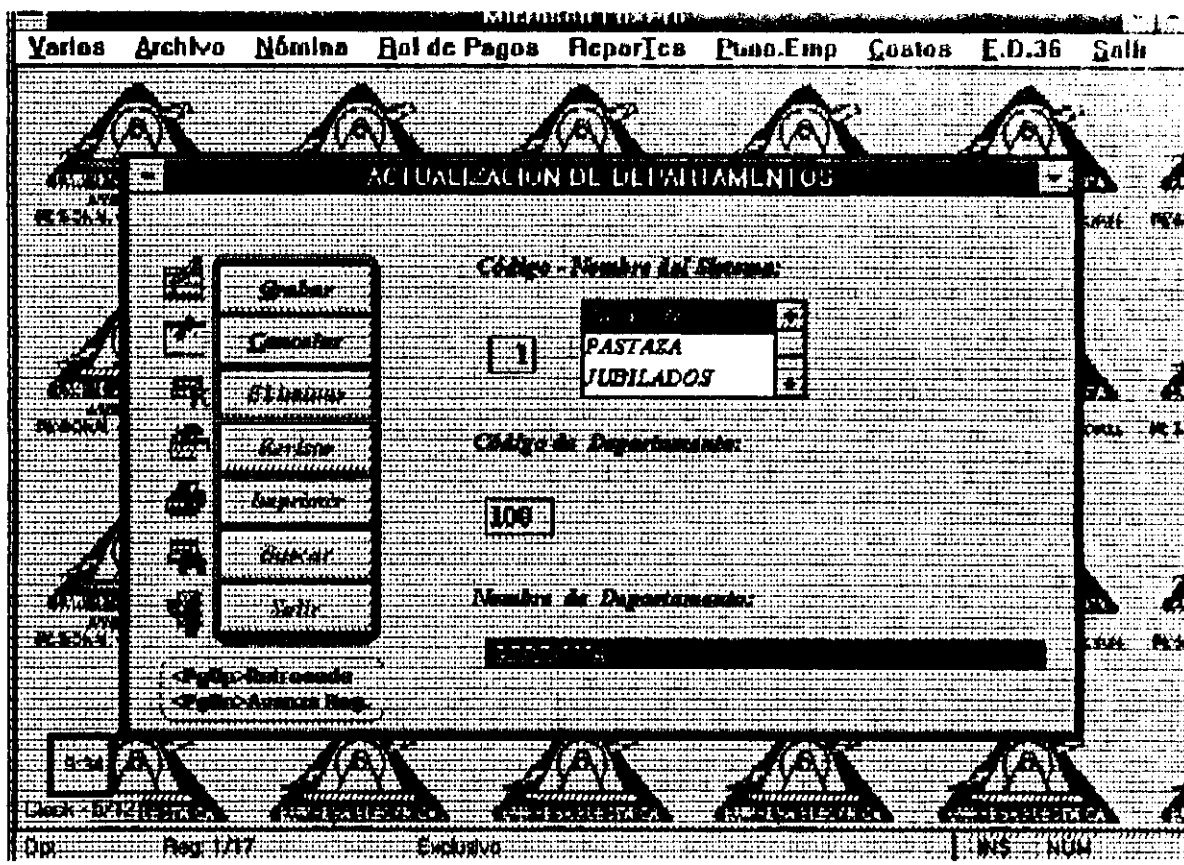
```
  721:  *!                : JUBILAR              (procedure in
PERSONAL.PRG)
  722:  *!
  723:  *!           Calls: MSCATTER             (procedure in
AACT_SST.SPR)
  724:  *!                : ACT_KEYS             (procedure in
AACT_SST.SPR)
  725:  *!                : DESPLEGA             (procedure in
PERSONAL.PRG)
  726:  *!                : BORRA_TMP            (procedure in
MO_PDLAB.SPR)
  727:  *!                : ASIGNAM              (procedure in
MO_PDLAB.SPR)
  728:  *!                : REF_SCR              (procedure in
MO_PDLAB.SPR)
  729:  *!
  730:
*!*************************************************************
**************
  731:      PROCEDURE cancelar
  732:          ******************
  733:          GO reg_ant
  734:          DO mscatter
  735:          SHOW GETS
  736:          SHOW GET opc_scr,1 PROMPT '\<Crear'
  737:          SHOW GET opc_scr,2 PROMPT '\<Editar'
  738:          SHOW GET opc_scr ENABLE
  739:          DO act_keys
  740: <=======RETURN
  741:
  742:          ******************
  743:
*!*************************************************************
**************
  744:  *!
  745:  *!       Procedure: MSCATTER
  746:  *!
  747:  *!       Called by: AACT_SST.SPR
  748:  *!                : AACT_DPT.SPR
  749:  *!                : AACT_SEC.SPR
  750:  *!                : ELIMINAR             (procedure in
AACT_SST.SPR)
  751:  *!                : REVISAR              (procedure in
AACT_SST.SPR)
  752:  *!                : VAL_BUSCAR           (procedure in
AACT_SST.SPR)
  753:  *!                : CTR_POSP             (procedure in
AACT_SST.SPR)
  754:  *!                : CANCELAR             (procedure in
AACT_SST.SPR)
  755:  *!
  756:
*!*************************************************************
**************
```

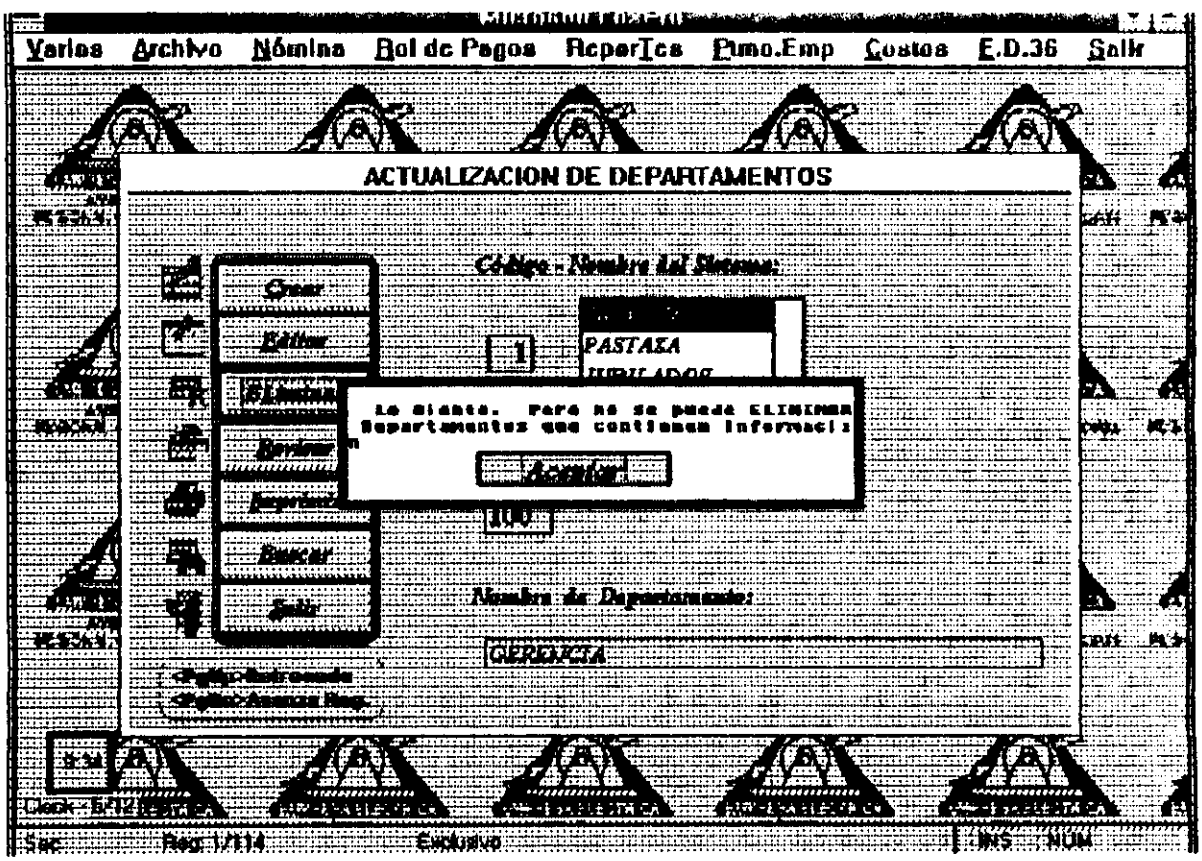
```
  757:      PROCEDURE mscatter
  758:          ******************
  759:          SCATTER MEMVAR
  760:          STOR m.tipo + 1 TO m.tipo
  761: <=======RETURN
  762:
  763:          *
************************************************************
  764:          *        *
  765:          *        * _R9R163S5N            opc_scr VALID
  766:          *        *
  767:          *        * Function Origin:
  768:          *        *
  769:          *        * From Platform:        Windows
  770:          *        * From Screen:          AACT_SST,
Record Number:    6
  771:          *        * Variable:             opc_scr
  772:          *        * Called By:            VALID Clause
  773:          *        * Object Type:          Push Button
  774:          *        * Snippet Number:       1
  775:          *        *
  776:          *
************************************************************
  777:          *
  778:
*!*************************************************************************
**************
  779:  *!
  780:  *!        Function: _R9R163S5N
  781:  *!
  782:  *!      Called by: AACT_SST.SPR
  783:  *!
  784:  *!           Calls: &OPC_SCR.PRG
  785:  *!
  786:
*!*************************************************************************
**************
  787:      FUNCTION _r9r163s5n      &&  opc_scr VALID
  788:          #REGION 1
  789:          DO &opc_scr
  790:
  791:          *
************************************************************
  792:          *        *
  793:          *        * _R9R163SSE            m.cod_ciu VALID
  794:          *        *
  795:          *        * Function Origin:
  796:          *        *
  797:          *        * From Platform:        Windows
  798:          *        * From Screen:          AACT_SST,
Record Number:   17
  799:          *        * Variable:             m.cod_ciu
  800:          *        * Called By:            VALID Clause
  801:          *        * Object Type:          Field
```

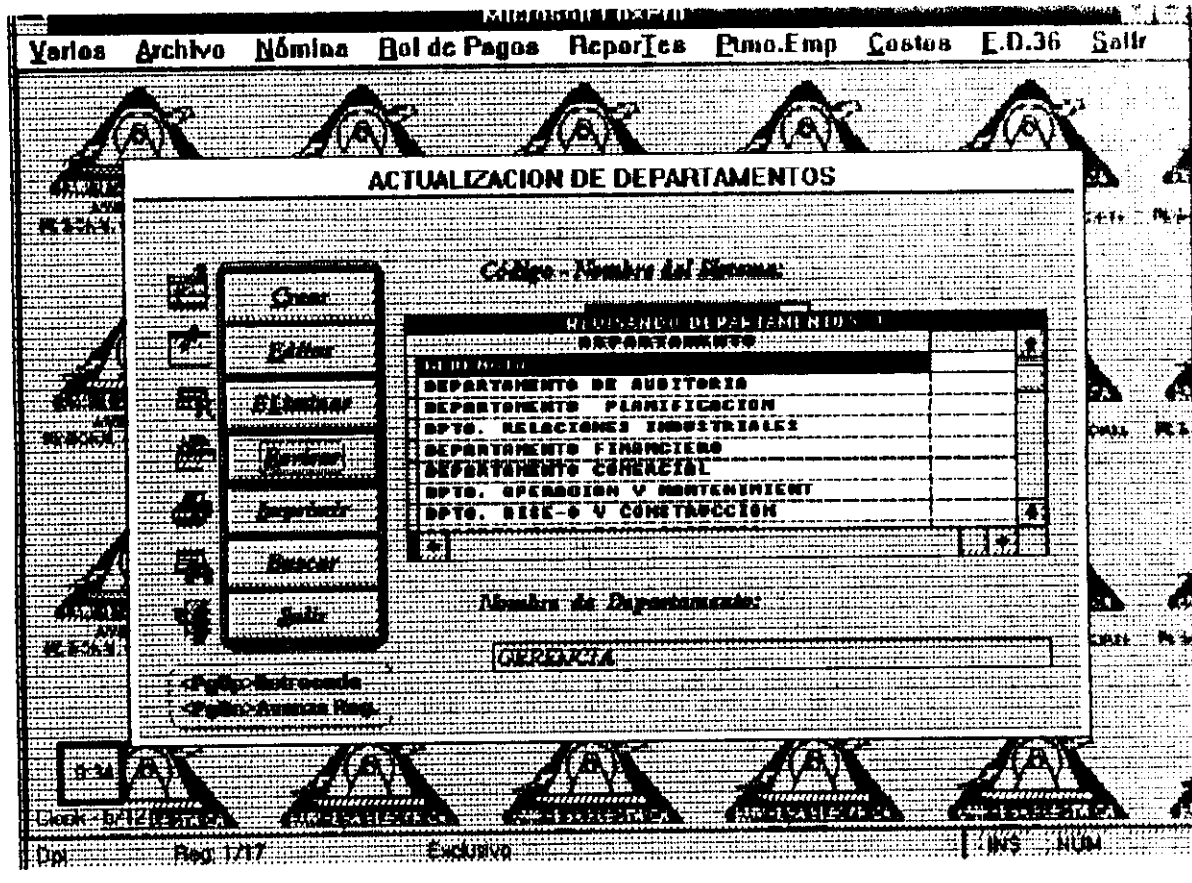
```
  802:          *        * Snippet Number:       2
  803:          *        *
  804:          *
************************************************************
  805:          *
  806:
*!*************************************************************************
**************
  807:  *!
  808:  *!        Function: _R9R163SSE
  809:  *!
  810:  *!       Called by: AACT_SST.SPR
  811:  *!
  812:
*!*************************************************************************
**************
  813:      FUNCTION _r9r163sse      &&  m.cod_ciu VALID
  814:          #REGION 1
  815:          SEEK m.cod_ciu
  816: <=======RETURN !FOUND()
  817:
  818:
  819:          *
************************************************************
  820:          *        *
  821:          *        * _R9R163SWQ            m.tipo VALID
  822:          *        *
  823:          *        * Function Origin:
  824:          *        *
  825:          *        * From Platform:        Windows
  826:          *        * From Screen:          AACT_SST,
Record Number:   19
  827:          *        * Variable:             m.tipo
  828:          *        * Called By:            VALID Clause
  829:          *        * Object Type:          List
  830:          *        * Snippet Number:       3
  831:          *        *
  832:          *
************************************************************
  833:          *
  834:
*!*************************************************************************
**************
  835:  *!
  836:  *!        Function: _R9R163SWQ
  837:  *!
  838:  *!       Called by: AACT_SST.SPR
  839:  *!
  840:
*!*************************************************************************
**************
  841:      FUNCTION _r9r163swq      &&  m.tipo VALID
  842:          #REGION 1
  843:          STOR tsst.tipo TO m.tipo
```

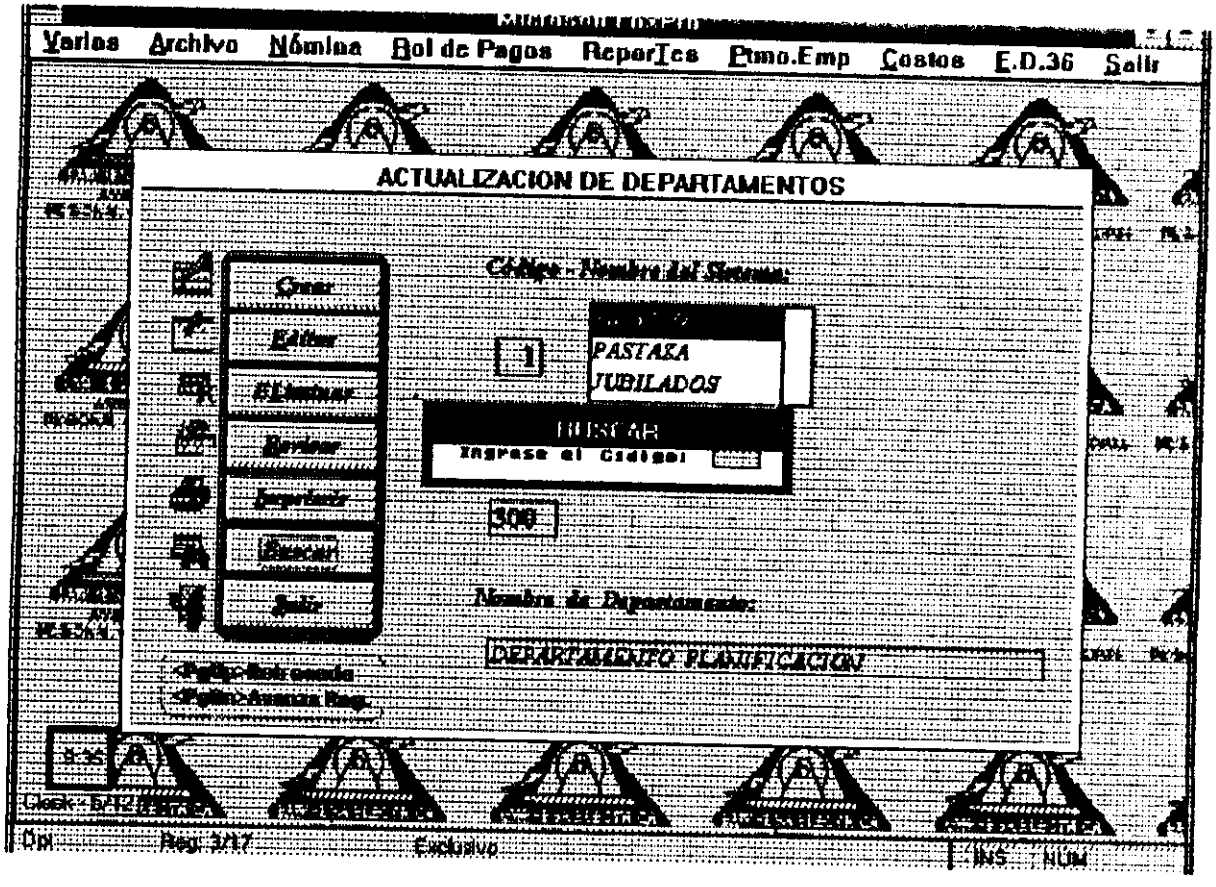
```
  516:          reg_ant = RECNO()
  517:          STORE 0 TO mcod_ciu
  518:          @ 1,1 SAY 'Ingrese el Cσdigo: ' GET mcod_ciu PICT
'9' VALID val_buscar(mcod_ciu) ERROR 'Cσdigo no Existe'
  519:          READ
  520:          SHOW GETS
  521:          DEACTIVATE WINDOW buscar
  522: <=======RETURN
  523:
  524:          ********************
  525:
*!*************************************************************************
**************
  526:  *!
  527:  *!       Procedure: VAL_BUSCAR
  528:  *!
  529:  *!       Called by: BUSCAR              (procedure in
AACT_SST.SPR)
  530:  *!
  531:  *!           Calls: MSCATTER            (procedure in
AACT_SST.SPR)
  532:  *!
  533:
*!*************************************************************************
**************
  534:      PROCEDURE val_buscar
  535:          ********************
  536:          PARAMETER mcod_ciu
  537:          STOR .T. TO vret
  538:        ┌─IF LAST() = 27
  539: <======┼═══RETURN TO buscar
  540:        └─ENDIF
  541:          SEEK mcod_ciu
  542:        ┌─IF !FOUND()
  543:        │   GOTO reg_ant
  544:        │   STOR .F. TO vret
  545:        └─ENDIF
  546:          DO mscatter
  547:          reg_ant = RECNO()
  548: <=======RETURN vret
  549:
  550:          ***************
  551:
*!*************************************************************************
**************
  552:  *!
  553:  *!       Procedure: SALIR
  554:  *!
  555:  *!       Called by: _RAA1732TG()        (function in
AT_ESCLS.SPR)
  556:  *!                : DA_ASISG.PRG
  557:  *!                : DE_EGRQ1.PRG
  558:  *!                : DE_EGRQ2.PRG
  559:  *!                : DE_EGRMN.PRG
```


Varios Archivo Nómina Bol de Pagos ReperTes Plan.Emp Costos E.D.36 Salir
ACTUALIZACION DE DEPARTAMENTOS
Crear
Editar
Eliminar
Revisar
Imprimir
Buscar
Salir
Código - Nombre del Sistema:
1
PASTAZA
JUBILADOS
Código de Departamento:
100
Nombre de Departamento:
GERENCIA
Reg: 1/17
Exclusivo
NUM


Varios Archivo Nómina Rol de Pagos Reportes Pano.Emp Costos E.D.36 Salir
ACTUALIZACION DE DEPARTAMENTOS
Código - Nombre del Sistema:
1
PASTAZA
JUBILADOS
Código de Departamento:
109
Nombre de Departamento:
Grabar
Cancelar
Eliminar
Consulta
Imprimir
Buscar
Salir
Disp
Reg: 1/17
Exclusivo
MAYS
NUM


Varios Archivo Nómina Bol de Pagos ReporTes Pmo.Emp Costos E.D.36 Salir
ACTUALIZACION DE DEPARTAMENTOS
Crear
Editar
Eliminar
Reportes
Buscar
Salir
1
PASTASA
Nombre de Departamento:
GERENCIA


Varios Archivo Nómina Bol de Pagos Reportes Pmo.Emp Costos E.D.36 Salir
ACTUALIZACION DE DEPARTAMENTOS
Crear
Editar
Eliminar
Imprimir
Buscar
Salir
DEPARTAMENTO DE AUDITORIA
DEPARTAMENTO PLANIFICACION
DPTO. RELACIONES INDUSTRIALES
DEPARTAMENTO FINANCIERO
DEPARTAMENTO COMERCIAL
DPTO. OPERACION Y MANTENIMIENT
DPTO. DISE-O Y CONSTRUCCION
GERENCIA
Dpi
Reg 1/17
Exclusivo


Varios Archivo Nómina Bol de Pagos ReporTes Ptmo.Emp Costos E.D.36 Salir
ACTUALIZACION DE DEPARTAMENTOS
Crear
Editar
Eliminar
Buscar
Imprimir
Buscar
Salir
Código - Nombre del Sistema:
1
PASTAZA
JUBILADOS
BUSCAR
Ingrese el Código:
309
Nombre de Departamento:
DEPARTAMENTO PLANIFICACION
Opt
Reg: 3/17
Exclusivo
INS
NUM

```
    1:
*:*************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\AACT_DPT.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrel Ambato - Ecuador
    7:  *:  Last modified: 03/05/1995 at 21:27:22
    8:  *:
    9:  *:  Procs & Fncts: _RA419ZKEV()
   10:  *:                : _RA419ZKI9()
   11:  *:                : _RA419ZKLY()
   12:  *:                : _RA419ZKOO()
   13:  *:
   14:  *:          Set by: EMPRESA.MPR
   15:  *:
   16:  *:           Calls: MSCATTER            (procedure in
AACT_SST.SPR)
   17:  *:                : ACT_KEYS            (procedure in
AACT_SST.SPR)
   18:  *:                : _RA419ZKEV()        (function in
AACT_DPT.SPR)
   19:  *:                : _RA419ZKI9()        (function in
AACT_DPT.SPR)
   20:  *:                : _RA419ZKLY()        (function in
AACT_DPT.SPR)
   21:  *:                : _RA419ZKOO()        (function in
AACT_DPT.SPR)
   22:  *:
   23:  *:       Documented 19:14:26
FoxDoc version 3.00a
   24:
*:*************************************************************
**************
   25:      *
************************************************************
   26:      *        *
   27:      *        * 03/05/1995           AACT_DPT.SPR
 21:27:19
   28:      *        *
   29:      *
************************************************************
   30:      *        *
   31:      *        * Nombre del autor
   32:      *        *
   33:      *        * Copyright (c) 1996 Nombre de la empresa
   34:      *        * Direcci¥n
   35:      *        * Ciudad,     C.P.
   36:      *        * Pa¥s
   37:      *        *
   38:      *        * Description:
   39:      *        * Este programa lo ha generado autom®ticamente
GENSCRN.
   40:      *        *
   41:      *
```

```
*************************************************************
   42:
   43:
   44:      *
*************************************************************
   45:      *        *
   46:      *        *  AACT_DPT/Windows Código de configuración -
SECCION 1
   47:      *        *
   48:      *
*************************************************************
   49:      *
   50:
   51:      #REGION 1
   52:      **********
   53:      REGIONAL opc_scr, creando, editando, reg_ant
   54:
   55:      STOR ''    TO opc_scr
   56:      STOR .F. TO creando, editando
   57:
   58:      **********
   59:
   60:      #REGION 0
   61:      REGIONAL m.currarea, m.talkstat, m.compstat
   62:
   63:    ┌─IF SET("TALK") = "ON"
   64:    │    SET TALK OFF
   65:    │    m.talkstat = "ON"
   66:    ├─ELSE
   67:    │    m.talkstat = "OFF"
   68:    └─ENDIF
   69:      m.compstat = SET("COMPATIBLE")
   70:      SET COMPATIBLE FOXPLUS
   71:
   72:      m.rborder = SET("READBORDER")
   73:      SET readborder ON
   74:
   75:      m.currarea = SELECT()
   76:
   77:
   78:      *
*************************************************************
   79:      *        *
   80:      *        *  AACT_DPT/Windows Bases de datos, Yndices,
relaciones
   81:      *        *
   82:      *
*************************************************************
   83:      *
   84:
   85:    ┌─IF USED("sst")
   86:    │    SELECT sst
   87:    │    SET ORDER TO TAG "sist_cod"
   88:    ├─ELSE
   89:    │    SELECT 0
   90:    │    USE (LOCFILE("srol\dbfs\sistema.dbf","DBF","Buscar
sistema?"));
```

```
 91:         AGAIN ALIAS sst ;
 92:         ORDER TAG "sist_cod"
 93:     ENDIF
 94:
 95:     IF USED("dpt")
 96:         SELECT dpt
 97:         SET ORDER TO TAG "dpto_cod"
 98:     ELSE
 99:         SELECT 0
100:         USE (LOCFILE("sroll\dbfs\departam.dbf","DBF","Buscar
departam?"));
101:         AGAIN ALIAS dpt ;
102:         ORDER TAG "dpto_cod"
103:     ENDIF
104:
105:     IF USED("sec")
106:         SELECT sec
107:         SET ORDER TO TAG "cod_dpto"
108:     ELSE
109:         SELECT 0
110:         USE (LOCFILE("sroll\dbfs\seccion.dbf","DBF","Buscar
seccion?"));
111:         AGAIN ALIAS sec ;
112:         ORDER TAG "cod_dpto"
113:     ENDIF
114:
115:     SELECT sst
116:
117:     SELECT dpt
118:     SELECT dpt
119:
120:     SET RELATION OFF INTO sst
121:     SET RELATION TO cod_ciu INTO sst ADDITIVE
122:
123:     SELECT sst
124:
125:     *
**********************************************************
126:     *        *
127:     *        *              Windows Definiciones de ventana
128:     *        *
129:     *
**********************************************************
130:     *
131:
132:     IF NOT WEXIST("aact_dpt") ;
133:         OR UPPER(WTITLE("AACT_DPT")) == "AACT_DPT.PJX" ;
134:         OR UPPER(WTITLE("AACT_DPT")) == "AACT_DPT.SCX" ;
135:         OR UPPER(WTITLE("AACT_DPT")) == "AACT_DPT.MNX" ;
136:         OR UPPER(WTITLE("AACT_DPT")) == "AACT_DPT.PRG" ;
137:         OR UPPER(WTITLE("AACT_DPT")) == "AACT_DPT.FRX" ;
138:         OR UPPER(WTITLE("AACT_DPT")) == "AACT_DPT.QPR"
139:         DEFINE WINDOW aact_dpt ;
140:         AT  0.000, 0.000  ;
141:         SIZE 15.100,72.571 ;
142:         TITLE "ACTUALIZACION DE DEPARTAMENTOS" ;
143:         FONT "Times New Roman", 12 ;
```

```
  144:          STYLE "BI" ;
  145:          FLOAT ;
  146:          NOCLOSE ;
  147:          MINIMIZE ;
  148:          SYSTEM ;
  149:          COLOR RGB(,,,192,192,192)
  150:       MOVE WINDOW aact_dpt CENTER
  151:    ENDIF
  152:
  153:
  154:    *
************************************************************
  155:    *        *
  156:    *        *  AACT_DPT/Windows Código de configuración -
SECCION 2
  157:    *        *
  158:    *
************************************************************
  159:    *
  160:
  161:    #REGION 1
  162:    **********
  163:    SELE dpt
  164:    SCATTER MEMVAR BLANK
  165:    DO mscatter
  166:    STOR RECNO() TO reg_ant
  167:    DO act_keys
  168:    IF NOT WEXIST('waiting')
  169:       DEFI WIND waiting FROM 0,0 TO 8,40 PANEL
  170:       MOVE WIND waiting CENTER
  171:    ENDIF
  172:    IF NOT WEXIST('BUSCAR') && o definir en el principal
  173:       DEFINE WINDOW buscar FROM 0,0 TO 3,28 TITLE
'BUSCAR' SYSTEM
  174:       MOVE WIND buscar CENTER
  175:    ENDIF
  176:    DEFINE WINDOW revisar FROM 0,0 TO 15,50 TITLE
'REVISANDO DEPARTAMENTOS ..!'
  177:
  178:    MOVE WINDOW revisar CENTER
  179:
  180:
  181:    #REGION 1
  182:    DEFINE POPUP _ra419zjtk ;
  183:       PROMPT FIELD sst.ciudad ;
  184:       SCROLL
  185:
  186:    *
************************************************************
  187:    *        *
  188:    *        *        AACT_DPT/Windows Distribución de
pantalla
  189:    *        *
  190:    *
************************************************************
  191:    *
  192:
```

```
  193:      #REGION 1
  194:     ┌─IF WVISIBLE("aact_dpt")
  195:     │     ACTIVATE WINDOW aact_dpt SAME
  196:     ├─ELSE
  197:     │     ACTIVATE WINDOW aact_dpt NOSHOW
  198:     └─ENDIF
  199:      @ 1.700,6.429 TO 12.700,19.286 ;
  200:        PATTERN 1 ;
  201:        PEN 1, 8 ;
  202:        STYLE "16" ;
  203:        COLOR RGB(,,,128,128,0)
  204:      @ 3.400,2.571 SAY
(LOCFILE("bitm\program.bmp","BMP|ICO|PCT|ICN", "┐Dónde está
program?" )) BITMAP ;
  205:        SIZE 1.250,3.571 ;
  206:        STRETCH ;
  207:        STYLE "T"
  208:      @ 5.000,2.857 SAY
(LOCFILE("bitm\wzdelete.bmp","BMP|ICO|PCT|ICN", "┐Dónde está
wzdelete?" )) BITMAP ;
  209:        SIZE 1.250,3.571 ;
  210:        STRETCH ;
  211:        STYLE "T"
  212:      @ 6.350,2.857 SAY
(LOCFILE("bitm\query.bmp","BMP|ICO|PCT|ICN", "┐Dónde está query?"
)) BITMAP ;
  213:        SIZE 1.250,3.571 ;
  214:        STRETCH ;
  215:        STYLE "T"
  216:      @ 8.050,2.429 SAY
(LOCFILE("bitm\wzprint.bmp","BMP|ICO|PCT|ICN", "┐Dónde está
wzprint?" )) BITMAP ;
  217:        SIZE 1.250,3.571 ;
  218:        STRETCH ;
  219:        STYLE "T"
  220:      @ 9.600,2.571 SAY
(LOCFILE("bitm\wzlocate.bmp","BMP|ICO|PCT|ICN", "┐Dónde está
wzlocate?" )) BITMAP ;
  221:        SIZE 1.250,3.571 ;
  222:        STRETCH ;
  223:        STYLE "T"
  224:      @ 11.200,2.286 SAY
(LOCFILE("bitm\close.bmp","BMP|ICO|PCT|ICN", "┐Dónde está close?"
)) BITMAP ;
  225:        SIZE 1.250,3.571 ;
  226:        STRETCH ;
  227:        STYLE "T"
  228:      @ 1.650,26.714 SAY "Código - Nombre del Sistema:" ;
  229:        FONT "Times New Roman", 10 ;
  230:        STYLE "BIT"
  231:      @ 6.650,26.429 SAY "Código de  Departamento:" ;
  232:        FONT "Times New Roman", 10 ;
  233:        STYLE "BIT"
  234:      @ 2.000,6.857 GET opc_scr ;
  235:        PICTURE "@*VN
|<Crear;|<Editar;E|<Liminar;|<Revisar;|<Imprimir;|<Buscar;|<Salir
" ;
```

```
  236:         SIZE 1.647,16.800,0.118 ;
  237:         DEFAULT 1 ;
  238:         FONT "Times New Roman", 10 ;
  239:         STYLE "BI" ;
  240:         VALID _ra419zkev()
  241:      @ 4.100,28.000 GET m.cod_ciu ;
  242:         SIZE 0.950,2.857 ;
  243:         RANGE 1, 8 ;
  244:         DEFAULT 0 ;
  245:         FONT "Times New Roman", 12 ;
  246:         STYLE "B" ;
  247:         PICTURE "@K 9" ;
  248:         WHEN creando ;
  249:         VALID _ra419zki9() ;
  250:         ERROR ' CODIGO DE SISTEMA NO EXISTE '
  251:      @ 3.000,34.857 GET sst.ciudad ;
  252:         PICTURE "@&N" ;
  253:         POPUP _ra419zjtk ;
  254:         SIZE 3.375,23.800 ;
  255:         DEFAULT " " ;
  256:         FONT "Times New Roman", 9 ;
  257:         STYLE "BI" ;
  258:         WHEN .F. ;
  259:         VALID _ra419zkly()
  260:      @ 8.650,27.714 GET m.cod_dep ;
  261:         SIZE 0.950,4.571 ;
  262:         DEFAULT " " ;
  263:         FONT "Times New Roman", 12 ;
  264:         STYLE "B" ;
  265:         PICTURE "@KT 999" ;
  266:         WHEN creando ;
  267:         VALID _ra419zkoo() ;
  268:         ERROR ' CODIGO DE DEPARTAMENTO YA HA SIDO
REGISTRADO'
  269:      @ 12.600,27.714 GET m.nombre ;
  270:         SIZE 0.938,59.200 ;
  271:         DEFAULT " " ;
  272:         FONT "Times New Roman", 9 ;
  273:         STYLE "BI" ;
  274:         PICTURE "@K!" ;
  275:         WHEN creando OR editando
  276:      @ 13.250,3.429 SAY "<PgUp>Retrocede" + CHR(13) + ;
  277:         "<PgDn>Avanza Reg." ;
  278:         SIZE 2.000,22.800, 0.000 ;
  279:         FONT "Arial", 8 ;
  280:         STYLE "BT"
  281:      @ 13.050,2.429 TO 14.900,19.715 ;
  282:         PEN 1, 8 ;
  283:         STYLE "16" ;
  284:         COLOR RGB(128,128,0,,,,)
  285:      @ 1.800,1.714 SAY
(LOCFILE("bitm\memo.bmp","BMP|ICO|PCT|ICN", "¬Dónde está memo?"
)) BITMAP ;
  286:         SIZE 1.250,5.286 ;
  287:         STRETCH ;
  288:         STYLE "T"
  289:      @ 11.000,26.429 SAY "Nombre  de  Departamento:" ;
```

```
290:         FONT "Times New Roman", 10 ;
291:         STYLE "BIT"
292:
293:     IF NOT WVISIBLE("aact_dpt")
294:         ACTIVATE WINDOW aact_dpt
295:     ENDIF
296:
297:      READ CYCLE ;
298:         VALID .F.
299:
300:      RELEASE WINDOW aact_dpt
301:
302:      *
************************************************************
303:      *        *
304:      *        *              Windows Cerrando bases de datos
305:      *        *
306:      *
************************************************************
307:      *
308:
309:     IF USED("sst")
310:         SELECT sst
311:         USE
312:     ENDIF
313:
314:     IF USED("dpt")
315:         SELECT dpt
316:         USE
317:     ENDIF
318:
319:     IF USED("sec")
320:         SELECT sec
321:         USE
322:     ENDIF
323:
324:      SELECT (m.currarea)
325:
326:      RELEASE POPUPS _ra419zjtk
327:
328:      #REGION 0
329:
330:      SET readborder &rborder
331:
332:     IF m.talkstat = "ON"
333:         SET TALK ON
334:     ENDIF
335:     IF m.compstat = "ON"
336:         SET COMPATIBLE ON
337:     ENDIF
338:
339:
340:      *
************************************************************
341:      *        *
342:      *        *            AACT_DPT/Windows Código de
limpieza
```

```
  343:      *        *
  344:      *
 ************************************************************
  345:      *
  346:
  347:      #REGION 1
  348:      ***************
  349:
  350:
  351:      *
 ************************************************************
  352:      *        *
  353:      *        * AACT_DPT/Windows Procedimientos y funciones
de soporte
  354:      *        *
  355:      *
 ************************************************************
  356:      *
  357:
  358:      #REGION 1
  359:
*!*************************************************************************
**************
  360:   *!
  361:   *!       Procedure: CREAR
  362:   *!
  363:   *!           Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
  364:   *!                : BUSC_CLV()         (function in
DGP_CARG.SPR)
  365:   *!                : EDITAR             (procedure in
CTRL_RES.SPR)
  366:   *!                : PROCESO            (procedure in
DSC_GRTR.SPR)
  367:   *!                : INGRESO            (procedure in
ACT_PTMO.SPR)
  368:   *!                : CREA_TMP           (procedure in
MO_PDLAB.SPR)
  369:   *!                : MBROW              (procedure in
MO_PDLAB.SPR)
  370:   *!                : MENSAJES           (procedure in
MO_PDLAB.SPR)
  371:   *!                : REF_SCR            (procedure in
MO_PDLAB.SPR)
  372:   *!
  373:
*!*************************************************************************
**************
  374:      PROCEDURE crear
  375:          ***************
  376:          DO deac_keys
  377:          reg_ant = RECNO()
  378:          SCATTER MEMV BLANK
  379:          STOR .T. TO creando
  380:          _CUROBJ = OBJNUM(m.cod_ciu)
  381:          SHOW GET opc_scr DISABLE
  382:          SHOW GET opc_scr,1 PROMPT '\<Grabar' ENABLE
```

```
  383:          SHOW GET opc_scr,2 PROMPT '|<Cancelar' ENABLE
  384: <========RETURN
  385:
  386:          ****************
  387:
*!*************************************************************************
**************
  388:  *!
  389:  *!       Procedure: EDITAR
  390:  *!
  391:  *!       Called by: CREAR              (procedure in
AACT_SST.SPR)
  392:  *!
  393:  *!            Calls: DEAC_KEYS         (procedure in
AACT_SST.SPR)
  394:  *!                 : CABEZA            (procedure in
PERSONAL.PRG)
  395:  *!                 : INGRESO           (procedure in
ACT_PTMO.SPR)
  396:  *!                 : CREA_TMP          (procedure in
MO_PDLAB.SPR)
  397:  *!                 : MBROW             (procedure in
MO_PDLAB.SPR)
  398:  *!                 : REF_SCR           (procedure in
MO_PDLAB.SPR)
  399:  *!                 : MENSAJES          (procedure in
MO_PDLAB.SPR)
  400:  *!
  401:
*!*************************************************************************
**************
  402:      PROCEDURE editar
  403:          ****************
  404:          DO deac_keys
  405:          STOR .T. TO editando
  406:          _CUROBJ = OBJNUM(m.nombre)
  407:          SHOW GET opc_scr DISABLE
  408:          SHOW GET opc_scr,1 PROMPT '|<Grabar' ENABLE
  409:          SHOW GET opc_scr,2 PROMPT '|<Cancelar' ENABLE
  410: <========RETURN
  411:
  412:          ******************
  413:
*!*************************************************************************
**************
  414:  *!
  415:  *!       Procedure: ELIMINAR
  416:  *!
  417:  *!            Calls: MSCATTER          (procedure in
AACT_SST.SPR)
  418:  *!                 : DEAC_KEYS         (procedure in
AACT_SST.SPR)
  419:  *!                 : ACT_KEYS          (procedure in
AACT_SST.SPR)
  420:  *!                 : BORRAR            (procedure in
DSC_GRTR.SPR)
  421:  *!                 : ELIM_MPL          (procedure in
```

```
MO_PDLAB.SPR)
  422:  *!                  : CTR_POSP            (procedure in
AACT_SST.SPR)
  423:  *!
  424:
*!*****************************************************************
**************
  425:      PROCEDURE eliminar
  426:          ******************
  427:          SELECT sec
  428:          SEEK m.cod_dep
  429:          ACTIVATE WINDOW waiting
  430:          ?CHR(7)
  431:        ┌─IF FOUND()
  432:        │     @1,1 SAY ' Lo Siento.  Pero no se puede
ELIMINAR '
  433:        │     @2,1 SAY 'Departamentos que contienen
informaci¥n'
  434:        │     STOR 1 TO mcont
  435:        │     ?CHR(7)
  436:        │     @ 4,10 GET mcont;
  437:        │        PICTURE "@*h |<Aceptar" ;
  438:        │        SIZE 1,15 FONT 'R',12 STYLE "BI"
  439:        │     READ MODAL
  440:        ├─ELSE
  441:        │     @1,0 SAY " El Departamento: " + m.nombre
  442:        │     @2,0 SAY ' ser8 ELIMINADO ...!'
  443:        │     STOR 1 TO mcont
  444:        │     ?CHR(7)
  445:        │     @ 4, 8 GET mcont;
  446:        │        PICTURE "@*h |<Cancelar;|<Eliminar" ;
  447:        │        SIZE 1,10 FONT 'R',12 STYLE "BI"
  448:        │     READ MODAL
  449:        │   ┌─IF mcont = 2
  450:        │   │     WAIT( "Borrando Departamento ....!")WINDOW
TIME(1)
  451:        │   │     SELE dpt
  452:        │   │     DELE
  453:        │   │     GO TOP
  454:        │   │     DO mscatter
  455:        │   │     reg_ant = RECNO()
  456:        │   │     SHOW GETS
  457:        │   └─ENDIF
  458:        └─ENDIF
  459:          DEACT WIND waiting
  460:          SELECT dpt
  461:  <═══════RETURN
  462:          ******************
  463:
*!*****************************************************************
**************
  464:  *!
  465:  *!       Procedure: IMPRIMIR
  466:  *!
  467:  *!           Calls: MENSAJE2           (procedure in
ACT_PTMO.SPR)
  468:  *!
```

```
  469:  *!   Report Forms: DSC_GRTR.FRX
  470:  *!                : CTA_REP.FRX
  471:  *!
  472:
*!*************************************************************
**************
  473:      PROCEDURE imprimir
  474:          ******************
  475:          ?CHR(7)
  476:          WAIT WIND 'PROCESO NO PREPARADO'
  477: <=======RETURN
  478:          ******************
  479:
*!*************************************************************
**************
  480:  *!
  481:  *!       Procedure: REVISAR
  482:  *!
  483:  *!           Calls: MSCATTER            (procedure in
AACT_SST.SPR)
  484:  *!                : DEAC_KEYS           (procedure in
AACT_SST.SPR)
  485:  *!                : ACT_KEYS            (procedure in
AACT_SST.SPR)
  486:  *!                : VAL_CAMPO()         (function in
AP_PRMCT.SPR)
  487:  *!                : REFRESH()           (function in
AP_PRMCT.SPR)
  488:  *!                : MBROW               (procedure in
MD_PDLAB.SPR)
  489:  *!
  490:
*!*************************************************************
**************
  491:      PROCEDURE revisar
  492:          ******************
  493:          MOVE WIND revisar TO 3.250, 20.857
  494:          ACTI WIND revisar IN WIND aact_dpt
  495:          BROW FIELD sst.ciudad :H = 'SISTEMA',
dpt.cod_dep:H='COD. DPTO.',  dpt.nombre:H='DEPARTAMENTO';
  496:             NOEDIT NODELE NOMENU
  497:          DEAC WIND revisar
  498:          reg_ant = RECNO()
  499:          DO mscatter
  500:          SHOW GETS
  501: <=======RETURN
  502:
  503:          ****************
  504:
*!*************************************************************
**************
  505:  *!
  506:  *!       Procedure: BUSCAR
  507:  *!
  508:  *!       Called by: SECC()             (function in
PERSONAL.PRG)
  509:  *!                : _R390JW4G3()       (function in
```

```
                    DSC_GRTR.SPR)
  510:  *!                  : _RA81B8ER0()        (function in
ACT_PTMO.SPR)
  511:  *!                  : _RA81B8GQD()        (function in
ACT_PTMO.SPR)
  512:  *!                  : _R6108FBLN()        (function in
MO_PDLAB.SPR)
  513:  *!                  : _R5K0M9Z09()        (function in
DSC_CTA.SPR)
  514:  *!
  515:  *!           Calls: VAL_BUSCAR            (procedure in
AACT_SST.SPR)
  516:  *!                  : BUSC_CLV()          (function in
DGP_CARG.SPR)
  517:  *!                  : DESPLEGA            (procedure in
PERSONAL.PRG)
  518:  *!                  : BUSCARP()           (function in
ACT_PTMO.SPR)
  519:  *!                  : REF_SCR             (procedure in
MO_PDLAB.SPR)
  520:  *!                  : MENSAJES            (procedure in
MO_PDLAB.SPR)
  521:  *!                  : ASIGNAM             (procedure in
MO_PDLAB.SPR)
  522:  *!
  523:
*!*************************************************************
**************
  524:      PROCEDURE buscar
  525:          ****************
  526:          ACTIVATE WINDOW buscar
  527:          reg_ant = RECNO()
  528:          STORE SPAC(3) TO mcod_dep
  529:          @ 0,2 SAY 'Ingrese el Código: ' GET mcod_dep PICT
'999' VALID val_buscar(mcod_dep) ERROR 'Código no Existe'
  530:          READ
  531:          SHOW GETS
  532:          DEACTIVATE WINDOW buscar
  533:  <=======RETURN
  534:
  535:          *********************
  536:
*!*************************************************************
**************
  537:  *!
  538:  *!       Procedure: VAL_BUSCAR
  539:  *!
  540:  *!       Called by: BUSCAR             (procedure in
AACT_SST.SPR)
  541:  *!
  542:  *!            Calls: MSCATTER          (procedure in
AACT_SST.SPR)
  543:  *!
  544:
*!*************************************************************
**************
  545:      PROCEDURE val_buscar
```

```
   595:  *!                    : DSC_GRTR.SPR
   596:  *!                    : ELIMINAR              (procedure in
AACT_SST.SPR)
   597:  *!                    : REVISAR               (procedure in
AACT_SST.SPR)
   598:  *!                    : CTR_POSP              (procedure in
AACT_SST.SPR)
   599:  *!                    : GRABAR                (procedure in
AACT_SST.SPR)
   600:  *!                    : CANCELAR              (procedure in
AACT_SST.SPR)
   601:  *!                    : ROL_Q1.SPR
   602:  *!                    : ROL_Q2.SPR
   603:  *!                    : ACT_PTMO.SPR
   604:  *!                    : MO_PDLAB.SPR
   605:  *!                    : DSC_CTA.SPR
   606:  *!                    : _R6108FBI7()          (function in
MO_PDLAB.SPR)
   607:  *!
   608:  *!          Calls: CTR_POSP                   (procedure in
AACT_SST.SPR)
   609:  *!
   610:
*!*****************************************************************
**************
   611:      PROCEDURE act_keys
   612:          ******************
   613:          ON KEY LABEL pgup DO ctr_posp WITH
'ANTERIOR','SKIP -1'
   614:          ON KEY LABEL pgdn DO ctr_posp WITH
'SIGUIENTE','SKIP'
   615:          ON KEY LABEL home DO ctr_posp WITH 'PRIMERO','GOTO
TOP'
   616:          ON KEY LABEL END DO ctr_posp WITH 'ULTIMO','GOTO
BOTTOM'
   617: <=======RETURN
   618:
   619:          ********************
   620:
*!*****************************************************************
**************
   621:  *!
   622:  *!       Procedure: DEAC_KEYS
   623:  *!
   624:  *!       Called by: EDITAR               (procedure in
CTRL_RES.SPR)
   625:  *!                : SALIR                (procedure in
CTRL_RES.SPR)
   626:  *!                : CREAR                (procedure in
AACT_SST.SPR)
   627:  *!                : ELIMINAR             (procedure in
AACT_SST.SPR)
   628:  *!                : REVISAR              (procedure in
AACT_SST.SPR)
   629:  *!                : CTR_POSP             (procedure in
AACT_SST.SPR)
   630:  *!                : _R6108FBI7()         (function in
```

```
MO_PDLAB.SPR)
  631:  *!
  632:
*!*****************************************************************
**************
  633:      PROCEDURE deac_keys
  634:          ********************
  635:          ON KEY LABEL pgup
  636:          ON KEY LABEL pgdn
  637:          ON KEY LABEL home
  638:          ON KEY LABEL END
  639: <=======RETURN
  640:
  641:          ******************
  642:
*!*****************************************************************
**************
  643:  *!
  644:  *!       Procedure: CTR_POSP
  645:  *!
  646:  *!       Called by: ACT_KEYS           (procedure in
AACT_SST.SPR)
  647:  *!                : ELIMINAR           (procedure in
AACT_SST.SPR)
  648:  *!
  649:  *!           Calls: MSCATTER           (procedure in
AACT_SST.SPR)
  650:  *!                : DEAC_KEYS          (procedure in
AACT_SST.SPR)
  651:  *!                : ASIGNAM            (procedure in
MO_PDLAB.SPR)
  652:  *!                : REF_SCR            (procedure in
MO_PDLAB.SPR)
  653:  *!                : ACT_KEYS           (procedure in
AACT_SST.SPR)
  654:  *!
  655:
*!*****************************************************************
**************
  656:      PROCEDURE ctr_posp
  657:          ******************
  658:          PARAMETER opc_posp, comando
  659:        1=DO CASE
  660:        à=====CASE BOF() AND comando = 'SKIP -1'
  661:        |         GOTO TOP
  662:        à=====CASE EOF() AND comando = 'SKIP'
  663:        |         GOTO BOTTOM
  664:        à=====OTHERWISE
  665:        |         &comando
  666:        ß=ENDCASE
  667:        1=DO CASE
  668:        à=====CASE BOF()
  669:        |         GOTO TOP
  670:        |         WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
  671:        à=====CASE EOF()
  672:        |         GOTO BOTTOM
  673:        |         WAIT WIND NOWAIT 'FIN DE ARCHIVO'
```

```
  674:        ENDCASE
  675:        DO CASE
  676:        CASE opc_posp = 'PRIMERO'
  677:             WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
  678:        CASE opc_posp = 'SIGUIENTE'
  679:             IF EOF()
  680:                GO BOTT
  681:                WAIT WIND NOWAIT 'FIN DE ARCHIVO'
  682:             ENDIF
  683:        CASE opc_posp = 'ANTERIOR'
  684:             IF BOF()
  685:                GOTO TOP
  686:                WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
  687:             ENDIF
  688:        CASE opc_posp = 'ULTIMO'
  689:             WAIT WIND NOWAIT 'FIN DE ARCHIVO'
  690:        ENDCASE
  691:          DO mscatter
  692:          SHOW GETS
  693:        RETURN
  694:
  695:          ****************
  696:
*!*************************************************************************
**************
  697:  *!
  698:  *!      Procedure: GRABAR
  699:  *!
  700:  *!      Called by: ING                 (procedure in
PERSONAL.PRG)
  701:  *!               : MERCE()             (function in
PERSONAL.PRG)
  702:  *!               : JUBILAR             (procedure in
PERSONAL.PRG)
  703:  *!
  704:  *!          Calls: ACT_KEYS            (procedure in
AACT_SST.SPR)
  705:  *!               : ELIM_MPL            (procedure in
MO_PDLAB.SPR)
  706:  *!               : INSERTAR()          (function in
MO_PDLAB.SPR)
  707:  *!               : BORRA_TMP           (procedure in
MO_PDLAB.SPR)
  708:  *!               : REF_SCR             (procedure in
MO_PDLAB.SPR)
  709:  *!
  710:  *!           Uses: TMP.DBF
  711:  *!
  712:
*!*************************************************************************
**************
  713:     PROCEDURE grabar
  714:          ****************
  715:          IF creando
  716:             APPEN BLANK
  717:          ENDIF
  718:          reg_ant = RECNO()
```

```
    1:
*:*************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\AACT_SEC.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 03/02/1995 at 20:54:36
    8:  *:
    9:  *:  Procs & Fncts: _RA418TFQ9()
   10:  *:                : _RA418TFUD()
   11:  *:                : _RA418TFYU()
   12:  *:                : _RA418TG1K()
   13:  *:                : _RA418TG5H()
   14:  *:                : _RA418TG9Z()
   15:  *:
   16:  *:          Set by: EMPRESA.MPR
   17:  *:
   18:  *:           Calls: MSCATTER           (procedure in
AACT_SST.SPR)
   19:  *:                : ACT_KEYS           (procedure in
AACT_SST.SPR)
   20:  *:                : _RA418TFQ9()       (function in
AACT_SEC.SPR)
   21:  *:                : _RA418TFUD()       (function in
AACT_SEC.SPR)
   22:  *:                : _RA418TFYU()       (function in
AACT_SEC.SPR)
   23:  *:                : _RA418TG1K()       (function in
AACT_SEC.SPR)
   24:  *:                : _RA418TG5H()       (function in
AACT_SEC.SPR)
   25:  *:                : _RA418TG9Z()       (function in
AACT_SEC.SPR)
   26:  *:
   27:  *:       Documented 19:14:28
FoxDoc version 3.00a
   28:
*:*************************************************************
**************
   29:      *
************************************************************
   30:      *        *
   31:      *        * 03/05/1996           AACT_SEC.SPR
 20:54:33
   32:      *        *
   33:      *
************************************************************
   34:      *        *
   35:      *        * Nombre del autor
   36:      *        *
   37:      *        * Copyright (c) 1995 Nombre de la empresa
   38:      *        * Direcci¾n
   39:      *        * Ciudad,     C.P.
   40:      *        * Paŷs
   41:      *        *
```

```
   42:      *        * Description:
   43:      *        * Este programa lo ha generado automßticamente
GENSCRN.
   44:      *        *
   45:      *
************************************************************
   46:
   47:
   48:      *
************************************************************
   49:      *        *
   50:      *        *  AACT_SEC/Windows C¾digo de configuraci¾n -
SECCION 1
   51:      *        *
   52:      *
************************************************************
   53:      *
   54:
   55:      #REGION 1
   56:      **********
   57:      REGIONAL opc_scr, creando, editando, reg_ant
   58:
   59:      STOR ''   TO opc_scr
   60:      STOR .F. TO creando, editando
   61:
   62:      **********
   63:
   64:      #REGION 0
   65:      REGIONAL m.currarea, m.talkstat, m.compstat
   66:
   67:    ┌─IF SET("TALK") = "ON"
   68:    │    SET TALK OFF
   69:    │    m.talkstat = "ON"
   70:    ├─ELSE
   71:    │    m.talkstat = "OFF"
   72:    └─ENDIF
   73:      m.compstat = SET("COMPATIBLE")
   74:      SET COMPATIBLE FOXPLUS
   75:
   76:      m.rborder = SET("READBORDER")
   77:      SET readborder ON
   78:
   79:      m.currarea = SELECT()
   80:
   81:
   82:      *
************************************************************
   83:      *        *
   84:      *        *  AACT_SEC/Windows Bases de datos, Ýndices,
relaciones
   85:      *        *
   86:      *
************************************************************
   87:      *
   88:
   89:    ┌─IF USED("sst")
   90:    │    SELECT sst
   91:    │    SET ORDER TO TAG "sist_cod"
```

```
 92:    ELSE
 93:        SELECT 0
 94:        USE (LOCFILE("sro|dbfs|sistema.dbf","DBF","Buscar
sistema?"));
 95:            AGAIN ALIAS sst ;
 96:            ORDER TAG "sist_cod"
 97:    ENDIF
 98:
 99:    IF USED("dpt")
100:        SELECT dpt
101:        SET ORDER TO TAG "dpto_cod"
102:    ELSE
103:        SELECT 0
104:        USE (LOCFILE("sro|dbfs|departam.dbf","DBF","Buscar
departam?"));
105:            AGAIN ALIAS dpt ;
106:            ORDER TAG "dpto_cod"
107:    ENDIF
108:
109:    IF USED("dgp")
110:        SELECT dgp
111:        SET ORDER TO TAG "num_sec"
112:    ELSE
113:        SELECT 0
114:        USE (LOCFILE("sro|dbfs|personal.dbf","DBF","Buscar
personal?"));
115:            AGAIN ALIAS dgp ;
116:            ORDER TAG "num_sec"
117:    ENDIF
118:
119:    IF USED("sec")
120:        SELECT sec
121:        SET ORDER TO TAG "secc_cod"
122:    ELSE
123:        SELECT 0
124:        USE (LOCFILE("sro|dbfs|seccion.dbf","DBF","Buscar
seccion?"));
125:            AGAIN ALIAS sec ;
126:            ORDER TAG "secc_cod"
127:    ENDIF
128:
129:    IF USED("dgt")
130:        SELECT dgt
131:        SET ORDER TO TAG "grupo_real"
132:    ELSE
133:        SELECT 0
134:        USE (LOCFILE("sdmo|dbfs|dsc_grtr.dbf","DBF","Buscar
dsc_grtr?"));
135:            AGAIN ALIAS dgt ;
136:            ORDER TAG "grupo_real"
137:    ENDIF
138:
139:    SELECT dgt
140:
141:    SELECT sec
142:    SELECT sec
143:
144:    SET RELATION OFF INTO sst
```

```
145:      SET RELATION TO sec.cod_ciu INTO sst ADDITIVE
146:
147:      SELECT sec
148:
149:      SET RELATION OFF INTO dpt
150:      SET RELATION TO cod_dpto INTO dpt ADDITIVE
151:
152:      SELECT sec
153:
154:      SET RELATION OFF INTO dgt
155:      SET RELATION TO grupotra INTO dgt ADDITIVE
156:
157:      SELECT dgt
158:
159:      *
**********************************************************
160:      *        *
161:      *        *              Windows Definiciones de ventana
162:      *        *
163:      *
**********************************************************
164:      *
165:
166:      IF NOT WEXIST("aact_sec") ;
167:            OR UPPER(WTITLE("AACT_SEC")) == "AACT_SEC.PJX" ;
168:            OR UPPER(WTITLE("AACT_SEC")) == "AACT_SEC.SCX" ;
169:            OR UPPER(WTITLE("AACT_SEC")) == "AACT_SEC.MNX" ;
170:            OR UPPER(WTITLE("AACT_SEC")) == "AACT_SEC.PRG" ;
171:            OR UPPER(WTITLE("AACT_SEC")) == "AACT_SEC.FRX" ;
172:            OR UPPER(WTITLE("AACT_SEC")) == "AACT_SEC.QPR"
173:         DEFINE WINDOW aact_sec ;
174:            AT  0.000, 0.000  ;
175:            SIZE 22.800,63.875 ;
176:            TITLE "ACTUALIZACION DE SECCIONES" ;
177:            FONT "Fixedsys", 9 ;
178:            FLOAT ;
179:            NOCLOSE ;
180:            MINIMIZE ;
181:            SYSTEM ;
182:            COLOR RGB(,,,192,192,192)
183:         MOVE WINDOW aact_sec CENTER
184:      ENDIF
185:
186:
187:      *
**********************************************************
188:      *        *
189:      *        *  AACT_SEC/Windows Código de configuración -
SECCION 2
190:      *        *
191:      *
**********************************************************
192:      *
193:
194:      #REGION 1
195:      **********
196:      SELE sec
197:      SCATTER MEMVAR BLANK
```

```
  198:      STOR m.grupotra TO m.cod_grupo
  199:      DO mscatter
  200:      STOR RECNO() TO reg_ant
  201:      DO act_keys
  202:     IF NOT WEXIST('waiting')
  203:        DEFI WIND waiting FROM 0,0 TO 8,40 PANEL
  204:        MOVE WIND waiting CENTER
  205:     ENDIF
  206:     IF NOT WEXIST('BUSCAR') && o definir en el principal
  207:        DEFINE WINDOW buscar FROM 0,0 TO 3,28 TITLE
'BUSCAR' SYSTEM
  208:        MOVE WIND buscar CENTER
  209:     ENDIF
  210:      DEFINE WINDOW revisar FROM 0,0 TO 15,50 TITLE
'REVISANDO SECCIONES ..!'
  211:      MOVE WINDOW revisar CENTER
  212:
  213:
  214:      #REGION 1
  215:      DEFINE POPUP _ra418texu ;
  216:         PROMPT FIELD dpt.nombre ;
  217:         SCROLL
  218:      DEFINE POPUP _ra418tey4 ;
  219:         PROMPT FIELD dgt.descrip ;
  220:         SCROLL
  221:      DEFINE POPUP _ra418tey8 ;
  222:         PROMPT FIELD sst.ciudad ;
  223:         SCROLL
  224:
  225:      *
************************************************************
  226:      *        *
  227:      *        *         AACT_SEC/Windows Distribuci¥n de
pantalla
  228:      *        *
  229:      *
************************************************************
  230:      *
  231:
  232:      #REGION 1
  233:     IF WVISIBLE("aact_sec")
  234:        ACTIVATE WINDOW aact_sec SAME
  235:     ELSE
  236:        ACTIVATE WINDOW aact_sec NOSHOW
  237:     ENDIF
  238:      @ 6.200,0.750 SAY
(LOCFILE("bitm\program.bmp","BMP¦ICO¦PCT¦ICN", "¬Dónde est8
program?" )) BITMAP ;
  239:         SIZE 1.667,3.125 ;
  240:         STRETCH ;
  241:         STYLE "T"
  242:      @ 4.333,0.000 SAY
(LOCFILE("bitm\memo.bmp","BMP¦ICO¦PCT¦ICN", "¬Dónde est8 memo?"
)) BITMAP ;
  243:         SIZE 1.667,4.625 ;
  244:         STRETCH ;
  245:         STYLE "T"
  246:      @ 8.333,1.000 SAY
```

```
(LOCFILE("bitm\wzdelete.bmp","BMP¦ICO¦PCT¦ICN", "┐Dónde está
wzdelete?" )) BITMAP ;
  247:          SIZE 1.667,3.125 ;
  248:          STRETCH ;
  249:          STYLE "T"
  250:       @ 10.133,1.000 SAY
(LOCFILE("bitm\query.bmp","BMP¦ICO¦PCT¦ICN", "┐Dónde está query?"
)) BITMAP ;
  251:          SIZE 1.667,3.125 ;
  252:          STRETCH ;
  253:          STYLE "T"
  254:       @ 12.400,0.625 SAY
(LOCFILE("bitm\wzprint.bmp","BMP¦ICO¦PCT¦ICN", "┐Dónde está
wzprint?" )) BITMAP ;
  255:          SIZE 1.667,3.125 ;
  256:          STRETCH ;
  257:          STYLE "T"
  258:       @ 14.467,0.750 SAY
(LOCFILE("bitm\wzlocate.bmp","BMP¦ICO¦PCT¦ICN", "┐Dónde está
wzlocate?" )) BITMAP ;
  259:          SIZE 1.667,3.125 ;
  260:          STRETCH ;
  261:          STYLE "T"
  262:       @ 16.600,0.500 SAY
(LOCFILE("bitm\close.bmp","BMP¦ICO¦PCT¦ICN", "┐Dónde está close?"
)) BITMAP ;
  263:          SIZE 1.667,3.125 ;
  264:          STRETCH ;
  265:          STYLE "T"
  266:       @ 16.800,17.125 SAY "Sistema de Rol:" ;
  267:          FONT "Times New Roman", 10 ;
  268:          STYLE "BIT"
  269:       @ 1.667,17.375 SAY "Código:" ;
  270:          FONT "Times New Roman", 10 ;
  271:          STYLE "BIT"
  272:       @ 5.667,16.875 SAY "Código -  Nombre de Departamento:"
;
  273:          FONT "Times New Roman", 10 ;
  274:          STYLE "BIT"
  275:       @ 1.667,24.625 SAY "Nombre de Sección:" ;
  276:          FONT "Times New Roman", 10 ;
  277:          STYLE "BIT"
  278:       @ 13.733,17.125 SAY "Cuenta Contable:" ;
  279:          FONT "Times New Roman", 8 ;
  280:          STYLE "BT"
  281:       @ 11.133,17.500 SAY "Código   -    Nombre de Grupo
 Trabajo:" ;
  282:          FONT "Times New Roman", 10 ;
  283:          STYLE "BIT"
  284:       @ 3.533,4.125 TO 18.466,15.375 ;
  285:          PATTERN 1 ;
  286:          PEN 1, 8 ;
  287:          STYLE "24" ;
  288:          COLOR RGB(,,,128,128,0)
  289:       @ 3.200,18.000 GET m.cod_sec ;
  290:          SIZE 0.950,4.571 ;
  291:          DEFAULT " " ;
  292:          FONT "Times New Roman", 12 ;
```

```
293:          STYLE "B" ;
294:          PICTURE "@KT 999" ;
295:          WHEN creando ;
296:          VALID _ra418tfq9() ;
297:          ERROR ' CODIGO DE SECCION YA HA SIDO REGISTRADO'
298:       @ 3.400,23.875 GET m.nombre ;
299:          SIZE 0.938,58.800 ;
300:          DEFAULT " " ;
301:          FONT "Times New Roman", 9 ;
302:          STYLE "B" ;
303:          PICTURE "@K!" ;
304:          WHEN creando OR editando
305:       @ 8.000,18.000 GET m.cod_dpto ;
306:          SIZE 0.950,4.571 ;
307:          DEFAULT " " ;
308:          FONT "Times New Roman", 12 ;
309:          STYLE "B" ;
310:          PICTURE "@KT 999" ;
311:          WHEN creando ;
312:          VALID _ra418tfud() ;
313:          ERROR ' CODIGO DE SECCION YA HA SIDO REGISTRADO'
314:       @ 7.267,24.125 GET dpt.nombre ;
315:          PICTURE "@&N" ;
316:          POPUP _ra418texu ;
317:          SIZE 3.462,44.600 ;
318:          DEFAULT " " ;
319:          FONT "Times New Roman", 8 ;
320:          STYLE "B" ;
321:          WHEN .F. ;
322:          VALID _ra418tfyu()
323:       @ 12.400,18.250 GET m.cod_grupo ;
324:          SIZE 0.938,6.400 ;
325:          DEFAULT 0 ;
326:          FONT "Times New Roman", 9 ;
327:          STYLE "B" ;
328:          PICTURE "@KZ 99" ;
329:          WHEN creando ;
330:          VALID _ra418tg1k() ;
331:          ERROR ' GRUPO DE TRABAJO NO EXISTE'
332:       @ 12.933,33.750 GET dgt.descrip ;
333:          PICTURE "@&N" ;
334:          POPUP _ra418tey4 ;
335:          SIZE 3.462,44.600 ;
336:          DEFAULT " " ;
337:          FONT "Times New Roman", 8 ;
338:          STYLE "B" ;
339:          WHEN .F. ;
340:          VALID _ra418tg5h()
341:       @ 14.933,18.000 GET dgt.cta_ctble ;
342:          SIZE 1.000,23.000 ;
343:          DEFAULT " " ;
344:          FONT "Times New Roman", 8 ;
345:          STYLE "B" ;
346:          PICTURE "@K" ;
347:          WHEN .F.
348:       @ 18.467,18.000 GET sst.ciudad ;
349:          PICTURE "@&N" ;
350:          POPUP _ra418tey8 ;
```

```
351:          SIZE 3.462,23.800 ;
352:          DEFAULT " " ;
353:          FONT "Times New Roman", 8 ;
354:          STYLE "B" ;
355:          WHEN .F.
356:       @ 4.200,4.500 GET opc_scr ;
357:          PICTURE "@*VN
|<Crear;|<Editar;E|<Liminar;|<Revisar;|<Imprimir;|<Buscar;|<Salir
" ;
358:          SIZE 1.647,16.800,0.118 ;
359:          DEFAULT 1 ;
360:          FONT "Times New Roman", 10 ;
361:          STYLE "BI" ;
362:          VALID _ra418tg9z()
363:       @ 19.267,48.000 SAY "<PgUp>Retrocede" + CHR(13) + ;
364:          "<PgDn>Avanza Reg." ;
365:          SIZE 2.000,22.800, 0.000 ;
366:          FONT "Arial", 8 ;
367:          STYLE "BT"
368:       @ 19.000,46.500 TO 21.467,61.625 ;
369:          PEN 1, 8 ;
370:          STYLE "16" ;
371:          COLOR RGB(128,128,0,,,,)
372:
373:       IF NOT WVISIBLE("aact_sec")
374:          ACTIVATE WINDOW aact_sec
375:       ENDIF
376:
377:       READ CYCLE ;
378:          VALID .F.
379:
380:       RELEASE WINDOW aact_sec
381:
382:       *
************************************************************
383:       *        *
384:       *        *              Windows Cerrando bases de datos
385:       *        *
386:       *
************************************************************
387:       *
388:
389:       IF USED("sst")
390:          SELECT sst
391:          USE
392:       ENDIF
393:
394:       IF USED("dpt")
395:          SELECT dpt
396:          USE
397:       ENDIF
398:
399:       IF USED("dgp")
400:          SELECT dgp
401:          USE
402:       ENDIF
403:
404:       IF USED("sec")
```

```
  405:     |    SELECT sec
  406:     |    USE
  407:     |ENDIF
  408:
  409:     |IF USED("dgt")
  410:     |    SELECT dgt
  411:     |    USE
  412:     |ENDIF
  413:
  414:     SELECT (m.currarea)
  415:
  416:     RELEASE POPUPS _ra418texu, _ra418tey4, _ra418tey8
  417:
  418:     #REGION 0
  419:
  420:     SET readborder &rborder
  421:
  422:     |IF m.talkstat = "ON"
  423:     |    SET TALK ON
  424:     |ENDIF
  425:     |IF m.compstat = "ON"
  426:     |    SET COMPATIBLE ON
  427:     |ENDIF
  428:
  429:
  430:     *
************************************************************
  431:     *        *
  432:     *        *            AACT_SEC/Windows Código de
limpieza
  433:     *        *
  434:     *
************************************************************
  435:     *
  436:
  437:     #REGION 1
  438:     ***************
  439:
  440:
  441:     *
************************************************************
  442:     *        *
  443:     *        * AACT_SEC/Windows Procedimientos y funciones
de soporte
  444:     *        *
  445:     *
************************************************************
  446:     *
  447:
  448:     #REGION 1
  449:
*!*************************************************************************
***************
  450:  *!
  451:  *!      Procedure: CREAR
  452:  *!
  453:  *!           Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
```

```
  454:  *!                : BUSC_CLV()          (function in
DGP_CARG.SPR)
  455:  *!                : EDITAR              (procedure in
CTRL_RES.SPR)
  456:  *!                : PROCESO             (procedure in
DSC_GRTR.SPR)
  457:  *!                : INGRESO             (procedure in
ACT_PTMO.SPR)
  458:  *!                : CREA_TMP            (procedure in
MO_PDLAB.SPR)
  459:  *!                : MBROW               (procedure in
MO_PDLAB.SPR)
  460:  *!                : MENSAJES            (procedure in
MO_PDLAB.SPR)
  461:  *!                : REF_SCR             (procedure in
MO_PDLAB.SPR)
  462:  *!
  463:
*!*************************************************************
**************
  464:      PROCEDURE crear
  465:          ***************
  466:          DO deac_keys
  467:          reg_ant = RECNO()
  468:          SCATTER MEMV BLANK
  469:          STOR .T. TO creando
  470:          _CUROBJ = OBJNUM(m.cod_sec)
  471:          SHOW GET opc_scr DISABLE
  472:          SHOW GET opc_scr,1 PROMPT '\<Grabar' ENABLE
  473:          SHOW GET opc_scr,2 PROMPT '\<Cancelar' ENABLE
  474: <=======RETURN
  475:
  476:          ****************
  477:
*!*************************************************************
**************
  478:  *!
  479:  *!       Procedure: EDITAR
  480:  *!
  481:  *!       Called by: CREAR              (procedure in
AACT_SST.SPR)
  482:  *!
  483:  *!            Calls: DEAC_KEYS         (procedure in
AACT_SST.SPR)
  484:  *!                 : CABEZA            (procedure in
PERSONAL.PRG)
  485:  *!                 : INGRESO           (procedure in
ACT_PTMO.SPR)
  486:  *!                 : CREA_TMP          (procedure in
MO_PDLAB.SPR)
  487:  *!                 : MBROW             (procedure in
MO_PDLAB.SPR)
  488:  *!                 : REF_SCR           (procedure in
MO_PDLAB.SPR)
  489:  *!                 : MENSAJES          (procedure in
MO_PDLAB.SPR)
  490:  *!
  491:
```

```
*!*************************************************************
**************
  492:      PROCEDURE editar
  493:          ****************
  494:          DO deac_keys
  495:          STOR .T. TO editando
  496:          _CUROBJ = OBJNUM(m.nombre)
  497:          SHOW GET opc_scr DISABLE
  498:          SHOW GET opc_scr,1 PROMPT '\<Grabar' ENABLE
  499:          SHOW GET opc_scr,2 PROMPT '\<Cancelar' ENABLE
  500: <========RETURN
  501:
  502:          ******************
  503:
*!*************************************************************
**************
  504:   *!
  505:   *!       Procedure: ELIMINAR
  506:   *!
  507:   *!           Calls: MSCATTER            (procedure in
AACT_SST.SPR)
  508:   *!                : DEAC_KEYS           (procedure in
AACT_SST.SPR)
  509:   *!                : ACT_KEYS            (procedure in
AACT_SST.SPR)
  510:   *!                : BORRAR              (procedure in
DSC_GRTR.SPR)
  511:   *!                : ELIM_MPL            (procedure in
MO_PDLAB.SPR)
  512:   *!                : CTR_POSP            (procedure in
AACT_SST.SPR)
  513:   *!
  514:
*!*************************************************************
**************
  515:      PROCEDURE eliminar
  516:          ******************
  517:          SELECT dgp
  518:          SEEK m.cod_sec
  519:          ACTIVATE WINDOW waiting
  520:          ?CHR(7)
  521:        ┌─IF FOUND()
  522:        │     @1,1 SAY 'Lo Siento. Pero no se puede ELIMINAR
'
  523:        │     @2,1 SAY ' Secciones que contienen informaci¥n'
  524:        │     STOR 1 TO mcont
  525:        │     ?CHR(7)
  526:        │     @ 4,10 GET mcont;
  527:        │        PICTURE "@*h \<Aceptar" SIZE 2,10
  528:        │     READ MODAL
  529:        ├─ELSE
  530:        │     @1,0 SAY " La Secci¥n: " + m.nombre
  531:        │     @2,0 SAY ' ser¡ ELIMINADA ...!'
  532:        │     STOR 1 TO mcont
  533:        │     ?CHR(7)
  534:        │     @ 4, 8 GET mcont;
  535:        │        PICTURE "@*h \<Cancelar;\<Eliminar" SIZE
2,10
```

```
  536:          READ MODAL
  537:          IF mcont = 2
  538:              WAIT( "Borrando Secci¤n ....!")WINDOW
TIME(1)
  539:              SELE sec
  540:              DELE
  541:              GO TOP
  542:              DO mscatter
  543:              reg_ant = RECNO()
  544:              SHOW GETS
  545:          ENDIF
  546:      ENDIF
  547:      DEACT WIND waiting
  548:      SELECT sec
  549: <====RETURN
  550:      ******************
  551:
*!*************************************************************
**************
  552:  *!
  553:  *!       Procedure: IMPRIMIR
  554:  *!
  555:  *!           Calls: MENSAJE2            (procedure in
ACT_PTMO.SPR)
  556:  *!
  557:  *!    Report Forms: DSC_GRTR.FRX
  558:  *!                : CTA_REP.FRX
  559:  *!
  560:
*!*************************************************************
**************
  561:  PROCEDURE imprimir
  562:      ******************
  563:      ?CHR(7)
  564:      WAIT WIND 'PROCESO NO PREPARADO'
  565: <====RETURN
  566:      ******************
  567:
*!*************************************************************
**************
  568:  *!
  569:  *!       Procedure: REVISAR
  570:  *!
  571:  *!           Calls: MSCATTER            (procedure in
AACT_SST.SPR)
  572:  *!                : DEAC_KEYS           (procedure in
AACT_SST.SPR)
  573:  *!                : ACT_KEYS            (procedure in
AACT_SST.SPR)
  574:  *!                : VAL_CAMPO()         (function in
AP_PRMCT.SPR)
  575:  *!                : REFRESH()           (function in
AP_PRMCT.SPR)
  576:  *!                : MBROW               (procedure in
MO_PDLAB.SPR)
  577:  *!
  578:
*!*************************************************************
```

```
**************
  579:      PROCEDURE revisar
  580:           ******************
  581:           DO deac_keys
  582:           MOVE WIND revisar TO 3.250, 20.857
  583:           ACTI WIND revisar IN WIND aact_sec
  584:           BROW FIELD sec.cod_sec:H='CODIGO',
sec.nombre:H='SECCION',;
  585:                dpt.nombre:H='DEPARTAMENTO', sst.ciudad :H =
'SISTEMA';
  586:                NOEDIT NODELE NOMENU
  587:           DEACT WIND revisar
  588:           reg_ant = RECNO()
  589:           DO mscatter
  590:           SHOW GETS
  591:           DO act_keys
  592: <=======RETURN
  593:
  594:           ****************
  595:
*!*********************************************************************
**************
  596:   *!
  597:   *!       Procedure: BUSCAR
  598:   *!
  599:   *!       Called by: SECC()              (function in
PERSONAL.PRG)
  600:   *!                : _R390JW4G3()        (function in
DSC_GRTR.SPR)
  601:   *!                : _RA81B8ER0()        (function in
ACT_PTMO.SPR)
  602:   *!                : _RA81B8GQD()        (function in
ACT_PTMO.SPR)
  603:   *!                : _R6108FBLN()        (function in
MO_PDLAB.SPR)
  604:   *!                : _R5K0M9Z09()        (function in
DSC_CTA.SPR)
  605:   *!
  606:   *!           Calls: VAL_BUSCAR          (procedure in
AACT_SST.SPR)
  607:   *!                : BUSC_CLV()          (function in
DGP_CARG.SPR)
  608:   *!                : DESPLEGA            (procedure in
PERSONAL.PRG)
  609:   *!                : BUSCARP()           (function in
ACT_PTMO.SPR)
  610:   *!                : REF_SCR             (procedure in
MO_PDLAB.SPR)
  611:   *!                : MENSAJES            (procedure in
MO_PDLAB.SPR)
  612:   *!                : ASIGNAM             (procedure in
MO_PDLAB.SPR)
  613:   *!
  614:
*!*********************************************************************
**************
  615:      PROCEDURE buscar
  616:           ****************
```

```
  617:          ACTIVATE WINDOW buscar
  618:          reg_ant = RECNO()
  619:          STORE SPAC(3) TO mcod_sec
  620:          @ 0,2 SAY 'Ingrese el C¾digo: ' GET mcod_sec PICT
'999' VALID val_buscar(mcod_sec) ERROR 'C¾digo no Existe'
  621:          READ
  622:          SHOW GETS
  623:          DEACTIVATE WINDOW buscar
  624: <=======RETURN
  625:
  626:          ********************
  627:
*!*************************************************************************
**************
  628:  *!
  629:  *!       Procedure: VAL_BUSCAR
  630:  *!
  631:  *!       Called by: BUSCAR              (procedure in
AACT_SST.SPR)
  632:  *!
  633:  *!            Calls: MSCATTER           (procedure in
AACT_SST.SPR)
  634:  *!
  635:
*!*************************************************************************
**************
  636:      PROCEDURE val_buscar
  637:          ********************
  638:          PARAMETER mcod_sec
  639:          STOR .T. TO vret
  640:        ┌─IF LAST() = 27
  641: <======┴═══RETURN TO buscar
  642:        └─ENDIF
  643:          SEEK mcod_sec
  644:        ┌─IF !FOUND()
  645:        │    GOTO reg_ant
  646:        │    STOR .F. TO vret
  647:        └─ENDIF
  648:          DO mscatter
  649:          reg_ant = RECNO()
  650: <=======RETURN vret
  651:
  652:          ***************
  653:
*!*************************************************************************
**************
  654:  *!
  655:  *!       Procedure: SALIR
  656:  *!
  657:  *!       Called by: _RAA1732TG()        (function in
AT_ESCLS.SPR)
  658:  *!                : DA_ASISG.PRG
  659:  *!                : DE_EGRQ1.PRG
  660:  *!                : DE_EGRQ2.PRG
  661:  *!                : DE_EGRMN.PRG
  662:  *!                : DE_IEJUB.PRG
  663:  *!
  664:  *!            Calls: DEAC_KEYS           (procedure in
```

```
AACT_SST.SPR)
  665:  *!
  666:
*!*****************************************************************
**************
  667:      PROCEDURE salir
  668:          ***************
  669:          RELE WIND waiting
  670:          RELE WIND buscar
  671:          RELE WIND revisar
  672:          DO deac_keys
  673:          CLEAR READ
  674: <=======RETURN
  675:
  676:          ******************
  677:
*!*****************************************************************
**************
  678:  *!
  679:  *!       Procedure: ACT_KEYS
  680:  *!
  681:  *!       Called by: AACT_SST.SPR
  682:  *!                : AACT_DPT.SPR
  683:  *!                : AACT_SEC.SPR
  684:  *!                : DGP_CARG.SPR
  685:  *!                : AC_IMPRT.SPR
  686:  *!                : DSC_GRTR.SPR
  687:  *!                : ELIMINAR           (procedure in
AACT_SST.SPR)
  688:  *!                : REVISAR            (procedure in
AACT_SST.SPR)
  689:  *!                : CTR_POSP           (procedure in
AACT_SST.SPR)
  690:  *!                : GRABAR             (procedure in
AACT_SST.SPR)
  691:  *!                : CANCELAR           (procedure in
AACT_SST.SPR)
  692:  *!                : ROL_Q1.SPR
  693:  *!                : ROL_Q2.SPR
  694:  *!                : ACT_PTMO.SPR
  695:  *!                : MO_PDLAB.SPR
  696:  *!                : DSC_CTA.SPR
  697:  *!                : _R6108FBI7()       (function in
MO_PDLAB.SPR)
  698:  *!
  699:  *!           Calls: CTR_POSP           (procedure in
AACT_SST.SPR)
  700:  *!
  701:
*!*****************************************************************
**************
  702:      PROCEDURE act_keys
  703:          ******************
  704:          ON KEY LABEL pgup DO ctr_posp WITH
'ANTERIOR','SKIP -1'
  705:          ON KEY LABEL pgdn DO ctr_posp WITH
'SIGUIENTE','SKIP'
  706:          ON KEY LABEL home DO ctr_posp WITH 'PRIMERO','GOTO
```

```
TOP'
  707:          ON KEY LABEL END DO ctr_posp WITH 'ULTIMO','GOTO
BOTTOM'
  708: <=======RETURN
  709:
  710:          ********************
  711:
*!*************************************************************
**************
  712:  *!
  713:  *!       Procedure: DEAC_KEYS
  714:  *!
  715:  *!       Called by: EDITAR              (procedure in
CTRL_RES.SPR)
  716:  *!                : SALIR               (procedure in
CTRL_RES.SPR)
  717:  *!                : CREAR               (procedure in
AACT_SST.SPR)
  718:  *!                : ELIMINAR            (procedure in
AACT_SST.SPR)
  719:  *!                : REVISAR             (procedure in
AACT_SST.SPR)
  720:  *!                : CTR_POSP            (procedure in
AACT_SST.SPR)
  721:  *!                : _R6108FBI7()        (function in
MO_PDLAB.SPR)
  722:  *!
  723:
*!*************************************************************
**************
  724:      PROCEDURE deac_keys
  725:          ********************
  726:          ON KEY LABEL pgup
  727:          ON KEY LABEL pgdn
  728:          ON KEY LABEL home
  729:          ON KEY LABEL END
  730: <=======RETURN
  731:
  732:          ******************
  733:
*!*************************************************************
**************
  734:  *!
  735:  *!       Procedure: CTR_POSP
  736:  *!
  737:  *!       Called by: ACT_KEYS            (procedure in
AACT_SST.SPR)
  738:  *!                : ELIMINAR            (procedure in
AACT_SST.SPR)
  739:  *!
  740:  *!           Calls: MSCATTER            (procedure in
AACT_SST.SPR)
  741:  *!                : DEAC_KEYS           (procedure in
AACT_SST.SPR)
  742:  *!                : ASIGNAM             (procedure in
MO_PDLAB.SPR)
  743:  *!                : REF_SCR             (procedure in
MO_PDLAB.SPR)
```

```
  744:  *!                : ACT_KEYS            (procedure in
AACT_SST.SPR)
  745:  *!
  746:
*!*************************************************************
**************
  747:      PROCEDURE ctr_posp
  748:          ******************
  749:          PARAMETER opc_posp, comando
  750:        1=DO CASE
  751:        â════CASE BOF() AND comando = 'SKIP -1'
  752:        │       GOTO TOP
  753:        â════CASE EOF() AND comando = 'SKIP'
  754:        │       GOTO BOTTOM
  755:        â════OTHERWISE
  756:        │       &comando
  757:        E=ENDCASE
  758:        1=DO CASE
  759:        â════CASE BOF()
  760:        │       GOTO TOP
  761:        │       WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
  762:        â════CASE EOF()
  763:        │       GOTO BOTTOM
  764:        │       WAIT WIND NOWAIT 'FIN DE ARCHIVO'
  765:        E=ENDCASE
  766:        1=DO CASE
  767:        â════CASE opc_posp = 'PRIMERO'
  768:        │       WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
  769:        â════CASE opc_posp = 'SIGUIENTE'
  770:        │       ┌IF EOF()
  771:        │       │   GO BOTT
  772:        │       │   WAIT WIND NOWAIT 'FIN DE ARCHIVO'
  773:        │       └ENDIF
  774:        â════CASE opc_posp = 'ANTERIOR'
  775:        │       ┌IF BOF()
  776:        │       │   GOTO TOP
  777:        │       │   WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
  778:        │       └ENDIF
  779:        â════CASE opc_posp = 'ULTIMO'
  780:        │       WAIT WIND NOWAIT 'FIN DE ARCHIVO'
  781:        E=ENDCASE
  782:          DO mscatter
  783:          SHOW GETS
  784:  <═══════RETURN
  785:
  786:          ****************
  787:
*!*************************************************************
**************
  788:  *!
  789:  *!      Procedure: GRABAR
  790:  *!
  791:  *!      Called by: ING                (procedure in
PERSONAL.PRG)
  792:  *!               : MERCE()            (function in
PERSONAL.PRG)
  793:  *!               : JUBILAR            (procedure in
PERSONAL.PRG)
```

```
  794:   *!
  795:   *!          Calls: ACT_KEYS          (procedure in
AACT_SST.SPR)
  796:   *!               : ELIM_MPL          (procedure in
MO_PDLAB.SPR)
  797:   *!               : INSERTAR()        (function in
MO_PDLAB.SPR)
  798:   *!               : BORRA_TMP         (procedure in
MO_PDLAB.SPR)
  799:   *!               : REF_SCR           (procedure in
MO_PDLAB.SPR)
  800:   *!
  801:   *!           Uses: TMP.DBF
  802:   *!
  803:
*!*****************************************************************
**************
  804:      PROCEDURE grabar
  805:          ****************
  806:        ┌─IF creando
  807:        │    APPEN BLANK
  808:        └─ENDIF
  809:          reg_ant = RECNO()
  810:          GATHER MEMVAR
  811:          SHOW GETS
  812:          STOR .F. TO creando, editando
  813:          SHOW GET opc_scr,1 PROMPT '|<Crear'
  814:          SHOW GET opc_scr,2 PROMPT '|<Editar'
  815:          SHOW GET opc_scr ENABLE
  816:          DO act_keys
  817: <═══════RETURN
  818:
  819:          ******************
  820:
*!*****************************************************************
**************
  821:   *!
  822:   *!       Procedure: CANCELAR
  823:   *!
  824:   *!       Called by: ING               (procedure in
PERSONAL.PRG)
  825:   *!                : JUBILAR           (procedure in
PERSONAL.PRG)
  826:   *!
  827:   *!           Calls: MSCATTER          (procedure in
AACT_SST.SPR)
  828:   *!                : ACT_KEYS          (procedure in
AACT_SST.SPR)
  829:   *!                : DESPLEGA          (procedure in
PERSONAL.PRG)
  830:   *!                : BORRA_TMP         (procedure in
MO_PDLAB.SPR)
  831:   *!                : ASIGNAM           (procedure in
MO_PDLAB.SPR)
  832:   *!                : REF_SCR           (procedure in
MO_PDLAB.SPR)
  833:   *!
  834:
```

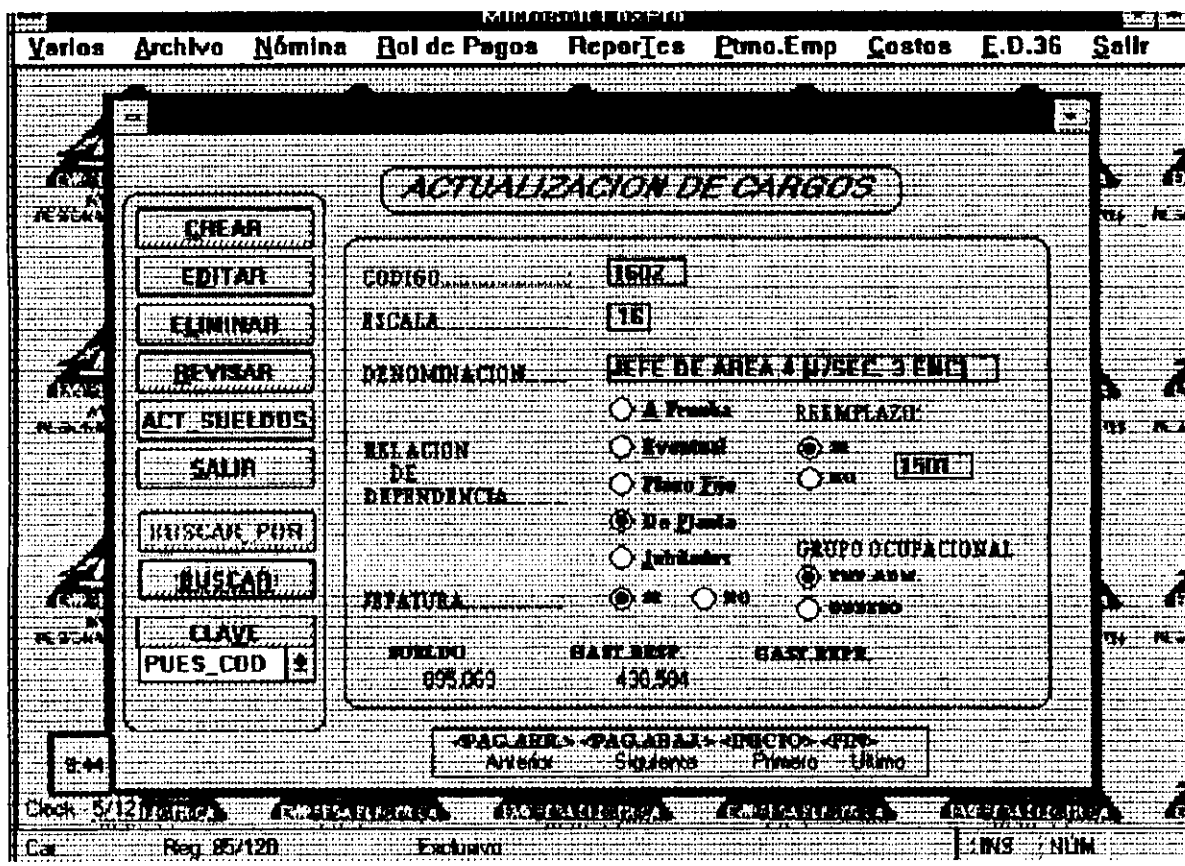
```
*!*************************************************************
**************
  835:      PROCEDURE cancelar
  836:          ******************
  837:          GO reg_ant
  838:          DO mscatter
  839:          SHOW GETS
  840:          STOR .F. TO creando, editando
  841:          SHOW GET opc_scr,1 PROMPT '\<Crear'
  842:          SHOW GET opc_scr,2 PROMPT '\<Editar'
  843:          SHOW GET opc_scr ENABLE
  844:          DO act_keys
  845: <=======RETURN
  846:
  847:          ******************
  848:
*!*************************************************************
**************
  849:  *!
  850:  *!       Procedure: MSCATTER
  851:  *!
  852:  *!       Called by: AACT_SST.SPR
  853:  *!                : AACT_DPT.SPR
  854:  *!                : AACT_SEC.SPR
  855:  *!                : ELIMINAR              (procedure in
AACT_SST.SPR)
  856:  *!                : REVISAR               (procedure in
AACT_SST.SPR)
  857:  *!                : VAL_BUSCAR            (procedure in
AACT_SST.SPR)
  858:  *!                : CTR_POSP              (procedure in
AACT_SST.SPR)
  859:  *!                : CANCELAR              (procedure in
AACT_SST.SPR)
  860:  *!
  861:
*!*************************************************************
**************
  862:      PROCEDURE mscatter
  863:          ******************
  864:          SCATTER MEMVAR
  865:          STOR m.grupotra TO m.cod_grupo
  866: <=======RETURN
  867:
  868:
  869:
  870:          *
*************************************************************
  871:          *        *
  872:          *        * _RA418TFQ9            m.cod_sec VALID
  873:          *        *
  874:          *        * Function Origin:
  875:          *        *
  876:          *        * From Platform:        Windows
  877:          *        * From Screen:          AACT_SEC,
Record Number:   24
  878:          *        * Variable:             m.cod_sec
  879:          *        * Called By:            VALID Clause
```

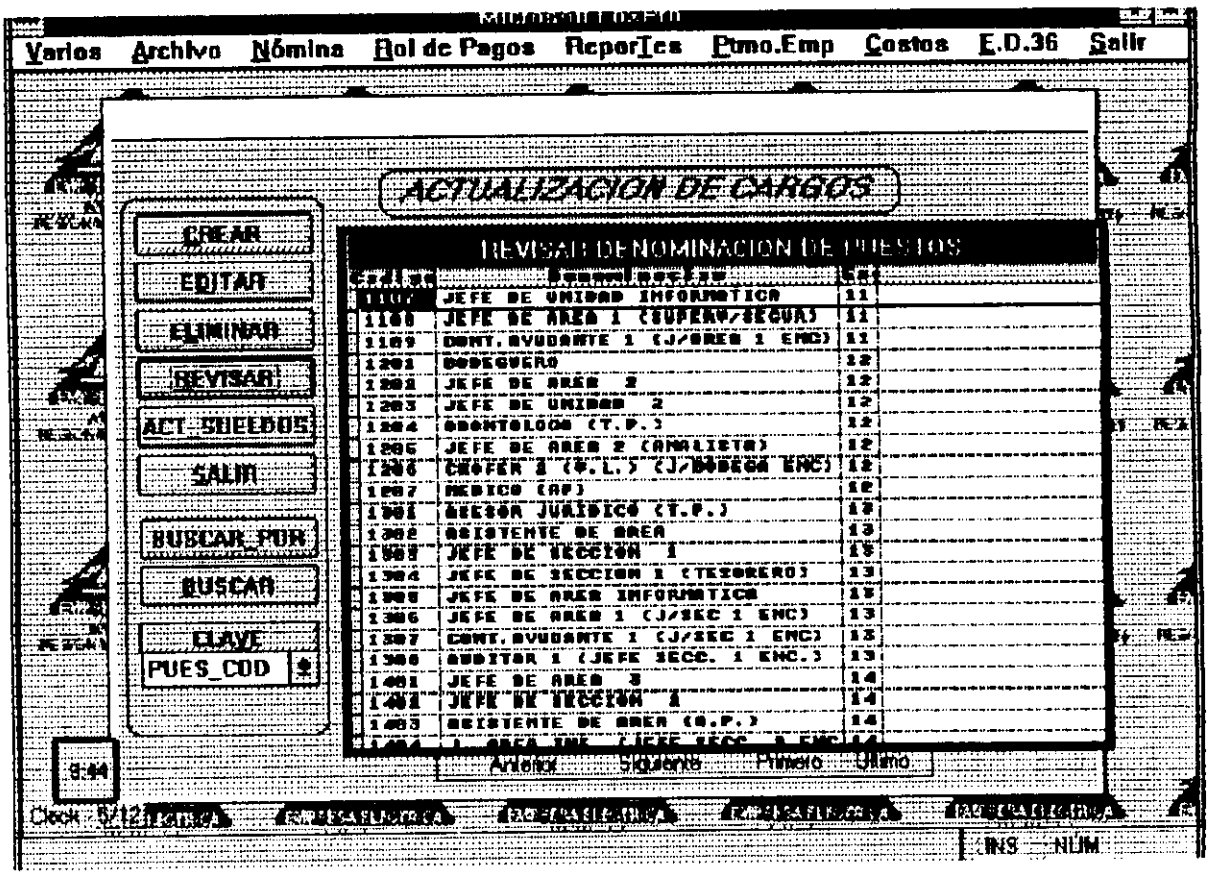
```
880:          *        * Object Type:          Field
881:          *        * Snippet Number:       1
882:          *        *
883:          *
*************************************************************
884:          *
885:
*!*************************************************************
**************
886:   *!
887:   *!        Function: _RA418TFQ9
888:   *!
889:   *!       Called by: AACT_SEC.SPR
890:   *!
891:
*!*************************************************************
**************
892:      FUNCTION _ra418tfq9      &&  m.cod_sec VALID
893:          #REGION 1
894:          SEEK m.cod_sec
895: <=======RETURN !FOUND()
896:
897:          *
*************************************************************
898:          *        *
899:          *        * _RA418TFUD            m.cod_dpto VALID
900:          *        *
901:          *        * Function Origin:
902:          *        *
903:          *        * From Platform:        Windows
904:          *        * From Screen:          AACT_SEC,
Record Number:   26
905:          *        * Variable:             m.cod_dpto
906:          *        * Called By:            VALID Clause
907:          *        * Object Type:          Field
908:          *        * Snippet Number:       2
909:          *        *
910:          *
*************************************************************
911:          *
912:
*!*************************************************************
**************
913:   *!
914:   *!        Function: _RA418TFUD
915:   *!
916:   *!       Called by: AACT_SEC.SPR
917:   *!
918:
*!*************************************************************
**************
919:      FUNCTION _ra418tfud      &&  m.cod_dpto VALID
920:          #REGION 1
921:          SET RELATION OFF INTO dpt
922:          SELECT dpt
923:          SEEK m.cod_dpto
924:         ┌IF !FOUND()
925:         │    ? CHR(7)
```

```
  926:          |    WAIT WIND 'DEPARTAMENTO NO EXISTE'
  927:          |    ACTI WIND revisar
  928:          |    BROW FIELD dpt.cod_dep, dpt.nombre NOEDIT
NODELE
  929:          |    RELE WIND revisar
  930:          |ENDIF
  931:          SHOW GET dpt.nombre
  932:          STOR dpt.cod_dep TO m.cod_dpto
  933:          STOR dpt.cod_ciu TO m.cod_ciu
  934:          SELE sec
  935:          SET RELATION TO cod_dpto INTO dpt ADDITIVE
  936: <========RETURN .T.
  937:
  938:          *
************************************************************
  939:          *       *
  940:          *       * _RA418TFYU            DPT.NOMBRE VALID
  941:          *       *
  942:          *       * Function Origin:
  943:          *       *
  944:          *       * From Platform:        Windows
  945:          *       * From Screen:          AACT_SEC,
Record Number:   27
  946:          *       * Variable:             DPT.NOMBRE
  947:          *       * Called By:            VALID Clause
  948:          *       * Object Type:          List
  949:          *       * Snippet Number:       3
  950:          *       *
  951:          *
************************************************************
  952:          *
  953:
*!*************************************************************************
**************
  954:  *!
  955:  *!       Function: _RA418TFYU
  956:  *!
  957:  *!      Called by: AACT_SEC.SPR
  958:  *!
  959:
*!*************************************************************************
**************
  960:     FUNCTION _ra418tfyu      &&  DPT.NOMBRE VALID
  961:         #REGION 1
  962:         STOR dpt.cod_dpto  TO m.cod_dpto
  963:         SHOW GETS
  964:
  965:          *
************************************************************
  966:          *       *
  967:          *       * _RA418TG1K            m.cod_grupo VALID
  968:          *       *
  969:          *       * Function Origin:
  970:          *       *
  971:          *       * From Platform:        Windows
  972:          *       * From Screen:          AACT_SEC,
Record Number:   28
  973:          *       * Variable:             m.cod_grupo
```

```
 974:          *        * Called By:            VALID Clause
 975:          *        * Object Type:          Field
 976:          *        * Snippet Number:       4
 977:          *        *
 978:          *
*************************************************************
 979:          *
 980:
*!*************************************************************************
**************
 981:  *!
 982:  *!       Function: _RA418TG1K
 983:  *!
 984:  *!      Called by: AACT_SEC.SPR
 985:  *!
 986:
*!*************************************************************************
**************
 987:     FUNCTION _ra418tg1k     &&  m.cod_grupo VALID
 988:         #REGION 1
 989:         malias = ALIAS()
 990:         SET RELATION OFF INTO dgt
 991:         SELE dgt
 992:         SEEK m.cod_grupo
 993:         STOR FOUND() TO vret
 994:         STOR dgt.cta_ctble TO m.cuent_cont
 995:         STOR m.cod_grupo TO m.grupotra
 996:         SHOW GET dgt.descrip
 997:         SELE &malias
 998:         SET RELATION TO grupotra INTO dgt ADDITIVE
 999: <=======RETURN vret
1000:
1001:
1002:          *
*************************************************************
1003:          *        *
1004:          *        * _RA418TG5H            Dgt.descrip VALID
1005:          *        *
1006:          *        * Function Origin:
1007:          *        *
1008:          *        * From Platform:        Windows
1009:          *        * From Screen:          AACT_SEC,
Record Number:   29
1010:          *        * Variable:             Dgt.descrip
1011:          *        * Called By:            VALID Clause
1012:          *        * Object Type:          List
1013:          *        * Snippet Number:       5
1014:          *        *
1015:          *
*************************************************************
1016:         *
1017:
*!*************************************************************************
**************
1018:  *!
1019:  *!       Function: _RA418TG5H
1020:  *!
1021:  *!      Called by: AACT_SEC.SPR
```

```
1022:  *!
1023:
*!*****************************************************************
**************
1024:      FUNCTION _ra418tg5h      &&  Dgt.descrip VALID
1025:          #REGION 1
1026:          STOR dgt.cod_grupo  TO m.grupotra
1027:          STOR dgt.cta_ctble TO m.cuent_cont
1028:          SHOW GETS
1029:
1030:          *
*************************************************************
1031:          *        *
1032:          *        * _RA418TG9Z            opc_scr VALID
1033:          *        *
1034:          *        * Function Origin:
1035:          *        *
1036:          *        * From Platform:        Windows
1037:          *        * From Screen:          AACT_SEC,
Record Number:   32
1038:          *        * Variable:             opc_scr
1039:          *        * Called By:            VALID Clause
1040:          *        * Object Type:          Push Button
1041:          *        * Snippet Number:       6
1042:          *        *
1043:          *
*************************************************************
1044:          *
1045:
*!*****************************************************************
**************
1046:  *!
1047:  *!        Function: _RA418TG9Z
1048:  *!
1049:  *!       Called by: AACT_SEC.SPR
1050:  *!
1051:  *!           Calls: &OPC_SCR.PRG
1052:  *!
1053:
*!*****************************************************************
**************
1054:      FUNCTION _ra418tg9z      &&  opc_scr VALID
1055:          #REGION 1
1056:          DO &opc_scr
1057:  *:  EOF: AACT_SEC.acl
```


Varios Archivo Nómina Bol de Pagos ReporTes Pmo.Emp Costos E.D.36 Salir
ACTUALIZACION DE CARGOS
CREAR
EDITAR
ELIMINAR
REVISAR
ACT SUELDOS
SALIR
BUSCAR POR
BUSCAR
CLAVE
PUES_COD
CODIGO
ESCALA
DENOMINACION
RELACION DE DEPENDENCIA
REEMPLAZO
GRUPO OCUPACIONAL
SUELDO
GAST REPR.
GAST REPR.
Anterior
Siguiente
Primero
Ultimo
Reg 85/120
Exclusivo


Varios Archivo Nómina Rol de Pagos Reportes Ptmo.Emp Costos E.D.36 Salir
ACTUALIZACION DE CARGOS
CREAR
EDITAR
ELIMINAR
REVISAR
ACT_SUELDOS
SALIR
BUSCAR_POR
BUSCAR
CLAVE
PUES_COD
REVISAR DENOMINACION DE PUESTOS
1100 JEFE DE UNIDAD INFORMATICA 11
1160 JEFE DE AREA 1 (SUPERV/SEGUR) 11
1169 CONT. AYUDANTE 1 (J/AREA 1 ENC) 11
1201 BODEGUERO 12
1202 JEFE DE AREA 3 12
1203 JEFE DE UNIDAD 2 12
1204 ODONTOLOGO (T.P.) 12
1205 JEFE DE AREA 2 (ANALISTA) 12
1206 CHOFER 1 (V.L.) (J/BODEGA ENC) 12
1207 MEDICO (AP) 12
1301 SECRET JURIDICO (T.P.) 12
1302 ASISTENTE DE AREA 13
1303 JEFE DE SECCION 1 13
1304 JEFE DE SECCION 1 (TESORERO) 13
1305 JEFE DE AREA INFORMATICA 13
1306 JEFE DE AREA 1 (J/SEC 1 ENC) 13
1307 CONT. AYUDANTE 1 (J/SEC 1 ENC) 13
1308 AUDITOR 1 (JEFE SECC. 1 ENC.) 13
1401 JEFE DE AREA 3 14
1402 JEFE DE SECCION 2 14
1403 ASISTENTE DE AREA (S.P.) 14
Anterior Siguiente Primero Ultimo
9:44
Clock
INS NUM

```
    1:
*:*****************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\DGP_CARG.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-06-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 12/12/1995 at 18:22:52
    8:  *:
    9:  *:  Procs & Fncts: _RAD13E8NZ()
   10:  *:                : _RAD13E8RD()
   11:  *:                : _RAD13E8U1()
   12:  *:                : _RAD13E8X2()
   13:  *:                : _RAD13E916()
   14:  *:                : _RAD13E951()
   15:  *:                : _RAD13E99D()
   16:  *:                : _RAD13E9CQ()
   17:  *:                : _RAD13E9FR()
   18:  *:                : _RAD13E9J9()
   19:  *:                : _RAD13E9LW()
   20:  *:                : BUSCAR_POR
   21:  *:                : BUSC_CLV()
   22:  *:                : ACT_SUELDOS
   23:  *:
   24:  *:          Set by: EMPRESA.MPR
   25:  *:
   26:  *:           Calls: TAG                  (procedure in
GRAFICOS.PRG)
   27:  *:                : ACT_KEYS             (procedure in
AACT_SST.SPR)
   28:  *:                : _RAD13E8NZ()         (function in
DGP_CARG.SPR)
   29:  *:                : _RAD13E8RD()         (function in
DGP_CARG.SPR)
   30:  *:                : _RAD13E8U1()         (function in
DGP_CARG.SPR)
   31:  *:                : _RAD13E8X2()         (function in
DGP_CARG.SPR)
   32:  *:                : _RAD13E916()         (function in
DGP_CARG.SPR)
   33:  *:                : _RAD13E951()         (function in
DGP_CARG.SPR)
   34:  *:                : _RAD13E99D()         (function in
DGP_CARG.SPR)
   35:  *:                : _RAD13E9CQ()         (function in
DGP_CARG.SPR)
   36:  *:                : _RAD13E9FR()         (function in
DGP_CARG.SPR)
   37:  *:                : _RAD13E9J9()         (function in
DGP_CARG.SPR)
   38:  *:                : _RAD13E9LW()         (function in
DGP_CARG.SPR)
   39:  *:
   40:  *:       Documented 19:14:31
FoxDoc version 3.00a
   41:
```

```
*:*************************************************************
**************
   42:      *
************************************************************
   43:      *        *
   44:      *        * 05/12/96             DGP_CARG.SPR
 18:22:48
   45:      *        *
   46:      *
************************************************************
   47:      *        *
   48:      *        * Nombre del autor
   49:      *        *
   50:      *        * Copyright (c) 1995 Nombre de la empresa
   51:      *        * Direcci¾n
   52:      *        * Ciudad,      C.P.
   53:      *        * Paŷs
   54:      *        *
   55:      *        * Description:
   56:      *        * Este programa lo ha generado automßticamente
GENSCRN.
   57:      *        *
   58:      *
************************************************************
   59:
   60:
   61:      *
************************************************************
   62:      *        *
   63:      *        *  DGP_CARG/Windows C¾digo de configuraci¾n -
SECCION 1
   64:      *        *
   65:      *
************************************************************
   66:      *
   67:
   68:      #REGION 1
   69:      **********
   70:      REGIONAL opc_scr, opc_busc, opc_jef, opc_tag, vartag,
mtag, opc_reldep,;
   71:           editando, creando, mclave, mcod_pues, reg_ant,
reg_act, opc_sn,;
   72:           reg_act1,vcod_real
   73:      STOR 1 TO opc_jef, opc_reldep, reg_ant,
reg_act,reg_act1,opc_sn
   74:      STOR '' TO opc_scr, opc_busc, opc_tag, vartag, mtag
   75:      STOR .F. TO editando, creando, mclave,vcod_real
   76:      STOR SPAC(5) TO mcod_pues
   77:      DIMENSION vartag(256)
   78:
   79:      **********
   80:
   81:      #REGION 0
   82:      REGIONAL m.currarea, m.talkstat, m.compstat
   83:
   84:      IF SET("TALK") = "ON"
   85:          SET TALK OFF
   86:          m.talkstat = "ON"
```

```
   87:     ELSE
   88:        m.talkstat = "OFF"
   89:     ENDIF
   90:     m.compstat = SET("COMPATIBLE")
   91:     SET COMPATIBLE FOXPLUS
   92:
   93:     m.rborder = SET("READBORDER")
   94:     SET readborder ON
   95:
   96:     m.currarea = SELECT()
   97:
   98:
   99:     *
************************************************************
  100:     *        *
  101:     *        *  DGP_CARG/Windows Bases de datos, Ýndices,
relaciones
  102:     *        *
  103:     *
************************************************************
  104:     *
  105:
  106:     IF USED("car")
  107:        SELECT car
  108:        SET ORDER TO TAG "pues_cod"
  109:     ELSE
  110:        SELECT 0
  111:        USE (LOCFILE("\dbfs\puestos.dbf","DBF","Buscar
puestos?"));
  112:           AGAIN ALIAS car ;
  113:           ORDER TAG "pues_cod"
  114:     ENDIF
  115:
  116:     IF USED("dgp")
  117:        SELECT dgp
  118:        SET ORDER TO 0
  119:     ELSE
  120:        SELECT 0
  121:        USE (LOCFILE("\dbfs\personal.dbf","DBF","Buscar
personal?"));
  122:           AGAIN ALIAS dgp ;
  123:           ORDER 0
  124:     ENDIF
  125:
  126:     IF USED("esc")
  127:        SELECT esc
  128:        SET ORDER TO TAG "escala"
  129:     ELSE
  130:        SELECT 0
  131:        USE (LOCFILE("\dbfs\escala.dbf","DBF","Buscar
escala?"));
  132:           AGAIN ALIAS esc ;
  133:           ORDER TAG "escala"
  134:     ENDIF
  135:
  136:     IF USED("lcc")
  137:        SELECT lcc
  138:        SET ORDER TO 0
```

```
   139:     ELSE
   140:         SELECT 0
   141:         USE (LOCFILE("\dbfs\ley_cont.dbf","DBF","Buscar
 ley_cont?"));
   142:             AGAIN ALIAS lcc ;
   143:             ORDER 0
   144:     ENDIF
   145:
   146:     SELECT dgp
   147:
   148:     SELECT car
   149:     SELECT car
   150:
   151:     SET RELATION OFF INTO esc
   152:     SET RELATION TO escala INTO esc ADDITIVE
   153:
   154:     SELECT dgp
   155:     SELECT dgp
   156:
   157:     SET RELATION OFF INTO car
   158:     SET RELATION TO dgp.cod_pues INTO car ADDITIVE
   159:
   160:     SELECT dgp
   161:
   162:     *
*************************************************************
   163:     *         *
   164:     *         *              Windows Definiciones de ventana
   165:     *         *
   166:     *
*************************************************************
   167:     *
   168:
   169:     IF NOT WEXIST("dgp_carg") ;
   170:           OR UPPER(WTITLE("DGP_CARG")) == "DGP_CARG.PJX" ;
   171:           OR UPPER(WTITLE("DGP_CARG")) == "DGP_CARG.SCX" ;
   172:           OR UPPER(WTITLE("DGP_CARG")) == "DGP_CARG.MNX" ;
   173:           OR UPPER(WTITLE("DGP_CARG")) == "DGP_CARG.PRG" ;
   174:           OR UPPER(WTITLE("DGP_CARG")) == "DGP_CARG.FRX" ;
   175:           OR UPPER(WTITLE("DGP_CARG")) == "DGP_CARG.QPR"
   176:        DEFINE WINDOW dgp_carg ;
   177:           AT  0.000, 0.000  ;
   178:           SIZE 28.308,104.800 ;
   179:           FONT "MS Sans Serif", 8 ;
   180:           FLOAT ;
   181:           NOCLOSE ;
   182:           MINIMIZE ;
   183:           SYSTEM ;
   184:           COLOR RGB(,,,192,192,192)
   185:        MOVE WINDOW dgp_carg CENTER
   186:     ENDIF
   187:
   188:
   189:     *
*************************************************************
   190:     *         *
   191:     *         *  DGP_CARG/Windows C#digo de configuraci#n -
SECCION 2
```

```
  192:      *       *
  193:      *
************************************************************
  194:      *
  195:
  196:      #REGION 1
  197:      **********
  198:      SELECT car
  199:      DO TAG
  200:      STOR vartag(1) TO mtag
  201:      DO act_keys
  202:      GO TOP
  203:      SCATTER TO reg_dgp BLANK
  204:
  205:      *
************************************************************
  206:      *       *
  207:      *       *        DGP_CARG/Windows Distribución de
pantalla
  208:      *       *
  209:      *
************************************************************
  210:      *
  211:
  212:      #REGION 1
  213:     ┌IF WVISIBLE("dgp_carg")
  214:     │   ACTIVATE WINDOW dgp_carg SAME
  215:     ├ELSE
  216:     │   ACTIVATE WINDOW dgp_carg NOSHOW
  217:     └ENDIF
  218:      @ 1.538,30.600 SAY "ACTUALIZACION DE CARGOS" ;
  219:         FONT "MS Sans Serif", 12 ;
  220:         STYLE "BIT"
  221:      @ 1.385,28.800 TO 3.462,84.800 ;
  222:         PEN 1, 8 ;
  223:         STYLE "16"
  224:      @ 5.769,26.600 SAY
"CODIGO.....:" ;
  225:         FONT "MS Serif", 8 ;
  226:         STYLE "T"
  227:      @ 7.615,26.600 SAY
"ESCALA.....:" ;
  228:         FONT "MS Serif", 8 ;
  229:         STYLE "T"
  230:      @ 9.923,26.600 SAY "DENOMINACION.....:" ;
  231:         FONT "MS Serif", 8 ;
  232:         STYLE "T"
  233:      @ 19.769,26.600 SAY
"JEFATURA.....:" ;
  234:         FONT "MS Serif", 8 ;
  235:         STYLE "T"
  236:      @ 13.231,26.800 SAY "REL.ACION " + CHR(13) + ;
  237:         "       DE" + CHR(13) + ;
  238:         "DEPENDENCIA.....:" + CHR(13) + ;
  239:         "" ;
  240:         SIZE 3.000,22.800, 0.000 ;
  241:         FONT "MS Serif", 8 ;
  242:         STYLE "T"
```

```
  243:      @ 2.538,1.000 TO 25.923,22.800 ;
  244:         PEN 1, 8 ;
  245:         STYLE "16"
  246:      @ 25.769,36.400 SAY "<PAG.ARR.> <PAG.ABAJ.> <INICIO>
<FIN>" ;
  247:         FONT "Futura Md BT", 8 ;
  248:         STYLE "BT"
  249:      @ 26.692,40.000 SAY "Anterior           Siguiente
   Primero      Ultimo" ;
  250:         FONT "MS Sans Serif", 8 ;
  251:         STYLE "T"
  252:      @ 22.077,29.400 SAY "SUELDO" ;
  253:         FONT "MS Serif", 7 ;
  254:         STYLE "BT"
  255:      @ 22.077,49.000 SAY "GAST.RESP." ;
  256:         FONT "MS Serif", 7 ;
  257:         STYLE "BT"
  258:      @ 22.154,68.800 SAY "GAST.REPR." ;
  259:         FONT "MS Serif", 7 ;
  260:         STYLE "BT"
  261:      @ 25.769,34.200 TO 27.923,87.600 ;
  262:         PEN 1, 8
  263:      @ 4.538,24.800 TO 24.923,100.600 ;
  264:         PEN 1, 8 ;
  265:         STYLE "16"
  266:      @ 11.615,73.200 SAY "REEMPLAZO:" ;
  267:         FONT "MS Serif", 8 ;
  268:         STYLE "T"
  269:      @ 17.538,73.200 SAY "GRUPO OCUPACIONAL" ;
  270:         FONT "MS Serif", 8 ;
  271:         STYLE "T"
  272:      @ 3.231,2.400 GET opc_scr ;
  273:         PICTURE "@*VN
!<CREAR;E!<DITAR;E!<LIMINAR;!<REVISAR;!<ACT_SUELDOS;!<SALIR" ;
  274:         SIZE 1.769,16.000,0.308 ;
  275:         DEFAULT 1 ;
  276:         FONT "MS Sans Serif", 8 ;
  277:         STYLE "B" ;
  278:         VALID _rad13e8nz()
  279:      @ 20.923,2.400 GET opc_tag ;
  280:         PICTURE "@*VN CLA!<VE" ;
  281:         SIZE 1.462,16.167,0.308 ;
  282:         DEFAULT 1 ;
  283:         FONT "MS Sans Serif", 8 ;
  284:         STYLE "B" ;
  285:         VALID _rad13e8rd()
  286:      @ 16.385,2.600 GET opc_busc ;
  287:         PICTURE "@*VN !!BUSCAR_POR;!<BUSCAR" ;
  288:         SIZE 1.769,16.000,0.308 ;
  289:         DEFAULT 1 ;
  290:         FONT "MS Sans Serif", 8 ;
  291:         STYLE "B" ;
  292:         VALID _rad13e8u1()
  293:      @ 22.231,2.600 GET mtag ;
  294:         PICTURE "@^" ;
  295:         FROM vartag ;
  296:         SIZE 1.538,16.000 ;
  297:         DEFAULT 1 ;
```

```
298:          FONT "MS Sans Serif", 8 ;
299:          STYLE "B" ;
300:          WHEN mclave ;
301:          VALID _rad13e8x2()
302:       @ 5.538,53.600 GET car.cod_pues ;
303:          SIZE 1.000,6.167 ;
304:          DEFAULT " " ;
305:          FONT "MS Sans Serif", 8 ;
306:          STYLE "B" ;
307:          WHEN .F. ;
308:          MESSAGE 'El C#digo se Forma con la Escala y Un No.'
309:       @ 7.462,53.600 GET car.escala ;
310:          SIZE 1.000,3.000 ;
311:          DEFAULT 0 ;
312:          FONT "MS Sans Serif", 8 ;
313:          STYLE "B" ;
314:          PICTURE "@K 99" ;
315:          WHEN editando ;
316:          VALID _rad13e916() ;
317:          MESSAGE ' El Rango de  Escala va entre 4..18' ;
318:          ERROR "Especificaci#n Invalida de Escala"
319:       @ 9.692,53.600 GET car.descrip ;
320:          SIZE 1.000,34.000 ;
321:          DEFAULT " " ;
322:          FONT "MS Sans Serif", 8 ;
323:          STYLE "B" ;
324:          PICTURE "@K!" ;
325:          WHEN editando ;
326:          VALID _rad13e951() ;
327:          ERROR 'Denominaci#n no Valida'
328:       @ 11.308,53.400 GET car.rel_dep ;
329:          PICTURE "@*RVN |<A Prueba;|<Eventual;Plazo
|<Fijo;De |<Planta;|<Jubilados" ;
330:          SIZE 1.308,12.833,0.308 ;
331:          DEFAULT 1 ;
332:          FONT "MS Serif", 8 ;
333:          STYLE "BT" ;
334:          WHEN editando ;
335:          VALID _rad13e99d() ;
336:          MESSAGE 'Jefatura Solo Puede Tener Personal de
Planta'
337:       @ 19.538,53.400 GET car.jefatura ;
338:          PICTURE "@*RHN SI;NO" ;
339:          SIZE 1.700,8.400,0.400 ;
340:          DEFAULT 1 ;
341:          FONT "MS Serif", 6 ;
342:          STYLE "BT" ;
343:          WHEN editando AND car.rel_dep = 4 ;
344:          VALID _rad13e9cq() ;
345:          MESSAGE ' Solo Personal de Planta Puede Tener
Jefatura'
346:       @ 13.000,73.600 GET opc_sn ;
347:          PICTURE "@*RVN SI;NO" ;
348:          SIZE 1.700,8.000,0.000 ;
349:          DEFAULT 1 ;
350:          FONT "MS Serif", 6 ;
351:          STYLE "BT" ;
352:          WHEN editando AND car.rel_dep = 4 ;
```

```
  353:         VALID _rad13e9fr() ;
  354:         MESSAGE ' Solo Personal de Planta Puede Tener
Jefatura'
  355:      @ 13.923,84.400 GET car.catego ;
  356:         SIZE 1.000,6.167 ;
  357:         DEFAULT " " ;
  358:         FONT "MS Sans Serif", 8 ;
  359:         STYLE "B" ;
  360:         WHEN vcod_real ;
  361:         VALID _rad13e9j9() ;
  362:         MESSAGE 'INGRESE EL CODIGO ORIGINAL DE LA PERSONA
QUE ESTA REEMPLAZANDO' ;
  363:         ERROR _rad13e9lw()
  364:      @ 18.615,73.400 GET car.gru_ocu ;
  365:         PICTURE "@*RVN EMP.ADM.;OBRERO" ;
  366:         SIZE 1.700,15.200,0.100 ;
  367:         DEFAULT 1 ;
  368:         FONT "MS Serif", 6 ;
  369:         STYLE "BT" ;
  370:         WHEN editando
  371:
  372:      IF NOT WVISIBLE("dgp_carg")
  373:         ACTIVATE WINDOW dgp_carg
  374:      ENDIF
  375:
  376:      READ CYCLE ;
  377:         VALID .F.
  378:
  379:      RELEASE WINDOW dgp_carg
  380:
  381:      *
************************************************************
  382:      *        *
  383:      *        *              Windows Cerrando bases de datos
  384:      *        *
  385:      *
************************************************************
  386:      *
  387:
  388:      IF USED("car")
  389:         SELECT car
  390:         USE
  391:      ENDIF
  392:
  393:      IF USED("dgp")
  394:         SELECT dgp
  395:         USE
  396:      ENDIF
  397:
  398:      IF USED("esc")
  399:         SELECT esc
  400:         USE
  401:      ENDIF
  402:
  403:      IF USED("lcc")
  404:         SELECT lcc
  405:         USE
  406:      ENDIF
  407:
```

```
  408:      SELECT (m.currarea)
  409:
  410:
  411:      #REGION 0
  412:
  413:      SET readborder &rborder
  414:
  415:     ┌IF m.talkstat = "ON"
  416:     │   SET TALK ON
  417:     └ENDIF
  418:     ┌IF m.compstat = "ON"
  419:     │   SET COMPATIBLE ON
  420:     └ENDIF
  421:
  422:
  423:      *
************************************************************
  424:      *        *
  425:      *        *           DGP_CARG/Windows Código de
limpieza
  426:      *        *
  427:      *
************************************************************
  428:      *
  429:
  430:      #REGION 1
  431:      ****************
  432:
  433:
  434:      *
************************************************************
  435:      *        *
  436:      *        * DGP_CARG/Windows Procedimientos y funciones
de soporte
  437:      *        *
  438:      *
************************************************************
  439:      *
  440:
  441:      #REGION 1
  442:
*!*************************************************************************
**************
  443:  *!
  444:  *!       Procedure: EDITAR
  445:  *!
  446:  *!       Called by: CREAR              (procedure in
AACT_SST.SPR)
  447:  *!
  448:  *!            Calls: DEAC_KEYS         (procedure in
AACT_SST.SPR)
  449:  *!                 : CABEZA            (procedure in
PERSONAL.PRG)
  450:  *!                 : INGRESO           (procedure in
ACT_PTMO.SPR)
  451:  *!                 : CREA_TMP          (procedure in
MO_PDLAB.SPR)
  452:  *!                 : MBROW             (procedure in
MO_PDLAB.SPR)
```

```
  453:  *!                : REF_SCR             (procedure in
MO_PDLAB.SPR)
  454:  *!                : MENSAJES            (procedure in
MO_PDLAB.SPR)
  455:  *!
  456:
*!*****************************************************************
**************
  457:      PROCEDURE editar
  458:          ****************
  459:          DO deac_keys
  460:          editando = .T.
  461:          SHOW GET opc_scr DISABLE
  462:          SHOW GET opc_busc DISABLE
  463:          SHOW GET opc_tag DISABLE
  464:          SHOW GET opc_scr,3 PROMPT '|<GRABAR'
  465:          SHOW GET opc_scr,4 PROMPT '|<CANCELAR'
  466:          STOR .T. TO editando
  467:          _CUROBJ = OBJNUM(car.escala)
  468:          SCATTER TO reg_dgp
  469: <========RETURN
  470:
  471:          ***************
  472:
*!*****************************************************************
**************
  473:  *!
  474:  *!       Procedure: CREAR
  475:  *!
  476:  *!            Calls: DEAC_KEYS         (procedure in
AACT_SST.SPR)
  477:  *!                 : BUSC_CLV()        (function in
DGP_CARG.SPR)
  478:  *!                 : EDITAR            (procedure in
CTRL_RES.SPR)
  479:  *!                 : PROCESO           (procedure in
DSC_GRTR.SPR)
  480:  *!                 : INGRESO           (procedure in
ACT_PTMO.SPR)
  481:  *!                 : CREA_TMP          (procedure in
MO_PDLAB.SPR)
  482:  *!                 : MBROW             (procedure in
MO_PDLAB.SPR)
  483:  *!                 : MENSAJES          (procedure in
MO_PDLAB.SPR)
  484:  *!                 : REF_SCR           (procedure in
MO_PDLAB.SPR)
  485:  *!
  486:
*!*****************************************************************
**************
  487:      PROCEDURE crear
  488:          ***************
  489:          DO deac_keys
  490:          reg_ant= RECNO()
  491:          creando = .T.
  492:          SET ORDER TO pues_cod
  493:        ┌─IF !busc_clv()
  494:        │    SET DELE OFF
```

```
  495:          |    LOCATE FOR DELE()
  496:          |   IF FOUND()
  497:          |      RECALL
  498:          |   ELSE
  499:          |      APPEND BLANK
  500:          |   ENDIF
  501:          |    SET DELETE ON
  502:          |    REPLACE  cod_pues WITH mcod_pues,;
  503:          |       escala   WITH VAL(SUBS(mcod_pues,1,2)),;
  504:          |       jefatura WITH 2,;
  505:          |       rel_dep  WITH 1
  506:          |    SHOW GETS
  507:          |    DO editar
  508:          |  ELSE
  509:          |       WAIT  'Código ya Existe' WINDOW NOWAIT
  510:          |       SHOW GETS
  511:          |  ENDIF
  512:  <=======RETURN
  513:
  514:          ****************
  515:
*!*********************************************************************
**************
  516:  *!
  517:  *!      Procedure: GRABAR
  518:  *!
  519:  *!      Called by: ING                 (procedure in
PERSONAL.PRG)
  520:  *!               : MERCE()             (function in
PERSONAL.PRG)
  521:  *!               : JUBILAR             (procedure in
PERSONAL.PRG)
  522:  *!
  523:  *!          Calls: ACT_KEYS            (procedure in
AACT_SST.SPR)
  524:  *!               : ELIM_MPL            (procedure in
MO_PDLAB.SPR)
  525:  *!               : INSERTAR()          (function in
MO_PDLAB.SPR)
  526:  *!               : BORRA_TMP           (procedure in
MO_PDLAB.SPR)
  527:  *!               : REF_SCR             (procedure in
MO_PDLAB.SPR)
  528:  *!
  529:  *!           Uses: TMP.DBF
  530:  *!
  531:
*!*********************************************************************
**************
  532:      PROCEDURE grabar
  533:          ****************
  534:          STOR .F. TO editando,creando
  535:          SHOW GET opc_scr,3 PROMPT 'E\<LIMINAR'
  536:          SHOW GET opc_scr,4 PROMPT '\<REVISAR'
  537:          SHOW GET opc_scr  ENABLE
  538:          SHOW GET opc_busc ENABLE
  539:          SHOW GET opc_tag  ENABLE
  540:          SHOW GETS
  541:          FLUSH
```

```
  542:          DO act_keys
  543:          SELE dgp
  544:          SCAN FOR dgp.cod_pues = car.cod_pues
  545:             REPLACE dgp.cargo    WITH car.descrip
  546:             REPLACE dgp.categor  WITH car.escala
  547:             REPLACE dgp.rel_dep  WITH car.rel_dep
  548:             REPLACE dgp.sueldo   WITH IIF(car.rel_dep = 4,
esc.suel_act,;
  549:                IIF(car.rel_dep = 5,;
  550:                lcc.cc_sjubil, esc.suel_ant))
  551:             REPLACE dgp.gast_res WITH IIF(car.rel_dep = 4;
  552:                AND car.jefatura = 1,esc.vgres_act,0)
  553:             REPLACE dgp.gast_rep WITH IIF(car.rel_dep = 4;
  554:                AND car.jefatura = 1,esc.vgrep_act,0)
  555:             REPLACE dgp.cod_reempl WITH car.catego
  556:          ENDSCAN
  557:          SELE car
  558:          REPLACE car.sueldo WITH IIF(car.rel_dep = 4,
esc.suel_act, IIF(car.rel_dep = 5, lcc.cc_sjubil,esc.suel_ant))
  559:          RETURN
  560:
  561:          ******************
  562:
*!*************************************************************
**************
  563:  *!
  564:  *!       Procedure: CANCELAR
  565:  *!
  566:  *!       Called by: ING                (procedure in
PERSONAL.PRG)
  567:  *!                : JUBILAR            (procedure in
PERSONAL.PRG)
  568:  *!
  569:  *!           Calls: MSCATTER           (procedure in
AACT_SST.SPR)
  570:  *!                : ACT_KEYS           (procedure in
AACT_SST.SPR)
  571:  *!                : DESPLEGA           (procedure in
PERSONAL.PRG)
  572:  *!                : BORRA_TMP          (procedure in
MO_PDLAB.SPR)
  573:  *!                : ASIGNAM            (procedure in
MO_PDLAB.SPR)
  574:  *!                : REF_SCR            (procedure in
MO_PDLAB.SPR)
  575:  *!
  576:
*!*************************************************************
**************
  577:      PROCEDURE cancelar
  578:          ******************
  579:          SHOW GET opc_scr,3 PROMPT 'E\<LIMINAR'
  580:          SHOW GET opc_scr,4 PROMPT '\<REVISAR'
  581:          SHOW GET opc_scr  ENABLE
  582:          SHOW GET opc_busc ENABLE
  583:          SHOW GET opc_tag  ENABLE
  584:          IF creando
  585:             SCATTER TO reg_dgp BLANK
  586:             GATHER FROM reg_dgp
```

```
  587:          |    DELETE
  588:          |    GOTO reg_ant
  589:          |-ELSE
  590:          |    GATHER FROM reg_dgp
  591:          |-ENDIF
  592:            STOR .F. TO editando,creando
  593:            SHOW GETS
  594:            FLUSH
  595:            DO act_keys
  596: <========RETURN
  597:            ******************
  598:
*!*************************************************************************
**************
  599:  *!
  600:  *!       Procedure: ELIMINAR
  601:  *!
  602:  *!           Calls: MSCATTER            (procedure in
AACT_SST.SPR)
  603:  *!                : DEAC_KEYS          (procedure in
AACT_SST.SPR)
  604:  *!                : ACT_KEYS           (procedure in
AACT_SST.SPR)
  605:  *!                : BORRAR             (procedure in
DSC_GRTR.SPR)
  606:  *!                : ELIM_MPL           (procedure in
MO_PDLAB.SPR)
  607:  *!                : CTR_POSP           (procedure in
AACT_SST.SPR)
  608:  *!
  609:
*!*************************************************************************
**************
  610:      PROCEDURE eliminar
  611:          ******************
  612:          DO deac_keys
  613:        ┌-IF NOT WEXIST('BUSCAR') && o definir en el
principal
  614:        |    DEFINE WINDOW buscar FROM 0,0 TO 5,30 TITLE
'BUSCAR' SYSTEM
  615:        └-ENDIF
  616:          MOVE WINDOW buscar CENTER
  617:          ACTIVATE WINDOW buscar
  618:          SELE dgp
  619:          LOCATE FOR ALLT(dgp.cod_pues) = ALLT(car.cod_pues)
  620:          BROW
  621:          ?FOUND()
  622:        ┌-IF FOUND()
  623:        |    ?CHR(7)
  624:        |    @ 0,1 SAY 'Código Con Movimoientos'
  625:        |    @ 1,1 SAY 'No Puede Ser Eliminado'
  626:        |    STOR 1 TO opc_ec
  627:        |    @ 2,10 GET opc_ec ;
  628:        |       PICTURE "@*h |<ACEPTAR" ;
  629:        |       SIZE 1.3,15 ;
  630:        |       DEFAULT 1 ;
  631:        |       FONT "MS Sans Serif", 8 ;
  632:        |       STYLE "N"
  633:        |    READ
```

```
  634:          |    SELE car
  635:          |ELSE
  636:          |    ?CHR(7)
  637:          |    STOR 2 TO opc_ec
  638:          |    @ 0,1 SAY 'Eliminar El Registro ...?'
  639:          |    @ 1,2 GET opc_ec ;
  640:          |       PICTURE "@*h E\<LIMINAR;\<CANCELAR" :
  641:          |       SIZE 1.5,10 ;
  642:          |       DEFAULT 1 ;
  643:          |       FONT "MS Sans Serif", 8 :
  644:          |       STYLE "B"
  645:          |    READ
  646:          |    SELE car
  647:          |  IF opc_ec = 1
  648:          |  |    DELE
  649:          |  |    GO reg_ant
  650:          |  |    SHOW GETS
  651:          |  ENDIF
  652:          |ENDIF
  653:          DEACTIVATE WINDOW buscar
  654:          DO act_keys
  655: <========RETURN
  656:          *****************
  657:
*!*****************************************************************
**************
  658:  *!
  659:  *!      Procedure: REVISAR
  660:  *!
  661:  *!          Calls: MSCATTER           (procedure in
AACT_SST.SPR)
  662:  *!               : DEAC_KEYS          (procedure in
AACT_SST.SPR)
  663:  *!               : ACT_KEYS           (procedure in
AACT_SST.SPR)
  664:  *!               : VAL_CAMPO()        (function in
AP_PRMCT.SPR)
  665:  *!               : REFRESH()          (function in
AP_PRMCT.SPR)
  666:  *!               : MBROW              (procedure in
MO_PDLAB.SPR)
  667:  *!
  668:
*!*****************************************************************
**************
  669:     PROCEDURE revisar
  670:          *****************
  671:          DO deac_keys
  672:       IF NOT WEXIST('REVISAR') && o definir en el
principal
  673:       |    DEFINE WIND revisar FROM 10,24.800 TO
40.923,83.400 ;
  674:       |       TITLE 'REVISAR DENOMINACION DE PUESTOS'
SYSTEM
  675:       ENDIF
  676:          ACTI WIND revisar
  677:          BROW FIELDS car.cod_pues:H='Código':5,
car.descrip:H='Denominación':30,;
  678:               car.escala:2:p='99';
```

```
  679:             IN WIND revisar NOEDI NOMENU
  680:          RELE WIND revisar
  681:          DO act_keys
  682:          SHOW GETS
  683: <=======RETURN
  684:          ***************
  685:
*!*****************************************************************
**************
  686:  *!
  687:  *!       Procedure: SALIR
  688:  *!
  689:  *!       Called by: _RAA1732TG()        (function in
AT_ESCLS.SPR)
  690:  *!                : DA_ASISG.PRG
  691:  *!                : DE_EGRQ1.PRG
  692:  *!                : DE_EGRQ2.PRG
  693:  *!                : DE_EGRMN.PRG
  694:  *!                : DE_IEJUB.PRG
  695:  *!
  696:  *!           Calls: DEAC_KEYS           (procedure in
AACT_SST.SPR)
  697:  *!
  698:
*!*****************************************************************
**************
  699:      PROCEDURE salir
  700:          ***************
  701:          DO deac_keys
  702:          CLEAR READ
  703: <=======RETURN
  704:          ********************
  705:
*!*****************************************************************
**************
  706:  *!
  707:  *!       Procedure: BUSCAR_POR
  708:  *!
  709:
*!*****************************************************************
**************
  710:      PROCEDURE buscar_por
  711:          ********************
  712: <=======RETURN
  713:          ****************
  714:
*!*****************************************************************
**************
  715:  *!
  716:  *!       Procedure: BUSCAR
  717:  *!
  718:  *!       Called by: SECC()              (function in
PERSONAL.PRG)
  719:  *!                : _R390JW4G3()        (function in
DSC_GRTR.SPR)
  720:  *!                : _RA81B8ER0()        (function in
ACT_PTMO.SPR)
  721:  *!                : _RA81B8GQD()        (function in
ACT_PTMO.SPR)
```

```
  722:  *!                : _R6108FBLN()        (function in
MO_PDLAB.SPR)
  723:  *!                : _R5K0M9Z09()        (function in
DSC_CTA.SPR)
  724:  *!
  725:  *!           Calls: VAL_BUSCAR          (procedure in
AACT_SST.SPR)
  726:  *!                : BUSC_CLV()          (function in
DGP_CARG.SPR)
  727:  *!                : DESPLEGA            (procedure in
PERSONAL.PRG)
  728:  *!                : BUSCARP()           (function in
ACT_PTMO.SPR)
  729:  *!                : REF_SCR             (procedure in
MO_PDLAB.SPR)
  730:  *!                : MENSAJES            (procedure in
MO_PDLAB.SPR)
  731:  *!                : ASIGNAM             (procedure in
MO_PDLAB.SPR)
  732:  *!
  733:
*!*****************************************************************
**************
  734:      PROCEDURE buscar
  735:          ****************
  736:        ┌─IF NOT WEXIST('BUSCAR') && o definir en el
principal
  737:        │     DEFINE WINDOW buscar FROM 0,0 TO 5,30 TITLE
'BUSCAR' SYSTEM
  738:        └─ENDIF
  739:          ACTIVATE WINDOW buscar
  740:          MOVE WINDOW buscar CENTER
  741:        ┌─IF !busc_clv()
  742:        │     ?CHR(7)
  743:        │     @ 1,1 SAY 'Código no Existe'
  744:        │     READ
  745:        └─ENDIF
  746:          SHOW GETS
  747: <═══════RETURN
  748:          ******************
  749:
*!*****************************************************************
**************
  750:  *!
  751:  *!        Function: BUSC_CLV
  752:  *!
  753:  *!      Called by: CREAR              (procedure in
AACT_SST.SPR)
  754:  *!                : BUSCAR            (procedure in
AACT_SST.SPR)
  755:  *!                : OPC_DES()         (function in
DA_ASISG.PRG)
  756:  *!                : INGRESO           (procedure in
ACT_PTMO.SPR)
  757:  *!
  758:  *!           Calls: RUBROS            (procedure in
DA_ASISG.PRG)
  759:  *!                : AYU_DPTO          (procedure in
MO_PDLAB.SPR)
```

```
  760:  *!                : BUSC_GRUPO()        (function in
MO_PDLAB.SPR)
  761:  *!
  762:
*!*************************************************************
**************
  763:      FUNCTION busc_clv
  764:          ******************
  765:        ┌─IF NOT WEXIST('BUSCAR') && o definir en el
principal
  766:        │     DEFINE WINDOW buscar FROM 0,0 TO 5,30 TITLE
'BUSCAR' SYSTEM
  767:        └─ENDIF
  768:          ACTIVATE WINDOW buscar
  769:          MOVE WINDOW buscar CENTER
  770:          reg_ant = RECNO()
  771:          SET ORDER TO pues_cod
  772:          mcod_pues = SPAC(5)
  773:        ┌─DO WHILE .T.
  774:        │     @ 1,1 SAY 'Ingrese el C¾digo: ' GET mcod_pues
PICT '99999'
  775:        │     READ MODAL
  776:        │   ┌─IF creando AND LAST() = 27
  777:        │ <═══LOOP
  778:        │   └─ENDIF
  779:        │   ┌─IF LAST() = 27
  780:        │   │     GOTO reg_ant
  781:        │   │     SHOW GETS
  782:        │   │     vret = .F.
  783:        v═══════EXIT
  784:        │   └─ENDIF
  785:        └─ENDIF
  786:          SEEK mcod_pues
  787:        ┌─IF !FOUND()
  788:        │     vret = .F.
  789:        │   ┌─IF creando
*:EXIT error detected
EXIT
  791:        │   ├─ELSE
  792:        │   │     WAIT  'C¾digo no Existe' WINDOW NOWAIT
  793:        │   │     GOTO reg_ant
*:LOOP error detected
LOOP
  795:        │   └─ENDIF
  796:        ├─ELSE
  797:        │     vret = .T.
  798:        │     SHOW GETS
*:EXIT error detected
EXIT
  800:        └─ENDIF
  801:      ENDDO
  802:      DEACTIVATE WINDOW buscar
  803: <═══RETURN vret
  804:
  805:       ******************
  806:
 *!*************************************************************
 **************
  807:  *!
```

```
  808:  *!       Procedure: ACT_KEYS
  809:  *!
  810:  *!       Called by: AACT_SST.SPR
  811:  *!                : AACT_DPT.SPR
  812:  *!                : AACT_SEC.SPR
  813:  *!                : DGP_CARG.SPR
  814:  *!                : AC_IMPRT.SPR
  815:  *!                : DSC_GRTR.SPR
  816:  *!                : ELIMINAR            (procedure in
AACT_SST.SPR)
  817:  *!                : REVISAR             (procedure in
AACT_SST.SPR)
  818:  *!                : CTR_POSP            (procedure in
AACT_SST.SPR)
  819:  *!                : GRABAR              (procedure in
AACT_SST.SPR)
  820:  *!                : CANCELAR            (procedure in
AACT_SST.SPR)
  821:  *!                : ROL_Q1.SPR
  822:  *!                : ROL_Q2.SPR
  823:  *!                : ACT_PTMO.SPR
  824:  *!                : MO_PDLAB.SPR
  825:  *!                : DSC_CTA.SPR
  826:  *!                : _R6108FB17()        (function in
MO_PDLAB.SPR)
  827:  *!
  828:  *!          Calls: CTR_POSP             (procedure in
AACT_SST.SPR)
  829:  *!
  830:
*!*****************************************************************
**************
  831:      PROCEDURE act_keys
  832:          ******************
  833:          ON KEY LABEL pgup DO ctr_posp WITH
'ANTERIOR','SKIP -1'
  834:          ON KEY LABEL pgdn DO ctr_posp WITH
'SIGUIENTE','SKIP'
  835:          ON KEY LABEL home DO ctr_posp WITH 'PRIMERO','GOTO
TOP'
  836:          ON KEY LABEL END DO ctr_posp WITH 'ULTIMO','GOTO
BOTTOM'
  837: <=======RETURN
  838:
  839:          *********************
  840:
*!*****************************************************************
**************
  841:  *!
  842:  *!       Procedure: DEAC_KEYS
  843:  *!
  844:  *!       Called by: EDITAR             (procedure in
CTRL_RES.SPR)
  845:  *!                : SALIR              (procedure in
CTRL_RES.SPR)
  846:  *!                : CREAR              (procedure in
AACT_SST.SPR)
  847:  *!                : ELIMINAR           (procedure in
AACT_SST.SPR)
```

```
  848:  *!                  : REVISAR           (procedure in
AACT_SST.SPR)
  849:  *!                  : CTR_POSP          (procedure in
AACT_SST.SPR)
  850:  *!                  : _R6108FBI7()      (function in
MO_PDLAB.SPR)
  851:  *!
  852:
*!*****************************************************************
**************
  853:      PROCEDURE deac_keys
  854:          ********************
  855:          ON KEY LABEL pgup
  856:          ON KEY LABEL pgdn
  857:          ON KEY LABEL home
  858:          ON KEY LABEL END
  859: <=======RETURN
  860:
  861:          ********************
  862:
*!*****************************************************************
**************
  863:  *!
  864:  *!      Procedure: CTR_POSP
  865:  *!
  866:  *!      Called by: ACT_KEYS           (procedure in
AACT_SST.SPR)
  867:  *!                 : ELIMINAR         (procedure in
AACT_SST.SPR)
  868:  *!
  869:  *!           Calls: MSCATTER          (procedure in
AACT_SST.SPR)
  870:  *!                 : DEAC_KEYS        (procedure in
AACT_SST.SPR)
  871:  *!                 : ASIGNAM          (procedure in
MO_PDLAB.SPR)
  872:  *!                 : REF_SCR          (procedure in
MO_PDLAB.SPR)
  873:  *!                 : ACT_KEYS         (procedure in
AACT_SST.SPR)
  874:  *!
  875:
*!*****************************************************************
**************
  876:      PROCEDURE ctr_posp
  877:          ********************
  878:          PARAMETER opc_posp, comando
  879:        ┌─IF BOF()
  880:        │   GOTO TOP
  881:        └─ENDIF
  882:        ┌─IF EOF()
  883:        │   GOTO BOTTOM
  884:        └─ENDIF
  885:          &comando
  886:        ı=DO CASE
  887:        ã═══CASE opc_posp = 'PRIMERO'
  888:        │       WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
  889:        ã═══CASE opc_posp = 'SIGUIENTE'
  890:        │     ┌─IF EOF()
```

```
 891:                 GOTO BOTTOM
 892:                 WAIT WIND NOWAIT 'FIN DE ARCHIVO'
 893:              ENDIF
 894:           CASE opc_posp = 'ANTERIOR'
 895:              IF BOF()
 896:                 GOTO TOP
 897:                 WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
 898:              ENDIF
 899:           CASE opc_posp = 'ULTIMO'
 900:                 WAIT WIND NOWAIT 'FIN DE ARCHIVO'
 901:           ENDCASE
 902:           SHOW GETS
 903:        RETURN
 904:
 905:           **********************
 906:
*!*************************************************************
**************
 907:  *!
 908:  *!       Procedure: ACT_SUELDOS
 909:  *!
 910:
*!*************************************************************
**************
 911:     PROCEDURE act_sueldos
 912:           **********************
 913:        IF NOT WEXIST('BUSCAR') && o definir en el
principal
 914:           DEFINE WINDOW buscar FROM 0,0 TO 5,40 TITLE
'BUSCAR' SYSTEM
 915:        ENDIF
 916:        MOVE WINDOW buscar CENTER
 917:        ACTIVATE WINDOW buscar
 918:        ?CHR(7)
 919:        @ 0,1 SAY 'Este Proceso Actualizarß Los Sueldos'
 920:        @ 1,3 SAY 'En El Archivo Maestro de Personal'
 921:        STOR 1 TO opc_ec
 922:        ?CHR(7)
 923:        @ 2,8 GET opc_ec ;
 924:           PICTURE "@*h |<ACEPTAR;|<CANCELAR" ;
 925:           SIZE 1,10 ;
 926:           DEFAULT 1 ;
 927:           FONT "MS Sans Serif", 8 ;
 928:           STYLE "B"
 929:        READ
 930:        DEACTIVATE WINDOW buscar
 931:        IF opc_ec = 1
 932:           SELE dgp
 933:           SCAN FOR estado # 2
 934:              REPLACE dgp.cargo    WITH car.descrip
 935:              REPLACE dgp.categor  WITH car.escala
 936:              REPLACE dgp.rel_dep  WITH car.rel_dep
 937:              REPLACE dgp.sueldo   WITH IIF(car.rel_dep =
4, esc.suel_act,;
 938:                 IIF(car.rel_dep = 5,;
 939:                 lcc.cc_sjubil, esc.suel_ant))
 940:              REPLACE dgp.gast_res WITH IIF(car.rel_dep =
4;
 941:                 AND car.jefatura = 1,esc.vgres_act,0)
```

```
  942:        |  ||    REPLACE dgp.gast_rep WITH IIF(car.rel_dep =
4;
  943:        |  ||       AND car.jefatura = 1,esc.vgrep_act,0)
  944:        |  ||    REPLACE dgp.cod_reempl WITH car.catego
  945:        |  |└═ENDSCAN
  946:        |  |  SELE car
  947:        |  |  REPLACE car.sueldo WITH IIF(car.rel_dep = 4,
esc.suel_act, IIF(car.rel_dep = 5, lcc.cc_sjubil,esc.suel_ant))
  948:        |└═ENDIF
  949: <═══════RETURN
  950:
  951:        *
************************************************************
  952:        *       *
  953:        *       * _RAD13E8NZ            OPC_SCR VALID
  954:        *       *
  955:        *       * Function Origin:
  956:        *       *
  957:        *       * From Platform:        Windows
  958:        *       * From Screen:          DGP_CARG,
Record Number:   25
  959:        *       * Variable:             OPC_SCR
  960:        *       * Called By:            VALID Clause
  961:        *       * Object Type:          Push Button
  962:        *       * Snippet Number:       1
  963:        *       *
  964:        *
************************************************************
  965:        *
  966:
*!*************************************************************************
**************
  967:  *!
  968:  *!       Function: _RAD13E8NZ
  969:  *!
  970:  *!      Called by: DGP_CARG.SPR
  971:  *!
  972:  *!          Calls: &OPC_SCR.PRG
  973:  *!
  974:
*!*************************************************************************
**************
  975:     FUNCTION _rad13e8nz     &&  OPC_SCR VALID
  976:         #REGION 1
  977:         DO &opc_scr
  978:
  979:        *
************************************************************
  980:        *       *
  981:        *       * _RAD13E8RD            OPC_TAG VALID
  982:        *       *
  983:        *       * Function Origin:
  984:        *       *
  985:        *       * From Platform:        Windows
  986:        *       * From Screen:          DGP_CARG,
Record Number:   26
  987:        *       * Variable:             OPC_TAG
  988:        *       * Called By:            VALID Clause
  989:        *       * Object Type:          Push Button
```

```
 990:          *        * Snippet Number:       2
 991:          *        *
 992:          *
************************************************************
 993:          *
 994:
*!*************************************************************************
**************
 995:   *!
 996:   *!        Function: _RAD13E8RD
 997:   *!
 998:   *!       Called by: DGP_CARG.SPR
 999:   *!
1000:
*!*************************************************************************
**************
1001:      FUNCTION _rad13e8rd      &&  OPC_TAG VALID
1002:          #REGION 1
1003:          STOR .T. TO mclave
1004:          _CUROBJ= OBJNUM(mtag)
1005:
1006:          *
************************************************************
1007:          *        *
1008:          *        * _RAD13E8U1            OPC_BUSC VALID
1009:          *        *
1010:          *        * Function Origin:
1011:          *        *
1012:          *        * From Platform:        Windows
1013:          *        * From Screen:          DGP_CARG,
Record Number:   27
1014:          *        * Variable:             OPC_BUSC
1015:          *        * Called By:            VALID Clause
1016:          *        * Object Type:          Push Button
1017:          *        * Snippet Number:       3
1018:          *        *
1019:          *
************************************************************
1020:          *
1021:
*!*************************************************************************
**************
1022:   *!
1023:   *!        Function: _RAD13E8U1
1024:   *!
1025:   *!       Called by: DGP_CARG.SPR
1026:   *!
1027:   *!           Calls: &OPC_BUSC.PRG
1028:   *!
1029:
*!*************************************************************************
**************
1030:      FUNCTION _rad13e8u1      &&  OPC_BUSC VALID
1031:          #REGION 1
1032:          DO &opc_busc
1033:
1034:          *
************************************************************
1035:          *        *
```

```
 1036:          *        * _RAD13E8X2           MTAG VALID
 1037:          *        *
 1038:          *        * Function Origin:
 1039:          *        *
 1040:          *        * From Platform:       Windows
 1041:          *        * From Screen:         DGP_CARG,
Record Number:   28
 1042:          *        * Variable:            MTAG
 1043:          *        * Called By:           VALID Clause
 1044:          *        * Object Type:         Popup
 1045:          *        * Snippet Number:      4
 1046:          *        *
 1047:          *
*************************************************************
 1048:          *
 1049:
*!*************************************************************************
**************
 1050:   *!
 1051:   *!       Function: _RAD13E8X2
 1052:   *!
 1053:   *!      Called by: DGP_CARG.SPR
 1054:   *!
 1055:
*!*************************************************************************
**************
 1056:      FUNCTION _rad13e8x2     &&  MTAG VALID
 1057:          #REGION 1
 1058:          SET ORDER TO &mtag
 1059:          STOR .F. TO mclave
 1060:
 1061:          *
*************************************************************
 1062:          *        *
 1063:          *        * _RAD13E916           Car.escala VALID
 1064:          *        *
 1065:          *        * Function Origin:
 1066:          *        *
 1067:          *        * From Platform:       Windows
 1068:          *        * From Screen:         DGP_CARG,
Record Number:   30
 1069:          *        * Variable:            Car.escala
 1070:          *        * Called By:           VALID Clause
 1071:          *        * Object Type:         Field
 1072:          *        * Snippet Number:      5
 1073:          *        *
 1074:          *
*************************************************************
 1075:          *
 1076:
*!*************************************************************************
**************
 1077:   *!
 1078:   *!       Function: _RAD13E916
 1079:   *!
 1080:   *!      Called by: DGP_CARG.SPR
 1081:   *!
 1082:
*!*************************************************************************
```

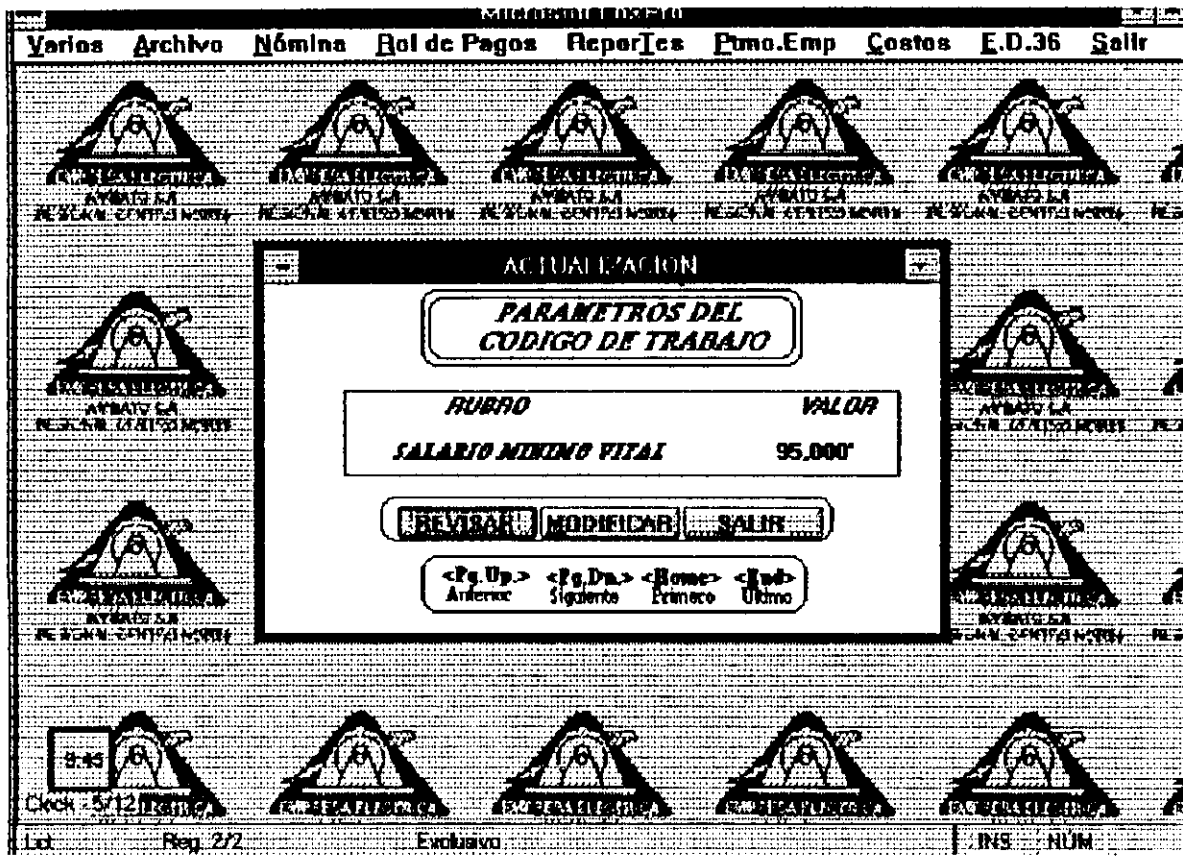
```
1083:      FUNCTION _rad13e916      && Car.escala VALID
1084:          #REGION 1
1085:         IF (car.escala > 3 AND car.escala < 19) OR
(car.escala = 1)
1086:            IF (car.escala = 1)
1087:               REPLACE car.rel_dep WITH 5
1088:               REPLACE car.jefatura WITH 2
1089:               SHOW GETS
1090:               SHOW GET rel_dep DISA
1091:               SHOW GET opc_jef DISA
1092:            ENDIF
1093:         ELSE
1094:
1095:         ENDIF
1096:
1097:          *
************************************************************
1098:          *        *
1099:          *        * _RAD13E951            Car.descrip VALID
1100:          *        *
1101:          *        * Function Origin:
1102:          *        *
1103:          *        * From Platform:        Windows
1104:          *        * From Screen:          DGP_CARG,
Record Number:   31
1105:          *        * Variable:             Car.descrip
1106:          *        * Called By:            VALID Clause
1107:          *        * Object Type:          Field
1108:          *        * Snippet Number:       6
1109:          *        *
1110:          *
************************************************************
1111:          *
1112:
*!*****************************************************************
**************
1113:  *!
1114:  *!       Function: _RAD13E951
1115:  *!
1116:  *!      Called by: DGP_CARG.SPR
1117:  *!
1118:
*!*****************************************************************
**************
1119:      FUNCTION _rad13e951      && Car.descrip VALID
1120:          #REGION 1
1121:          vret = .F.
1122:         IF EMPTY(car.descrip)
1123:            WAIT WIND 'Ingrese la Denominaci¥n'
1124: <=========RETURN .F.
1125:         ENDIF
1126:          reg_act = RECNO()
1127:          STOR car.descrip TO mdescrip
1128:          LOCATE FOR car.descrip = mdescrip
1129:         IF FOUND() AND RECNO() # reg_act
1130:            WAIT WIND 'Cargo ya Existe'
1131:            GO reg_act
1132:            STOR .F. TO vret
1133:         ELSE
```

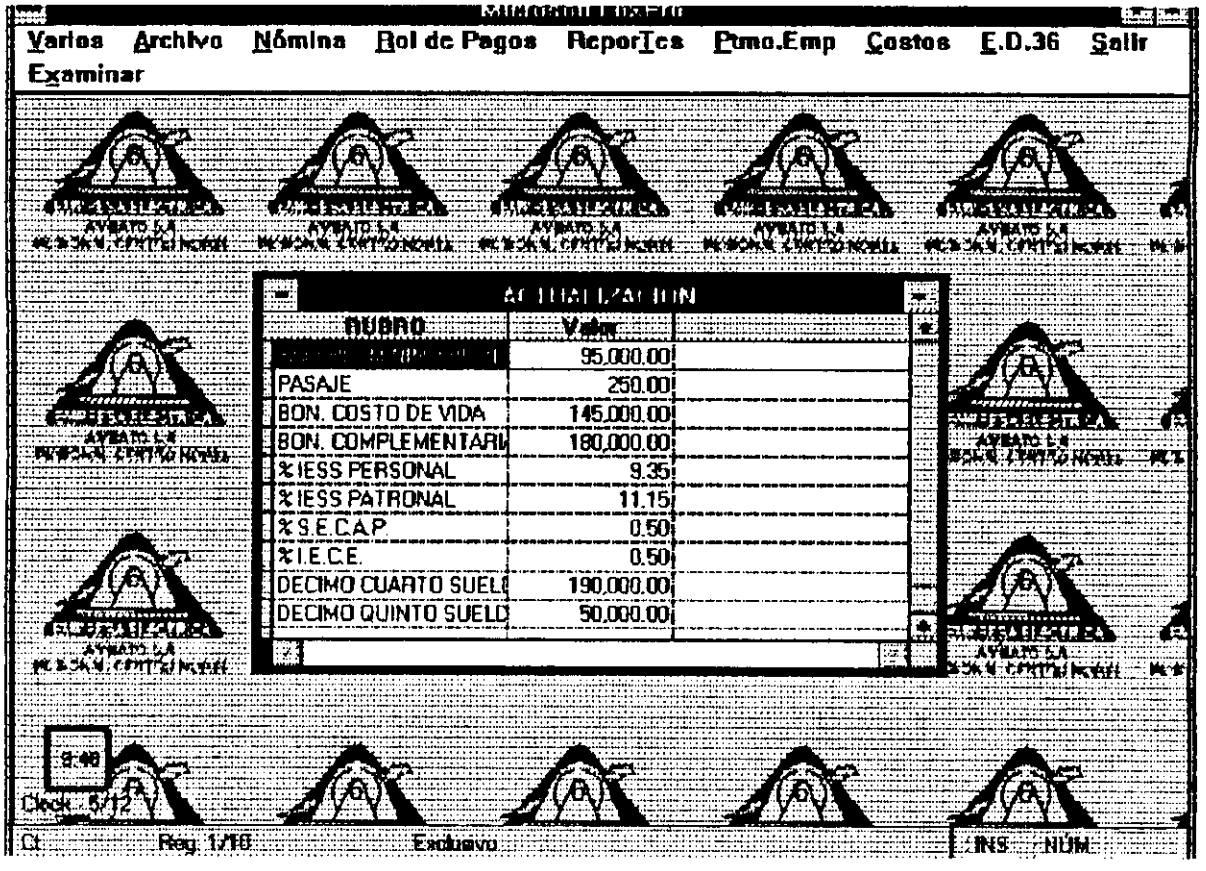
```
1134:          |    STOR .T. TO vret
1135:          |ENDIF
1136: <=======RETURN vret
1137:
1138:          *
************************************************************
1139:          *       *
1140:          *       * _RAD13E99D            Car.rel_dep VALID
1141:          *       *
1142:          *       * Function Origin:
1143:          *       *
1144:          *       * From Platform:        Windows
1145:          *       * From Screen:          DGP_CARG,
Record Number:   32
1146:          *       * Variable:             Car.rel_dep
1147:          *       * Called By:            VALID Clause
1148:          *       * Object Type:          Radio Button
1149:          *       * Snippet Number:       7
1150:          *       *
1151:          *
************************************************************
1152:          *
1153:
*!*************************************************************************
**************
1154:  *!
1155:  *!       Function: _RAD13E99D
1156:  *!
1157:  *!      Called by: DGP_CARG.SPR
1158:  *!
1159:
*!*************************************************************************
**************
1160:     FUNCTION _rad13e99d     &&  Car.rel_dep VALID
1161:         #REGION 1
1162:        |IF car.rel_dep # 4
1163:        |   REPLACE car.jefatura WITH 2
1164:        |   SHOW GETS
1165:        |   SHOW GET opc_jef DISA
1166:        |ENDIF
1167:         REPLACE car.nivel  WITH IIF(car.rel_dep =
4,'0','1')
1168:
1169:          *
************************************************************
1170:          *       *
1171:          *       * _RAD13E9CQ            Car.jefatura VALID
1172:          *       *
1173:          *       * Function Origin:
1174:          *       *
1175:          *       * From Platform:        Windows
1176:          *       * From Screen:          DGP_CARG,
Record Number:   33
1177:          *       * Variable:             Car.jefatura
1178:          *       * Called By:            VALID Clause
1179:          *       * Object Type:          Radio Button
1180:          *       * Snippet Number:       8
1181:          *       *
1182:          *
```

```
*************************************************************
 1183:          *
 1184:
*!*************************************************************************
**************
 1185:   *!
 1186:   *!        Function: _RAD13E9CQ
 1187:   *!
 1188:   *!       Called by: DGP_CARG.SPR
 1189:   *!
 1190:
*!*************************************************************************
**************
 1191:      FUNCTION _rad13e9cq      &&  Car.jefatura VALID
 1192:          #REGION 1
  1193:        ┌─IF car.rel_dep # 4 AND car.jefatura = 1
?CHR(7)
 1194:        │     WAIT 'Dato Erróneo ...!'
 1195:        └─ENDIF
 1196:
 1197:          *
*************************************************************
 1198:          *        *
 1199:          *        * _RAD13E9FR            OPC_SN VALID
 1200:          *        *
 1201:          *        * Function Origin:
 1202:          *        *
 1203:          *        * From Platform:        Windows
 1204:          *        * From Screen:          DGP_CARG.
Record Number:   34
 1205:          *        * Variable:             OPC_SN
 1206:          *        * Called By:            VALID Clause
 1207:          *        * Object Type:          Radio Button
 1208:          *        * Snippet Number:       9
 1209:          *        *
 1210:          *
*************************************************************
 1211:          *
 1212:
*!*************************************************************************
**************
 1213:   *!
 1214:   *!        Function: _RAD13E9FR
 1215:   *!
 1216:   *!       Called by: DGP_CARG.SPR
 1217:   *!
 1218:
*!*************************************************************************
**************
 1219:      FUNCTION _rad13e9fr      &&  OPC_SN VALID
 1220:          #REGION 1
 1221:          STOR .F. TO vcod_real
  1222:        ┌─IF car.rel_dep # 4 AND car.jefatura = 1
?CHR(7)
 1223:        │     WAIT 'Dato Erróneo ...!'
 1224:        └─ENDIF
 1225:        ┌─IF opc_sn = 1
 1226:        │     STOR .T. TO vcod_real
 1227:        └─ENDIF
```

```
1228:
1229:
1230:          *
*************************************************************
1231:          *        *
1232:          *        * _RAD13E9J9            Car.catego VALID
1233:          *        *
1234:          *        * Function Origin:
1235:          *        *
1236:          *        * From Platform:        Windows
1237:          *        * From Screen:          DGP_CARG,
Record Number:   35
1238:          *        * Variable:             Car.catego
1239:          *        * Called By:            VALID Clause
1240:          *        * Object Type:          Field
1241:          *        * Snippet Number:       10
1242:          *        *
1243:          *
*************************************************************
1244:          *
1245:
*!*****************************************************************
**************
1246:   *!
1247:   *!        Function: _RAD13E9J9
1248:   *!
1249:   *!       Called by: DGP_CARG.SPR
1250:   *!
1251:
*!*****************************************************************
**************
1252:      FUNCTION _rad13e9j9     &&  Car.catego VALID
1253:          #REGION 1
1254:          IF  !EMPTY(car.catego)
1255:              reg_act1 = RECNO()
1256:              SEEK car.catego
1257:              STOR FOUND() TO vrf
1258:              GO reg_act1
1259:          ENDIF
1260:  <=======RETURN vrf
1261:
1262:
1263:          *
*************************************************************
1264:          *        *
1265:          *        * _RAD13E9LW            Car.catego ERROR
1266:          *        *
1267:          *        * Function Origin:
1268:          *        *
1269:          *        * From Platform:        Windows
1270:          *        * From Screen:          DGP_CARG,
Record Number:   35
1271:          *        * Variable:             Car.catego
1272:          *        * Called By:            ERROR Clause
1273:          *        * Object Type:          Field
1274:          *        * Snippet Number:       11
1275:          *        *
1276:          *
*************************************************************
```

```
1277:          *
1278:
*!*****************************************************************
**************
1279:   *!
1280:   *!        Function: _RAD13E9LW
1281:   *!
1282:   *!       Called by: DGP_CARG.SPR
1283:   *!
1284:
*!*****************************************************************
**************
1285:      FUNCTION _rad13e9lw      &&  Car.catego ERROR
1286:          #REGION 1
1287:          'CODIGO DE PUESTO NO EXISTE'
1288:   *:  EOF: DGP_CARG.acl
```


Varios Archivo Nómina Rol de Pagos Reportes Pmo.Emp Costos E.D.36 Salir
ACTUALIZACION
PARAMETROS DEL CODIGO DE TRABAJO
RUBRO
VALOR
SALARIO MINIMO VITAL
95,000
REVISAR
MODIFICAR
SALIR
<Pg.Up> Anterior
<Pg.Dn> Siguiente
<Home> Primero
<End> Ultimo
Exclusivo
INS
NUM


Varios Archivo Nómina Rol de Pagos ReporTes Ptmo.Emp Costos E.D.36 Salir
Examinar
AUMENTOS
NUMERO Valor
95,000.00
PASAJE 250.00
BON. COSTO DE VIDA 145,000.00
BON. COMPLEMENTARI 180,000.00
%IESS PERSONAL 9.35
%IESS PATRONAL 11.15
%S.E.C.A.P 0.50
%I.E.C.E. 0.50
DECIMO CUARTO SUELO 190,000.00
DECIMO QUINTO SUELO 50,000.00
9:48
Clock: 5/12
Ct
Reg: 1/16
Exclusivo
INS
NUM

```
    1:
*:*************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\AP_PRMCT.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 15/09/1995 at 14:49:46
    8:  *:
    9:  *:  Procs & Fncts: ANTERIOR
   10:  *:                : SIGUIENTE
   11:  *:                : PRIMERO
   12:  *:                : ULTIMO
   13:  *:                : _R3N0VS8FB()
   14:  *:                : _R3N0VS8RL()
   15:  *:                : MODIFICAR
   16:  *:                : VAL_CAMPO()
   17:  *:                : REFRESH()
   18:  *:
   19:  *:          Set by: EMPRESA.MPR
   20:  *:
   21:  *:           Calls: ANTERIOR           (procedure in
AP_PRMCT.SPR)
   22:  *:                : SIGUIENTE          (procedure in
AP_PRMCT.SPR)
   23:  *:                : PRIMERO            (procedure in
AP_PRMCT.SPR)
   24:  *:                : ULTIMO             (procedure in
AP_PRMCT.SPR)
   25:  *:                : _R3N0VS8FB()       (function in
AP_PRMCT.SPR)
   26:  *:                : _R3N0VS8RL()       (function in
AP_PRMCT.SPR)
   27:  *:
   28:  *:       Documented 19:14:34
FoxDoc version 3.00a
   29:
*:*************************************************************
**************
   30:      *
************************************************************
   31:      *        *
   32:      *        * 09/15/1995            ACT_PCT.SPR
 14:49:44
   33:      *        *
   34:      *
************************************************************
   35:      *        *
   36:      *        * Author's Name
   37:      *        *
   38:      *        * Copyright (c) 1995 Company Name
   39:      *        * Address
   40:      *        * City,     Zip
   41:      *        *
   42:      *        * Description:
   43:      *        * This program was automatically generated by
GENSCRN.
```

```
   44:      *        *
   45:      *
************************************************************
   46:
   47:
   48:      *
************************************************************
   49:      *        *
   50:      *        *          ACT_PCT/Windows Setup Code - SECTION
1
   51:      *        *
   52:      *
************************************************************
   53:      *
   54:
   55:      #REGION 1
   56:      **********
   57:      REGIONAL mcampo,mformato,opc_scr
   58:      STOR '' TO mcampo,opc_scr,mformato
   59:      ON KEY LABEL pgup DO anterior
   60:      ON KEY LABEL pgdn DO siguiente
   61:      ON KEY LABEL home DO primero
   62:      ON KEY LABEL END  DO ultimo
   63:
   64:      **********
   65:
   66:      #REGION 0
   67:      REGIONAL m.currarea, m.talkstat, m.compstat
   68:
   69:     ┌IF SET("TALK") = "ON"
   70:     │   SET TALK OFF
   71:     │   m.talkstat = "ON"
   72:     ├ELSE
   73:     │   m.talkstat = "OFF"
   74:     └ENDIF
   75:      m.compstat = SET("COMPATIBLE")
   76:      SET COMPATIBLE FOXPLUS
   77:
   78:      m.rborder = SET("READBORDER")
   79:      SET readborder ON
   80:
   81:      m.currarea = SELECT()
   82:
   83:
   84:      *
************************************************************
   85:      *        *
   86:      *        *       ACT_PCT/Windows Databases, Indexes,
Relations
   87:      *        *
   88:      *
************************************************************
   89:      *
   90:
   91:     ┌IF USED("lct")
   92:     │   SELECT lct
   93:     │   SET ORDER TO 0
   94:     ├ELSE
   95:     │   SELECT 0
```

```
   96:     |    USE (LOCFILE("dbfs\ley_labo.dbf","DBF","Where is
ley_labo?"));
   97:     |        AGAIN ALIAS lct ;
   98:     |        ORDER 0
   99:     |ENDIF
  100:
  101:     |IF USED("ct")
  102:     |    SELECT ct
  103:     |    SET ORDER TO 0
  104:     |ELSE
  105:     |    SELECT 0
  106:     |    USE (LOCFILE("dbfs\codtra.par","DBF","Where is
codtra?"));
  107:     |        AGAIN ALIAS ct ;
  108:     |        ORDER 0
  109:     |ENDIF
  110:
  111:
  112:
  113:     *
************************************************************
  114:     *        *
  115:     *        *                Windows Window definitions
  116:     *        *
  117:     *
************************************************************
  118:     *
  119:
  120:     |IF NOT WEXIST("act_pct") ;
  121:     |        OR UPPER(WTITLE("ACT_PCT")) == "ACT_PCT.PJX" ;
  122:     |        OR UPPER(WTITLE("ACT_PCT")) == "ACT_PCT.SCX" ;
  123:     |        OR UPPER(WTITLE("ACT_PCT")) == "ACT_PCT.MNX" ;
  124:     |        OR UPPER(WTITLE("ACT_PCT")) == "ACT_PCT.PRG" ;
  125:     |        OR UPPER(WTITLE("ACT_PCT")) == "ACT_PCT.FRX" ;
  126:     |        OR UPPER(WTITLE("ACT_PCT")) == "ACT_PCT.QPR"
  127:     |    DEFINE WINDOW act_pct ;
  128:     |        AT  0.000, 0.000  ;
  129:     |        SIZE 15.154,72.600 ;
  130:     |        TITLE "ACTUALIZACION" ;
  131:     |        FONT "MS Sans Serif", 8 ;
  132:     |        FLOAT ;
  133:     |        NOCLOSE ;
  134:     |        MINIMIZE ;
  135:     |        SYSTEM
  136:     |    MOVE WINDOW act_pct CENTER
  137:     |ENDIF
  138:
  139:
  140:     *
************************************************************
  141:     *        *
  142:     *        *         ACT_PCT/Windows Setup Code - SECTION
2
  143:     *        *
  144:     *
************************************************************
  145:     *
  146:
  147:     #REGION 1
```

```
148:      **********
149:      SELE ct
150:      GO TOP
151:      STOR ct.campo TO mcampo
152:      STOR ct.formato TO mformato
153:      SHOW GETS ENABLE
154:      SELE lct
155:
156:      *
************************************************************
157:      *        *
158:      *        *                ACT_PCT/Windows Screen Layout
159:      *        *
160:      *
************************************************************
161:      *
162:
163:      #REGION 1
164:     ┌IF WVISIBLE("act_pct")
165:     │   ACTIVATE WINDOW act_pct SAME
166:     ├ELSE
167:     │   ACTIVATE WINDOW act_pct NOSHOW
168:     └ENDIF
169:      @ 6.769,13.200 SAY ct.detalle ;
170:         SIZE 1.000,25.667 ;
171:         FONT "MS Serif", 8 ;
172:         STYLE "BI"
173:      @ 6.769,50.600 SAY &mcampo ;
174:         SIZE 1.000,12.000 ;
175:         FONT "MS Sans Serif", 8 ;
176:         STYLE "B" ;
177:         PICTURE "&MFORMATO"
178:      @ 9.692,14.400 GET opc_scr ;
179:         PICTURE "@*HN |<REVISAR;|<MODIFICAR;|<SALIR" ;
180:         SIZE 1.385,12.500,0.333 ;
181:         DEFAULT 1 ;
182:         FONT "MS Sans Serif", 8 ;
183:         STYLE "B" ;
184:         VALID _r3n0vs8fb()
185:      @ 12.154,19.000 SAY "<Pg.Up.>"  ;
186:         FONT "MS Serif", 8 ;
187:         STYLE "BT"
188:      @ 13.000,19.400 SAY "Anterior"  ;
189:         FONT "MS Serif", 8 ;
190:         STYLE "T"
191:      @ 12.231,30.200 SAY "<Pg.Dn.> <Home>  <End>"  ;
192:         FONT "MS Serif", 8 ;
193:         STYLE "BT"
194:      @ 13.077,30.600 SAY "Siguiente"  ;
195:         FONT "MS Serif", 8 ;
196:         STYLE "T"
197:      @ 13.077,41.600 SAY "Primero"  ;
198:         FONT "MS Serif", 8 ;
199:         STYLE "T"
200:      @ 13.077,51.200 SAY "Ultimo"  ;
201:         FONT "MS Serif", 8 ;
202:         STYLE "T"
203:      @ 11.846,17.000 TO 14.308,58.400 ;
204:         PEN 1, 8 ;
```

```
205:        STYLE "16"
206:     @ 0.615,21.400 SAY "PARAMETROS DEL " + CHR(13) + ;
207:        "CODIGO DE TRABAJO"  ;
208:        SIZE 2.000,24.143, 0.000 ;
209:        PICTURE "@!" ;
210:        FONT "MS Serif", 10 ;
211:        STYLE "BIT" ;
212:        COLOR RGB(0,0,0,,,,)
213:     @ 0.231,17.000 TO 3.539,58.000 ;
214:        PEN 1, 8 ;
215:        STYLE "16"
216:     @ 4.615,8.600 TO 8.384,68.400 ;
217:        PEN 1, 8
218:     @ 4.769,19.200 SAY "RUBRO"  ;
219:        FONT "MS Sans Serif", 8 ;
220:        STYLE "BIT"
221:     @ 4.769,57.200 SAY "VALOR"  ;
222:        FONT "MS Sans Serif", 8 ;
223:        STYLE "BIT"
224:     @ 9.231,12.600 TO 11.231,61.200 ;
225:        PEN 1, 8 ;
226:        STYLE "16"
227:     @ 0.538,17.800 TO 3.230,57.200 ;
228:        PEN 1, 8 ;
229:        STYLE "16"
230:
231:     IF NOT WVISIBLE("act_pct")
232:        ACTIVATE WINDOW act_pct
233:     ENDIF
234:
235:     READ CYCLE ;
236:        SHOW _r3n0vs8rl()
237:
238:     RELEASE WINDOW act_pct
239:
240:     *
************************************************************
241:     *        *
242:     *        *                  Windows Closing Databases
243:     *        *
244:     *
************************************************************
245:     *
246:
247:     IF USED("lct")
248:        SELECT lct
249:        USE
250:     ENDIF
251:
252:     IF USED("ct")
253:        SELECT ct
254:        USE
255:     ENDIF
256:
257:     SELECT (m.currarea)
258:
259:
260:     #REGION 0
261:
```

```
  262:      SET readborder &rborder
  263:
  264:     IF m.talkstat = "ON"
  265:        SET TALK ON
  266:     ENDIF
  267:     IF m.compstat = "ON"
  268:        SET COMPATIBLE ON
  269:     ENDIF
  270:
  271:
  272:      *
************************************************************
  273:      *        *
  274:      *        *                ACT_PCT/Windows Cleanup Code
  275:      *        *
  276:      *
************************************************************
  277:      *
  278:
  279:      #REGION 1
  280:      ********************
  281:
  282:
  283:      *
************************************************************
  284:      *        *
  285:      *        *  ACT_PCT/Windows Supporting Procedures and
Functions
  286:      *        *
  287:      *
************************************************************
  288:      *
  289:
  290:      #REGION 1
  291:
*!*************************************************************************
**************
  292:   *!
  293:   *!       Procedure: SIGUIENTE
  294:   *!
  295:   *!       Called by: AP_PRMCT.SPR
  296:   *!                : AP_PRMCC.SPR
  297:   *!                : AP_PRMVR.SPR
  298:   *!                : SECC()              (function in
PERSONAL.PRG)
  299:   *!
  300:   *!           Calls: REFRESH()          (function in
AP_PRMCT.SPR)
  301:   *!                : DESPLEGA           (procedure in
PERSONAL.PRG)
  302:   *!
  303:
*!*************************************************************************
**************
  304:      PROCEDURE siguiente
  305:          ********************
  306:          SELE ct
  307:         IF !EOF()
  308:            SKIP
```

```
 309:          |   IF EOF()
 310:          |      GOTO BOTTOM
 311:          |   ENDIF
 312:          |ELSE
 313:          |   GOTO BOTTOM
 314:          |   WAIT WIND 'FIN DE ARCHIVO'
 315:          |ENDIF
 316:          DO REFRESH
 317: <========RETURN
 318:          ******************
 319:
*!*************************************************************
**************
  320:   *!
  321:   *!       Procedure: ANTERIOR
  322:   *!
  323:   *!       Called by: AP_PRMCT.SPR
  324:   *!                : AP_PRMCC.SPR
  325:   *!                : AP_PRMVR.SPR
  326:   *!                : SECC()              (function in
PERSONAL.PRG)
  327:   *!
  328:   *!           Calls: REFRESH()          (function in
AP_PRMCT.SPR)
  329:   *!                : DESPLEGA           (procedure in
PERSONAL.PRG)
  330:   *!
  331:
*!*************************************************************
**************
  332:      PROCEDURE anterior
  333:          ******************
  334:          SELE ct
  335:          IF !BOF()
  336:             SKIP -1
  337:             IF BOF()
  338:                GOTO TOP
  339:             ENDIF
  340:          ELSE
  341:             GOTO TOP
  342:             WAIT WIND 'INICIO DE ARCHIVO'
  343:          ENDIF
  344:          DO REFRESH
  345: <========RETURN
  346:          ******************
  347:
*!*************************************************************
**************
  348:   *!
  349:   *!       Procedure: PRIMERO
  350:   *!
  351:   *!       Called by: AP_PRMCT.SPR
  352:   *!                : AP_PRMCC.SPR
  353:   *!                : AP_PRMVR.SPR
  354:   *!
  355:   *!           Calls: REFRESH()          (function in
AP_PRMCT.SPR)
  356:   *!
  357:
```

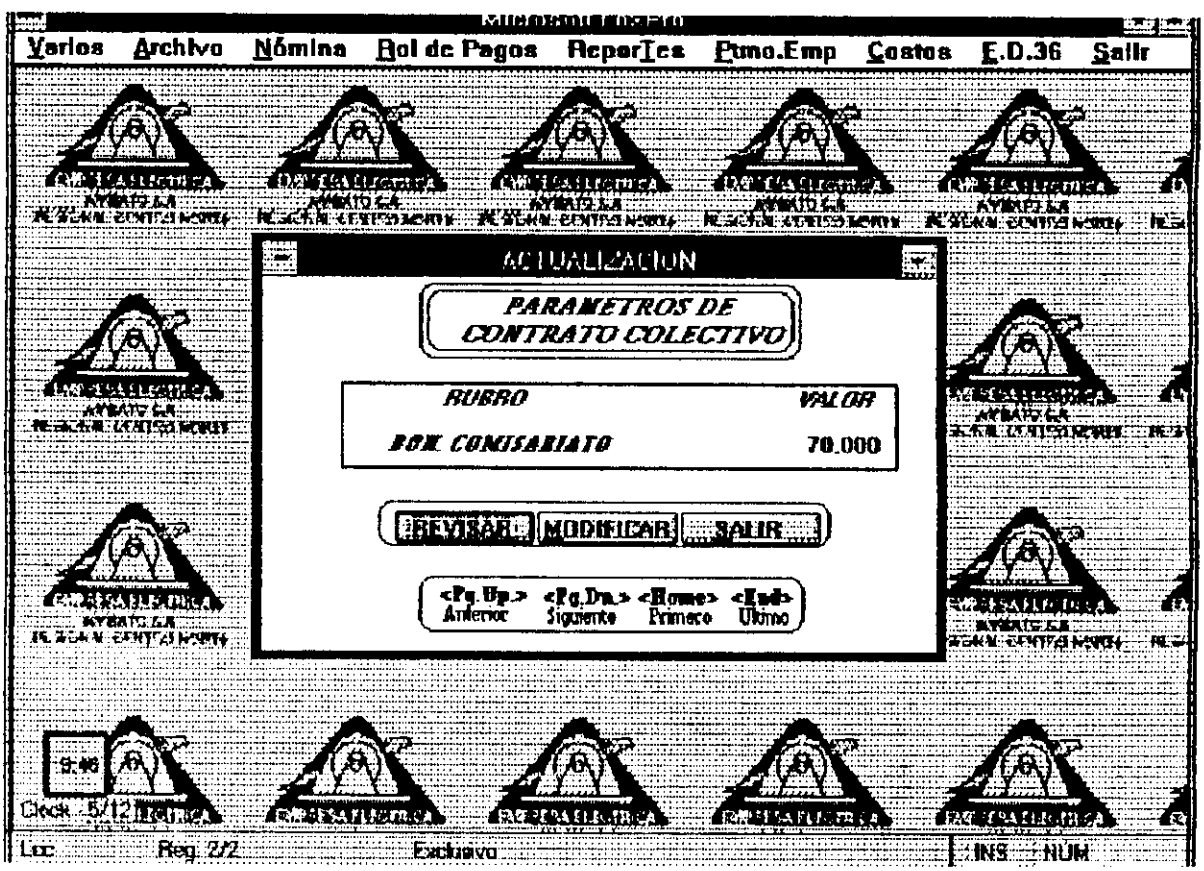
```
*!*************************************************************************
**************
  358:     PROCEDURE primero
  359:          ******************
  360:          SELE ct
  361:          GOTO TOP
  362:          WAIT WIND 'INICIO DE ARCHIVO'
  363:          DO REFRESH
  364: <========RETURN
  365:          ******************
  366:
*!*************************************************************************
**************
  367:   *!
  368:   *!       Procedure: ULTIMO
  369:   *!
  370:   *!       Called by: AP_PRMCT.SPR
  371:   *!                : AP_PRMCC.SPR
  372:   *!                : AP_PRMVR.SPR
  373:   *!
  374:   *!           Calls: REFRESH()          (function in
AP_PRMCT.SPR)
  375:   *!
  376:
*!*************************************************************************
**************
  377:      PROCEDURE ultimo
  378:          ******************
  379:          SELE ct
  380:          GOTO BOTTOM
  381:          WAIT WIND 'FIN DE ARCHIVO'
  382:          DO REFRESH
  383: <========RETURN
  384:
  385:
  386:          ******************
  387:
*!*************************************************************************
**************
  388:   *!
  389:   *!       Procedure: MODIFICAR
  390:   *!
  391:
*!*************************************************************************
**************
  392:      PROCEDURE modificar
  393:          ******************
  394:          A = ct.campo
  395:          @ 6.769,49.200 GET &mcampo ;
  396:             SIZE 1.000,14.400 ;
  397:             PICT &mformato;
  398:             FONT "MS Sans Serif", 8 ;
  399:             STYLE "B"
  400:          READ
  401: <========RETURN
  402:          ******************
  403:
*!*************************************************************************
**************
```

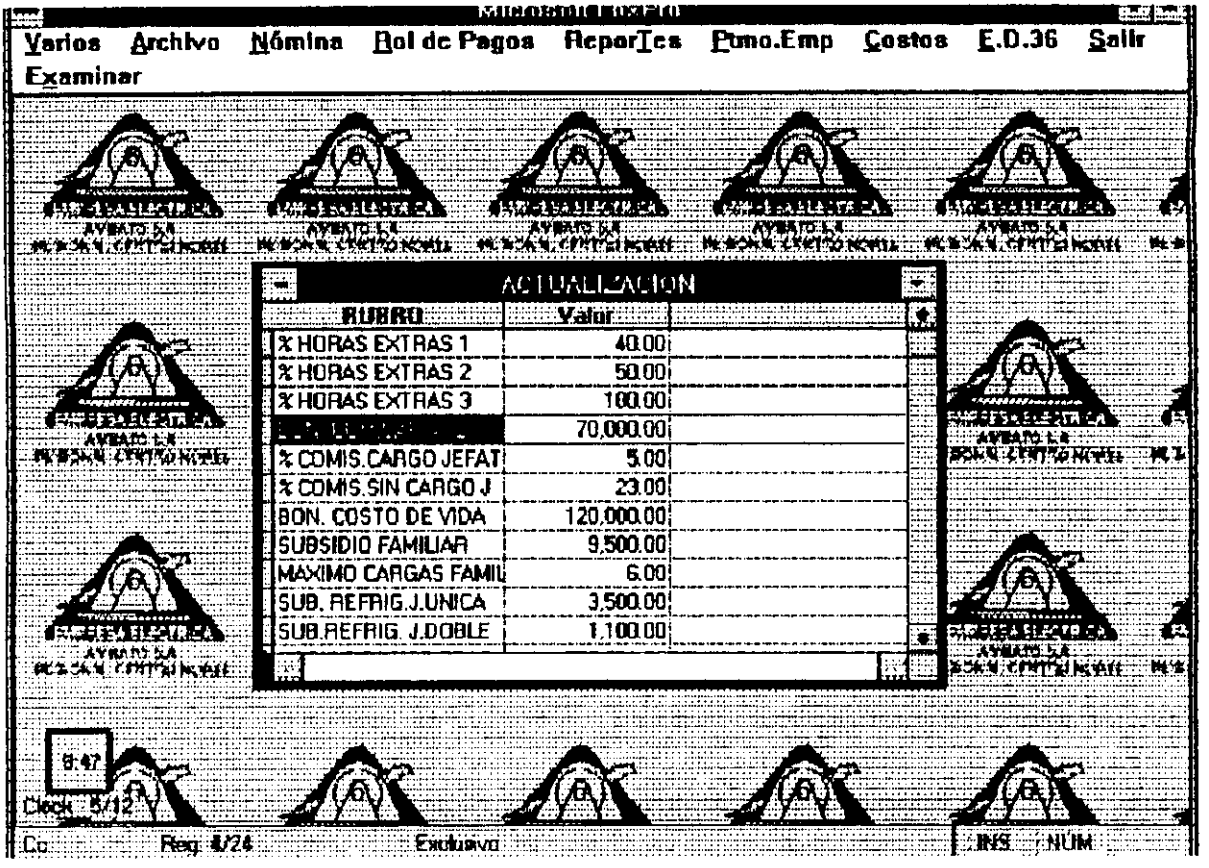
```
  404:  *!
  405:  *!       Procedure: REVISAR
  406:  *!
  407:  *!           Calls: MSCATTER           (procedure in
AACT_SST.SPR)
  408:  *!                : DEAC_KEYS          (procedure in
AACT_SST.SPR)
  409:  *!                : ACT_KEYS           (procedure in
AACT_SST.SPR)
  410:  *!                : VAL_CAMPO()        (function in
AP_PRMCT.SPR)
  411:  *!                : REFRESH()          (function in
AP_PRMCT.SPR)
  412:  *!                : MBROW              (procedure in
MO_PDLAB.SPR)
  413:  *!
  414:
*!*************************************************************
**************
  415:      PROCEDURE revisar
  416:          *****************
  417:          SELE ct
  418:          BROW FIELDS
ct.detalle:H='RUBRO',valor=val_campo():p='999,999,999.99';
  419:             NOMODIFY
  420:          DO REFRESH
  421: <=======RETURN
  422:          ******************
  423:
*!*************************************************************
**************
  424:  *!
  425:  *!        Function: VAL_CAMPO
  426:  *!
  427:  *!       Called by: REVISAR            (procedure in
AACT_SST.SPR)
  428:  *!
  429:
*!*************************************************************
**************
  430:      FUNCTION val_campo
  431:          ******************
  432:          STOR ct.campo TO mcampo
  433:          SELE lct
  434:          STOR &mcampo TO valor
  435:          SELE ct
  436: <=======RETURN valor
  437:
  438:          ***************
  439:
*!*************************************************************
**************
  440:  *!
  441:  *!       Procedure: SALIR
  442:  *!
  443:  *!       Called by: _RAA1732TG()       (function in
AT_ESCLS.SPR)
  444:  *!                : DA_ASISG.PRG
  445:  *!                : DE_EGRQ1.PRG
```

```
  446:  *!                  : DE_EGRQ2.PRG
  447:  *!                  : DE_EGRMN.PRG
  448:  *!                  : DE_IEJUB.PRG
  449:  *!
  450:  *!          Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
  451:  *!
  452:
*!*********************************************************************
**************
  453:      PROCEDURE salir
  454:          ***************
  455:          ON KEY LABEL pgup
  456:          ON KEY LABEL pgdn
  457:          ON KEY LABEL END
  458:          ON KEY LABEL home
  459:          CLEAR READ
  460: <=======RETURN
  461:          ********************
  462:
*!*********************************************************************
**************
  463:  *!
  464:  *!       Function: REFRESH
  465:  *!
  466:  *!      Called by: REVISAR            (procedure in
AACT_SST.SPR)
  467:  *!                : ANTERIOR          (procedure in
AP_PRMCT.SPR)
  468:  *!                : SIGUIENTE         (procedure in
AP_PRMCT.SPR)
  469:  *!                : PRIMERO           (procedure in
AP_PRMCT.SPR)
  470:  *!                : ULTIMO            (procedure in
AP_PRMCT.SPR)
  471:  *!
  472:
*!*********************************************************************
**************
  473:      PROCEDURE REFRESH
  474:          ********************
  475:          SHOW GETS ENABLE
  476:          SELE ct
  477:          STOR ct.campo TO mcampo
  478:          STOR ct.formato TO mformato
  479:          SELE lct
  480:          @ 6.769,49.200 SAY &mcampo :
  481:             SIZE 1.000,14.400 :
  482:             PICT &mformato;
  483:             FONT "MS Sans Serif", 8;
  484:             STYLE "B"
  485: <=======RETURN
  486:
  487:          *
 ************************************************************
  488:          *        *
  489:          *        * _R3N0VS8FB          OPC_SCR VALID
  490:          *        *
  491:          *        * Function Origin:
```

```
  492:          *        *
  493:          *        * From Platform:        Windows
  494:          *        * From Screen:          ACT_PCT,      Record
Number:     6
  495:          *        * Variable:             OPC_SCR
  496:          *        * Called By:            VALID Clause
  497:          *        * Object Type:          Push Button
  498:          *        * Snippet Number:       1
  499:          *        *
  500:          *
************************************************************
  501:          *
  502:
*!*************************************************************************
**************
  503:  *!
  504:  *!        Function: _R3N0VS8FB
  505:  *!
  506:  *!       Called by: AP_PRMCT.SPR
  507:  *!
  508:  *!           Calls: &OPC_SCR.PRG
  509:  *!
  510:
*!*************************************************************************
**************
  511:      FUNCTION _r3n0vs8fb     &&  OPC_SCR VALID
  512:          #REGION 1
  513:          DO &opc_scr
  514:
  515:          *
************************************************************
  516:          *        *
  517:          *        * _R3N0VS8RL            Read Level Show
  518:          *        *
  519:          *        * Function Origin:
  520:          *        *
  521:          *        *
  522:          *        * From Platform:        Windows
  523:          *        * From Screen:          ACT_PCT
  524:          *        * Called By:            READ Statement
  525:          *        * Snippet Number:       2
  526:          *        *
  527:          *
************************************************************
  528:          *
  529:
*!*************************************************************************
**************
  530:  *!
  531:  *!        Function: _R3N0VS8RL
  532:  *!
  533:  *!       Called by: AP_PRMCT.SPR
  534:  *!
  535:
*!*************************************************************************
**************
  536:      FUNCTION _r3n0vs8rl     && Read Level Show
  537:          PRIVATE currwind
  538:          STORE WOUTPUT() TO currwind
```

```
539:          *
540:          * Show Code from screen: ACT_PCT
541:          *
542:          #REGION 1
543:         ┌─IF SYS(2016) = "ACT_PCT" OR SYS(2016) = "*"
544:         │     ACTIVATE WINDOW act_pct SAME
545:         │     @ 6.769,13.200 SAY ct.detalle ;
546:         │        SIZE 1.000,25.667, 0.000 ;
547:         │        FONT "MS Serif", 8 ;
548:         │        STYLE "BI"
549:         └─ENDIF
550:         ┌─IF NOT EMPTY(currwind)
551:         │     ACTIVATE WINDOW (currwind) SAME
552:         └─ENDIF
553:   *:  EOF: AP_PRMCT.acl
```


Varios Archivo Nómina Rol de Pagos ReporTes Pmo.Emp Costos E.D.36 Salir
ACTUALIZACION
PARAMETROS DE CONTRATO COLECTIVO
RUBRO
VALOR
BON. COMISARIATO
70.000
REVISAR
MODIFICAR
SALIR
<Pg.Up.> Anterior
<Pg.Dn.> Siguiente
<Home> Primero
<End> Ultimo
Loc
Reg 2/2
Exclusivo
INS
NUM


Varios Archivo Nómina Rol de Pagos Reportes Func.Emp Costos E.D.36 Salir
Examinar
ACTUALIZACION
RUBRO
Valor
% HORAS EXTRAS 1
40.00
% HORAS EXTRAS 2
50.00
% HORAS EXTRAS 3
100.00
70,000.00
% COMIS.CARGO JEFAT
5.00
% COMIS.SIN CARGO J
23.00
BON. COSTO DE VIDA
120,000.00
SUBSIDIO FAMILIAR
9,500.00
MAXIMO CARGAS FAMIL
6.00
SUB. REFRIG.J.UNICA
3,500.00
SUB.REFRIG. J.DOBLE
1,100.00
8:47
Clock 5/12
Reg: 4/24
Exclusivo
INS
NUM

```
    1:
*:*************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\AP_PRMCC.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 15/09/1995 at 14:47:56
    8:  *:
    9:  *:  Procs & Fncts: _R3N0VPW82()
   10:  *:                : _R3N0VPWJX()
   11:  *:
   12:  *:          Set by: EMPRESA.MPR
   13:  *:
   14:  *:           Calls: ANTERIOR           (procedure in
AP_PRMCT.SPR)
   15:  *:                : SIGUIENTE          (procedure in
AP_PRMCT.SPR)
   16:  *:                : PRIMERO            (procedure in
AP_PRMCT.SPR)
   17:  *:                : ULTIMO             (procedure in
AP_PRMCT.SPR)
   18:  *:                : _R3N0VPW82()       (function in
AP_PRMCC.SPR)
   19:  *:                : _R3N0VPWJX()       (function in
AP_PRMCC.SPR)
   20:  *:
   21:  *:       Documented 19:14:35
FoxDoc version 3.00a
   22:
*:*************************************************************
**************
   23:      *
************************************************************
   24:      *        *
   25:      *        * 09/15/1995            ACT_PCC.SPR
 14:47:55
   26:      *        *
   27:      *
************************************************************
   28:      *        *
   29:      *        * Author's Name
   30:      *        *
   31:      *        * Copyright (c) 1995 Company Name
   32:      *        * Address
   33:      *        * City,     Zip
   34:      *        *
   35:      *        * Description:
   36:      *        * This program was automatically generated by
GENSCRN.
   37:      *        *
   38:      *
************************************************************
   39:
   40:
   41:      *
************************************************************
```

```
   42:      *        *
   43:      *        *          ACT_PCC/Windows Setup Code - SECTION
1
   44:      *        *
   45:      *
**********************************************************
   46:      *
   47:
   48:      #REGION 1
   49:      **********
   50:      REGIONAL mcampo,mformato,opc_scr
   51:      STOR '' TO mcampo,opc_scr,mformato
   52:      ON KEY LABEL pgup DO anterior
   53:      ON KEY LABEL pgdn DO siguiente
   54:      ON KEY LABEL home DO primero
   55:      ON KEY LABEL END  DO ultimo
   56:
   57:      **********
   58:
   59:      #REGION 0
   60:      REGIONAL m.currarea, m.talkstat, m.compstat
   61:
   62:    ┌─IF SET("TALK") = "ON"
   63:    │    SET TALK OFF
   64:    │    m.talkstat = "ON"
   65:    ├─ELSE
   66:    │    m.talkstat = "OFF"
   67:    └─ENDIF
   68:      m.compstat = SET("COMPATIBLE")
   69:      SET COMPATIBLE FOXPLUS
   70:
   71:      m.rborder = SET("READBORDER")
   72:      SET readborder ON
   73:
   74:      m.currarea = SELECT()
   75:
   76:
   77:      *
**********************************************************
   78:      *        *
   79:      *        *      ACT_PCC/Windows Databases, Indexes,
Relations
   80:      *        *
   81:      *
**********************************************************
   82:      *
   83:
   84:    ┌─IF USED("lcc")
   85:    │    SELECT lcc
   86:    │    SET ORDER TO 0
   87:    ├─ELSE
   88:    │    SELECT 0
   89:    │    USE (LOCFILE("dbfs|ley_cont.dbf","DBF","Where is
ley_cont?"));
   90:    │         AGAIN ALIAS lcc ;
   91:    │         ORDER 0
   92:    └─ENDIF
   93:
   94:    ┌─IF USED("cc")
```

```
   95:       SELECT cc
   96:       SET ORDER TO 0
   97:    ELSE
   98:       SELECT 0
   99:       USE (LOCFILE("dbfs\concol.par","DBF","Where is
concol?"));
  100:          AGAIN ALIAS cc ;
  101:          ORDER 0
  102:    ENDIF
  103:
  104:    SELECT lcc
  105:
  106:
  107:    *
************************************************************
  108:    *        *
  109:    *        *                Windows Window definitions
  110:    *        *
  111:    *
************************************************************
  112:    *
  113:
  114:    IF NOT WEXIST("act_pcc") :
  115:          OR UPPER(WTITLE("ACT_PCC")) == "ACT_PCC.PJX" ;
  116:          OR UPPER(WTITLE("ACT_PCC")) == "ACT_PCC.SCX" ;
  117:          OR UPPER(WTITLE("ACT_PCC")) == "ACT_PCC.MNX" ;
  118:          OR UPPER(WTITLE("ACT_PCC")) == "ACT_PCC.PRG" ;
  119:          OR UPPER(WTITLE("ACT_PCC")) == "ACT_PCC.FRX" ;
  120:          OR UPPER(WTITLE("ACT_PCC")) == "ACT_PCC.QPR"
  121:       DEFINE WINDOW act_pcc :
  122:          AT  0.000, 0.000  ;
  123:          SIZE 16.231,72.400 ;
  124:          TITLE "ACTUALIZACION" ;
  125:          FONT "MS Sans Serif", 8 ;
  126:          FLOAT ;
  127:          NOCLOSE ;
  128:          MINIMIZE ;
  129:          SYSTEM
  130:       MOVE WINDOW act_pcc CENTER
  131:    ENDIF
  132:
  133:
  134:    *
************************************************************
  135:    *        *
  136:    *        *          ACT_PCC/Windows Setup Code - SECTION
2
  137:    *        *
  138:    *
************************************************************
  139:    *
  140:
  141:    #REGION 1
  142:    **********
  143:    SELE cc
  144:    GO TOP
  145:    STOR cc.campo TO mcampo
  146:    STOR cc.formato TO mformato
  147:    SHOW GETS ENABLE
```

```
148:      SELE lcc
149:
150:      *
**************************************************************
151:      *        *
152:      *        *                ACT_PCC/Windows Screen Layout
153:      *        *
154:      *
**************************************************************
155:      *
156:
157:      #REGION 1
158:     ┌IF WVISIBLE("act_pcc")
159:     │   ACTIVATE WINDOW act_pcc SAME
160:     ├ELSE
161:     │   ACTIVATE WINDOW act_pcc NOSHOW
162:     └ENDIF
163:      @ 6.769,13.200 SAY cc.detalle ;
164:         SIZE 1.000,25.667 ;
165:         FONT "MS Serif", 8 ;
166:         STYLE "BI"
167:      @ 6.769,50.600 SAY &mcampo ;
168:         SIZE 1.000,12.000 ;
169:         FONT "MS Sans Serif", 8 ;
170:         STYLE "B" ;
171:         PICTURE "&MFORMATO"
172:      @ 10.231,14.400 GET opc_scr ;
173:         PICTURE "@*HN \<REVISAR;\<MODIFICAR;\<SALIR" ;
174:         SIZE 1.385,12.500,0.333 ;
175:         DEFAULT 1 ;
176:         FONT "MS Sans Serif", 8 ;
177:         STYLE "B" ;
178:         VALID _r3n0vpw82()
179:      @ 13.308,19.000 SAY "<Pg.Up.>"  ;
180:         FONT "MS Serif", 8 ;
181:         STYLE "BT"
182:      @ 14.154,19.400 SAY "Anterior"  ;
183:         FONT "MS Serif", 8 ;
184:         STYLE "T"
185:      @ 13.385,30.200 SAY "<Pg.Dn.> <Home>  <End>"  ;
186:         FONT "MS Serif", 8 ;
187:         STYLE "BT"
188:      @ 14.231,30.600 SAY "Siguiente"  ;
189:         FONT "MS Serif", 8 ;
190:         STYLE "T"
191:      @ 14.231,41.600 SAY "Primero"  ;
192:         FONT "MS Serif", 8 ;
193:         STYLE "T"
194:      @ 14.231,51.200 SAY "Ultimo"  ;
195:         FONT "MS Serif", 8 ;
196:         STYLE "T"
197:      @ 13.000,17.000 TO 15.462,58.400 ;
198:         PEN 1, 8 ;
199:         STYLE "16"
200:      @ 0.615,19.600 SAY "PARAMETROS DE " + CHR(13) + ;
201:         "CONTRATO COLECTIVO"  ;
202:         SIZE 2.000,26.571, 0.000 ;
203:         PICTURE "@I" ;
204:         FONT "MS Serif", 10 ;
```

```
205:          STYLE "BIT" ;
206:          COLOR RGB(0,0,0,,,,)
207:       @ 0.231,17.000 TO 3.539,58.000 ;
208:          PEN 1, 8 ;
209:          STYLE "16"
210:       @ 4.615,8.600 TO 8.384,68.400 ;
211:          PEN 1, 8
212:       @ 4.769,19.200 SAY "RUBRO"  ;
213:          FONT "MS Sans Serif", 8 ;
214:          STYLE "BIT"
215:       @ 4.769,57.200 SAY "VALOR"  ;
216:          FONT "MS Sans Serif", 8 ;
217:          STYLE "BIT"
218:       @ 9.769,12.600 TO 11.769,61.200 ;
219:          PEN 1, 8 ;
220:          STYLE "16"
221:       @ 0.538,17.800 TO 3.230,57.200 ;
222:          PEN 1, 8 ;
223:          STYLE "16"
224:
225:    ┌─IF NOT WVISIBLE("act_pcc")
226:    │     ACTIVATE WINDOW act_pcc
227:    └─ENDIF
228:
229:       READ CYCLE ;
230:          SHOW _r3n0vpwjx()
231:
232:       RELEASE WINDOW act_pcc
233:
234:       *
************************************************************
235:       *        *
236:       *        *                  Windows Closing Databases
237:       *        *
238:       *
************************************************************
239:       *
240:
241:    ┌─IF USED("lcc")
242:    │     SELECT lcc
243:    │     USE
244:    └─ENDIF
245:
246:    ┌─IF USED("cc")
247:    │     SELECT cc
248:    │     USE
249:    └─ENDIF
250:
251:       SELECT (m.currarea)
252:
253:
254:       #REGION 0
255:
256:       SET readborder &rborder
257:
258:    ┌─IF m.talkstat = "ON"
259:    │     SET TALK ON
260:    └─ENDIF
261:    ┌─IF m.compstat = "ON"
```

```
  262:     |    SET COMPATIBLE ON
  263:     |ENDIF
  264:
  265:
  266:     *
************************************************************
  267:     *        *
  268:     *        *                ACT_PCC/Windows Cleanup Code
  269:     *        *
  270:     *
************************************************************
  271:     *
  272:
  273:     #REGION 1
  274:     ********************
  275:
  276:
  277:     *
************************************************************
  278:     *        *
  279:     *        *  ACT_PCC/Windows Supporting Procedures and
Functions
  280:     *        *
  281:     *
************************************************************
  282:     *
  283:
  284:     #REGION 1
  285:
*!*************************************************************************
**************
  286:  *!
  287:  *!       Procedure: SIGUIENTE
  288:  *!
  289:  *!       Called by: AP_PRMCT.SPR
  290:  *!                : AP_PRMCC.SPR
  291:  *!                : AP_PRMVR.SPR
  292:  *!                : SECC()              (function in
PERSONAL.PRG)
  293:  *!
  294:  *!           Calls: REFRESH()          (function in
AP_PRMCT.SPR)
  295:  *!                : DESPLEGA           (procedure in
PERSONAL.PRG)
  296:  *!
  297:
*!*************************************************************************
**************
  298:     PROCEDURE siguiente
  299:         ********************
  300:         SELE cc
  301:       |-IF !EOF()
  302:       |    SKIP
  303:       |  |-IF EOF()
  304:       |  |    GOTO BOTTOM
  305:       |  |ENDIF
  306:       |-ELSE
  307:       |    GOTO BOTTOM
  308:       |    WAIT WIND 'FIN DE ARCHIVO'
```

```
  309:          ENDIF
  310:          DO REFRESH
  311:  <=======RETURN
  312:          ******************
  313:
*!*****************************************************************
**************
  314:  *!
  315:  *!       Procedure: ANTERIOR
  316:  *!
  317:  *!       Called by: AP_PRMCT.SPR
  318:  *!                : AP_PRMCC.SPR
  319:  *!                : AP_PRMVR.SPR
  320:  *!                : SECC()              (function in
PERSONAL.PRG)
  321:  *!
  322:  *!           Calls: REFRESH()          (function in
AP_PRMCT.SPR)
  323:  *!                : DESPLEGA           (procedure in
PERSONAL.PRG)
  324:  *!
  325:
*!*****************************************************************
**************
  326:      PROCEDURE anterior
  327:          ******************
  328:          SELE cc
  329:          IF !BOF()
  330:             SKIP -1
  331:             IF BOF()
  332:                GOTO TOP
  333:             ENDIF
  334:          ELSE
  335:             GOTO TOP
  336:             WAIT WIND 'INICIO DE ARCHIVO'
  337:          ENDIF
  338:          DO REFRESH
  339:  <=======RETURN
  340:          ******************
  341:
*!*****************************************************************
**************
  342:  *!
  343:  *!       Procedure: PRIMERO
  344:  *!
  345:  *!       Called by: AP_PRMCT.SPR
  346:  *!                : AP_PRMCC.SPR
  347:  *!                : AP_PRMVR.SPR
  348:  *!
  349:  *!           Calls: REFRESH()          (function in
AP_PRMCT.SPR)
  350:  *!
  351:
*!*****************************************************************
**************
  352:      PROCEDURE primero
  353:          ******************
  354:          SELE cc
  355:          GOTO TOP
```

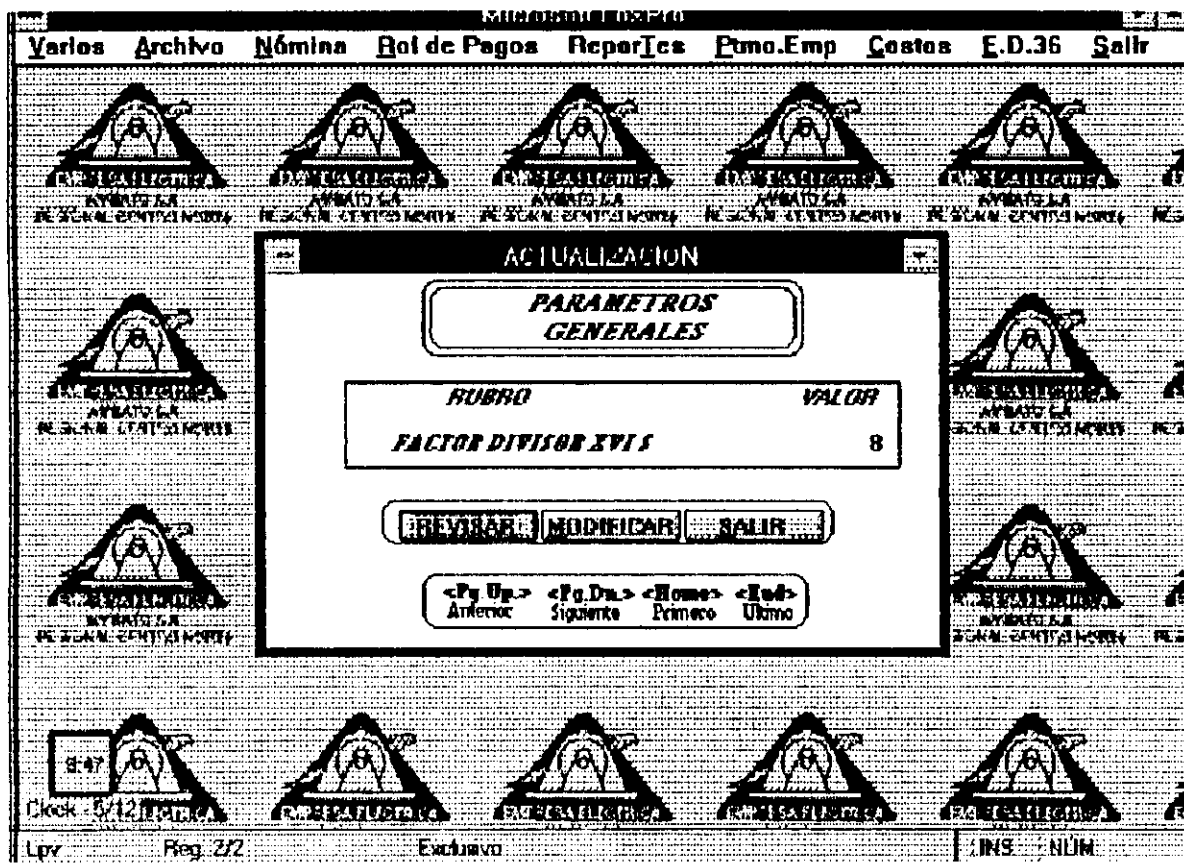
```
356:          WAIT WIND 'INICIO DE ARCHIVO'
357:          DO REFRESH
358: <=======RETURN
359:          ******************
360:
*!*************************************************************************
**************
361:  *!
362:  *!       Procedure: ULTIMO
363:  *!
364:  *!       Called by: AP_PRMCT.SPR
365:  *!                : AP_PRMCC.SPR
366:  *!                : AP_PRMVR.SPR
367:  *!
368:  *!           Calls: REFRESH()          (function in
AP_PRMCT.SPR)
369:  *!
370:
*!*************************************************************************
**************
371:      PROCEDURE ultimo
372:          ******************
373:          SELE cc
374:          GOTO BOTTOM
375:          WAIT WIND 'FIN DE ARCHIVO'
376:          DO REFRESH
377: <=======RETURN
378:
379:
380:          ********************
381:
*!*************************************************************************
**************
382:  *!
383:  *!       Procedure: MODIFICAR
384:  *!
385:
*!*************************************************************************
**************
386:      PROCEDURE modificar
387:          ********************
388:          A = cc.campo
389:          @ 6.769,49.200 GET &mcampo :
390:             SIZE 1.000,14.400 :
391:             PICT &mformato:
392:             FONT "MS Sans Serif", 8 :
393:             STYLE "B"
394:          READ
395: <=======RETURN
396:          ******************
397:
*!*************************************************************************
**************
398:  *!
399:  *!       Procedure: REVISAR
400:  *!
401:  *!           Calls: MSCATTER           (procedure in
AACT_SST.SPR)
402:  *!                : DEAC_KEYS          (procedure in
```

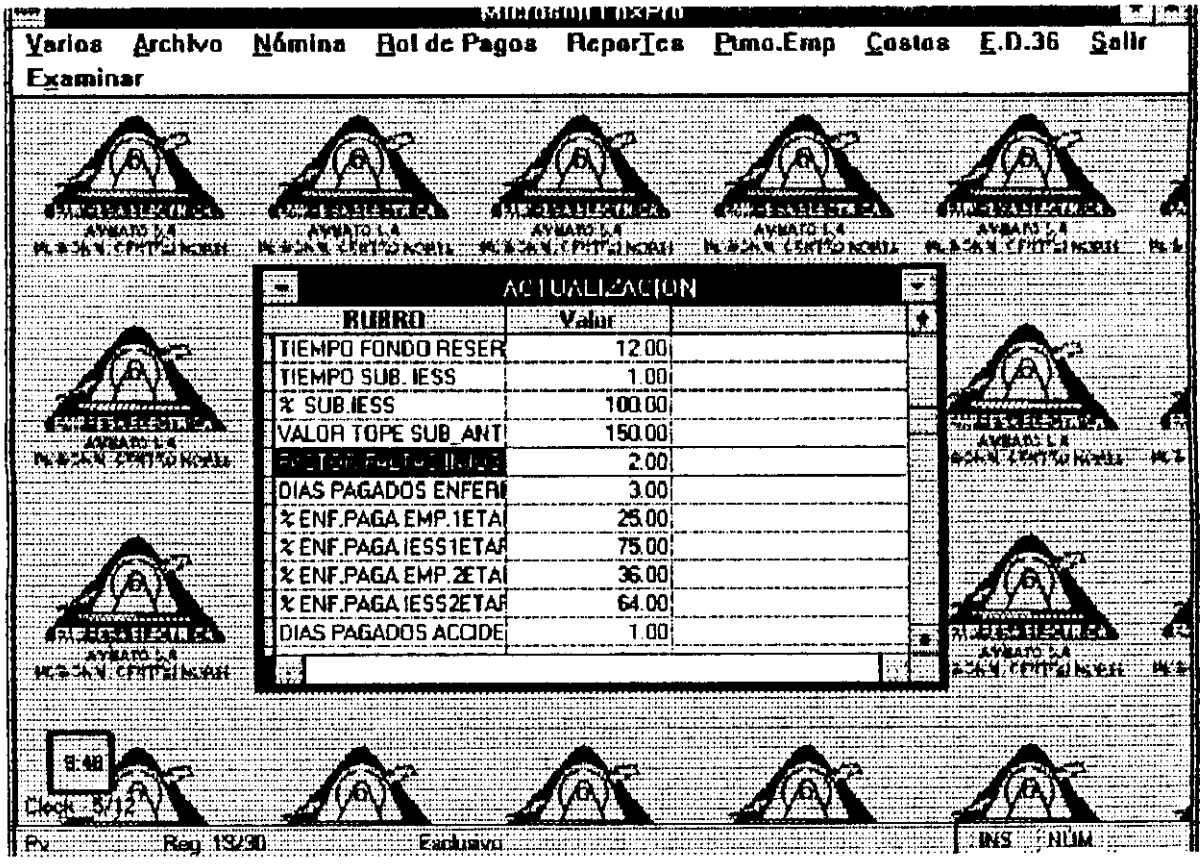
```
AACT_SST.SPR)
  403:  *!                    : ACT_KEYS          (procedure in
AACT_SST.SPR)
  404:  *!                    : VAL_CAMPO()       (function in
AP_PRMCT.SPR)
  405:  *!                    : REFRESH()         (function in
AP_PRMCT.SPR)
  406:  *!                    : MBROW             (procedure in
MO_PDLAB.SPR)
  407:  *!
  408:
*!*********************************************************************
**************
  409:      PROCEDURE revisar
  410:           ******************
  411:           SELE cc
  412:           BROW FIELDS
cc.detalle:H='RUBRO',valor=val_campo():p='999,999,999.99';
  413:              NOMODIFY
  414:           DO REFRESH
  415: <=======RETURN
  416:           ******************
  417:
*!*********************************************************************
**************
  418:  *!
  419:  *!        Function: VAL_CAMPO
  420:  *!
  421:  *!      Called by: REVISAR            (procedure in
AACT_SST.SPR)
  422:  *!
  423:
*!*********************************************************************
**************
  424:      FUNCTION val_campo
  425:           ******************
  426:           STOR cc.campo TO mcampo
  427:           SELE lcc
  428:           STOR &mcampo TO valor
  429:           SELE cc
  430: <=======RETURN valor
  431:
  432:           ***************
  433:
*!*********************************************************************
**************
  434:  *!
  435:  *!       Procedure: SALIR
  436:  *!
  437:  *!       Called by: _RAA1732TG()       (function in
AT_ESCLS.SPR)
  438:  *!                  : DA_ASISG.PRG
  439:  *!                  : DE_EGRQ1.PRG
  440:  *!                  : DE_EGRQ2.PRG
  441:  *!                  : DE_EGRMN.PRG
  442:  *!                  : DE_IEJUB.PRG
  443:  *!
  444:  *!            Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
```

```
  445:  *!
  446:
*!*****************************************************************
**************
  447:      PROCEDURE salir
  448:           ***************
  449:           ON KEY LABEL pgup
  450:           ON KEY LABEL pgdn
  451:           ON KEY LABEL END
  452:           ON KEY LABEL home
  453:           CLEAR READ
  454:  <========RETURN
  455:           ********************
  456:
*!*****************************************************************
**************
  457:  *!
  458:  *!        Function: REFRESH
  459:  *!
  460:  *!      Called by: REVISAR              (procedure in
AACT_SST.SPR)
  461:  *!               : ANTERIOR             (procedure in
AP_PRMCT.SPR)
  462:  *!               : SIGUIENTE            (procedure in
AP_PRMCT.SPR)
  463:  *!               : PRIMERO              (procedure in
AP_PRMCT.SPR)
  464:  *!               : ULTIMO               (procedure in
AP_PRMCT.SPR)
  465:  *!
  466:
*!*****************************************************************
**************
  467:      PROCEDURE REFRESH
  468:           ********************
  469:           SHOW GETS ENABLE
  470:           SELE cc
  471:           STOR cc.campo TO mcampo
  472:           STOR cc.formato TO mformato
  473:           SELE lcc
  474:           @ 6.769,49.200 SAY &mcampo ;
  475:              SIZE 1.000,14.400 ;
  476:              PICT &mformato;
  477:              FONT "MS Sans Serif", 8;
  478:              STYLE "B"
  479:  <========RETURN
  480:
  481:           *
*************************************************************
  482:           *        *
  483:           *        * _R3N0VPW62           OPC_SCR VALID
  484:           *        *
  485:           *        * Function Origin:
  486:           *        *
  487:           *        * From Platform:       Windows
  488:           *        * From Screen:         ACT_PCC.      Record
Number:     6
  489:           *        * Variable:            OPC_SCR
  490:           *        * Called By:           VALID Clause
```

```
491:          *        * Object Type:          Push Button
492:          *        * Snippet Number:       1
493:          *        *
494:          *
**************************************************************
495:          *
496:
*!*************************************************************
**************
497:  *!
498:  *!       Function: _R3N0VPW82
499:  *!
500:  *!      Called by: AP_PRMCC.SPR
501:  *!
502:  *!           Calls: &OPC_SCR.PRG
503:  *!
504:
*!*************************************************************
**************
505:      FUNCTION _r3n0vpw82      &&  OPC_SCR VALID
506:          #REGION 1
507:          DO &opc_scr
508:
509:          *
**************************************************************
510:          *        *
511:          *        * _R3N0VPWJX            Read Level Show
512:          *        *
513:          *        * Function Origin:
514:          *        *
515:          *        *
516:          *        * From Platform:        Windows
517:          *        * From Screen:          ACT_PCC
518:          *        * Called By:            READ Statement
519:          *        * Snippet Number:       2
520:          *        *
521:          *
**************************************************************
522:          *
523:
*!*************************************************************
**************
524:  *!
525:  *!       Function: _R3N0VPWJX
526:  *!
527:  *!      Called by: AP_PRMCC.SPR
528:  *!
529:
*!*************************************************************
**************
530:      FUNCTION _r3n0vpwjx      && Read Level Show
531:          PRIVATE currwind
532:          STORE WOUTPUT() TO currwind
533:          *
534:          * Show Code from screen: ACT_PCC
535:          *
536:          #REGION 1
537:          IF SYS(2016) = "ACT_PCC" OR SYS(2016) = "*"
538:              ACTIVATE WINDOW act_pcc SAME
```

```
539:          |    @ 6.769,13.200 SAY cc.detalle ;
540:          |       SIZE 1.000,25.667, 0.000 ;
541:          |       FONT "MS Serif", 8 ;
542:          |       STYLE "BI"
543:          |ENDIF
544:          |IF NOT EMPTY(currwind)
545:          |    ACTIVATE WINDOW (currwind) SAME
546:          |ENDIF
547:  *:  EOF: AP_PRMCC.ac1
```


Varios Archivo Nómina Bol de Pagos ReporTes Pmo.Emp Costos E.D.36 Salir
ACTUALIZACION
PARAMETROS GENERALES
RUBRO
VALOR
FACTOR DIVISOR XVI S
8
PREVISARR
MODIFICAR
SALIR
<Pg Up> Anterior
<Pg Dn> Siguiente
<Home> Primero
<End> Ultimo
Reg: 2/2
Exclusivo
INS
NUM


Varios Archivo Nómina Rol de Pagos Reportes Pmo.Emp Costos E.D.36 Salir
Examinar
ACTUALIZACION
RUBRO
Valor
TIEMPO FONDO RESER 12.00
TIEMPO SUB. IESS 1.00
% SUB.IESS 100.00
VALOR TOPE SUB_ANT 150.00
DIAS PAGADOS ENFERI 3.00
% ENF.PAGA EMP.1ETAI 25.00
% ENF.PAGA IESS1ETAF 75.00
% ENF.PAGA EMP.2ETAI 36.00
% ENF.PAGA IESS2ETAF 64.00
DIAS PAGADOS ACCIDE 1.00
9:48
Reg 13/30
Exclusivo
INS
NUM

```
    1:
*:**********************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\AP_PRMVR.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-03-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 15/09/1995 at 14:56:48
    8:  *:
    9:  *:  Procs & Fncts: _R3N0W1AYC()
   10:  *:                : _R3N0W1BAF()
   11:  *:
   12:  *:          Set by: EMPRESA.MPR
   13:  *:
   14:  *:           Calls: ANTERIOR            (procedure in
AP_PRMCT.SPR)
   15:  *:                : SIGUIENTE           (procedure in
AP_PRMCT.SPR)
   16:  *:                : PRIMERO             (procedure in
AP_PRMCT.SPR)
   17:  *:                : ULTIMO              (procedure in
AP_PRMCT.SPR)
   18:  *:                : _R3N0W1AYC()        (function in
AP_PRMVR.SPR)
   19:  *:                : _R3N0W1BAF()        (function in
AP_PRMVR.SPR)
   20:  *:
   21:  *:       Documented 19:14:37
FoxDoc version 3.00a
   22:
*:**********************************************************************
**************
   23:      *
************************************************************
   24:      *        *
   25:      *        * 09/15/1995           ACT_PVAR.SPR
 14:56:47
   26:      *        *
   27:      *
************************************************************
   28:      *        *
   29:      *        * Author's Name
   30:      *        *
   31:      *        * Copyright (c) 1995 Company Name
   32:      *        * Address
   33:      *        * City.      Zip
   34:      *        *
   35:      *        * Description:
   36:      *        * This program was automatically generated by
GENSCRN.
   37:      *        *
   38:      *
************************************************************
   39:
   40:
   41:      *
************************************************************
```

```
   42:      *        *
   43:      *        *          ACT_PVAR/Windows Setup Code -
SECTION 1
   44:      *        *
   45:      *
************************************************************
   46:      *
   47:
   48:      #REGION 1
   49:      **********
   50:      REGIONAL mcampo,mformato,opc_scr
   51:      STOR '' TO mcampo,opc_scr,mformato
   52:      ON KEY LABEL pgup DO anterior
   53:      ON KEY LABEL pgdn DO siguiente
   54:      ON KEY LABEL home DO primero
   55:      ON KEY LABEL END  DO ultimo
   56:
   57:      **********
   58:
   59:      #REGION 0
   60:      REGIONAL m.currarea, m.talkstat, m.compstat
   61:
   62:    ┌─IF SET("TALK") = "ON"
   63:    │    SET TALK OFF
   64:    │    m.talkstat = "ON"
   65:    ├─ELSE
   66:    │    m.talkstat = "OFF"
   67:    └─ENDIF
   68:      m.compstat = SET("COMPATIBLE")
   69:      SET COMPATIBLE FOXPLUS
   70:
   71:      m.rborder = SET("READBORDER")
   72:      SET readborder ON
   73:
   74:      m.currarea = SELECT()
   75:
   76:
   77:      *
************************************************************
   78:      *        *
   79:      *        *      ACT_PVAR/Windows Databases, Indexes,
Relations
   80:      *        *
   81:      *
************************************************************
   82:      *
   83:
   84:    ┌─IF USED("lpv")
   85:    │    SELECT lpv
   86:    │    SET ORDER TO 0
   87:    ├─ELSE
   88:    │    SELECT 0
   89:    │    USE (LOCFILE("dbfs\paramvar.dbf","DBF","Where is
paramvar?"));
   90:    │         AGAIN ALIAS lpv ;
   91:    │         ORDER 0
   92:    └─ENDIF
   93:
   94:    ┌─IF USED("pv")
```

```
   95:     |    SELECT PV
   96:     |    SET ORDER TO 0
   97:     |-ELSE
   98:     |    SELECT 0
   99:     |    USE (LOCFILE("dbfs|para_var.par","DBF","Where is
para_var?"));
  100:     |        AGAIN ALIAS PV ;
  101:     |        ORDER 0
  102:     |-ENDIF
  103:
  104:
  105:
  106:     *
************************************************************
  107:     *       *
  108:     *       *                Windows Window definitions
  109:     *       *
  110:     *
************************************************************
  111:     *
  112:
  113:     |-IF NOT WEXIST("act_pvar") ;
  114:     |       OR UPPER(WTITLE("ACT_PVAR")) == "ACT_PVAR.PJX" ;
  115:     |       OR UPPER(WTITLE("ACT_PVAR")) == "ACT_PVAR.SCX" ;
  116:     |       OR UPPER(WTITLE("ACT_PVAR")) == "ACT_PVAR.MNX" ;
  117:     |       OR UPPER(WTITLE("ACT_PVAR")) == "ACT_PVAR.PRG" ;
  118:     |       OR UPPER(WTITLE("ACT_PVAR")) == "ACT_PVAR.FRX" ;
  119:     |       OR UPPER(WTITLE("ACT_PVAR")) == "ACT_PVAR.QPR"
  120:     |    DEFINE WINDOW act_pvar ;
  121:     |       AT  0.000, 0.000  ;
  122:     |       SIZE 16.231,72.400 ;
  123:     |       TITLE "ACTUALIZACION" ;
  124:     |       FONT "MS Sans Serif", 8 ;
  125:     |       FLOAT ;
  126:     |       NOCLOSE ;
  127:     |       MINIMIZE ;
  128:     |       SYSTEM
  129:     |    MOVE WINDOW act_pvar CENTER
  130:     |-ENDIF
  131:
  132:
  133:     *
************************************************************
  134:     *       *
  135:     *       *          ACT_PVAR/Windows Setup Code -
SECTION 2
  136:     *       *
  137:     *
************************************************************
  138:     *
  139:
  140:     #REGION 1
  141:     **********
  142:     SELE PV
  143:     GO TOP
  144:     STOR pv.campo TO mcampo
  145:     STOR pv.formato TO mformato
  146:     SHOW GETS ENABLE
  147:     SELE lpv
```

```
148:
149:      *
**********************************************************
150:      *        *
151:      *        *              ACT_PVAR/Windows Screen Layout
152:      *        *
153:      *
**********************************************************
154:      *
155:
156:      #REGION 1
157:      IF WVISIBLE("act_pvar")
158:         ACTIVATE WINDOW act_pvar SAME
159:      ELSE
160:         ACTIVATE WINDOW act_pvar NOSHOW
161:      ENDIF
162:      @ 6.769,13.200 SAY pv.detalle ;
163:         SIZE 1.000,25.667 ;
164:         FONT "MS Serif", 8 ;
165:         STYLE "BI"
166:      @ 6.769,50.600 SAY &mcampo ;
167:         SIZE 1.000,12.000 ;
168:         FONT "MS Sans Serif", 8 ;
169:         STYLE "B" ;
170:         PICTURE "&MFORMATO"
171:      @ 10.231,14.400 GET opc_scr ;
172:         PICTURE "@*HN \<REVISAR;\<MODIFICAR;\<SALIR" ;
173:         SIZE 1.385,12.500,0.333 ;
174:         DEFAULT 1 ;
175:         FONT "MS Sans Serif", 8 ;
176:         STYLE "B" ;
177:         VALID _r3n0w1ayc()
178:      @ 13.308,19.000 SAY "<Pg.Up.>"  ;
179:         FONT "MS Serif", 8 ;
180:         STYLE "BT"
181:      @ 14.154,19.400 SAY "Anterior"  ;
182:         FONT "MS Serif", 8 ;
183:         STYLE "T"
184:      @ 13.385,30.200 SAY "<Pg.Dn.> <Home>  <End>"  ;
185:         FONT "MS Serif", 8 ;
186:         STYLE "BT"
187:      @ 14.231,30.600 SAY "Siguiente"  ;
188:         FONT "MS Serif", 8 ;
189:         STYLE "T"
190:      @ 14.231,41.600 SAY "Primero"  ;
191:         FONT "MS Serif", 8 ;
192:         STYLE "T"
193:      @ 14.231,51.200 SAY "Ultimo"  ;
194:         FONT "MS Serif", 8 ;
195:         STYLE "T"
196:      @ 13.000,17.000 TO 15.462,58.400 ;
197:         PEN 1, 8 ;
198:         STYLE "16"
199:      @ 0.615,27.200 SAY "PARAMETROS " + CHR(13) + ;
200:         "GENERALES"  ;
201:         SIZE 2.000,15.857, 0.000 ;
202:         PICTURE "@I" ;
203:         FONT "MS Serif", 10 ;
204:         STYLE "BIT" ;
```

```
205:          COLOR RGB(0,0,0,,,,)
206:       @ 0.231,17.000 TO 3.539,58.000 ;
207:          PEN 1, 8 ;
208:          STYLE "16"
209:       @ 4.615,8.600 TO 8.384,68.400 ;
210:          PEN 1, 8
211:       @ 4.769,19.200 SAY "RUBRO"  ;
212:          FONT "MS Sans Serif", 8 ;
213:          STYLE "BIT"
214:       @ 4.769,57.200 SAY "VALOR"  ;
215:          FONT "MS Sans Serif", 8 ;
216:          STYLE "BIT"
217:       @ 9.769,12.600 TO 11.769,61.200 ;
218:          PEN 1, 8 ;
219:          STYLE "16"
220:       @ 0.538,17.800 TO 3.230,57.200 ;
221:          PEN 1, 8 ;
222:          STYLE "16"
223:
224:       IF NOT WVISIBLE("act_pvar")
225:          ACTIVATE WINDOW act_pvar
226:       ENDIF
227:
228:       READ CYCLE ;
229:          SHOW _r3n0w1baf()
230:
231:       RELEASE WINDOW act_pvar
232:
233:       *
**********************************************************
234:       *        *
235:       *        *                  Windows Closing Databases
236:       *        *
237:       *
**********************************************************
238:       *
239:
240:       IF USED("lpv")
241:          SELECT lpv
242:          USE
243:       ENDIF
244:
245:       IF USED("pv")
246:          SELECT PV
247:          USE
248:       ENDIF
249:
250:       SELECT (m.currarea)
251:
252:
253:       #REGION 0
254:
255:       SET readborder &rborder
256:
257:       IF m.talkstat = "ON"
258:          SET TALK ON
259:       ENDIF
260:       IF m.compstat = "ON"
261:          SET COMPATIBLE ON
```

```
  262:    └─ENDIF
  263:
  264:
  265:      *
************************************************************
  266:      *        *
  267:      *        *                ACT_PVAR/Windows Cleanup Code
  268:      *        *
  269:      *
************************************************************
  270:      *
  271:
  272:      #REGION 1
  273:      ********************
  274:
  275:
  276:      *
************************************************************
  277:      *        *
  278:      *        *  ACT_PVAR/Windows Supporting Procedures and
Functions
  279:      *        *
  280:      *
************************************************************
  281:      *
  282:
  283:      #REGION 1
  284:
*!*************************************************************************
**************
  285:   *!
  286:   *!       Procedure: SIGUIENTE
  287:   *!
  288:   *!       Called by: AP_PRMCT.SPR
  289:   *!                : AP_PRMCC.SPR
  290:   *!                : AP_PRMVR.SPR
  291:   *!                : SECC()              (function in
PERSONAL.PRG)
  292:   *!
  293:   *!           Calls: REFRESH()          (function in
AP_PRMCT.SPR)
  294:   *!                : DESPLEGA           (procedure in
PERSONAL.PRG)
  295:   *!
  296:
*!*************************************************************************
**************
  297:      PROCEDURE siguiente
  298:          ********************
  299:          SELE PV
  300:        ┌─IF !EOF()
  301:        │    SKIP
  302:        │  ┌─IF EOF()
  303:        │  │    GOTO BOTTOM
  304:        │  └─ENDIF
  305:        ├─ELSE
  306:        │    GOTO BOTTOM
  307:        │    WAIT WIND 'FIN DE ARCHIVO'
  308:        └─ENDIF
```

```
  309:          DO REFRESH
  310: <=======RETURN
  311:          ******************
  312:
*!*************************************************************
**************
  313:  *!
  314:  *!       Procedure: ANTERIOR
  315:  *!
  316:  *!       Called by: AP_PRMCT.SPR
  317:  *!                : AP_PRMCC.SPR
  318:  *!                : AP_PRMVR.SPR
  319:  *!                : SECC()              (function in
PERSONAL.PRG)
  320:  *!
  321:  *!           Calls: REFRESH()          (function in
AP_PRMCT.SPR)
  322:  *!                : DESPLEGA           (procedure in
PERSONAL.PRG)
  323:  *!
  324:
*!*************************************************************
**************
  325:      PROCEDURE anterior
  326:          ******************
  327:          SELE PV
  328:          IF !BOF()
  329:              SKIP -1
  330:              IF BOF()
  331:                  GOTO TOP
  332:              ENDIF
  333:          ELSE
  334:              GOTO TOP
  335:              WAIT WIND 'INICIO DE ARCHIVO'
  336:          ENDIF
  337:          DO REFRESH
  338: <=======RETURN
  339:          ******************
  340:
*!*************************************************************
**************
  341:  *!
  342:  *!       Procedure: PRIMERO
  343:  *!
  344:  *!       Called by: AP_PRMCT.SPR
  345:  *!                : AP_PRMCC.SPR
  346:  *!                : AP_PRMVR.SPR
  347:  *!
  348:  *!           Calls: REFRESH()          (function in
AP_PRMCT.SPR)
  349:  *!
  350:
*!*************************************************************
**************
  351:      PROCEDURE primero
  352:          ******************
  353:          SELE PV
  354:          GOTO TOP
  355:          WAIT WIND 'INICIO DE ARCHIVO'
```

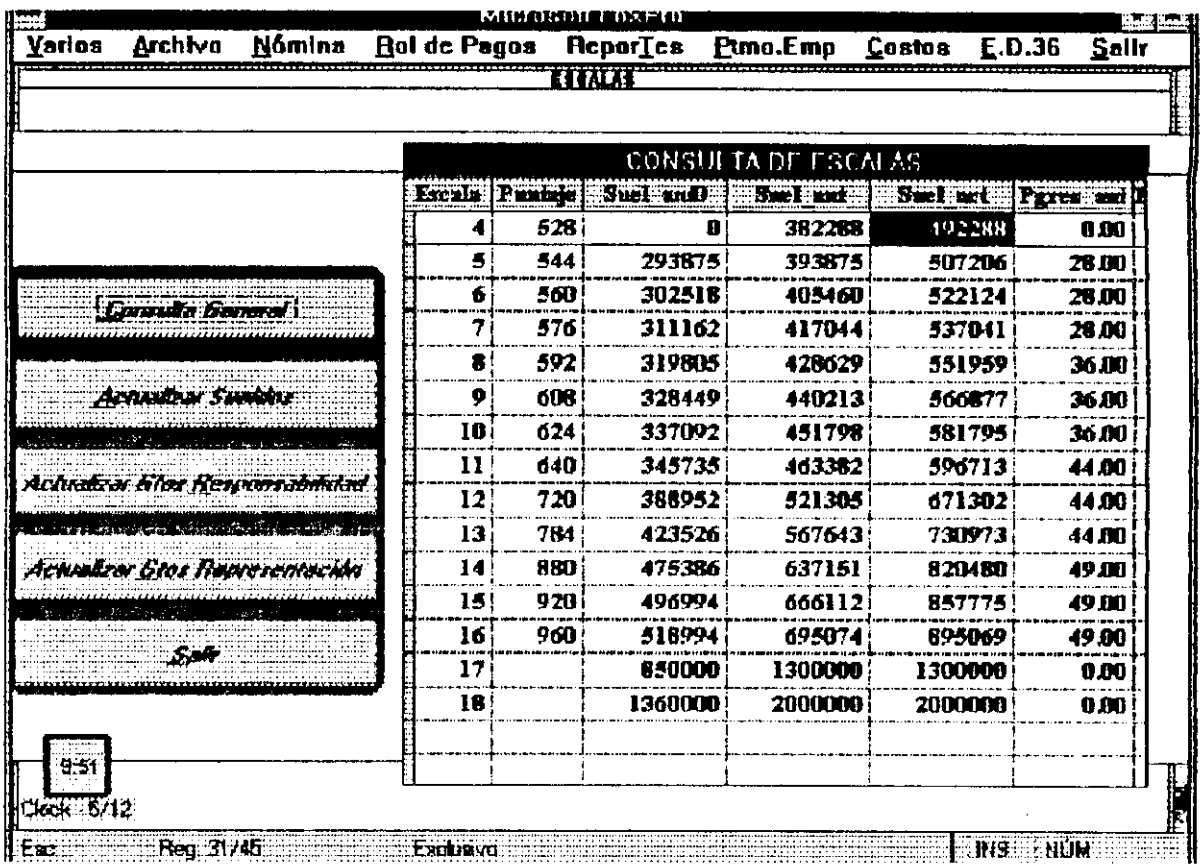
```
  356:          DO REFRESH
  357: <=======RETURN
  358:          ******************
  359:
*!*************************************************************************
**************
  360:  *!
  361:  *!       Procedure: ULTIMO
  362:  *!
  363:  *!       Called by: AP_PRMCT.SPR
  364:  *!                : AP_PRMCC.SPR
  365:  *!                : AP_PRMVR.SPR
  366:  *!
  367:  *!           Calls: REFRESH()          (function in
AP_PRMCT.SPR)
  368:  *!
  369:
*!*************************************************************************
**************
  370:      PROCEDURE ultimo
  371:          ******************
  372:          SELE PV
  373:          GOTO BOTTOM
  374:          WAIT WIND 'FIN DE ARCHIVO'
  375:          DO REFRESH
  376: <=======RETURN
  377:
  378:
  379:          ********************
  380:
*!*************************************************************************
**************
  381:  *!
  382:  *!       Procedure: MODIFICAR
  383:  *!
  384:
*!*************************************************************************
**************
  385:      PROCEDURE modificar
  386:          ********************
  387:          A = pv.campo
  388:          @ 6.769,49.200 GET &mcampo ;
  389:             SIZE 1.000,14.400 ;
  390:             PICT &mformato;
  391:             FONT "MS Sans Serif", 8 ;
  392:             STYLE "B"
  393:          READ
  394: <=======RETURN
  395:          ******************
  396:
*!*************************************************************************
**************
  397:  *!
  398:  *!       Procedure: REVISAR
  399:  *!
  400:  *!           Calls: MSCATTER           (procedure in
AACT_SST.SPR)
  401:  *!                : DEAC_KEYS          (procedure in
AACT_SST.SPR)
```

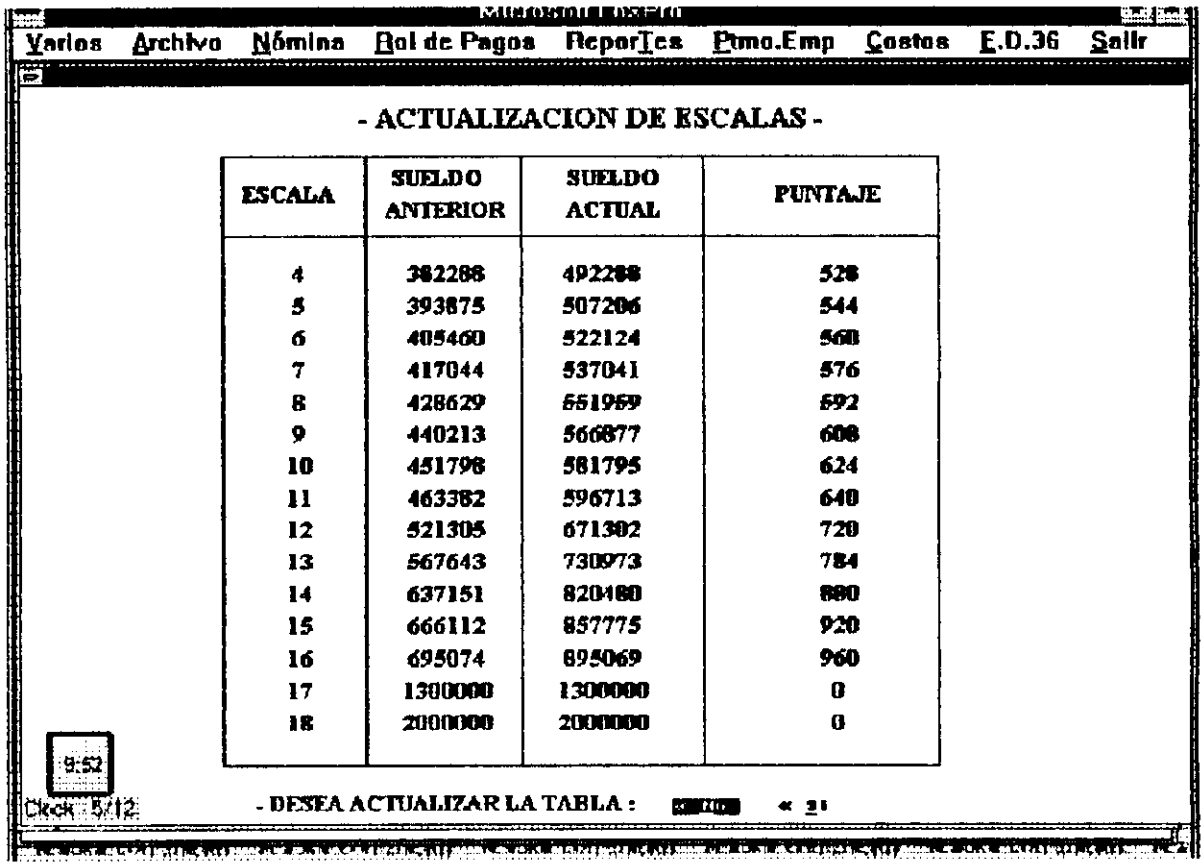
```
  402:  *!                : ACT_KEYS          (procedure in
AACT_SST.SPR)
  403:  *!                : VAL_CAMPO()       (function in
AP_PRMCT.SPR)
  404:  *!                : REFRESH()         (function in
AP_PRMCT.SPR)
  405:  *!                : MBROW             (procedure in
MO_PDLAB.SPR)
  406:  *!
  407:
*!*************************************************************************
**************
  408:      PROCEDURE revisar
  409:           ******************
  410:           SELE PV
  411:           BROW FIELDS
pv.detalle:H='RUBRO',valor=val_campo():p='999,999,999.99';
  412:              NOMODIFY
  413:           DO REFRESH
  414: <=======RETURN
  415:           ********************
  416:
*!*************************************************************************
**************
  417:  *!
  418:  *!        Function: VAL_CAMPO
  419:  *!
  420:  *!      Called by: REVISAR            (procedure in
AACT_SST.SPR)
  421:  *!
  422:
*!*************************************************************************
**************
  423:      FUNCTION val_campo
  424:           ******************
  425:           STOR pv.campo TO mcampo
  426:           SELE !pv
  427:           STOR &mcampo TO valor
  428:           SELE PV
  429: <=======RETURN valor
  430:
  431:           ****************
  432:
*!*************************************************************************
**************
  433:  *!
  434:  *!       Procedure: SALIR
  435:  *!
  436:  *!      Called by: _RAA1732TG()       (function in
AT_ESCLS.SPR)
  437:  *!                 : DA_ASISG.PRG
  438:  *!                 : DE_EGRQ1.PRG
  439:  *!                 : DE_EGRQ2.PRG
  440:  *!                 : DE_EGRMN.PRG
  441:  *!                 : DE_IEJUB.PRG
  442:  *!
  443:  *!           Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
```

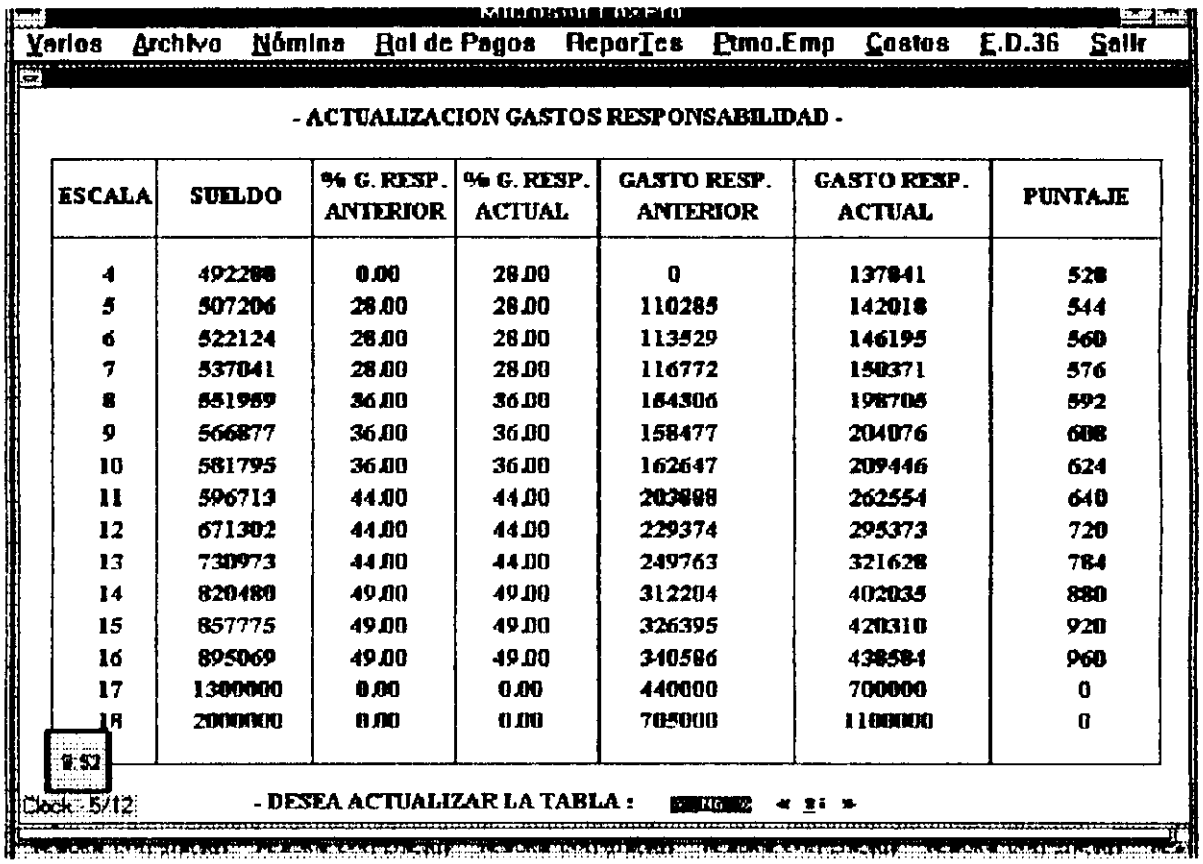
```
  444:   *!
  445:
*!*************************************************************************
**************
  446:      PROCEDURE salir
  447:           ***************
  448:           ON KEY LABEL pgup
  449:           ON KEY LABEL pgdn
  450:           ON KEY LABEL END
  451:           ON KEY LABEL home
  452:           CLEAR READ
  453: <=========RETURN
  454:           ********************
  455:
*!*************************************************************************
**************
  456:   *!
  457:   *!        Function: REFRESH
  458:   *!
  459:   *!      Called by: REVISAR              (procedure in
AACT_SST.SPR)
  460:   *!               : ANTERIOR             (procedure in
AP_PRMCT.SPR)
  461:   *!               : SIGUIENTE            (procedure in
AP_PRMCT.SPR)
  462:   *!               : PRIMERO              (procedure in
AP_PRMCT.SPR)
  463:   *!               : ULTIMO               (procedure in
AP_PRMCT.SPR)
  464:   *!
  465:
*!*************************************************************************
**************
  466:      PROCEDURE REFRESH
  467:           ********************
  468:           SHOW GETS ENABLE
  469:           SELE PV
  470:           STOR pv.campo TO mcampo
  471:           STOR pv.formato TO mformato
  472:           SELE lpv
  473:           @ 6.769,49.200 SAY &mcampo :
  474:              SIZE 1.000,14.400 :
  475:              PICT &mformato;
  476:              FONT "MS Sans Serif", 8;
  477:              STYLE "B"
  478: <=========RETURN
  479:
  480:           *
*************************************************************
  481:           *        *
  482:           *        * _R3N0W1AYC            OPC_SCR VALID
  483:           *        *
  484:           *        * Function Origin:
  485:           *        *
  486:           *        * From Platform:        Windows
  487:           *        * From Screen:          ACT_PVAR,
Record Number:    6
  488:           *        * Variable:             OPC_SCR
```

```
489:          *        * Called By:            VALID Clause
490:          *        * Object Type:          Push Button
491:          *        * Snippet Number:       1
492:          *        *
493:          *
************************************************************
494:          *
495:
*!*************************************************************************
**************
496:  *!
497:  *!        Function: _R3N0W1AYC
498:  *!
499:  *!      Called by: AP_PRMVR.SPR
500:  *!
501:  *!           Calls: &OPC_SCR.PRG
502:  *!
503:
*!*************************************************************************
**************
504:      FUNCTION _r3n0w1ayc      &&  OPC_SCR VALID
505:          #REGION 1
506:          DO &opc_scr
507:
508:          *
************************************************************
509:          *        *
510:          *        * _R3N0W1BAF            Read Level Show
511:          *        *
512:          *        * Function Origin:
513:          *        *
514:          *        *
515:          *        * From Platform:        Windows
516:          *        * From Screen:          ACT_PVAR
517:          *        * Called By:            READ Statement
518:          *        * Snippet Number:       2
519:          *        *
520:          *
************************************************************
521:          *
522:
*!*************************************************************************
**************
523:  *!
524:  *!        Function: _R3N0W1BAF
525:  *!
526:  *!      Called by: AP_PRMVR.SPR
527:  *!
528:
*!*************************************************************************
**************
529:      FUNCTION _r3n0w1baf      && Read Level Show
530:          PRIVATE currwind
531:          STORE WOUTPUT() TO currwind
532:          *
533:          * Show Code from screen: ACT_PVAR
534:          *
535:          #REGION 1
```

```
536:        ┌─IF SYS(2016) = "ACT_PVAR" OR SYS(2016) = "**"
537:        │     ACTIVATE WINDOW act_pvar SAME
538:        │     @ 6.769,13.200 SAY pv.detalle :
539:        │        SIZE 1.000,25.667, 0.000 ;
540:        │        FONT "MS Serif", 8 ;
541:        │        STYLE "BI"
542:        └─ENDIF
543:        ┌─IF NOT EMPTY(currwind)
544:        │     ACTIVATE WINDOW (currwind) SAME
545:        └─ENDIF
546:  *:  EOF: AP_PRMVR.acl
```


Varios Archivo Nómina Bol de Pagos ReporTes Pmo.Emp Costos E.D.36 Salir
ESCALAS
CONSULTA DE ESCALAS
Escala Puntaje Suel mín Suel med Suel sup Pgres anu
4 528 0 382288 422288 0.00
5 544 293875 393875 507206 28.00
6 560 302518 405460 522124 28.00
7 576 311162 417044 537041 28.00
8 592 319805 428629 551959 36.00
9 608 328449 440213 566877 36.00
10 624 337092 451798 581795 36.00
11 640 345735 463382 596713 44.00
12 720 388952 521305 671302 44.00
13 784 423526 567643 730973 44.00
14 880 475386 637151 820480 49.00
15 920 496994 666112 857775 49.00
16 960 518994 695074 895069 49.00
17 850000 1300000 1300000 0.00
18 1360000 2000000 2000000 0.00
Escala General
Actualizar Sueldos
Actualizar Años Reconocidos
Actualizar Otros Reconocidos
Salir
8:51
Clock 5/12
Esc Reg 31/45 Exclusivo INS NUM


Varios Archivo Nómina Bol de Pagos ReporTes Ptmo.Emp Costos E.D.36 Salir
- ACTUALIZACION DE ESCALAS -
ESCALA
SUELDO ANTERIOR
SUELDO ACTUAL
PUNTAJE
4 382288 492288 528
5 393875 507206 544
6 405460 522124 560
7 417044 537041 576
8 428629 551959 592
9 440213 566877 608
10 451798 581795 624
11 463382 596713 640
12 521305 671302 720
13 567643 730973 784
14 637151 820480 880
15 666112 857775 920
16 695074 895069 960
17 1300000 1300000 0
18 2000000 2000000 0
9:52
Click 5/12
- DESEA ACTUALIZAR LA TABLA :


Varios Archivo Nómina Rol de Pagos Reportes Ptmo.Emp Costos E.D.36 Salir
- ACTUALIZACION GASTOS RESPONSABILIDAD -
ESCALA | SUELDO | % G. RESP. ANTERIOR | % G. RESP. ACTUAL | GASTO RESP. ANTERIOR | GASTO RESP. ACTUAL | PUNTAJE
4 | 492288 | 0.00 | 28.00 | 0 | 137841 | 528
5 | 507206 | 28.00 | 28.00 | 110285 | 142018 | 544
6 | 522124 | 28.00 | 28.00 | 113529 | 146195 | 560
7 | 537041 | 28.00 | 28.00 | 116772 | 150371 | 576
8 | 551959 | 36.00 | 36.00 | 164306 | 198706 | 592
9 | 566877 | 36.00 | 36.00 | 158477 | 204076 | 608
10 | 581795 | 36.00 | 36.00 | 162647 | 209446 | 624
11 | 596713 | 44.00 | 44.00 | 203898 | 262554 | 640
12 | 671302 | 44.00 | 44.00 | 229374 | 295373 | 720
13 | 730973 | 44.00 | 44.00 | 249763 | 321628 | 784
14 | 820480 | 49.00 | 49.00 | 312204 | 402035 | 880
15 | 857775 | 49.00 | 49.00 | 326395 | 420310 | 920
16 | 895069 | 49.00 | 49.00 | 340586 | 438584 | 960
17 | 1300000 | 0.00 | 0.00 | 440000 | 700000 | 0
18 | 2000000 | 0.00 | 0.00 | 705000 | 1100000 | 0
8:52
Clock 5/12
- DESEA ACTUALIZAR LA TABLA :

```
    1:
*:*************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\AT_ESCLS.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-03-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 09/05/1995 at 20:06:08
    8:  *:
    9:  *:  Procs & Fncts: _RAA1732TG()
   10:  *:                : CON_ESCALA
   11:  *:                : ACT_SUELDO
   12:  *:                : ACT_G_REP
   13:  *:                : ACT_G_RES
   14:  *:                : CUADRO1
   15:  *:                : CUADRO2
   16:  *:                : CUADRO3
   17:  *:
   18:  *:          Set by: EMPRESA.MPR
   19:  *:
   20:  *:           Calls: _RAA1732TG()        (function in
AT_ESCLS.SPR)
   21:  *:
   22:  *:       Documented 19:14:38
FoxDoc version 3.00a
   23:
*:*************************************************************
**************
   24:      *
************************************************************
   25:      *        *
   26:      *        * 09/05/1996            AT_ESCLS.SPR
 20:06:04
   27:      *        *
   28:      *
************************************************************
   29:      *        *
   30:      *        * Nombre del autor
   31:      *        *
   32:      *        * Copyright (c) 1995 Nombre de la empresa
   33:      *        * Direcci¾n
   34:      *        * Ciudad,      C.P.
   35:      *        * Pa¡s
   36:      *        *
   37:      *        * Description:
   38:      *        * Este programa lo ha generado autom¿ticamente
 GENSCRN.
   39:      *        *
   40:      *
************************************************************
   41:
   42:
   43:      *
************************************************************
   44:      *        *
   45:      *        *  AT_ESCLS/Windows C¾digo de configuraci¾n -
```

```
SECCION 1
   46:      *        *
   47:      *
************************************************************
   48:      *
   49:
   50:      #REGION 1
   51:      ***********
   52:      REGIONAL opc_scr
   53:      STOR 1 TO opc_scr
   54:
   55:      **********
   56:
   57:      #REGION 0
   58:      REGIONAL m.currarea, m.talkstat, m.compstat
   59:
   60:     ┌IF SET("TALK") = "ON"
   61:     │   SET TALK OFF
   62:     │   m.talkstat = "ON"
   63:     ├ELSE
   64:     │   m.talkstat = "OFF"
   65:     └ENDIF
   66:      m.compstat = SET("COMPATIBLE")
   67:      SET COMPATIBLE FOXPLUS
   68:
   69:      m.rborder = SET("READBORDER")
   70:      SET readborder ON
   71:
   72:      m.currarea = SELECT()
   73:
   74:
   75:      *
************************************************************
   76:      *        *
   77:      *        *  AT_ESCLS/Windows Bases de datos, Ýndices,
relaciones
   78:      *        *
   79:      *
************************************************************
   80:      *
   81:
   82:     ┌IF USED("esc")
   83:     │   SELECT esc
   84:     │   SET ORDER TO 0
   85:     ├ELSE
   86:     │   SELECT 0
   87:     │   USE (LOCFILE("sroll\dbfs\escala.dbf","DBF","Buscar
escala?"));
   88:     │        AGAIN ALIAS esc ;
   89:     │        ORDER 0
   90:     └ENDIF
   91:
   92:     ┌IF USED("dgp")
   93:     │   SELECT dgp
   94:     │   SET ORDER TO TAG "categoria"
   95:     ├ELSE
   96:     │   SELECT 0
   97:     │   USE (LOCFILE("sroll\dbfs\personal.dbf","DBF","Buscar
```

```
personal?"));
   98:          AGAIN ALIAS dgp :
   99:          ORDER TAG "categoria"
  100:      ENDIF
  101:
  102:      SELECT dgp
  103:
  104:
  105:      *
************************************************************
  106:      *        *
  107:      *        *              Windows Definiciones de ventana
  108:      *        *
  109:      *
************************************************************
  110:      *
  111:
  112:      IF NOT WEXIST("at_escls") :
  113:           OR UPPER(WTITLE("AT_ESCLS")) == "AT_ESCLS.PJX" :
  114:           OR UPPER(WTITLE("AT_ESCLS")) == "AT_ESCLS.SCX" :
  115:           OR UPPER(WTITLE("AT_ESCLS")) == "AT_ESCLS.MNX" :
  116:           OR UPPER(WTITLE("AT_ESCLS")) == "AT_ESCLS.PRG" :
  117:           OR UPPER(WTITLE("AT_ESCLS")) == "AT_ESCLS.FRX" :
  118:           OR UPPER(WTITLE("AT_ESCLS")) == "AT_ESCLS.QPR"
  119:         DEFINE WINDOW at_escls :
  120:           AT  0.000, 0.000  ;
  121:           SIZE 24.923,129.200 :
  122:           TITLE "Actualizaci?n de Escalas" :
  123:           FONT "MS Sans Serif", 8 :
  124:           FLOAT :
  125:           NOCLOSE :
  126:           MINIMIZE :
  127:           SYSTEM
  128:         MOVE WINDOW at_escls CENTER
  129:      ENDIF
  130:
  131:
  132:      *
************************************************************
  133:      *        *
  134:      *        *  AT_ESCLS/Windows C?digo de configuraci?n -
SECCION 2
  135:      *        *
  136:      *
************************************************************
  137:      *
  138:
  139:      #REGION 1
  140:      **********
  141:
  142:      STORE ' FONT "N",12 style "B"'TO fsdesc && styl and
font of say coment
  143:      STORE ' FONT "N",10 STYLE "B"' TO fscampc && styl and
font of say CHARACTER fields
  144:      STORE ' FONT "N",10 STYLE "B"' TO fscampn && styl and
font of say NUMERIC fields
  145:
  146:      DEFINE WINDOW procesos FROM 0,0 TO 47,100 SHADOW FLOAT
```

```
GROW NOZOOM
  147:      DEFINE WINDOW consulta FROM 5,30 TO 45,87 SHADOW TITLE
" CONSULTA DE ESCALAS " NOGROW NOFLOAT FONT "r",10 STYLE "b"
  148:      DEFINE WINDOW ven FROM 0,0 TO 48,89 PANEL TITLE "
ESCALAS " SHADOW FLOAT GROW NOZOOM
  149:      ACTIVATE WINDOW ven
  150:      CLEAR
  151:      SELECT esc
  152:
  153:      *
************************************************************
  154:      *         *
  155:      *         *         AT_ESCLS/Windows Distribuci?n de
pantalla
  156:      *         *
  157:      *
************************************************************
  158:      *
  159:
  160:      #REGION 1
  161:     ┌IF WVISIBLE("at_escls")
  162:     │   ACTIVATE WINDOW at_escls SAME
  163:     ├ELSE
  164:     │   ACTIVATE WINDOW at_escls NOSHOW
  165:     └ENDIF
  166:      @ 4.077,0.800 TO 22.462,41.400 :
  167:         PATTERN 1 ;
  168:         PEN 1, 8 ;
  169:         STYLE "16" ;
  170:         COLOR RGB(,,,0,128,0)
  171:      @ 4.462,1.600 GET opc_scr :
  172:         PICTURE "@*VN |Consulta General:|Actualizar
Sueldos;Actualizar Gtos |Responsabilidad;Actualizar |Gtos
Representaci?n;|Salir" :
  173:         SIZE 2.923,32.667,0.769 :
  174:         DEFAULT 1 :
  175:         FONT "MS Sans Serif", 8 :
  176:         STYLE "BI" ;
  177:         VALID _raa1732tg()
  178:
  179:     ┌IF NOT WVISIBLE("at_escls")
  180:     │   ACTIVATE WINDOW at_escls
  181:     └ENDIF
  182:
  183:      READ CYCLE :
  184:         VALID .F.
  185:
  186:      RELEASE WINDOW at_escls
  187:
  188:      *
************************************************************
  189:      *         *
  190:      *         *              Windows Cerrando bases de datos
  191:      *         *
  192:      *
************************************************************
  193:      *
  194:
```

```
  195:     IF USED("esc")
  196:        SELECT esc
  197:        USE
  198:     ENDIF
  199:
  200:     IF USED("dgp")
  201:        SELECT dgp
  202:        USE
  203:     ENDIF
  204:
  205:     SELECT (m.currarea)
  206:
  207:
  208:     #REGION 0
  209:
  210:     SET readborder &rborder
  211:
  212:     IF m.talkstat = "ON"
  213:        SET TALK ON
  214:     ENDIF
  215:     IF m.compstat = "ON"
  216:        SET COMPATIBLE ON
  217:     ENDIF
  218:
  219:
  220:     *
************************************************************
  221:     *        *
  222:     *        *            AT_ESCLS/Windows Código de
limpieza
  223:     *        *
  224:     *
************************************************************
  225:     *
  226:
  227:     #REGION 1
  228:     ************************
  229:
  230:
  231:     *
************************************************************
  232:     *        *
  233:     *        * AT_ESCLS/Windows Procedimientos y funciones
de soporte
  234:     *        *
  235:     *
************************************************************
  236:     *
  237:
  238:     #REGION 1
  239:
*!*************************************************************************
**************
  240:  *!
  241:  *!      Procedure: CON_ESCALA
  242:  *!
  243:  *!      Called by: _RAA1732TG()        (function in
AT_ESCLS.SPR)
```

```
  244:  *!
  245:
*!*************************************************************************
**************
  246:      PROCEDURE con_escala
  247:          **********************
  248:          ACTIVATE WINDOW consulta
  249:          BROWSE NOAPPEND NOMODIFY NODELE IN WINDOW consulta
  250:          DEACTIVATE WINDOW consulta
  251: <========RETURN
  252:          **********************
  253:
*!*************************************************************************
**************
  254:  *!
  255:  *!       Procedure: ACT_SUELDO
  256:  *!
  257:  *!       Called by: _RAA1732TG()        (function in
AT_ESCLS.SPR)
  258:  *!
  259:  *!           Calls: CUADRO1             (procedure in
AT_ESCLS.SPR)
  260:  *!
  261:
*!*************************************************************************
**************
  262:      PROCEDURE act_sueldo
  263:          **********************
  264:          ACTIVATE WINDOW procesos
  265:          @ 4,15 TO 9,70
  266:          @ 9,15 TO 42,70
  267:          @ 4,15 TO 42,26
  268:          @ 4,26 TO 42,38
  269:          @ 4,38 TO 42,52
  270:          @ 4,52 TO 42,70
  271:          @ 1,26 SAY '- ACTUALIZACION DE ESCALAS - ' &fsdesc
  272:          @ 9,15 SAY '├'
  273:          @ 9,70 SAY '┤'
  274:          @ 4,26 SAY '┬'
  275:          @ 4,38 SAY '┬'
  276:          @ 4,52 SAY '┬'
  277:          @ 9,26 SAY '┼'
  278:          @ 9,38 SAY '┼'
  279:          @ 9,52 SAY '┼'
  280:          @ 42,26 SAY '┴'
  281:          @ 42,38 SAY '┴'
  282:          @ 42,52 SAY '┴'
  283:          @ 6,17 SAY 'ESCALA' &fscampn
  284:          @ 5,28 SAY ' SUELDO ' &fscampc
  285:          @ 7,28 SAY 'ANTERIOR' &fscampc
  286:          @ 5,42 SAY 'SUELDO' &fscampc
  287:          @ 7,42 SAY 'ACTUAL' &fscampc
  288:          @ 6,58 SAY 'PUNTAJE ' &fscampc
  289:          DO cuadro1
  290:
  291:          @ 44,1 CLEAR TO 46,98
  292:          @ 44,18 SAY '- DESEA ACTUALIZAR LA TABLA : '
&fscampc
```

```
293:
294:          op=0
295:          @ 44.5,50 PROMPT '« |<No '
296:          @ 44.5,58 PROMPT '« |<Si '
297:          MENU TO op
298:        ┌─IF op=0 .OR. op=1
299:        │    DEACTIVATE WINDOW procesos
300:  <═════╪═══RETURN
301:        └─ENDIF
302:
303:          STORE 0 TO incx
304:        ╔═DO WHILE .T.
305:        ║    @ 44,1 CLEAR TO 46,78
306:        ║    @ 44,34 SAY '- INCREMENTO : ' FONT 'NORMAL',10
307:        ║    incx = 0
308:        ║    @ 44.5,48 GET incx PICT '9999999'
309:        ║    READ
310:        ║    CLEAR GETS
311:        ║  ┌─IF incx = 0
312:        ║  │    @ 44,2 CLEAR TO 46,79
313:        ║  │    @ 44,25 SAY ' Presione una tecla para
retornar..' &fscampo
314:        ║  │    READ
315:        ║  │    @ 44,2 CLEAR TO 46,89
316:  <═════╪══╪═══RETURN
317:        ║  ├─ELSE
318:        ║  │  ╔═DO WHILE .T.
319:        ║  │  ║    op=0
320:        ║  │  ║    @ 44,1 CLEAR TO 46,78
321:        ║  │  ║    @ 44,22 SAY '- Datos correctos : ' FONT
'NORMAL',12
322:        ║  │  ║    @ 44.5,42 PROMPT '« |<No »'
323:        ║  │  ║    @ 44.5,50 PROMPT '« |<Si »'
324:        ║  │  ║    MENU TO op
325:        ║  │  ║  ┌─IF op=0
326:        ║  │  ╚══╪═══LOOP
327:        ║  │     ╟─ENDIF
328:        ║  │     v═══EXIT
329:        ║  │     ╚═ENDDO
330:        ║  │  ┌─IF op = 1
331:        ╚══╪══╪═══LOOP
332:           │  └─ENDIF
333:        v══╪═══EXIT
334:           └─ENDIF
335:        ╚═ENDDO
336:
337:
338:          @ 44.1 CLEAR TO 46,89
339:          SELE esc
340:          GO TOP
341:          STORE 0 TO rx,s_ax_17,s_ax_18
342:          LOCATE FOR escala=17
343:          s_ax_17 = suel_act
344:          LOCATE FOR escala=18
345:          s_ax_18 = suel_act
346:
347:        ╔═SCAN
348:        ║    REPLA suel_ant WITH suel_act
```

```
 349:          ENDSCAN
 350:
 351:          LOCATE FOR escala=4
 352:          REPLACE suel_act WITH suel_ant + incx
 353:          rx = suel_act / puntaje
 354:          SKIP
 355:
 356:          DO WHILE .NOT. EOF()
 357:             REPLA suel_act WITH puntaje * rx
 358:             SKIP
 359:          ENDDO
 360:
 361:          STORE 0 TO s_act_e4, s_act_e17, s_act_e18
 362:          SELE esc
 363:          LOCATE FOR escala=4
 364:          REPLACE suel_ant WITH suel_act
 365:          LOCATE FOR escala=17
 366:          REPLACE suel_ant WITH s_ax_17
 367:          LOCATE FOR escala=18
 368:          REPLACE suel_ant WITH s_ax_18
 369:          DO cuadro1
 370:
 371:          @ 11,41 GET s_act_e4  PICT '99999999999' FONT
'BOLD',9
 372:          @ 37,41 GET s_act_e17 PICT '99999999999' FONT
'BOLD',9
 373:          @ 39,41 GET s_act_e18 PICT '99999999999' FONT
'BOLD',9
 374:          READ
 375:          CLEAR GETS
 376:
 377:          LOCATE FOR escala=4
 378:          REPLACE suel_act WITH s_act_e4
 379:          LOCATE FOR escala=17
 380:          REPLACE suel_act WITH s_act_e17
 381:          LOCATE FOR escala=18
 382:          REPLACE suel_act WITH s_act_e18
 383:          @ 44,2 CLEAR TO 46,89
 384:
 385:          @ 44,25 SAY '* Actualizando sueldos de personal..'
FONT 'NORMAL',11
 386:          STORE 0 TO escx, suelx,px
 387:
 388:          DO WHILE .T.
 389:             op=0
 390:             @ 44,1 CLEAR TO 46,78
 391:             @ 44,17 SAY '- Actualizar estos datos en
personal : ' FONT 'NORMAL',12
 392:             @ 44.5,42 PROMPT '« \<No »'
 393:             @ 44.5,50 PROMPT '« \<Si »'
 394:             MENU TO op
 395:             IF op=0
 396:          =====EXIT
 397:             ENDIF
 398:
 399:             SELE esc
 400:             GO TOP
 401:             DO WHILE .NOT. EOF()
```

```
 402:                escx = escala
 403:                suelx = suel_act
 404:                px = (pgres_act/100) * suelx
 405:                px = ROUND(px,0)
 406:                REPLACE vgres_act WITH px
 407:                SELECT dgp
 408:                GO TOP
 409:             DO WHILE .NOT. EOF()
 410:                IF categor = escx .AND. estado=1
 411:                   REPLACE sueldo WITH suelx
 412:                   IF num_jer = 2 OR num_jer = 3
 413:                      REPLACE gast_res WITH px
 414:                   ENDIF
 415:                ENDIF
 416:                SKIP
 417:             ENDDO
 418:             SELECT esc
 419:             SKIP
 420:          ENDDO
 421:          EXIT
 422:          ENDDO
 423:
 424:          @ 44,2 CLEAR TO 46,89
 425:          @ 44,24 SAY '* Presione una tecla para retornar..
' FONT 'NORMAL',10
 426:          READ
 427:          @ 44,2 CLEAR TO 46,89
 428:          DEACTIVATE WINDOW procesos
 429:          RETURN
 430:
 431:          ***** ACTUALIZACION DE GASTO DE REPRESENTACION
********
 432:
 433:
*!*************************************************************************
**************
 434:    *!
 435:    *!       Procedure: ACT_G_REP
 436:    *!
 437:    *!       Called by: _RAA1732TG()        (function in
AT_ESCLS.SPR)
 438:    *!
 439:    *!           Calls: CUADRO2             (procedure in
AT_ESCLS.SPR)
 440:    *!
 441:
*!*************************************************************************
**************
 442:      PROCEDURE act_g_rep
 443:          SET DISP TO VGA50
 444:          ACTIVATE WINDOW procesos
 445:          @ 4,15 TO 9,70
 446:          @ 9,15 TO 42,70
 447:          @ 4,15 TO 42,26
 448:          @ 4,26 TO 42,38
 449:          @ 4,38 TO 42,52
 450:          @ 4,52 TO 42,70
 451:          @ 1,16 SAY '- ACTUALIZACION DE GASTOS DE
```

```
REPRESENTACION - ' &fscampc
  452:          @ 9,15 SAY '├'
  453:          @ 9,70 SAY '┤'
  454:          @ 4,26 SAY '┬'
  455:          @ 4,38 SAY '┬'
  456:          @ 4,52 SAY '┬'
  457:          @ 9,26 SAY '┴'
  458:          @ 9,38 SAY '┴'
  459:          @ 9,52 SAY '┴'
  460:          @ 42,26 SAY '┴'
  461:          @ 42,38 SAY '┴'
  462:          @ 42,52 SAY '┴'
  463:          @ 6,17 SAY 'ESCALA' &fscampn
  464:          @ 5,28 SAY 'G.REPRE.' &fscampn
  465:          @ 7,28 SAY 'ANTERIOR' &fscampn
  466:          @ 5,42 SAY 'G.REPRE.' &fscampn
  467:          @ 7,42 SAY 'ACTUAL' &fscampn
  468:          @ 6.58 SAY 'PUNTAJE ' &fscampn
  469:          DO cuadro2
  470:
  471:          @ 44,1 CLEAR TO 44,78
  472:          @ 44,18 SAY '- DESEA ACTUALIZAR LA TABLA : '
&fscampc
  473:          op=0
  474:          @ 44.5,50 PROMPT '« |<No »'
  475:          @ 44.5,58 PROMPT '« |<Si »'
  476:          MENU TO op
  477:          IF op=0 .OR. op=1
  478:              DEACTIVATE WINDOW procesos
  479: <=========RETURN
  480:          ENDIF
  481:
  482:          SCAN
  483:              REPLA vgrep_ant WITH vgrep_act
  494:          ENDSCAN
  485:          DO cuadro2
  486:          STORE 11 TO fil
  487:
  488:          FOR bx = 4 TO 18
  489:              LOCATE FOR escala=bx
  490:              REPLA vgrep_act WITH 0
  491:              @ fil,41 GET vgrep_act PICT '99999999999' FONT
'ECLD',9
  492:              READ
  493:              CLEAR GETS
  494:              @ fil,41 CLEAR TO fil+1,50
  495:              @ fil,41 SAY vgrep_act PICT '99999999999'
&fscampn
  496:              fil=fil+2
  497:          ENDFOR
  498:
  499:
  500:          STORE 0 TO g_repx,escx
  501:          DO WHILE .T.
  502:              op=0
  503:              @ 44,1 CLEAR TO 46,78
  504:              @ 44,17 SAY '- Actualizar estos datos en
personal : ' FONT 'NORMAL',12
```

```
  505:          @ 44.5,42 PROMPT '« \<No »'
  506:          @ 44.5,50 PROMPT '« \<Si »'
  507:          MENU TO op
  508:          IF op=0
  509:          EXIT
  510:          ENDIF
  511:
  512:          SELE esc
  513:          GO TOP
  514:          DO WHILE .NOT. EOF()
  515:             escx = escala
  516:             g_repx = vgrep_act
  517:             SELECT dgp
  518:             GO TOP
  519:             DO WHILE .NOT. EOF()
  520:                IF categor = escx .AND. estado=1
  521:                   IF num_jer = 1 OR num_jer = 3
  522:                      REPLA gast_rep WITH g_repx
  523:                   ENDIF
  524:                ENDIF
  525:                SKIP
  526:             ENDDO
  527:             SELECT esc
  528:             SKIP
  529:          ENDDO
  530:       ENDDO
  531:       @ 44,2 CLEAR TO 46,79
  532:       @ 44,25 SAY '* Presione una tecla para retornar..
' &fscampc
  533:       READ
  534:       @ 44,2 CLEAR TO 46,79
  535:       DEACTIVATE WINDOW procesos
  536: <     RETURN
  537:
  538:
  539:
  540:       *********   ACTUALIZACION DE GASTOS DE
RESPONSABILIDAD *******
  541:
  542:
*!*************************************************************************
**************
  543:  *!
  544:  *!       Procedure: ACT_G_RES
  545:  *!
  546:  *!       Called by: _RAA1732TG()        (function in
AT_ESCLS.SPR)
  547:  *!
  548:  *!            Calls: CUADRO3            (procedure in
AT_ESCLS.SPR)
  549:  *!
  550:
*!*************************************************************************
**************
  551:     PROCEDURE act_g_res          &&  Actualizar Gasto
Responsabilidad
  552:       SET DISP TO VGA50
  553:       ACTIVATE WINDOW procesos
```

```
554:          STORE 0 TO bx, rx
555:          @ 1,1 CLEAR TO 44,39
556:          @ 4,2 TO 9,87
557:          @ 9,2 TO 42,87
558:          @ 4,10 TO 42,22
559:          @ 4,22 TO 42,33
560:          @ 4,33 TO 42,44
561:          @ 4,44 TO 42,59
562:          @ 5,74 TO 41,74
563:          @ 1,21 SAY '- ACTUALIZACION GASTOS RESPONSABILIDAD
- '&fscampc
564:          @ 9,2  SAY '├'
565:          @ 9,87 SAY '┤'
566:          @ 4,10 SAY '┬'
567:          @ 4,22 SAY '┬'
568:          @ 4,33 SAY '┬'
569:          @ 4,44 SAY '┬'
570:          @ 4,59 SAY '┬'
571:          @ 4,74 SAY '┬'
572:          @ 9,10 SAY '┼'
573:          @ 9,22 SAY '┼'
574:          @ 9,33 SAY '┼'
575:          @ 9,44 SAY '┼'
576:          @ 9,59 SAY '┼'
577:          @ 9,74 SAY '┼'
578:          @ 42,10 SAY '┴'
579:          @ 42,22 SAY '┴'
580:          @ 42,33 SAY '┴'
581:          @ 42,44 SAY '┴'
582:          @ 42,59 SAY '┴'
583:          @ 42,74 SAY '┴'
584:          @ 6,3 SAY 'ESCALA' &fscampn
585:          @ 6,13 SAY 'SUELDO' &fscampn
586:          @ 5,23 SAY '% G. RESP.' &fscampn
587:          @ 7,23 SAY ' ANTERIOR' &fscampn
588:          @ 5,34 SAY '% G. RESP.' &fscampn
589:          @ 7,35 SAY 'ACTUAL' &fscampn
590:          @ 5,46 SAY 'GASTO RESP.' &fscampn
591:          @ 7,47 SAY ' ANTERIOR' &fscampn
592:          @ 5,61 SAY 'GASTO RESP.' &fscampn
593:          @ 7,62 SAY '  ACTUAL' &fscampn
594:          @ 6,77 SAY 'PUNTAJE' &fscampn
595:          DO cuadro3
596:
597:          @ 44,1 CLEAR TO 44,78
598:          @ 44,18 SAY '- DESEA ACTUALIZAR LA TABLA : '
&fscampc
599:          op=0
600:          @ 44,5,50 PROMPT '« I<No »'
601:          @ 44,5,58 PROMPT '« I<Si »'
602:          MENU TO op
603:        ┌─IF op=0 .OR. op=1
604:        │    DEACTIVATE WINDOW procesos
605:  <═════════RETURN
606:        └─ENDIF
607:          @ 43,2 CLEAR TO 46,80
608:
609:        ╔═SCAN
```

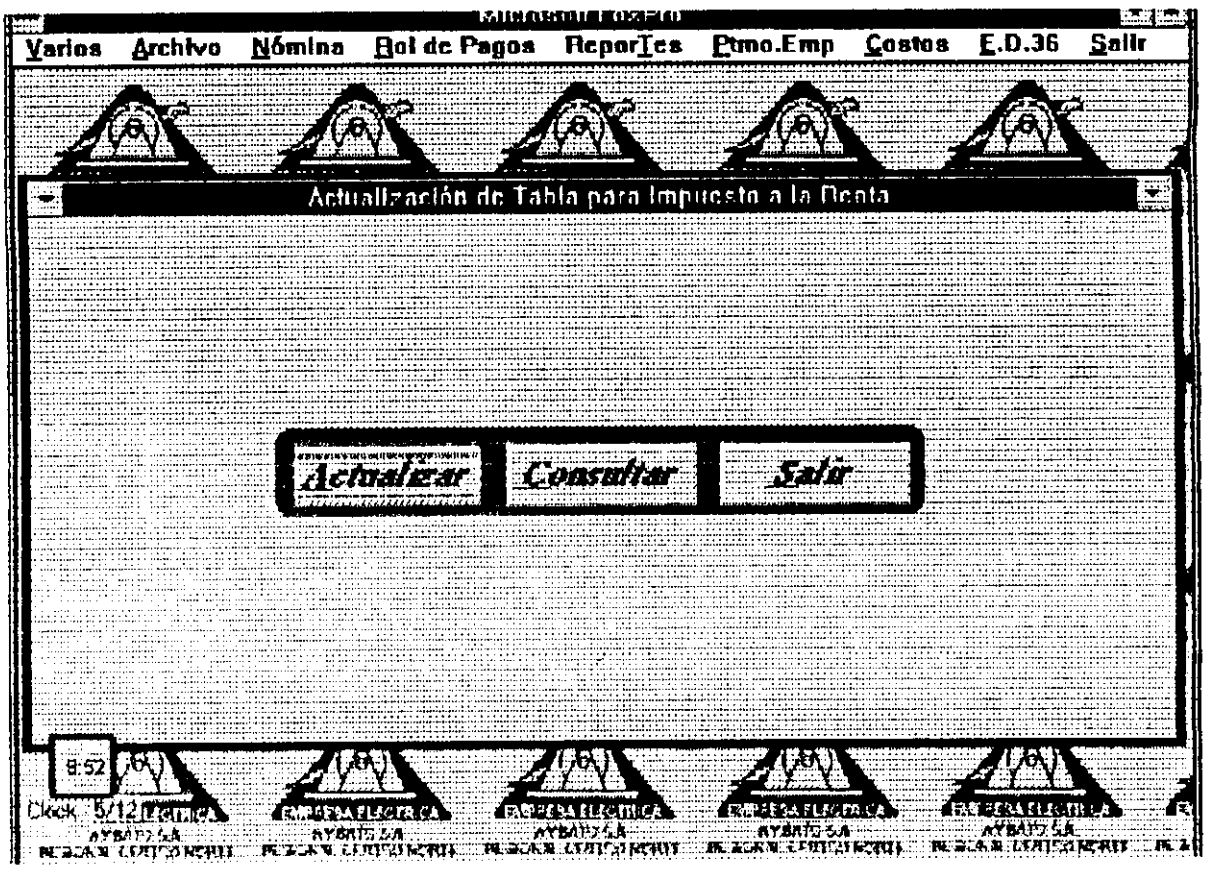
```
   610:          ||    REPLA pgres_ant WITH pgres_act
   611:          ||    REPLA vgres_ant WITH vgres_act
   612:          ||ENDSCAN
   613:          ||DO cuadro3
   614:          ||
   615:          ||STORE 13 TO fil
   616:          ||FOR bx = 4 TO 16
   617:          ||    LOCATE FOR escala=bx
   618:          ||    REPLA pgres_act WITH 0
   619:          ||    @ fil,36 CLEAR TO fil+1,41
   620:          ||    @ fil,36 GET pgres_act PICT '999.99' FONT
'BOLD',9
   621:          ||    READ
   622:          ||    CLEAR GETS
   623:          ||    @ fil,36 CLEAR TO fil+1,41
   624:          ||    @ fil,36 SAY pgres_act PICT '999.99' &fscampn
   625:          ||    rx = (pgres_act/100) * suel_act
   626:          ||    rx = ROUND(rx,0)
   627:          ||    REPLACE vgres_act WITH rx
   628:          ||    @ fil,62 CLEAR TO fil+1,73
   629:          ||    @ fil,62 SAY vgres_act PICT '9999999999'
&fscampn
   630:          ||    fil = fil+2
   631:          ||ENDFOR
   632:          ||
   633:          ||
   634:          ||STORE 0 TO v_gr_4, v_gr_17, v_gr_18
   635:          ||@ 11,62 CLEAR TO 12,73
   636:          ||@ 11,62 GET v_gr_4  PICT '9999999999' FONT
'BOLD',9
   637:          ||@ 37,62 CLEAR TO 38,73
   638:          ||@ 37,62 GET v_gr_17 PICT '9999999999' FONT
'BOLD',9
   639:          ||@ 39,62 CLEAR TO 40,73
   640:          ||@ 39,62 GET v_gr_18 PICT '9999999999' FONT
'BOLD',9
   641:          ||READ
   642:          ||CLEAR GETS
   643:          ||LOCATE FOR escala=4
   644:          ||REPLACE vgres_act WITH v_gr_4
   645:          ||LOCATE FOR escala=17
   646:          ||REPLACE vgres_act WITH v_gr_17
   647:          ||LOCATE FOR escala=18
   648:          ||REPLACE vgres_act WITH v_gr_18
   649:          ||@ 11,62 CLEAR TO 12,73
   650:          ||@ 37,62 CLEAR TO 40,73
   651:          ||DO cuadro3
   652:          ||
   653:          ||STORE 0 TO g_resx,escx
   654:          ||DO WHILE .T.
   655:          ||    op=0
   656:          ||    @ 44,1 CLEAR TO 46,78
   657:          ||    @ 44,17 SAY '- Actualizar estos datos en
personal : ' FONT 'NORMAL',12
   658:          ||    @ 44.5,42 PROMPT '« |<No »'
   659:          ||    @ 44.5,50 PROMPT '« |<Si »'
   660:          ||    MENU TO op
   661:          ||  IF op=0
```

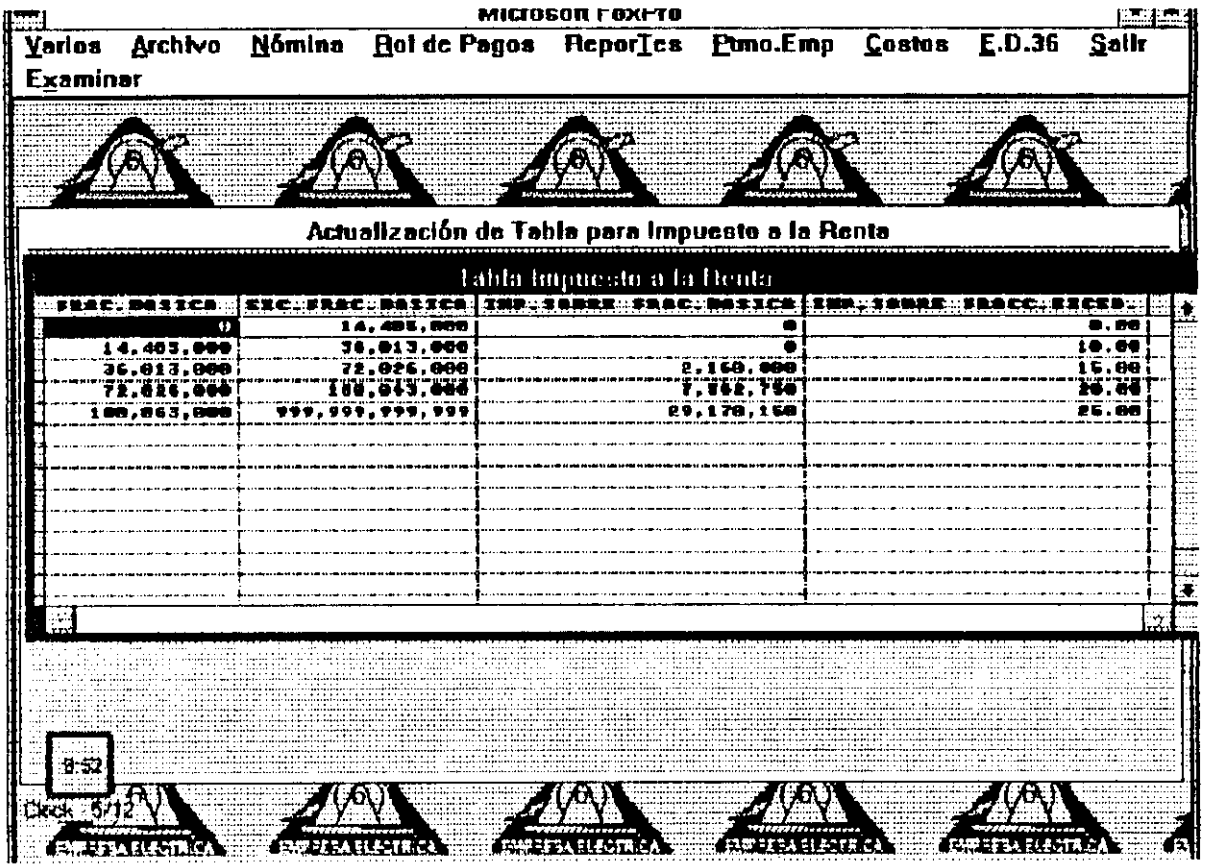
```
662:          v=======EXIT
663:          ||    └─ENDIF
664:          ||
665:          ||      SELE esc
666:          ||      GO TOP
667:          ||    ┌─DO WHILE .NOT. EOF()
668:          ||    │   escx = escala
669:          ||    │   g_resx = vgres_act
670:          ||    │   SELECT dgp
671:          ||    │   GO TOP
672:          ||    │ ┌─DO WHILE .NOT. EOF()
673:          ||    │ │   IF categor = escx .AND. estado=1
674:          ||    │ │     IF num_jer = 2 OR num_jer = 3
675:          ||    │ │       REPLA gast_res WITH g_resx
676:          ||    │ │     ENDIF
677:          ||    │ │   ENDIF
678:          ||    │ │   SKIP
679:          ||    │ └─ENDDO
680:          ||    │   SELECT esc
681:          ||    │   SKIP
682:          ||    └─ENDDO
683:          ||  └─ENDDO
684:          ||
685:          ||    @ 44,2 CLEAR TO 46,80
686:          ||    @ 44,25 SAY '* Presione una tecla para retornar..
' &fscampc
687:          ||    READ
688:          ||    @ 44,2 CLEAR TO 46,80
689:          ||    DEACTIVATE WINDOW procesos
690:          ||
691:          c=======RETURN
692:
693:
694:
*!*************************************************************************
**************
695:    *!
696:    *!       Procedure: CUADRO1
697:    *!
698:    *!       Called by: ACT_SUELDO          (procedure in
AT_ESCLS.SPR)
699:    *!
700:
*!*************************************************************************
**************
701:        PROCEDURE cuadro1
702:            SET DISP TO VGA50
703:
704:            SELE esc
705:            GO TOP
706:            F=11
707:          ┌─DO WHILE .NOT. EOF()
708:          │   @ F,20 SAY escala PICT '999' &fscampc
709:          │   @ F,28 SAY suel_ant PICT '9999999999' &fscampn
710:          │   @ F,40 SAY suel_act PICT '9999999999' &fscampn
711:          │   @ F,61 SAY puntaje  PICT '9999' &fscampn
712:          │   F=F+2
713:          │   SKIP
```

```
  714:         ENDDO
  715:  <=======RETURN
  716:
  717:
  718:
*!*************************************************************************
**************
  719:   *!
  720:   *!       Procedure: CUADRO2
  721:   *!
  722:   *!       Called by: ACT_G_REP          (procedure in
AT_ESCLS.SPR)
  723:   *!
  724:
*!*************************************************************************
**************
  725:      PROCEDURE cuadro2
  726:          SET DISP TO VGA50
  727:
  728:          SELE esc
  729:          GO TOP
  730:          F=11
  731:          DO WHILE .NOT. EOF()
  732:             @ F,20 SAY escala PICT '999' &fscampn
  733:             @ F,26 SAY vgrep_ant PICT '9999999999' &fscampn
  734:             @ F,40 SAY vgrep_act PICT '9999999999' &fscampn
  735:             @ F,61 SAY puntaje  PICT '9999' &fscampn
  736:             F=F+2
  737:             SKIP
  738:          ENDDO
  739:  <=======RETURN
  740:
  741:
  742:
*!*************************************************************************
**************
  743:   *!
  744:   *!       Procedure: CUADRO3
  745:   *!
  746:   *!       Called by: ACT_G_RES          (procedure in
AT_ESCLS.SPR)
  747:   *!
  748:
*!*************************************************************************
**************
  749:      PROCEDURE cuadro3
  750:          SET DISP TO VGA50
  751:          SELE esc
  752:          GO TOP
  753:          F=11
  754:          DO WHILE .NOT. EOF()
  755:             @ F,6  SAY escala PICT '99' &fscampn
  756:             @ F,12 SAY suel_act  PICT '9999999999' &fscampn
  757:             @ F,25 SAY pgres_ant PICT '999.99' &fscampn
  758:             @ F,36 SAY pgres_act PICT '999.99' &fscampn
  759:             @ F,46 SAY vgres_ant PICT '9999999999' &fscampn
  760:             @ F,62 SAY vgres_act PICT '9999999999' &fscampn
  761:             @ F,80 SAY puntaje  PICT '9999' &fscampn
```

```
  762:          ||    F=F+2
  763:          ||    SKIP
  764:          |ENDDO
  765: <========RETURN
  766:          ****************
  767:
*!*************************************************************************
**************
  768:  *!
  769:  *!       Procedure: SALIR
  770:  *!
  771:  *!       Called by: _RAA1732TG()        (function in
AT_ESCLS.SPR)
  772:  *!                : DA_ASISG.PRG
  773:  *!                : DE_EGRQ1.PRG
  774:  *!                : DE_EGRQ2.PRG
  775:  *!                : DE_EGRMN.PRG
  776:  *!                : DE_IEJUB.PRG
  777:  *!
  778:  *!           Calls: DEAC_KEYS           (procedure in
AACT_SST.SPR)
  779:  *!
  780:
*!*************************************************************************
**************
  781:      PROCEDURE salir
  782:          ****************
  783:          RELE WIND consulta
  784:          RELE WIND ven
  785:          RELE WIND procesos
  786:          CLEAR READ
  787: <========RETURN
  788:
  789:
  790:
  791:          *
************************************************************
  792:          *
  793:          *        * _RAA1732TG           opc_scr VALID
  794:          *        *
  795:          *        * Function Origin:
  796:          *        *
  797:          *        * From Platform:       Windows
  798:          *        * From Screen:         AT_ESCLS,
Record Number:    5
  799:          *        * Variable:            opc_scr
  800:          *        * Called By:           VALID Clause
  801:          *        * Object Type:         Push Button
  802:          *        * Snippet Number:      1
  803:          *        *
  804:          *
************************************************************
  805:          *
  806:
*!*************************************************************************
**************
  807:  *!
  808:  *!       Function: _RAA1732TG
```

```
809:   *!
810:   *!        Called by: AT_ESCLS.SPR
811:   *!
812:   *!             Calls: CON_ESCALA          (procedure in
AT_ESCLS.SPR)
813:   *!                  : ACT_SUELDO          (procedure in
AT_ESCLS.SPR)
814:   *!                  : ACT_G_RES           (procedure in
AT_ESCLS.SPR)
815:   *!                  : ACT_G_REP           (procedure in
AT_ESCLS.SPR)
816:   *!                  : SALIR               (procedure in
CTRL_RES.SPR)
817:   *!
818:
*!*************************************************************************
**************
819:      FUNCTION _raa1732tg      &&  opc_scr VALID
820:          #REGION 1
821:        1=DO CASE
822:        à═══CASE opc_scr=1
823:        │       DO con_escala
824:        à═══CASE opc_scr=2
825:        │       DO act_sueldo
826:        à═══CASE opc_scr=3
827:        │       DO act_g_res
828:        à═══CASE opc_scr=4
829:        │       DO act_g_rep
830:        à═══CASE opc_scr=5
831:        │       DO salir
832:        E=ENDCASE
833:   *:  EOF: AT_ESCLS.acl
```


Varios Archivo Nómina Rol de Pagos ReporTes Ptmo.Emp Costos E.D.36 Salir
Actualización de Tabla para Impuesto a la Renta
Actualizar
Consultar
Salir
8:52


MICROSOR FOXPTO
Varios Archivo Nómina Rol de Pagos Reportes Pmo.Emp Costos E.D.36 Salir
Examinar
Actualización de Tabla para Impuesto a la Renta
Tabla Impuesto a la Renta
FRAC. BASICA
EXC. FRAC. BASICA
IMP. SOBRE FRAC. BASICA
IMP. SOBRE FRACC. EXCED.
0 14,405,000 0 0.00
14,405,000 36,013,000 0 10.00
36,013,000 72,026,000 2,160,000 15.00
72,026,000 108,043,000 7,562,750 20.00
108,063,000 999,999,999,999 29,170,150 25.00
8:52
Click 5/12

```
    1:
*:*************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\AT_IMPRT.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-03-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 09/08/1996 at 19:26:16
    8:  *:
    9:  *:  Procs & Fncts: _RAA15NV7L()
   10:  *:                : ACTUALIZAR
   11:  *:                : CONSULTAR
   12:  *:                : ACTUALIZA
   13:  *:
   14:  *:          Set by: EMPRESA.MPR
   15:  *:
   16:  *:           Calls: _RAA15NV7L()        (function in
AT_IMPRT.SPR)
   17:  *:
   18:  *:       Documented 19:14:40
FoxDoc version 3.00a
   19:
*:*************************************************************
**************
   20:      *
************************************************************
   21:      *        *
   22:      *        * 09/05/1996           AT_IMPRT.SPR
 19:26:16
   23:      *        *
   24:      *
************************************************************
   25:      *        *
   26:      *        * Nombre del autor
   27:      *        *
   28:      *        * Copyright (c) 1995 Nombre de la empresa
   29:      *        * Direcci¥n
   30:      *        * Ciudad,      C.P.
   31:      *        * Pa¥s
   32:      *        *
   33:      *        * Description:
   34:      *        * Este programa lo ha generado autom®ticamente
GENSCRN.
   35:      *        *
   36:      *
************************************************************
   37:
   38:
   39:      *
************************************************************
   40:      *        *
   41:      *        *  AT_IMPRT/Windows C¢digo de configuraci¢n -
SECCION 1
   42:      *        *
   43:      *
************************************************************
```

```
44:      *
45:
46:      #REGION 1
47:      ************
48:      REGIONAL opc_scr
49:      STOR ''   TO opc_scr
50:
51:      **********
52:
53:      #REGION 0
54:      REGIONAL m.currarea, m.talkstat, m.compstat
55:
56:      IF SET("TALK") = "ON"
57:          SET TALK OFF
58:          m.talkstat = "ON"
59:      ELSE
60:          m.talkstat = "OFF"
61:      ENDIF
62:      m.compstat = SET("COMPATIBLE")
63:      SET COMPATIBLE FOXPLUS
64:
65:      m.rborder = SET("READBORDER")
66:      SET readborder ON
67:
68:      m.currarea = SELECT()
69:
70:
71:      *
************************************************************
72:      *        *
73:      *        *  AT_IMPRT/Windows Bases de datos, Yndices,
relaciones
74:      *        *
75:      *
************************************************************
76:      *
77:
78:      IF USED("ipr")
79:          SELECT ipr
80:          SET ORDER TO 0
81:      ELSE
82:          SELECT 0
83:          USE (LOCFILE("sroll\dbfs\imprenta.dbf","DBF","Buscar
imprenta?"));
84:              AGAIN ALIAS ipr :
85:              ORDER 0
86:      ENDIF
87:
88:
89:      *
************************************************************
90:      *        *
91:      *        *              Windows Definiciones de ventana
92:      *        *
93:      *
************************************************************
94:      *
95:
```

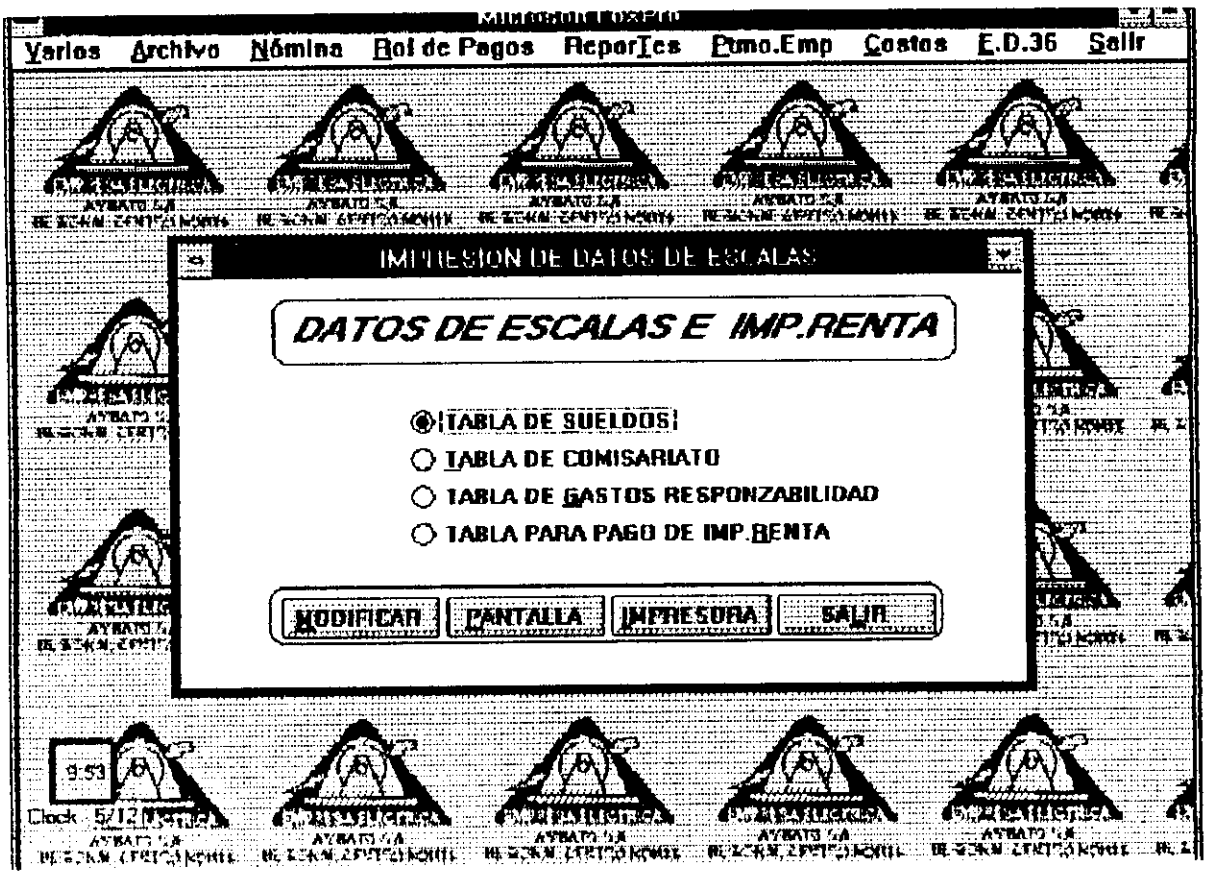
```
 96:     IF NOT WEXIST("at_imprt") :
 97:          OR UPPER(WTITLE("AT_IMPRT")) == "AT_IMPRT.PJX" :
 98:          OR UPPER(WTITLE("AT_IMPRT")) == "AT_IMPRT.SCX" :
 99:          OR UPPER(WTITLE("AT_IMPRT")) == "AT_IMPRT.MNX" :
100:          OR UPPER(WTITLE("AT_IMPRT")) == "AT_IMPRT.PRG" :
101:          OR UPPER(WTITLE("AT_IMPRT")) == "AT_IMPRT.FRX" :
102:          OR UPPER(WTITLE("AT_IMPRT")) == "AT_IMPRT.QPR"
103:        DEFINE WINDOW at_imprt ;
104:          AT  0.000, 0.000  ;
105:          SIZE 22.692,122.800 ;
106:          TITLE "Actualizaci¥n de Tabla para Impuesto a la
Renta" ;
107:          FONT "MS Sans Serif", 8 ;
108:          FLOAT ;
109:          NOCLOSE ;
110:          MINIMIZE ;
111:          SYSTEM ;
112:          COLOR RGB(,,,192,192,192)
113:        MOVE WINDOW at_imprt CENTER
114:     ENDIF
115:
116:
117:     *
************************************************************
118:     *        *
119:     *        *  AT_IMPRT/Windows C¾digo de configuraci¾n -
SECCION 2
120:     *        *
121:     *
************************************************************
122:     *
123:
124:     #REGION 1
125:     **********
126:     DEFI WIND imprenta FROM 15.2 TO 35.90 TITL ' Tabla
Impuesto a la Renta ':
127:        MDI SYSTEM IN DESKTOP
128:     MOVE WIND imprenta CENTER
129:
130:     *
************************************************************
131:     *        *
132:     *        *        AT_IMPRT/Windows Distribuci¾n de
pantalla
133:     *        *
134:     *
************************************************************
135:     *
136:
137:     #REGION 1
138:     IF WVISIBLE("at_imprt")
139:        ACTIVATE WINDOW at_imprt SAME
140:     ELSE
141:        ACTIVATE WINDOW at_imprt NOSHOW
142:     ENDIF
143:     @ 9.231,26.400 TO 13.000,96.200 ;
144:        PATTERN 1 ;
145:        PEN 1, 8 ;
```

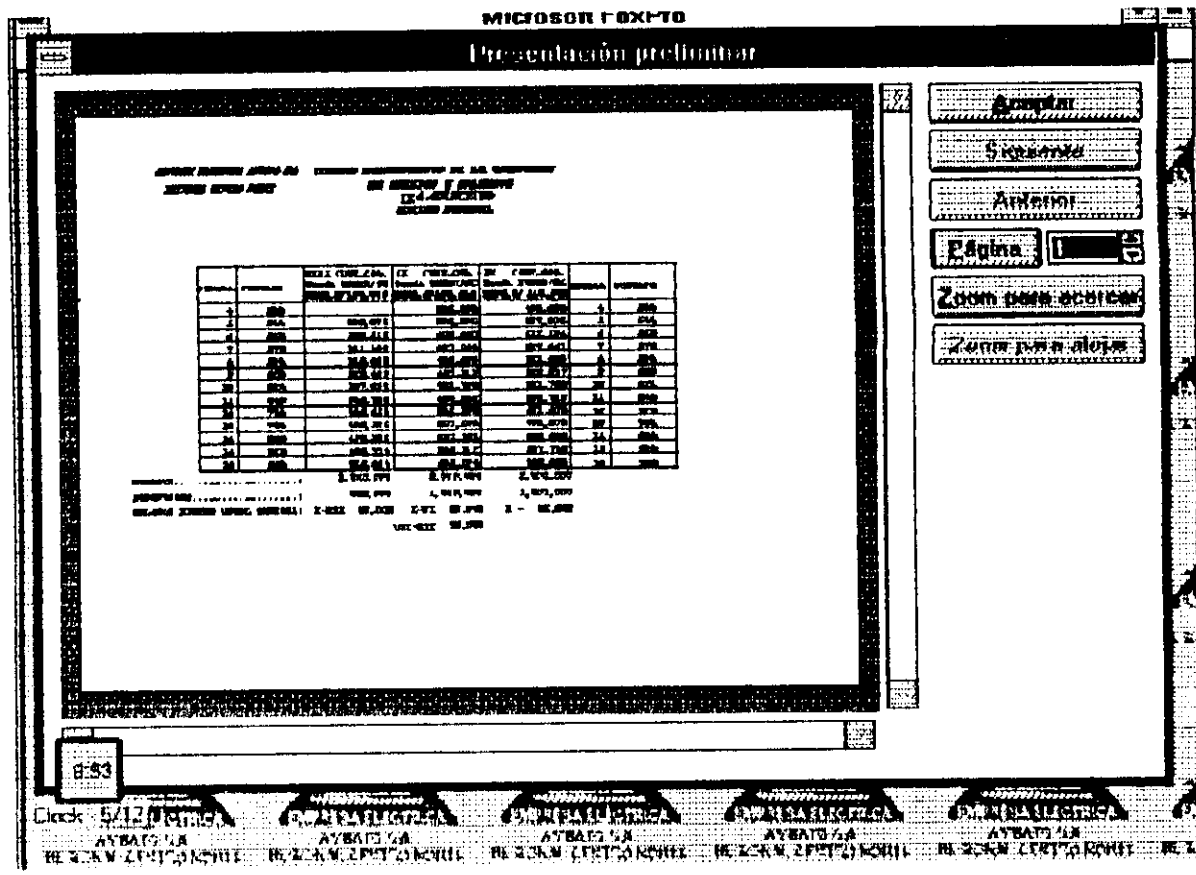
```
146:          STYLE "16" ;
147:          COLOR RGB(,,,0,128,0)
148:       @ 9.846,28.000 GET opc_scr ;
149:          PICTURE "@*HN \<Actualizar;\<Consultar;\<Salir" :
150:          SIZE 1.762,10.500,1.000 ;
151:          DEFAULT 1 ;
152:          FONT "MS Serif", 14 ;
153:          STYLE "BI" ;
154:          VALID _raa15nv7l()
155:
156:       IF NOT WVISIBLE("at_imprt")
157:          ACTIVATE WINDOW at_imprt
158:       ENDIF
159:
160:       READ CYCLE :
161:          VALID .F.
162:
163:       RELEASE WINDOW at_imprt
164:
165:       *
************************************************************
166:       *        *
167:       *        *              Windows Cerrando bases de datos
168:       *        *
169:       *
************************************************************
170:       *
171:
172:       IF USED("ipr")
173:          SELECT ipr
174:          USE
175:       ENDIF
176:
177:       SELECT (m.currarea)
178:
179:
180:       #REGION 0
181:
182:       SET readborder &rborder
183:
184:       IF m.talkstat = "ON"
185:          SET TALK ON
186:       ENDIF
187:       IF m.compstat = "ON"
188:          SET COMPATIBLE ON
189:       ENDIF
190:
191:
192:       *
************************************************************
193:       *        *
194:       *        *            AT_IMPRT/Windows Código de
limpieza
195:       *        *
196:       *
************************************************************
197:       *
198:
```

```
  199:      #REGION 1
  200:      ********************
  201:
  202:
  203:      *
*************************************************************
  204:      *        *
  205:      *        * AT_IMPRT/Windows Procedimientos y funciones
de soporte
  206:      *        *
  207:      *
*************************************************************
  208:      *
  209:
  210:      #REGION 1
  211:
*!*************************************************************************
**************
  212:  *!
  213:  *!       Procedure: ACTUALIZAR
  214:  *!
  215:  *!            Calls: ACTUALIZA           (procedure in
AT_IMPRT.SPR)
  216:  *!
  217:
*!*************************************************************************
**************
  218:      PROCEDURE actualizar
  219:          ********************
  220:          DO actualiza WITH 1
  221:  <=======RETURN
  222:
  223:          ********************
  224:
*!*************************************************************************
**************
  225:  *!
  226:  *!       Procedure: CONSULTAR
  227:  *!
  228:  *!            Calls: ACTUALIZA           (procedure in
AT_IMPRT.SPR)
  229:  *!
  230:
*!*************************************************************************
**************
  231:      PROCEDURE consultar
  232:          ********************
  233:          DO actualiza WITH 2
  234:  <=======RETURN
  235:
  236:          ***************
  237:
*!*************************************************************************
**************
  238:  *!
  239:  *!       Procedure: SALIR
  240:  *!
  241:  *!       Called by: _RAA1732TG()       (function in
```

```
AT_ESCLS.SPR)
  242:  *!                  : DA_ASISG.PRG
  243:  *!                  : DE_EGRQ1.PRG
  244:  *!                  : DE_EGRQ2.PRG
  245:  *!                  : DE_EGRMN.PRG
  246:  *!                  : DE_IEJUB.PRG
  247:  *!
  248:  *!          Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
  249:  *!
  250:
*!*****************************************************************
**************
  251:      PROCEDURE salir
  252:          ***************
  253:          RELEASE WIND impresa
  254:          CLEAR READ
  255: <=======RETURN
  256:
  257:          ******************
  258:
*!*****************************************************************
**************
  259:  *!
  260:  *!       Procedure: ACTUALIZA
  261:  *!
  262:  *!       Called by: ACTUALIZAR          (procedure in
AT_IMPRT.SPR)
  263:  *!                : CONSULTAR          (procedure in
AT_IMPRT.SPR)
  264:  *!
  265:
*!*****************************************************************
**************
  266:      PROCEDURE actualiza
  267:          ******************
  268:          PARA op
  269:          ACTI WIND impresa
  270:          STOR ' ' TO A
  271:        ┌─IF op = 2
  272:        │    STOR ' noedit ' TO A
  273:        └─ENDIF
  274:          BROW FIEL bas1: H= 'FRAC.BASICA':p =
'99,999,999,999', exe1:H='EXC.FRAC.BASICA' :p =
'999,999,999,999', :
  275:              vfb1:H = 'IMP.SOBRE FRAC.BASICA' :p =
'99,999,999,999' , pexe1:H = 'IMP.SOBRE FRACC.EXCED.':p =
'999.99';
  276:              IN WIND impresa COLO SCHE 10 NOAP NODE &a
  277:          DEACTI WIND impresa
  278: <=======RETURN
  279:
  280:          *
*****************************************************
  281:          *        *
  282:          *        * _RAA15NV7L          opc_scr VALID
  283:          *        *
  284:          *        * Function Origin:
```

```
  285:          *        *
  286:          *        * From Platform:        Windows
  287:          *        * From Screen:          AT_IMPRT,
Record Number:     4
  288:          *        * Variable:             opc_scr
  289:          *        * Called By:            VALID Clause
  290:          *        * Object Type:          Push Button
  291:          *        * Snippet Number:       1
  292:          *        *
  293:          *
************************************************************
  294:          *
  295:
*!*************************************************************************
**************
  296:   *!
  297:   *!        Function: _RAA15NV7L
  298:   *!
  299:   *!       Called by: AT_IMPRT.SPR
  300:   *!
  301:   *!            Calls: &OPC_SCR.PRG
  302:   *!
  303:
*!*************************************************************************
**************
  304:      FUNCTION _raa15nv7l      &&  opc_scr VALID
  305:           #REGION 1
  306:           DO &opc_scr
  307:   *:  EOF: AT_IMPRT.ac1
```


Varios Archivo Nómina Rol de Pagos ReporTes Ptmo.Emp Costos E.D.36 Salir
IMPRESION DE DATOS DE ESCALAS
DATOS DE ESCALAS E IMP.RENTA
TABLA DE SUELDOS
TABLA DE COMISARIATO
TABLA DE GASTOS RESPONZABILIDAD
TABLA PARA PAGO DE IMP.RENTA
MODIFICAR
PANTALLA
IMPRESORA
SALIR
9:53
Clock


MICROSOFT FOXPRO
Presentación preliminar
Anterior
Página
Zoom para acercar
Zoom para alejar
8:33

EMPRESA ELECTRICA AMBATO S.A.
REGIONAL CENTRO NORTE

CUADRO DEMOSTRATIVO DE LA VARIACION DE SUELDOS Y SALARIOS

IX C.COLECTIVO

SUELDO NOMINAL

ESCALA	PUNTAJE	VIII CONT.COL. Desde ENERO/94 INCR.S/100.000	IX CONT.COL. Desde ENERO/95 INCR.S/100.000	IX CONT.COL. Desde ENERO/96 INCR.S/ 110,000	ESCALA	PUNTAJE
4	528		382,288	492,288	4	528
5	544	293,875	393,875	507,206	5	544
6	560	302,518	405,460	522,124	6	560
7	576	311,162	417,044	537,041	7	576
8	592	319,805	428,629	551,959	8	592
9	608	328,449	440,213	566,877	9	608
10	624	337,092	451,798	581,795	10	624
11	640	345,735	463,382	596,713	11	640
12	720	388,952	521,305	671,302	12	720
13	784	423,526	567,643	730,973	13	784
14	880	475,386	637,151	820,489	14	880
15	920	496,994	666,112	857,775	15	920
16	960	518,994	695,074	895,069	16	960

GERENTE.....: 1,360,000 2,000,000 2,800,000

DIRECTORES.....: 850,000 1,300,000 1,700,000

SALARIO MINIMO VITAL GENERAL: I-XII 80,000 I-VI 86,000 I - 95,000

VII-XII 90,000

EMPRESA ELECTRICA AMBATO S.A.
REGIONAL CENTRO NORTE

BONIFICACION COMISARIATO
IX C.COLECTIVO

AÑO: 1.995 60,000 + 5 %
60,000 + 23 % No tienen cargo de Jefatura

AÑO: 1.996 70,000 + 5 %
70,000 + 23 % No tienen cargo de Jefatura

AÑO: 1.995				
ESCALA	PUNTAJE	SUELDO BASICO	60,000 BASICO	
			5 %	23 %
4	528	382,288	79,114	147,926
5	544	393,875	79,694	150,591
6	560	405,460	80,273	153,256
7	576	417,044	80,852	155,920
8	592	428,629	81,431	158,585
9	608	440,213	82,011	161,249
10	624	451,798	82,590	163,914
11	640	463,382	83,169	166,578
12	720	521,305	86,065	179,900
13	784	567,643	88,382	190,558
14	880	637,151	91,858	206,545
15	920	666,112	93,306	213,206
16	960	695,074	94,754	219,867

GERENTE: 2,000,000 160,000
DIRECTORES: 1,300,000 125,000
SAL.MINIMO VITAL GENERAL.....: 85,000

AÑO: 1.996				
ESCALA	PUNTAJE	SUELDO BASICO	70,000 BASICO	
			5 %	23 %
4	528	492,288	94,614	183,226
5	544	507,206	95,360	186,657
6	560	522,124	96,106	190,089
7	576	537,041	96,852	193,519
8	592	551,959	97,598	196,951
9	608	566,877	98,344	200,382
10	624	581,795	99,090	203,813
11	640	596,713	99,836	207,244
12	720	671,302	103,565	224,399
13	784	730,973	106,549	238,124
14	880	820,480	111,024	258,710
15	920	857,775	112,889	267,288
16	960	895,069	114,753	275,866

GERENTE: 2,600,000 200,000
DIRECTORES: 1,700,000 155,000
SAL.MINIMO VITAL GENERAL.....: 95,000

EMPRESA ELECTRICA AMBATO S.A.
REGIONAL CENTRO NORTE

GASTOS DE RESPONSABILIDAD
IX C.COLECTIVO

AÑO: 1.995				
ESCALA	PUNTAJE	SUELDO BASICO	%	VALOR
4	528	382,288	0 %	0
5	544	393,375	28 %	110,285
6	560	405,460	28 %	113,529
7	576	417,044	28 %	116,772
8	592	428,629	36 %	154,306
9	608	440,213	36 %	158,477
10	624	451,798	36 %	162,647
11	640	463,382	44 %	203,888
12	720	521,305	44 %	229,374
13	784	567,643	44 %	249,763
14	880	637,151	49 %	312,204
15	920	666,112	49 %	326,395
16	960	695,074	49 %	340,586

AÑO: 1.996				
ESCALA	PUNTAJE	SUELDO BASICO	%	VALOR
4	528	492,288	28 %	137,841
5	544	507,206	28 %	142,018
6	560	522,124	28 %	146,195
7	576	537,041	28 %	150,371
8	592	551,959	36 %	198,705
9	608	566,877	36 %	204,076
10	624	581,795	36 %	209,446
11	640	596,713	44 %	262,554
12	720	671,302	44 %	295,373
13	784	730,973	44 %	321,628
14	880	820,480	49 %	402,035
15	920	857,775	49 %	420,310
16	960	895,069	49 %	438,584

GERENTE....:	2,000,000	AÑO: 1.995	GASTOS REPRESENTACION..:	1,100,000	AÑO: 1.996	1,450,000
			GASTOS RESPONSABILIDAD.:	1,100,000		1,450,000
DIRECTORES.:	1,300,000		GASTOS REPRESENTACION..:	700,000		900,000
			GASTOS RESPONSABILIDAD.:	700,000		900,000

EMPRESA ELECTRICA AMBATO S.A.
REGIONAL CENTRO NORTE

TABLA PARA EL PAGO DEL IMPUESTO A LA RENTA

FRACCION BASICA	EXCESO SOBRE LA FRACCION BASICA - HASTA	IMPUESTO SOBRE LA FRACCION BASICA	IMPUESTO SOBRE LA FRACCION EXCEDENTE
0	14,405,000	0	+ 0 %
14,405,000	36,013,000	0	+ 10 %
36,013,000	72,026,000	2,160,800	+ 15 %
72,026,000	180,063,000	7,562,750	+ 20 %
180,063,000	999,999,999,999	29,170,150	+ 25 %

```
    1:
*:**************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\IT_IMPTB.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-03-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 10/08/1996 at  3:45:08
    8:  *:
    9:  *:  Procs & Fncts: _R6Y081JT8()
   10:  *:                : PANTALLA
   11:  *:                : IMPRESORA
   12:  *:
   13:  *:          Set by: EMPRESA.MPR
   14:  *:
   15:  *:           Calls: _R6Y081JT8()        (function in
IT_IMPTB.SPR)
   16:  *:
   17:  *:       Documented 19:14:41
FoxDoc version 3.00a
   18:
*:**************************************************************
**************
   19:      *
************************************************************
   20:      *        *
   21:      *        * 10/01/1996              TB_ESCL.SPR
 03:45:07
   22:      *        *
   23:      *
************************************************************
   24:      *        *
   25:      *        * Nombre del autor
   26:      *        *
   27:      *        * Copyright (c) 1995 Nombre de la empresa
   28:      *        * Direcci¥n
   29:      *        * Ciudad,      C.P.
   30:      *        * Pa¥s
   31:      *        *
   32:      *        * Description:
   33:      *        * Este programa lo ha generado automÆticamente
GENSCRN.
   34:      *        *
   35:      *
************************************************************
   36:
   37:
   38:      *
************************************************************
   39:      *        *
   40:      *        *   TB_ESCL/Windows C¥digo de configuraci¥n -
SECCION 1
   41:      *        *
   42:      *
************************************************************
   43:      *
```

```
   44:
   45:      #REGION 1
   46:      REGIONAL
opc_gen,opc_tbl,msuel_antd,msuel_antg,msuel_actd, msuel_actg,;
   47:         mgres_antg, mgres_antd, mgres_actg, mgres_actd,;
   48:         mgrep_antg, mgrep_antd, mgrep_actg, mgrep_actd
   49:      SET DECIMALS TO 0
   50:      STOR '' TO opc_gen
   51:      STOR 1 TO opc_tbl
   52:      STOR 0 TO msuel_antd,msuel_antg,msuel_actd, msuel_actg
   53:
   54:
   55:      #REGION 0
   56:      REGIONAL m.currarea, m.talkstat, m.compstat
   57:
   58:     IF SET("TALK") = "ON"
   59:        SET TALK OFF
   60:        m.talkstat = "ON"
   61:     ELSE
   62:        m.talkstat = "OFF"
   63:     ENDIF
   64:      m.compstat = SET("COMPATIBLE")
   65:      SET COMPATIBLE FOXPLUS
   66:
   67:      m.rborder = SET("READBORDER")
   68:      SET readborder ON
   69:
   70:      m.currarea = SELECT()
   71:
   72:
   73:      *
*************************************************************
   74:      *        *
   75:      *        *   TR_ESCL/Windows Bases de datos, Yndices,
relaciones
   76:      *        *
   77:      *
*************************************************************
   78:      *
   79:
   80:     IF USED("imprenta")
   81:        SELECT imprenta
   82:        SET ORDER TO 0
   83:     ELSE
   84:        SELECT 0
   85:        USE (LOCFILE("dbfs\imprenta.dbf","DBF","Buscar
imprenta?")):
   86:           AGAIN ALIAS imprenta :
   87:           ORDER 0
   88:     ENDIF
   89:
   90:     IF USED("ley_cont")
   91:        SELECT ley_cont
   92:        SET ORDER TO 0
   93:     ELSE
   94:        SELECT 0
   95:        USE (LOCFILE("dbfs\ley_cont.dbf","DBF","Buscar
ley_cont?")):
```

```
 96:          AGAIN ALIAS ley_cont :
 97:          ORDER 0
 98:      ENDIF
 99:
100:      IF USED("ley_labo")
101:          SELECT ley_labo
102:          SET ORDER TO 0
103:      ELSE
104:          SELECT 0
105:          USE (LOCFILE("dbfs\ley_labo.dbf","DBF","Buscar
ley_labo?"));
106:          AGAIN ALIAS ley_labo :
107:          ORDER 0
108:      ENDIF
109:
110:      IF USED("escala")
111:          SELECT escala
112:          SET ORDER TO 0
113:      ELSE
114:          SELECT 0
115:          USE (LOCFILE("dbfs\escala.dbf","DBF","Buscar
escala?"));
116:          AGAIN ALIAS escala :
117:          ORDER 0
118:      ENDIF
119:
120:      SELECT ley_cont
121:
122:
123:      *
*************************************************************
124:      *          *
125:      *          *              Windows Definiciones de ventana
126:      *          *
127:      *
*************************************************************
128:      *
129:
130:      IF NOT WEXIST("tb_escl") ;
131:          OR UPPER(WTITLE("TB_ESCL")) == "TB_ESCL.PJX" :
132:          OR UPPER(WTITLE("TB_ESCL")) == "TB_ESCL.SCX" :
133:          OR UPPER(WTITLE("TB_ESCL")) == "TB_ESCL.MNX" ;
134:          OR UPPER(WTITLE("TB_ESCL")) == "TB_ESCL.PRG" :
135:          OR UPPER(WTITLE("TB_ESCL")) == "TB_ESCL.FRX" :
136:          OR UPPER(WTITLE("TB_ESCL")) == "TB_ESCL.QPR"
137:          DEFINE WINDOW tb_escl ;
138:              AT  0.000, 0.000  :
139:              SIZE 17.615,90.600 ;
140:              TITLE "IMPRESION DE DATOS DE ESCALAS" :
141:              FONT "MS Sans Serif", 8 :
142:              FLOAT :
143:              NOCLOSE ;
144:              MINIMIZE :
145:              SYSTEM
146:          MOVE WINDOW tb_escl CENTER
147:      ENDIF
148:
149:
```

```
  150:      *
************************************************************
  151:      *        *
  152:      *        *   TB_ESCL/Windows C¾digo de configuraci¾n -
SECCION 2
  153:      *        *
  154:      *
************************************************************
  155:      *
  156:
  157:      #REGION 1
  158:      SELE ley_cont
  159:      SET FILTER TO anio = 1995
  160:      SELE ley_labo
  161:      SET FILTER TO anio = 1995
  162:
  163:      SELE escala
  164:      SET FILTER TO VAL(anio) = 1995
  165:      LOCATE FOR escala = 17
  166:      STOR suel_ant TO msuel_antd
  167:      STOR suel_act TO msuel_actd
  168:
  169:      STOR vgres_ant TO  mgres_antd
  170:      STOR vgres_act TO  mgres_actd
  171:      STOR vgrep_ant TO  mgrep_antd
  172:      STOR vgrep_act TO  mgrep_actd
  173:      *?MGRES_ANTD, MGRES_ACTD, MGREP_ANTD, MGREP_ACTD
  174:      LOCATE FOR escala = 18
  175:      STOR suel_ant TO msuel_antg
  176:      STOR suel_act TO msuel_actg
  177:
  178:      STOR vgres_ant TO  mgres_antg
  179:      STOR vgres_act TO  mgres_actg
  180:      STOR vgrep_ant TO  mgrep_antg
  181:      STOR vgrep_act TO  mgrep_actg
  182:      *?MGRES_ANTG, MGRES_ACTG, MGREP_ANTG, MGREP_ACTG
  183:
  184:      SET FILT TO escala > 3 AND escala < 17 AND VAL(anio) =
1995
  185:
  186:
  187:      *
************************************************************
  188:      *        *
  189:      *        *        TB_ESCL/Windows Distribuci¾n de
pantalla
  190:      *        *
  191:      *
************************************************************
  192:      *
  193:
  194:      #REGION 1
  195:      IF WVISIBLE("tb_escl")
  196:          ACTIVATE WINDOW tb_escl SAME
  197:      ELSE
  198:          ACTIVATE WINDOW tb_escl NOSHOW
  199:      ENDIF
  200:      @ 1.308,11.800 SAY "DATOS DE ESCALAS E  IMP.RENTA" ;
```

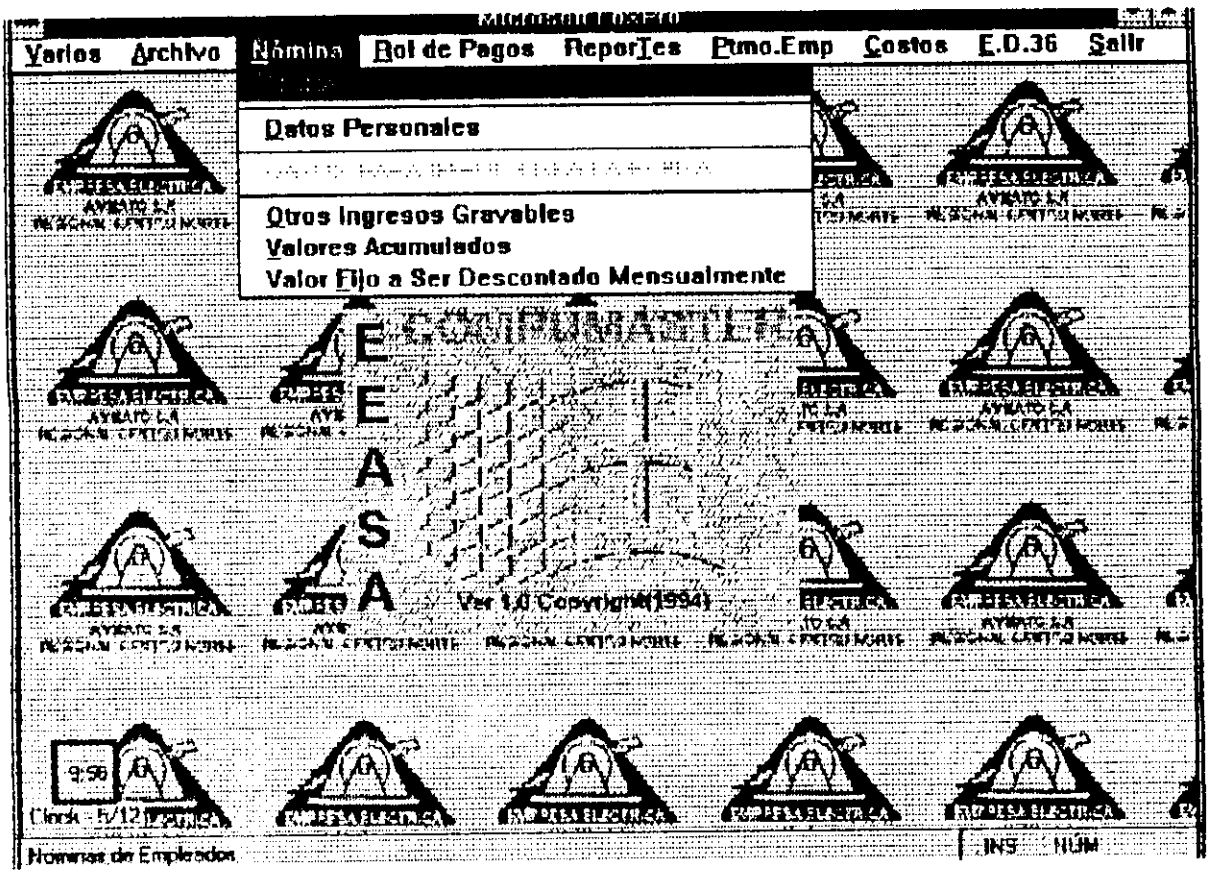
```
  201:          FONT "MS Sans Serif", 14 ;
  202:          STYLE "BIT"
  203:       @ 0.923,10.200 TO 3.769,83.800 ;
  204:          PEN 1, 8 ;
  205:          STYLE "16"
  206:       @ 5.615,25.000 GET opc_tbl ;
  207:          PICTURE "@*RVN TABLA DE |<SUELDOS;|<TABLA DE
COMISARIATO;TABLA DE |<GASTOS RESPONZABILIDAD;TABLA PARA PAGO DE
IMP.|<RENTA" ;
  208:          SIZE 1.308,43.167,0.308 ;
  209:          DEFAULT 1 ;
  210:          FONT "MS Sans Serif", 8 ;
  211:          STYLE "BT"
  212:       @ 13.846,11.000 GET opc_gen ;
  213:          PICTURE "@*HN
|<MODIFICAR;|<PANTALLA;|<IMPRESORA;SA|<LIR" ;
  214:          SIZE 1.769,14.167,0.667 ;
  215:          DEFAULT 1 ;
  216:          FONT "MS Sans Serif", 8 ;
  217:          STYLE "B" ;
  218:          VALID _r6y081jt8()
  219:       @ 13.385,9.600 TO 15.847,82.200 ;
  220:          PEN 1, 8 ;
  221:          STYLE "16"
  222:
  223:    ┌─IF NOT WVISIBLE("tb_escl")
  224:    │     ACTIVATE WINDOW tb_escl
  225:    └─ENDIF
  226:
  227:       READ CYCLE
  228:
  229:       RELEASE WINDOW tb_escl
  230:
  231:       *
**************************************************************
  232:       *        *
  233:       *        *              Windows Cerrando bases de datos
  234:       *        *
  235:       *
**************************************************************
  236:       *
  237:
  238:    ┌─IF USED("imprenta")
  239:    │     SELECT imprenta
  240:    │     USE
  241:    └─ENDIF
  242:
  243:    ┌─IF USED("ley_cont")
  244:    │     SELECT ley_cont
  245:    │     USE
  246:    └─ENDIF
  247:
  248:    ┌─IF USED("ley_labo")
  249:    │     SELECT ley_labo
  250:    │     USE
  251:    └─ENDIF
  252:
  253:    ┌─IF USED("escala")
```

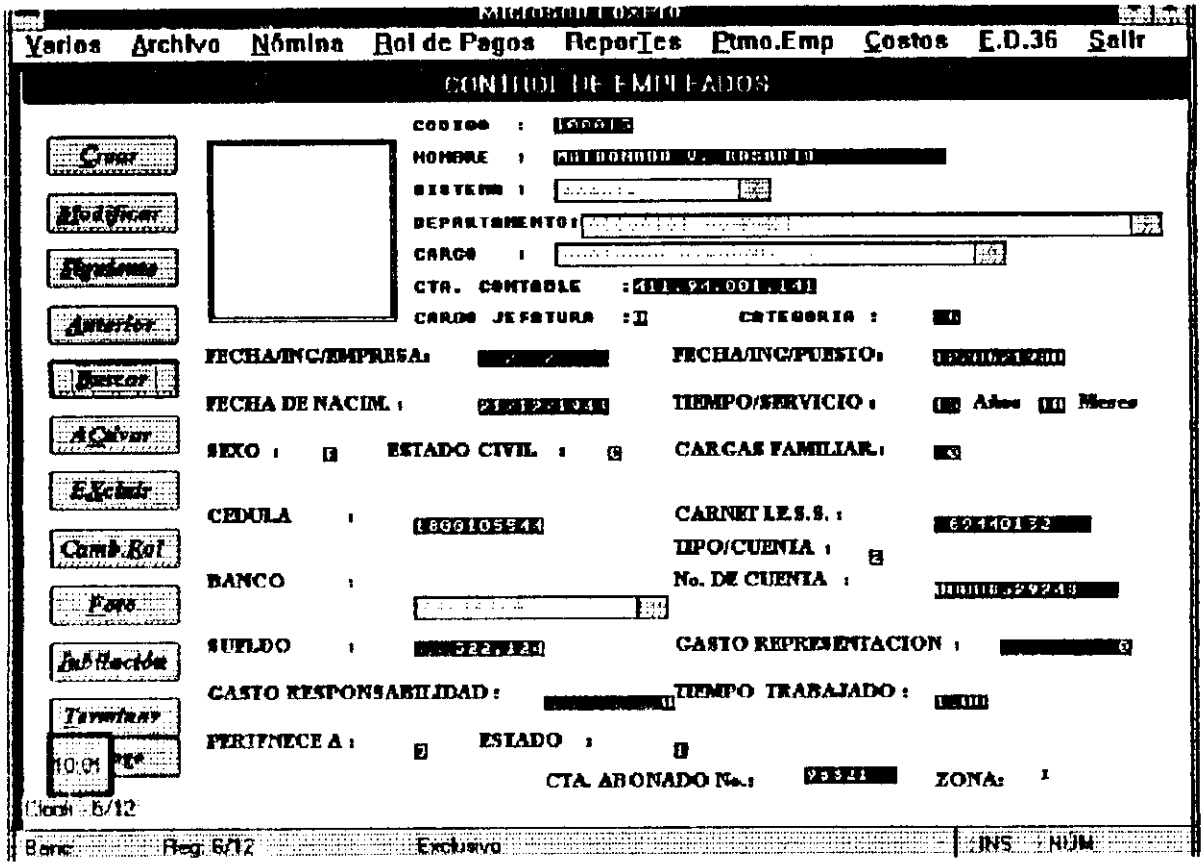
```
  254:     |    SELECT escala
  255:     |    USE
  256:     |ENDIF
  257:
  258:     SELECT (m.currarea)
  259:
  260:
  261:     #REGION 0
  262:
  263:     SET readborder &rborder
  264:
  265:     IF m.talkstat = "ON"
  266:        SET TALK ON
  267:     ENDIF
  268:     IF m.compstat = "ON"
  269:        SET COMPATIBLE ON
  270:     ENDIF
  271:
  272:
  273:     *
************************************************************
  274:     *        *
  275:     *        *           TB_ESCL/Windows Código de limpieza
  276:     *        *
  277:     *
************************************************************
  278:     *
  279:
  280:     #REGION 1
  281:     ********************
  282:
  283:
  284:     *
************************************************************
  285:     *        *
  286:     *        * TB_ESCL/Windows Procedimientos y funciones
de soporte
  287:     *        *
  288:     *
************************************************************
  289:     *
  290:
  291:     #REGION 1
  292:
*!*************************************************************************
**************
  293:  *!
  294:  *!       Procedure: MODIFICAR
  295:  *!
  296:
*!*************************************************************************
**************
  297:     PROCEDURE modificar
  298:         ********************
  299:       i=DO CASE
  300:       ³=====CASE opc_tbl = 1
  301:       ¦        MODI REPORT esc_suel
  302:       ³=====CASE opc_tbl = 2
```

```
303:          |        MODI REPORT esc_bcms
304:          à=====CASE opc_tbl = 3
305:          |        MODI REPORT esc_gast
306:          à=====CASE opc_tbl = 4
307:          |        SELE imprenta
308:          |        MODI REPORT tbl_impr
309:          |        SELE escala
310:          è=ENDCASE
311:  <=======RETURN
312:             ******************
313:
*!*************************************************************************
**************
314:    *!
315:    *!       Procedure: PANTALLA
316:    *!
317:    *!    Report Forms: ESC_SUEL.FRX
318:    *!                : ESC_BCMS.FRX
319:    *!                : ESC_GAST.FRX
320:    *!                : TBL_IMPR.FRX
321:    *!
322:
*!*************************************************************************
**************
323:       PROCEDURE pantalla
324:           ******************
325:          1=DO CASE
326:          à=====CASE opc_tbl = 1
327:          |        REPORT FORM esc_suel PREV NOCON
328:          à=====CASE opc_tbl = 2
329:          |        REPORT FORM esc_bcms PREV NOCON
330:          à=====CASE opc_tbl = 3
331:          |        REPORT FORM esc_gast PREV NOCON
332:          à=====CASE opc_tbl = 4
333:          |        SELE imprenta
334:          |        REPORT FORM tbl_impr PREV NOCON
335:          |        SELE escala
336:          è=ENDCASE
337:  <=======RETURN
338:             ******************
339:
*!*************************************************************************
**************
340:    *!
341:    *!       Procedure: IMPRESORA
342:    *!
343:    *!    Report Forms: ESC_SUEL.FRX
344:    *!                : ESC_BCMS.FRX
345:    *!                : ESC_GAST.FRX
346:    *!                : TBL_IMPR.FRX
347:    *!
348:
*!*************************************************************************
**************
349:       PROCEDURE impresora
350:           ******************
351:          1=DO CASE
352:          à=====CASE opc_tbl = 1
```

```
353:          |        REPORT FORM esc_suel TO PRINT NOCON
354:          |        *        modi repor ESC_SUEL
355:          à====CASE opc_tbl = 2
356:          |        REPORT FORM esc_bcms TO PRINT NOCON
357:          |        *        modi repor esc_bcms
358:          à====CASE opc_tbl = 3
359:          |        REPORT FORM esc_gast TO PRINT NOCON
360:          |        *        modi repor esc_gast
361:          à====CASE opc_tbl = 4
362:          |        SELE imprenta
363:          |        *        modi repor tbl_impr
364:          |        REPORT FORM tbl_impr TO PRINT NOCON
365:          |        SELE escala
366:          è=ENDCASE
367: <=======RETURN
368:           ***************
369:
*!*************************************************************************
**************
370:  *!
371:  *!       Procedure: SALIR
372:  *!
373:  *!       Called by: _RAA1732TG()        (function in
AT_ESCLS.SPR)
374:  *!                : DA_ASISG.PRG
375:  *!                : DE_EGRQ1.PRG
376:  *!                : DE_EGRQ2.PRG
377:  *!                : DE_EGRMN.PRG
378:  *!                : DE_IEJUB.PRG
379:  *!
380:  *!           Calls: DEAC_KEYS           (procedure in
AACT_SST.SPR)
381:  *!
382:
*!*************************************************************************
**************
383:      PROCEDURE salir
384:           ***************
385:          CLEAR READ
386: <========RETURN
387:
388:          *
************************************************************
389:          *        *
390:          *        * _R6Y081JT8            OPC_GEN VALID
391:          *        *
392:          *        * Function Origin:
393:          *        *
394:          *        * From Platform:        Windows
395:          *        * From Screen:          TB_ESCL,      Record
Number:     9
396:          *        * Variable:             OPC_GEN
397:          *        * Called By:            VALID Clause
398:          *        * Object Type:          Push Button
399:          *        * Snippet Number:       1
400:          *        *
401:          *
************************************************************
```

```
402:          *
403:
*!*************************************************************************
**************
404:   *!
405:   *!        Function: _R6Y081JT8
406:   *!
407:   *!       Called by: IT_IMPTB.SPR
408:   *!
409:   *!           Calls: &OPC_GEN.PRG
410:   *!
411:
*!*************************************************************************
**************
412:      FUNCTION _r6y081jt8      &&  OPC_GEN VALID
413:          #REGION 1
414:          DO &opc_gen
415:   *:  EOF: IT_IMPTB.ac1
```


Varios Archivo Nómina Rol de Pagos Reportes Pmo.Emp Costos E.D.36 Salir
Datos Personales
Otros Ingresos Gravables
Valores Acumulados
Valor Fijo a Ser Descontado Mensualmente
E
E
A
S
A
Ver 1.0 Copyright(1994)
9:56
Nombres de Empleados
INS
NUM


Varios Archivo Nómina Rol de Pagos Reportes Ptmo.Emp Costos E.D.36 Salir
CONTROL DE EMPLEADOS
Crear
Modificar
Eliminar
Anterior
Buscar
Salvar
Excluir
Camb.Rol
Fotos
Infracción
Terminar
10:01
Clock - 6/12
CODIGO :
NOMBRE :
SISTEMA :
DEPARTAMENTO:
CARGO :
CTA. CONTABLE :
CARGO JEFATURA :
CATEGORIA :
FECHA/ING/EMPRESA:
FECHA/ING/PUESTO:
FECHA DE NACIM. :
TIEMPO/SERVICIO :
Años
Meses
SEXO :
ESTADO CIVIL :
CARGAS FAMILIAR.:
CEDULA :
CARNET I.E.S.S. :
TIPO/CUENTA :
BANCO :
No. DE CUENTA :
SUELDO :
GASTO REPRESENTACION :
GASTO RESPONSABILIDAD :
TIEMPO TRABAJADO :
PERTENECE A :
ESTADO :
CTA. ABONADO No.:
ZONA:
Banc
Reg: 6/12
Exclusivo
INS
NUM

```
    1:
*:*****************************************************************
*****************
    2:  *:
    3:  *: Procedure file: C:\DOC\PERSONAL.PRG
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato -
Ecuador
    7:  *:  Last modified: 09/05/1996 at 20:11:02
    8:  *:
    9:  *:  Procs & Fncts: CABEZA
   10:  *:                : LIMPIA
   11:  *:                : SECC( )
   12:  *:                : TERMINAR( )
   13:  *:                : COMO( )
   14:  *:                : INGRESAR
   15:  *:                : CODIGO( )
   16:  *:                : COD_PERSONA
   17:  *:                : ING
   18:  *:                : DESPLEGA
   19:  *:                : VAL_SEC( )
   20:  *:                : VAL_PUE( )
   21:  *:                : VAL_BAN( )
   22:  *:                : CAMBIOS
   23:  *:                : MERCE( )
   24:  *:                : JUBILAR
   25:  *:                : ACTIVAR
   26:  *:                : EXCLUIR
   27:  *:                : CAMBIO_SIS
   28:  *:                : CAPTURAR
   29:  *:                : VAL_JEF( )
   30:  *:
   31:  *:          Set by: EMPRESA.MPR
   32:  *:
   33:  *:           Calls: TAG                    (procedure in
GRAFICOS.PRG)
   34:  *:                : ESTILOS                (procedure in
GRAFICOS.PRG)
   35:  *:                : CABEZA                 (procedure in
PERSONAL.PRG)
   36:  *:                : LIMPIA                 (procedure in
PERSONAL.PRG)
   37:  *:                : SECC( )                (function in
PERSONAL.PRG)
   38:  *:                : TERMINAR( )            (function in
PERSONAL.PRG)
   39:  *:                : COMO( )                (function in
PERSONAL.PRG)
   40:  *:
   41:  *:           Uses: SISTEMA.DBF                 Alias: SST
   42:  *:                : PUESTOS.DBF                Alias: PUES
   43:  *:                : DEPARTAM.DBF               Alias: DPTO
```

```
   44:  *:                  : PERSONAL.DBF            Alias: DGP
   45:  *:                  : SECCION.DBF             Alias: SEC
   46:  *:                  : MENSUAL.DBF             Alias: MEN
   47:  *:                  : DSCTOSVA.DBF            Alias: DVAR
   48:  *:                  : BANCOS.DBF              Alias: BANC
   49:  *:                  : ESCALA.DBF              Alias: ESCL
   50:  *:                  : LEY_CONT.DBF            Alias: LCC
   51:  *:                  : LEY_LABO.DBF            Alias: LEYL
   52:  *:                  : DHM_PERSONAL.DBF        Alias: DHM
   53:  *:
   54:  *:       CDX files: SISTEMA.CDX
   55:  *:                  : PUESTOS.CDX
   56:  *:                  : DEPARTAM.CDX
   57:  *:                  : PERSONAL.CDX
   58:  *:                  : SECCION.CDX
   59:  *:                  : MENSUAL.CDX
   60:  *:                  : DSCTOSVA.CDX
   61:  *:                  : BANCOS.CDX
   62:  *:                  : ESCALA.CDX
   63:  *:                  : LEY_CONT.CDX
   64:  *:                  : DHM_PERSONAL.CDX
   65:  *:
   66:  *:       Documented 19:14:42
FoxDoc version 3.00a
   67:
*:**********************************************************************
**************
   68:       **+-------------------------------------------- ------+
   69:       *! SISTEMA        : mantenimiento de archivos    !
   70:       *! PROGRAMA       : ACTUALIZACION DE EMPLEADOS   !
   71:       *+-------------- ---------------------------------------+
   72:       SET DISP TO VGA50
   73:       CLOSE DATA
   74:       SET STATUS BAR ON
   75:       SET DELE ON
   76:       ***** processo ****
   77:       REGIONAL
vnom,vcod,vcta,vfin,vfinemp,vfna,vcar,vced,vcar;
   78:            ,vcueb,vsueldo,vgrepre,vgrespo,ciclo,
vrsec,vciess,vcatel;
   79:            ,vrejn, vrdep,vcate,modo,k1,sistema,
mcodsis,sis1,sis2,vestado,;
   80:
vcat,vgruocu,vescala,ndep,vpue,vestciv,vsex,vtipcta,vper,vest,;
   81:
vban,vcodb,vcod_sec,vmedidor,vtiempo,como_dato,vcolor,K,A,vcodpue
,vjef,;
   82:            vtsa, vtsm, vrel_dep, vfecjub
   83:       STOR "Color W/R+/B" TO vcolor
   84:       STOR DATE() TO vfinemp,vfin,vfna,vfecjub
   85:       STOR SPAC(1) TO vpue,vestciv,vsex,vper,A
   86:       STOR SPAC(3) TO vcod_sec,vcod_dep,vrdep
   87:       STOR SPAC(2) TO vgruocu
   88:       STOR 0 TO
```

```
FGET,vcar,vced,vciess,vsueldo,vgrepre,vgrespo,vjef,vtsa,vtsm
   89:      STOR 4 TO vescala
   90:      STOR 1 TO
vtipcta,vest,como_dato,vcueb,vtiempo,vrciu,modo, vrel_dep
   91:      STOR SPAC(5) TO vcodpue
   92:      STOR SPAC(10) TO vcta,vmedidor
   93:      STOR SPAC(30) TO vnom
   94:      STOR 1 TO vzona
   95:      STOR 1 TO vcodb
   96:      SELECT A
   97:      USE sistema ALIAS sst
   98:      SET ORDER TO sist_cod
   99:      GO BOTT
  100:      STOR cod_ciu  TO sis2
  101:      GO TOP
  102:      STOR cod_ciu  TO sis1
  103:      SELECT B
  104:      USE puestos ALIAS pues
  105:      SET ORDER TO pues_des
  106:
  107:      SELECT C
  108:      USE departam ALIAS dpto
  109:      SET ORDER TO dpto_nom
  110:
  111:      SELECT D
  112:      USE personal  ALIAS dgp
  113:      SET ORDER TO cod_per
  114:      DIMENSION vartag(256)
  115:      DO TAG
  116:
  117:      SELECT E
  118:      USE seccion ALIAS sec
  119:      SET ORDER TO secc_nom
  120:
  121:      SELECT F
  122:      USE mensual ALIAS mens
  123:      SET ORDER TO mens_cod
  124:
  125:      SELECT G
  126:      USE  dsctosva ALIAS dvar
  127:      SET ORDER TO var_dsc
  128:
  129:      SELECT 17
  130:      USE bancos ALIAS banc
  131:      SET ORDER TO banc_cod
  132:
  133:      SELECT 19
  134:      USE escala ALIAS esc1
  135:      SET ORDER TO escala
  136:      SET FILTER TO VAL(esc1.anio) = 1996
  137:      SELECT 30
  138:      USE ley_cont ALIAS ley_cont
  139:      SELECT 31
  140:      USE ley_labo ALIAS ley_labo
```

```
  141:      USE dhm_personal ALIAS dhm IN 0
  142:      DO estilos
  143:      *tor "FONT "NBS",7    'to fsdesc
  && styl and font of say coment
  144:      STOR "   'TO fsdesc                                    && styl
and font of say coment*stor "  FONT "N",11 STYLE "B"' to fscampC
&& styl and font of say CHARACTER fields
  145:      *stor "  FONT "N",11 STYLE "B"' to fscampN   && styl
and font of say NUMERIC fields
  146:
  147:      DEFINE WINDOW ventana FROM 0,0 TO 50,91 TITLE "
CONTROL DE EMPLEADOS "  NOFLOAT NOGROW NOZOOM MDI SYSTEM
  148:      ACTIVATE WINDOW ventana
  149:      laso = .T.
  150:      STORE "  "  TO
m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,vestado,vcat,vsec
  151:      STOR 1 TO modo
  152:
  153:      SELECT pues          && **** Cargar Puestos***
  154:      COUNT TO r1
  155:      DIMENSION v1(r1)
  156:      STORE SPACE(30) TO v1
  157:      GO TOP
  158:    ┌─FOR i = 1 TO r1
  159:    │    STORE descrip TO v1(i)
  160:    │    SKIP
  161:    └─ENDFOR
  162:      STOR v1[1] TO vcat
  163:
  164:      SELECT sec    &&**** Cargar Seccion ***
  165:      COUNT TO r2
  166:      DIMENSION v2(r2)
  167:      STORE SPACE(40) TO v2
  168:      GO TOP
  169:    ┌─FOR i = 1 TO r2
  170:    │    STORE nombre TO v2(i)
  171:    │    SKIP
  172:    └─ENDFOR
  173:      STOR v2[1] TO vsec
  174:
  175:      SELECT sst  &&********** Cargar Sistema ****
  176:      COUNT TO r3
  177:      DIMENSION v3(r3)
  178:      STORE SPACE(12) TO v3
  179:      GO TOP
  180:    ┌─FOR i = 1 TO r3
  181:    │    STORE ciudad TO v3(i)
  182:    │    SKIP
  183:    └─ENDFOR
  184:      STOR v3[1] TO vciu
  185:
  186:      SELECT dpto  &&**************** Cargar Sistema ****
  187:      COUNT TO rd
  188:      DIMENSION vd(rd)
```

```
189:      GO TOP
190:      FOR i = 1 TO rd
191:         STORE nombre TO vd(i)
192:         SKIP
193:      ENDFOR
194:      STOR vd[1] TO ndep
195:
196:      SELECT banc &&********************* Cargar Bancos ****
197:      COUNT TO r5
198:      DIMENSION ban(r5)
199:      GO TOP
200:      FOR i = 1 TO r5
201:         STORE nom_banco TO ban(i)
202:         SKIP
203:      ENDF
204:      STOR ban(1) TO vban
205:      DIMENSION estciv(5)
206:      STORE "SOLTERO"     TO estciv(1)
207:      STORE "CASADO"      TO estciv(2)
208:      STORE "DIVORCIADO" TO estciv(3)
209:      STORE "VIUDO"       TO estciv(4)
210:      STORE "UNION LIBRE" TO estciv(5)
211:
212:      DIMENSION estcivcod(5)
213:      STORE "S" TO estcivcod(1)
214:      STORE "C" TO estcivcod(2)
215:      STORE "D" TO estcivcod(3)
216:      STORE "V" TO estcivcod(4)
217:      STORE "U" TO estcivcod(5)
218:
219:      DIMENSION tipcue(2)
220:      STORE "CORRIENTE" TO tipcue(1)
221:      STORE "AHORROS" TO tipcue(2)
222:
223:      DIMENSION perte(3)
224:      STORE "SINDICATO"      TO perte(1)
225:      STORE "ASOCIACION"     TO perte(2)
226:      STORE "INDEPENDIENTE" TO perte(3)
227:
228:      DIMENSION esta(2)
229:      STORE "ACTIVO" TO esta(1)
230:      STORE "PASIVO" TO esta(2)
231:      STOR esta(1) TO vesta
232:
233:      @ 2.14 TO 13.38 DOUBLE
234:      DO cabeza
235:      DO limpia
236:      STORE SPACE (6) TO vcod
237:      STOR -1 TO fila
238:      STOR ' SIZE 1.3.14 FONT "R",10 STYLE "BIS" ' TO
stilobot
239:      @39.02 GET como_dato PICT '@^ \<Codigo;\<Apellido'
SIZE 1.10 DISAB &vcolor
240:      DO WHILE laso
```

```
241:        @ fila+3,2 GET m1;
242:            PICTURE "@*N \<Crear " ;
243:            &stilobot;
244:            VALID secc(1)
245:        @ fila+6.5,2 GET m2;
246:            PICTURE "@*N \<Modificar " ;
247:            &stilobot;
248:            VALID secc(2)
249:        @ fila+10,2 GET m3;
250:            PICTURE "@*N \<Siguiente " ;
251:            &stilobot;
252:            VALID secc(3)
253:        @ fila+13.5,2 GET m4;
254:            PICTURE "@*N \<Anterior " ;
255:            &stilobot;
256:            VALID secc(4)
257:        @ fila+17,2 GET m5;
258:            PICTURE "@*N \<Buscar " ;
259:            &stilobot;
260:            VALID secc(5)
261:        @ fila+20.5,2 GET m6;
262:            PICTURE "@*N A\<Ctivar " ;
263:            &stilobot;
264:            VALID secc(6)
265:        @ fila+24,2 GET m7;
266:            PICTURE "@*N E\<Xcluir " ;
267:            &stilobot;
268:            VALID secc(7)
269:        @ fila+27.5,2 GET m8;
270:            PICTURE "@*N Camb.\<Rol " ;
271:            &stilobot;
272:            VALID secc(8)
273:        @ fila+31,2 GET m9;
274:            PICTURE "@*N \<Foto " ;
275:            &stilobot;
276:            VALID secc(9)
277:        @ fila+34.5,2 GET m10;
278:            PICTURE "@*N \<Jubilaci≤n " ;
279:            &stilobot;
280:            VALID secc(10)
281:        @ fila+38,2 GET m11;
282:            PICTURE "@*N \<Terminar " ;
283:            &stilobot;
284:            VALID terminar()
285:        @ fila+40.5,2 GET m12;
286:            PICTURE "@*N Cla\<ve" ;
287:            &stilobot;
288:            VALID como()
289:        READ CYCLE
290:    ENDDO
291:    SET DELE ON
292:    SHOW MENU _msysmenu
293:    RETURN
294:
```

```
295:
*!*************************************************************************
**************
296:  *!
297:  *!       Procedure: INGRESAR
298:  *!
299:  *!       Called by: SECC()              (function in
PERSONAL.PRG)
300:  *!
301:  *!           Calls: CODIGO()            (function in
PERSONAL.PRG)
302:  *!                : DESPLEGA            (procedure in
PERSONAL.PRG)
303:  *!                : ING                 (procedure in
PERSONAL.PRG)
304:  *!
305:
*!*************************************************************************
**************
306:      PROCEDURE ingresar
307:          STOR SPAC (6) TO vcod
308:          @ 1,41 GET vcod PICT "######" VALID codigo()
ERROR 'Cσdigo No Vβlido' &vcolor
309:          READ
310:        ┌─IF LASTKEY() = 27
311: <══════└═RETURN
312:        └─ENDIF
313:          SELECT dgp
314:
315:          SET ORDER TO cod_per
316:        ┌─IF modo = 2
317:        │     LOCATE FOR cod_per = vcod
318:        ├─ELSE
319:        │     SEEK vcod
320:        └─ENDIF
321:
322:        ┌─IF FOUND()
323:        │     WAIT "Registro Ya Existe....!" WINDOW TIME 1
324:        │     DO despliega
325:        ├─ELSE
326:        │     DO ing WITH 0
327:        └─ENDIF
328: <══════RETURN
329:
330:
*!*************************************************************************
**************
331:  *!
332:  *!        Function: CODIGO
333:  *!
334:  *!       Called by: INGRESAR            (procedure in
PERSONAL.PRG)
335:  *!
336:
```

```
*!*************************************************************************
**************
  337:      FUNCTION codigo
  338:        ┌─IF  (vcod = SPACE(6))
  339: <═══════╪═RETUR .F.
  340:        └─ENDIF
  341:          SELE sst
  342:          GO TOP
  343:          SEEK VAL(SUBS(vcod,1,1))
  344:        ┌─IF FOUND()
  345:        │   ┌─IF tipo  = 2
  346:        │   │    STOR 'disa' TO vestado
  347:        │   │    @ 41,42 GET vesta FROM esta  PICTURE "@^"
COLOR SCHEME 4  DISAB
  348:        │   │    STOR 3 TO vest
  349:        │   ├─ELSE
  350:        │   │    STOR 1 TO vest
  351:        │   │    STOR 'Enable' TO vestado
  352:        │   └─ENDI
  353:        │     STOR ciudad TO vciu
  354:        ├─ELSE
  355: <═══════╪═RETURN .F.
  356:        └─ENDIF
  357:          STOR 'enable ' TO vestado
  358:          @4,41  GET vciu FROM v3 PICTURE "@^" COLOR SCHEME
4 DISA
  359:          SELE sec
  360:          SET FILTER TO
  361:          SET FILTER TO cod_ciu = VAL(SUBS(vcod,1,1))
  362:          SELE dgp
  363: <═══════RETURN .T.
  364:
  365:
*!*************************************************************************
**************
  366:  *!
  367:  *!       Procedure: COD_PERSONA
  368:  *!
  369:  *!       Called by: ACTIVAR              (procedure in
PERSONAL.PRG)
  370:  *!                : EXCLUIR              (procedure in
PERSONAL.PRG)
  371:  *!                : CAMBIO_SIS           (procedure in
PERSONAL.PRG)
  372:  *!
  373:  *!           Calls: DESPLEGA             (procedure in
PERSONAL.PRG)
  374:  *!
  375:
*!*************************************************************************
**************
  376:      PROCEDURE cod_persona
  377:        ┌─IF vcod = SPACE(6)
  378:        │  ╔═DO WHILE .T.
```

```
  379:          |  ||     @ 1,41 GET vcod PICT "######" VALID vcod <>
SPACE(6)
  380:          |  ||     READ
  381:          |  ||   ┌─IF LASTKEY() = 27
  382:          |  ||   │     vcod = SPACE(6)
  383: <═══════════════════RETURN
  384:          |  ||   └─ENDIF
  385:          |  ||     SELECT dgp
  386:          |  ||     SET ORDER TO cod_per
  387:          |  ||     SEEK vcod
  388:          |  ||   ┌─IF FOUND()
  389:          |  ||   │     DO desplega
  390:          |  ||   v═══EXIT
  391:          |  ||   ├─ELSE
  392:          |  ||   │     WAIT "Registro No Existe....!" WINDOW
  393:          |  ||   └─ENDIF
  394:          |  ||═══ENDDO
  395:          └─ENDIF
  396:            SELECT dgp
  397:            SET ORDER TO cod_per
  398:          ┌─IF modo = 2
  399:          │     LOCATE FOR cod_per = vcod
  400:          ├─ELSE
  401:          │     SEEK vcod
  402:          └─ENDIF
  403: <═══════RETURN
  404:
  405:
*!*************************************************************************
**************
  406:  *!
  407:  *!       Procedure: ING
  408:  *!
  409:  *!       Called by: INGRESAR          (procedure in
PERSONAL.PRG)
  410:  *!                : CAMBIOS           (procedure in
PERSONAL.PRG)
  411:  *!
  412:  *!           Calls: LIMPIA            (procedure in
PERSONAL.PRG)
  413:  *!                : VAL_SEC()         (function in
PERSONAL.PRG)
  414:  *!                : VAL_FUE()         (function in
PERSONAL.PRG)
  415:  *!                : VAL_JEF()         (function in
PERSONAL.PRG)
  416:  *!                : VAL_BAN()         (function in
PERSONAL.PRG)
  417:  *!                : GRABAR            (procedure in
AACT_SST.SPR)
  418:  *!                : CANCELAR          (procedure in
AACT_SST.SPR)
  419:  *!
  420:
```

```
*!*************************************************************
**************
  421:      PROCEDURE ing
  422:           PARAMETERS sw
  423:           CLEAR GETS
  424:           STORE .T. TO ciclo
  425:           DO WHILE ciclo
  426:              IF sw <> 1
  427:                 DO limpia
  428:              ENDIF
  429:              STOR 0 TO FGET
  430:              @ FGET+3,     41 GET vnom  PICT "@!" VALID
ALLT(vnom) <> SPAC(1) &vcolor
  431:              *  @ fget+6.5, 42 Get Ndep From Vd Picture "@^"
Color Scheme 4 disa
  432:              @ FGET+6.2,   43 GET vsec FROM v2 PICTURE "@^"
COLOR SCHEME 4  VALID val_sec()
  433:              @ FGET+8,     41 GET vpue FROM v1 PICTURE "@^"
COLOR SCHEME 4  VALID val_pue()
  434:              @ FGET+11,    47 GET vcta DISABLE &vcolor
  435:              @ FGET+13,    47 GET vjef PICT '9' VALID
val_jef(vjef) ERROR '[0]Ninguno [1]Gast.Rep [2]Gast.Res [3]Los
Dos'&vcolor
  436:              @ FGET+13,    70 GET vescala pict'99' DISABLE
&vcolor
  437:
  438:              @ FGET+15.5, 35 GET vfinemp PICT "@D" &vcolor
  439:              @ FGET+15.5, 70 GET vfin PICT "@D" &vcolor
  440:              @ FGET+18.5, 35 GET vfna PICT "@D" &vcolor
  441:              *  @ fget+19, 65 Get VestCiv Picture "!" Color
Scheme 4:
  442:              VALID vestciv $'SCVDU' ;
  443:              ERROR 'El Estado Civil puede ser: (S)=Soltero,
(C)=Casado, (D)=Divorciado, (V)=Viudo, (U)=Unión Libre'
  444:              @ FGET+18.5, 70 GET vtsa PICTURE "99" COLOR
SCHEME 4:
  445:              WHEN .F.
  446:              @ FGET+18.5, 78 GET vtsm PICTURE "99" COLOR
SCHEME 4:
  447:              WHEN .F.
  448:              *  @ fget+22, 30  Get Vsex Picture "!" valid
Vsex$'MF' Color Scheme 4 ERROR 'El Sexo puede ser: (F) = Femenino
o (M) = Masculino'
  449:              @ FGET+21.5, 23  GET vsex PICTURE "!" VALID
vsex$'MF' COLOR SCHEME 4 ERROR 'El Sexo puede ser: (F) = Femenino
o (M) = Masculino'
  450:              @ FGET+21.5, 45 GET vestciv PICTURE "!" COLOR
SCHEME 4:
  451:              VALID vestciv $'SCVDU' :
  452:              ERROR 'El Estado Civil puede ser: (S)=Soltero,
(C)=Casado, (D)=Divorciado, (V)=Viudo, (U)=Unión Libre'
  453:              @ FGET+21.5,70 GET vcar PICT "99" &vcolor
  454:              @ FGET+26, 30 GET vced PICT "9999999999"
&vcolor
```

```
  455:          ||     @ FGET+26, 70 GET vciess PICT "##############"
&vcolor
  456:          ||     @ FGET+30, 30 GET vban FROM ban PICT '@^' COLOR
SCHEME 4  VALID val_ban()
  457:          ||     @ FGET+28, 65 GET vtipcta PICTURE "9" COLOR
SCHEME 4:
  458:          ||     VALID vtipcta >0 AND vtipcta <= 2 ;
  459:          ||     ERROR 'El Tipo de Cuenta puede ser: (1)=
Corriente, (2) = Ahorros '
  460:          ||     @ FGET+30, 70 GET vcueb PICT "################"
&vcolor
  461:          ||     @ FGET+33.5, 30 GET vsueldo PICT "99,999,999"
&vcolor
  462:          ||     @ FGET+33.5, 75 GET vgrepre PICT "99,999,999"
&vcolor
  463:          ||     @ FGET+37,   40 GET vgrespo PICT "99,999,999"
&vcolor
  464:          ||     @ FGET+37,   70 GET vtiempo PICT "9.99" &vcolor
  465:          ||     @ FGET+40, 30 GET vper PICT '9' COLOR SCHEME 4
;
  466:          ||     VALID vper $ '123';
  467:          ||     ERROR 'Puede Ser: (1)=Sindicato, (2) =
Asociaci¢n, (3) = Independiente '
  468:          ||    ┌─IF sst.tipo != 2 OR vest != 3
  469:          ||    │    @ FGET+40, 50 GET vest PICT '9';
  470:          ||    │    VALID vest > 0 AND vest < 4 ;
  471:          ||    │    ERROR 'El Tipo Puede Ser: (1) = Activo, (2)
= Pasivo ' &&vestado
  472:          ||    ├─ELSE
  473:          ||    │    STOR 3 TO vest
  474:          ||    └─ENDIF
  475:          ||     *  @ fget+40, 70 Get  VEscala pict'99' Disable
&Vcolor
  476:          ||     *  @ fget+41.5, 30 Get  VGruocu  Disable
&Vcolor
  477:          ||     @ FGET+41.5, 60 GET  vmedidor PICT '9999999'
&vcolor
  478:          ||     @ FGET+41.5, 78 GET  vzona PICT '9' VALID vzona
= 1 OR vzona = 2 ERROR 'La Zona Puede Ser: <1> Urbana <2> Rural '
&vcolor
  479:          ||
  480:          ||     STORE SPACE(1) TO op1
  481:          ||     @ 30,2 GET op1;
  482:          ||     PICTURE "@* \<Grabar " ;
  483:          ||     &stilobot;
  484:          ||     &vcolor;
  485:          ||     VALID grabar(sw)
  486:          ||
  487:          ||     @ 33.5,2 GET op1;
  488:          ||     PICTURE "@* \<Cancelar " ;
  489:          ||     &stilobot;
  490:          ||     &vcolor;
  491:          ||     VALID cancelar()
  492:          ||
```

```
493:            *SHOW GET M9,1 PROMPT "\<Grabar " ENABLE
494:            *SHOW GET M10,2 PROMPT "\<Cancelar " ENABLE
495:
496:            READ CYCLE
497:
498:         ENDDO
499:         SHOW GET m9 ENABLE
500:         SHOW GET m10 ENABLE
501:  <======RETURN
502:
503:
*!*************************************************************************
**************
504:  *!
505:  *!       Procedure: CANCELAR
506:  *!
507:  *!       Called by: ING                (procedure in
PERSONAL.PRG)
508:  *!                : JUBILAR            (procedure in
PERSONAL.PRG)
509:  *!
510:  *!          Calls: MSCATTER            (procedure in
AACT_SST.SPR)
511:  *!                 : ACT_KEYS          (procedure in
AACT_SST.SPR)
512:  *!                 : DESPLEGA          (procedure in
PERSONAL.PRG)
513:  *!                 : BORRA_TMP         (procedure in
MO_PDLAB.SPR)
514:  *!                 : ASIGNAM           (procedure in
MO_PDLAB.SPR)
515:  *!                 : REF_SCR           (procedure in
MO_PDLAB.SPR)
516:  *!
517:
*!*************************************************************************
**************
518:     FUNCTION cancelar
519:        SELE dep
520:        LOCATE FOR cod_per = vcod
521:        IF !FOUND()
522:           IF NOT EOF()
523:              SKIP
524:           ELSE
525:              SKIP - 1
526:           ENDIF
527:        ENDIF
528:        DO desplega
529:        STOR .F. TO ciclo
530:  <=====RETURN
531:
532:
*!*************************************************************************
**************
```

```
  533:  *!
  534:  *!        Procedure: DESPLEGA
  535:  *!
  536:  *!        Called by: BUSCAR              (procedure in
AACT_SST.SPR)
  537:  *!                 : CANCELAR            (procedure in
AACT_SST.SPR)
  538:  *!                 : ANTERIOR            (procedure in
AP_PRMCT.SPR)
  539:  *!                 : SIGUIENTE           (procedure in
AP_PRMCT.SPR)
  540:  *!                 : INGRESAR            (procedure in
PERSONAL.PRG)
  541:  *!                 : COD_PERSONA         (procedure in
PERSONAL.PRG)
  542:  *!                 : CAMBIOS             (procedure in
PERSONAL.PRG)
  543:  *!                 : ACTIVAR             (procedure in
PERSONAL.PRG)
  544:  *!                 : EXCLUIR             (procedure in
PERSONAL.PRG)
  545:  *!                 : CAMBIO_SIS          (procedure in
PERSONAL.PRG)
  546:  *!                 : CAPTURAR            (procedure in
PERSONAL.PRG)
  547:  *!                 : UNO                 (procedure in
ING_GRAV.PRG)
  548:  *!                 : DOS                 (procedure in
ING_GRAV.PRG)
  549:  *!
  550:  *!           Calls: VAL_SEC()           (function in
PERSONAL.PRG)
  551:  *!                : VAL_PUE()           (function in
PERSONAL.PRG)
  552:  *!
  553:
*!*************************************************************
**************
  554:      PROCEDURE despliega
  555:          STOR 0 TO FGET
  556:          SELECT dgp
  557:
  558:          vcod = cod_per
  559:          vnom = nombre
  560:          vpue = cargo
  561:          vreiu = num_eiu
  562:          vcod_sec = num_sec
  563:          vcodpue = cod_pues
  564:          vrel_dep = rel_dep
  565:          vrdep = num_dep
  566:          vcta = cuent_cont
  567:          vjef = num_jer
  568:          vfinemp = fec_ingpto
  569:          vfin = fec_ing
```

```
570:          vfna = fec_nac
571:          vfecjub = fec_jub
572:          vtsa = INT(tiempo_ser / 12)
573:          vtsm = tiempo_ser - (vtsa*12)
574:          vestciv = est_civ
575:          vsex =sexo
576:          vcar = num_car
577:          vciess = car_iess
578:          vcodb = cod_bco
579:          vtipcta = tipo_cts
580:          vcueb = num_cts
581:          vest = estado
582:          vsueldo = sueldo
583:          vgrepre = gast_rep
584:          vgrespo = gast_res
585:          vtiempo = tiempo
586:          vescala = categor
587:          vper    =  sindicato
588:          vced    = cedula
589:          vmedidor = medidor
590:          vzona    = zona
591:          @3,15 SAY foto SIZE 10,13 ISOM
592:          @ 1,41  SAY vcod &vcolor
593:          @ 3,41  SAY vnom  PICT "@!" &vcolor
594:          *  @ 41,65 say Vescala
595:          SELECT sec
596:          LOCATE FOR vcod_sec = cod_sec
597:          STOR nombre TO vsec
598:          @ FGET+6,2.  43 GET vsec FROM v2 PICTURE "@^"
COLOR SCHEME 4  VALID val_sec() DISAB
599:          @ FGET+8,    41 GET  vpue FROM v1 PICTURE "@^"
COLOR SCHEME 4  VALID val_pue() DISABLE
600:
601:          SELECT sst
602:          LOCATE FOR vrciu = cod_ciu
603:          STOR ciudad TO vciu
604:          @4,41  GET vciu FROM v3 PICTURE "@^" COLOR SCHEME
4 DISA
605:
606:          @ FGET+11, 47 SAY vcta &vcolor
607:          @ FGET+13, 47 SAY vjef PICT '9' &vcolor
608:          @ FGET+13, 70 SAY  vescala pict '99' &vcolor
609:          @ FGET+15.5, 35 SAY vfinemp PICT "@D" &vcolor
610:          @ FGET+15.5, 70 SAY vfin PICT "@D" &vcolor
611:          @ FGET+18.5, 35 SAY vfna PICT "@D" &vcolor
612:          *  @ fget+19, 65 say VestCiv Picture "!" &Vcolor
613:          @ FGET+18.5, 70 SAY vtsa PICTURE "99" &vcolor
614:          @ FGET+18.5, 78 SAY vtsm PICTURE "99" &vcolor
615:          @ FGET+21.5, 23  SAY vsex PICTURE "!" &vcolor
616:          @ FGET+21.5, 45 SAY vestciv PICTURE "!" &vcolor
617:          @ FGET+21.5,70 SAY vcar PICT "99" &vcolor
618:          @ FGET+26, 30 SAY vced PICT "##########" &vcolor
619:          @ FGET+26, 70 SAY vciess PICT "##############"
&vcolor
```

```
  620:          SELE banc
  621:          LOCATE FOR vcodb = codigo
  622:          IF FOUND()
  623:             STOR nom_banco TO vban
  624:          ENDIF
  625:          @ FGET+30, 30 GET vban FROM ban PICT '@^' &vcolor
DISA
  626:          @ FGET+28, 65 SAY vtipcta PICTURE "9" &vcolor
  627:          @ FGET+30, 70 SAY vcueb PICT "################"
&vcolor
  628:          @ FGET+33.5, 30 SAY vsueldo PICT "99,999,999"
&vcolor
  629:          @ FGET+33.5, 75 SAY vgrepre PICT "99,999,999"
&vcolor
  630:          @ FGET+37,   40 SAY vgrespo PICT "99,999,999"
&vcolor
  631:          @ FGET+37,   70 SAY vtiempo PICT "9.99" &vcolor
  632:          @ FGET+40, 30 SAY vper PICT '9' &vcolor
  633:          @ FGET+40, 50 SAY vest PICT '9' &vcolor
  634:          *  @ fget+40, 70 say  VEscala pict'99' &Vcolor
  635:          *  @ fget+41.5, 30 say  VGruocu  &Vcolor
  636:          @ FGET+41.5, 60 SAY  vmedidor PICT '9999999'
&vcolor
  637:          @ FGET+41.5, 78 SAY  vzona   PICT '9' FUNC '@z'
  638: <=======RETURN
  639:
  640:          ***** VALIDA SECCIONES ****
  641:
*!*************************************************************
**************
  642:  *!
  643:  *!        Function: VAL_SEC
  644:  *!
  645:  *!      Called by: ING                (procedure in
PERSONAL.PRG)
  646:  *!                : DESPLEGA          (procedure in
PERSONAL.PRG)
  647:  *!                : JUBILAR           (procedure in
PERSONAL.PRG)
  648:  *!                : CAMBIO_SIS        (procedure in
PERSONAL.PRG)
  649:  *!
  650:
*!*************************************************************
*************
  651:      FUNCTION val_sec
  652:          SELE sec
  653:          GO TOP
  654:          LOCATE FOR ALLT(vsec) = ALLT(nombre)
  655:          IF FOUND()
  656:             STOR cuent_cont TO vcta
  657:             @ FGET+11, 47 GET vcta DISABLE &vcolor
  658:             STOR grupotra   TO vgrutra
  659:             STOR cod_dpto   TO vdep,vrdep
```

```
660:            STOR cod_ciu     TO vrciu
661:            STOR cod_sec     TO vcod_sec
662:          ELSE
663:  <=========RETURN .F.
664:          ENDIF
665:          SELE dgp
666:  <=======RETURN
667:
668:          ****************
669:
*!*************************************************************************
**************
670:  *!
671:  *!        Function: VAL_PUE
672:  *!
673:  *!      Called by: ING                (procedure in
PERSONAL.PRG)
674:  *!               : DESPLEGA           (procedure in
PERSONAL.PRG)
675:  *!               : JUBILAR            (procedure in
PERSONAL.PRG)
676:  *!               : CAMBIO_SIS         (procedure in
PERSONAL.PRG)
677:  *!
678:
*!*************************************************************************
**************
679:      FUNCTION val_pue
680:          ****************
681:          SELE pues
682:          GO TOP
683:          LOCATE FOR ALLT( descrip) = ALLT(vpue)
684:          IF FOUND()
685:             STOR escala TO vescala
686:             STOR gru_ocu TO vgruocu
687:             STOR cod_pues TO vcodpue
688:             STOR rel_dep TO vrel_dep
689:             *       Select gru_ocu
690:             *         Seek vgruocu
691:             *         If Found()
692:             *            @ fget+42, 30 say subs(Grupo,1,10)
&Vcolor
693:             *         Else
694:             *            @ fget+42, 30 say  '*No definido*'
&Vcolor
695:             *         Endif
696:             IF vescala = 3
697:                SELE ley_cont
698:                STOR (cc_jubil/100) TO por
699:                SELE ley_labo
700:                STOR salminvit * por TO vsueldo
701:                STOR 0 TO vgrespo, vgrepre
702:             ELSE
703:                IF vescala != 3
```

```
704:                    SELECT escl
705:                    LOCATE FOR escala = vescala
706:                    IF pues.nivel = '0'
707:                        STOR suel_act TO vsueldo
708:                    ELSE
709:                        STOR suel_ant TO vsueldo
710:                    ENDIF
711:                    IF pues.jefatura = 1
712:                        STOR vgres_act TO vgrespo
713:                        STOR vgrep_act TO vgrepre
714:                    ELSE
715:                        STOR 0 TO vgrespo, vgrepre
716:                    ENDIF
717:                ELSE
718:                    STOR 0 TO vsueldo, vgrespo, vgrepre
719:                ENDIF
720:            ENDIF
721:        ENDIF
722:        SHOW GET vsueldo
723:        SHOW GET vescala
724:        SELE dgp
725:  <=====RETURN
726:
727:
*!*************************************************************************
**************
728:  *!
729:  *!          Function: VAL_BAN
730:  *!
731:  *!        Called by: ING                  (procedure in
PERSONAL.PRG)
732:  *!
733:
*!*************************************************************************
**************
734:      FUNCTION val_ban
735:          SELE banc
736:          LOCATE FOR ALLT(nom_banco) = ALLT(vban)
737:          IF FOUND()
738:              STOR codigo TO vcodb
739:          ENDIF
740:  <=======RETURN .T.
741:
742:
*!*************************************************************************
**************
743:  *!
744:  *!        Procedure: CAMBIOS
745:  *!
746:  *!        Called by: SECC()               (function in
PERSONAL.PRG)
747:  *!
748:  *!            Calls: DESPLEGA             (procedure in
PERSONAL.PRG)
```

```
  749:  *!                : ING                    (procedure in
PERSONAL.PRG)
  750:  *!
  751:
*!*************************************************************************
**************
  752:      PROCEDURE cambios
  753:          IF vcod = SPACE(6)
  754:              DO WHILE .T.
  755:                  @ 1,41 GET vcod PICT "######" VALID vcod <>
SPACE(6)
  756:                  READ
  757:                  IF LASTKEY() = 27
  758:                      vcod = SPACE(6)
  759:  <===============  RETURN
  760:                  ENDIF
  761:                  SELECT dgp
  762:                  IF modo = 2
  763:                      LOCATE FOR cod_per = vcod
  764:                  ELSE
  765:                      SEEK vcod
  766:                  ENDIF
  767:
  768:                  IF FOUND()
  769:                      DO despliega
  770:              v=======EXIT
  771:                  ELSE
  772:                      WAIT "Registro No Existe....!" WINDOW
  773:                  ENDIF
  774:              ENDDO
  775:          ENDIF
  776:          SELECT dgp
  777:          IF modo = 2
  778:              LOCATE FOR cod_per = vcod
  779:          ELSE
  780:              SEEK vcod
  781:          ENDIF
  782:          DO ing WITH 1
  783:  ^=======RETURN
  784:
  785:
*!*************************************************************************
**************
  786:  *!
  787:  *!          Function: MERCE
  788:  *!
  789:  *!             Calls: GRABAR               (procedure in
AACT_SST.SPR)
  790:  *!
  791:
*!*************************************************************************
**************
  792:      FUNCTION merce
  793:          PARAMETERS um,sw
```

```
794:          IF un = 1
795:             SELECT sec
796:             GO TOP
797:             LOCATE FOR ALLT(vsec) = ALLT(nombre)
798:             vcod_sec = cod_sec
799:             vcta = cuent_cont
800:             vrdep = cod_dpto
801:             SELECT sst
802:             LOCATE FOR ALLT(ciudad) =  ALLT(vciu)
803:             vrciu = cod_ciu
804:             =grabar(sw)
805:          ENDIF
806:          * Clear Gets
807:          STORE .F. TO ciclo
808: <=======RETURN
809:
810:
*!*************************************************************************
**************
811:  *!
812:  *!       Procedure: GRABAR
813:  *!
814:  *!       Called by: ING                (procedure in
PERSONAL.PRG)
815:  *!                : MERCE()            (function in
PERSONAL.PRG)
816:  *!                : JUBILAR            (procedure in
PERSONAL.PRG)
817:  *!
818:  *!           Calls: ACT_KEYS           (procedure in
AACT_SST.SPR)
819:  *!                : ELIM_MPL           (procedure in
MO_PDLAB.SPR)
820:  *!                : INSERTAR()         (function in
MO_PDLAB.SPR)
821:  *!                : BORRA_TMP          (procedure in
MO_PDLAB.SPR)
822:  *!                : REF_SCR            (procedure in
MO_PDLAB.SPR)
823:  *!
824:  *!            Uses: TMP.DBF
825:  *!
826:
*!*************************************************************************
**************
827:      FUNCTION grabar
828:          PARAMETERS sw
829:          SELECT dgp
830:          IF sw = 0
831:             APPEND BLANK
832:          ELSE
833:             SELE dhm
834:             m.anio = ALLT(STR(YEAR(DATE())))
835:             m.mes = ALLT(STR(MONTH(DATE())))
```

```
836:            m.fecha_mod = DATE()
837:            APPEN BLANK
838:            GATHER MEMVAR MEMO
839:            SELE dgp
840:          IF modo = 2
841:              LOCATE FOR cod_per = vcod
842:          ELSE
843:              SEEK vcod
844:          ENDIF
845:          IF NOT FOUND()
846:              SET ORDER TO cod_per
847:              SEEK vcod
848:          ENDIF
849:        ENDIF
850:        REPLACE cod_per     WITH vcod
851:        REPLACE nombre      WITH vnom
852:        REPLACE cod_pues    WITH vcodpue
853:        REPLACE rel_dep     WITH vrel_dep
854:        REPLACE cargo       WITH vpue
855:        REPLACE num_ciu     WITH VAL(SUBS(vcod,1,1))
&&vrciu&&1
856:        REPLACE num_sec     WITH vcod_sec && Sec.cod_sec &&
857:        REPLACE num_dep     WITH vrdep &&Sec.cod_dpto
858:        REPLACE num_jer     WITH vjef
859:        REPLACE fec_ingpto WITH vfinemp
860:        REPLACE fec_ing     WITH vfin
861:        REPLACE fec_jub     WITH vfecjub
862:        REPLACE tiempo_ser WITH IIF(sw = 0,
INT(((DATE()-vfinemp)/361)*12), tiempo_ser)
863:        REPLACE fec_nac     WITH vfna
864:        REPLACE est_civ     WITH vestciv
865:        REPLACE sexo        WITH vsex
866:        REPLACE num_car     WITH vcar
867:        REPLACE car_iess    WITH vciess
868:        REPLACE cod_bco     WITH vcodb
869:        REPLACE tipo_cts    WITH vtipcta
870:        REPLACE num_cts     WITH vcueb
871:        REPLACE estado      WITH vest
872:        REPLACE sueldo      WITH vsueldo
873:        REPLACE gast_rep    WITH vgrepre
874:        REPLACE gast_res    WITH vgrespo
875:        REPLACE categor     WITH vescala
876:        REPLACE cuent_cont WITH vcta  && Sec.cuent_cont &&
877:        REPLACE sindicato   WITH vper
878:        REPLACE tiempo      WITH IIF(vtiempo <= 0
,1,vtiempo)
879:        REPLACE cedula      WITH vced
880:        REPLACE gru_ocu     WITH vgruocu
881:        REPLACE medidor     WITH vmedidor
882:        REPLACE zona        WITH vzona
883:        SELECT mens
884:        IF sw = 0
885:            APPEND BLANK
886:        ELSE
```

```
887:          |    SEEK vcod
888:          |ENDIF
889:           REPLACE codigo    WITH vcod
890:           REPLACE ciudad    WITH vrciu
891:           SELECT dvar
892:          |IF sw = 0
893:          |    APPEND BLANK
894:          |ELSE
895:          |    SEEK vcod
896:          |ENDIF
897:           REPLACE sistema     WITH vrciu
898:           REPLACE cta_ctble   WITH vcta
899:           REPLACE codigo      WITH vcod
900:           REPLACE cod_dep     WITH dgp.num_dep
901:           REPLACE cod_sec     WITH dgp.num_sec
902:           REPLACE jerarquia   WITH dgp.categor
903:           REPLACE cta_ctble   WITH dgp.cuent_cont
904:           sw = 1
905:           STOR .F. TO ciclo
906: <=======RETURN
907:
908:
*!*************************************************************************
**************
909:  *!
910:  *!       Procedure: LIMPIA
911:  *!
912:  *!       Called by: PERSONAL.PRG
913:  *!                : ING                    (procedure in
PERSONAL.PRG)
914:  *!
915:
*!*************************************************************************
**************
916:      PROCEDURE limpia
917:          *Store SPACE(10) to VSEX
918:          STOR SPAC(1) TO vsex
919:          STORE SPACE(13) TO vpet
920:          STORE SPACE(25) TO vcate1
921:          STORE SPACE(4) TO vrsec,vrdep
922:          STORE SPAC(30) TO vnom
923:          *tore Space(30) to Vcat
924:          STORE SPAC(40) TO vsec
925:          STORE SPACE(12) TO vciu,vciess
926:          STORE 0 TO vjef
927:          STORE SPACE(30) TO vdpt
928:          STORE SPACE(14) TO vcta
929:          STORE DATE() TO vfin
930:          STORE DATE() TO vfinemp
931:          STORE DATE() TO vfna
932:          STORE 0 TO vcar,vrciu
933:          STORE SPACE(10) TO vced
934:          STORE SPACE(12) TO vcueb
935:          STORE 0 TO vgrepre,vgrespo,vsueldo,vcate
```

```
  936:          STORE 1 TO vtiempo
  937: <=======RETURN
  938:
  939:
  940:
*!*************************************************************************
**************
  941:  *!
  942:  *!        Function: SECC
  943:  *!
  944:  *!      Called by: PERSONAL.PRG
  945:  *!               : MEDIDOR.PRG
  946:  *!               : ING_GRAV.PRG
  947:  *!               : EGR_VARI.PRG
  948:  *!
  949:  *!           Calls: INGRESAR          (procedure in
PERSONAL.PRG)
  950:  *!                : CAMBIOS           (procedure in
PERSONAL.PRG)
  951:  *!                : SIGUIENTE         (procedure in
AP_PRMCT.SPR)
  952:  *!                : ANTERIOR          (procedure in
AP_PRMCT.SPR)
  953:  *!                : BUSCAR            (procedure in
AACT_SST.SPR)
  954:  *!                : ACTIVAR           (procedure in
PERSONAL.PRG)
  955:  *!                : EXCLUIR           (procedure in
PERSONAL.PRG)
  956:  *!                : CAMBIO_SIS        (procedure in
PERSONAL.PRG)
  957:  *!                : CAPTURAR          (procedure in
PERSONAL.PRG)
  958:  *!                : JUBILAR           (procedure in
PERSONAL.PRG)
  959:  *!                : COPIA_D           (procedure in
MEDIDOR.PRG)
  960:  *!                : ACT_MEDIDORES.PRG
  961:  *!                : UNO               (procedure in
ING_GRAV.PRG)
  962:  *!                : TOTALES           (procedure in
VF_IMPRTA.SPR)
  963:  *!
  964:
*!*************************************************************************
**************
  965:      FUNCTION secc
  966:          PARAMETER K
  967:        =DO CASE
  968:        ======CASE K=1
  969:              DO ingresar
  970:        ======CASE K=2
  971:              SELE dgp
  972:              SCATTER MEMVAR MEMO
```

```
 973:          |        DO cambios
 974:          |=====CASE K=3
 975:          |        DO siguiente
 976:          |=====CASE K=4
 977:          |        DO anterior
 978:          |=====CASE K=5
 979:          |        DO buscar
 980:          |=====CASE K = 6
 981:          |        DO activar
 982:          |=====CASE K = 7
 983:          |        DO excluir
 984:          |=====CASE K = 8
 985:          |        DO cambio_sis
 986:          |=====CASE K=9
 987:          |        DO capturar
 988:          |=====CASE K=10
 989:          |        DO jubilar
 990:          |=====CASE K=11
 991:          |        RELE WIND ventana
 992:          |    .   laso=.F.
 993:  <=======|=======RETURN
 994:          L=ENDCASE
 995:  <=========RETURN
 996:
 997:
*!*************************************************************************
**************
 998:  *!
 999:  *!         Function: TERMINAR
1000:  *!
1001:  *!       Called by: PERSONAL.PRG
1002:  *!
1003:
*!*************************************************************************
**************
1004:      FUNCTION terminar
1005:          CLEAR
1006:          RELE WIND ventana
1007:          CLOSE DATABASES
1008:          laso=.F.
1009:  <=======RETURN
1010:
1011:
*!*************************************************************************
**************
1012:  *!
1013:  *!         Function: COMO
1014:  *!
1015:  *!       Called by: PERSONAL.PRG
1016:  *!
1017:
*!*************************************************************************
**************
1018:      FUNCTION como
```

```
 1019:          STOR vartag(1) TO como_dato
 1020:          @39,02 GET como_dato FROM vartag PICTURE "@^" SIZE
1,12
 1021:          READ
 1022:          SELE dgp
 1023:          SET ORDER TO como_dato
 1024: <=======RETURN
 1025:
 1026:          ******************
 1027:
*!*************************************************************************
**************
 1028:  *!
 1029:  *!       Procedure: JUBILAR
 1030:  *!
 1031:  *!       Called by: SECC()              (function in
PERSONAL.PRG)
 1032:  *!
 1033:  *!            Calls: EXCLUIR            (procedure in
PERSONAL.PRG)
 1034:  *!                 : VAL_SEC()          (function in
PERSONAL.PRG)
 1035:  *!                 : VAL_PUE()          (function in
PERSONAL.PRG)
 1036:  *!                 : GRABAR             (procedure in
AACT_SST.SPR)
 1037:  *!                 : CANCELAR           (procedure in
AACT_SST.SPR)
 1038:  *!
 1039:
*!*************************************************************************
**************
 1040:      PROCEDURE jubilar
 1041:          ******************
 1042:          DO excluir
 1043:          STOR 0 TO sw
 1044:          SELE sst
 1045:          SET ORDER TO tipo_jub
 1046:          @ FGET+6,2,  43 GET vsec FROM v2 PICTURE "@^"
COLOR SCHEME 4  VALID val_sec()
 1047:          @ FGET+8,    41 GET vpue FROM v1 PICTURE "@^"
COLOR SCHEME 4  VALID val_pue()
 1048:          @ FGET+11,   47 GET veta DISABLE &vcolor
 1049:          @ FGET+13,   70 GET vescala pict'99' DISABLE
&vcolor
 1050:          @ 15,14 SAY "FECHA/JUBILACION:"&fsdesc
 1051:          @ FGET+15,5, 35 GET vfec_jub PICT "@D" &vcolor
 1052:          @ FGET+33,5, 30 GET vsueldo PICT "99,999,999"
&vcolor
 1053:          STOR 3 TO vest
 1054:          STOR STR(vreiu,1)+RIGHT(vcod,5) TO vcod
 1055:          ?vcod, vreiu
 1056:          WAIT
 1057:          STORE SPACE(1) TO op1
```

```
1058:
1059:          @ 30,2 GET op1:
1060:             PICTURE "@* \<Grabar " :
1061:             &stilobot:
1062:             &vcolor:
1063:             VALID grabar(sw)
1064:
1065:          @ 33,5,2 GET op1:
1066:             PICTURE "@* \<Cancelar " :
1067:             &stilobot:
1068:             &vcolor:
1069:             VALID cancelar()
1070:
1071:          READ CYCLE
1072:          @ 15,14 SAY "FECHA/ING/EMPRESA:"&fsdesc
1073:          SHOW GET m9 ENABLE
1074:          SHOW GET m10 ENABLE
1075: <=======RETURN
1076:
1077:          ******************
1078:
*!*************************************************************************
**************
 1079:  *!
 1080:  *!       Procedure: ACTIVAR
 1081:  *!
 1082:  *!       Called by: SECC()              (function in
PERSONAL.PRG)
 1083:  *!
 1084:  *!           Calls: COD_PERSONA         (procedure in
PERSONAL.PRG)
 1085:  *!                : DESPLEGA            (procedure in
PERSONAL.PRG)
 1086:  *!
 1087:
*!*************************************************************************
**************
 1088:      PROCEDURE activar
 1089:          ******************
 1090:          DO cod_persona
 1091:        ┌─IF NOT WEXIST('VARIOS') && o definir en el
principal
 1092:        │     DEFINE WINDOW varios FROM 0,0 TO 7,35 TITLE
'ALERTA ..!' SYSTEM
 1093:        └─ENDIF
 1094:          ACTIVATE WINDOW varios
 1095:          MOVE WINDOW varios CENTER
 1096:        ┌─IF dgp.estado = 2
 1097:        │     @ 1,2 SAY 'Desea Activar a:'+dgp.cod_per
 1098:        │     *       @ 2,6 SAY 'Ya ha sido generado.'
 1099:        ├─ELSE
 1100:        │     @ 1,2 SAY 'No se Puede Activar a:'+dgp.cod_per
 1101:        └─ENDIF
 1102:          STOR 2 TO opc_ac
```

```
1103:          @ 4,5 GET opc_ac PICT "@*H \<ACEPTAR;\<CANCELAR":
1104:             SIZE 1.308,12.500,0.308 :
1105:             DEFAULT 1 :
1106:             FONT "MS Serif", 8 :
1107:             STYLE "BT"
1108:          READ MODAL
1109:          RELE WIND varios
1110:          IF opc_ac = 2
1111:             STOR .F. TO vret
1112:             WAIT WIND "PERSONA NO FUE ACTIVADA ...!"
1113: <========RETURN
1114:          ENDIF
1115:
1116:          REPLACE estado WITH 1
1117:          DO despliega
1118:          SELECT mens
1119:          LOCATE FOR codigo = vcod
1120:          IF FOUND()
1121:             RECALL
1122:          ELSE
1123:             APPEN BLANK
1124:             REPLACE mens.codigo WITH vcod
1125:             REPLACE mens.ciudad WITH vrciu
1126:          ENDIF
1127:          SELECT  dvar
1128:          LOCATE FOR vcod = codigo
1129:          IF FOUND()
1130:             RECALL
1131:          ELSE
1132:             APPEND BLANK
1133:             REPLACE dvar.sistema    WITH vrciu
1134:             REPLACE dvar.cta_ctble  WITH vcta
1135:             REPLACE dvar.codigo     WITH vcod
1136:             REPLACE dvar.cod_dep    WITH vrdep
1137:             REPLACE dvar.cod_sec    WITH vcod_sec
1138:             REPLACE dvar.jerarquia  WITH vescala
1139:             REPLACE dvar.cta_ctble  WITH vcta
1140:          ENDIF
1141: <========RETURN
1142:
1143:
*!*************************************************************
**************
1144:  *!
1145:  *!       Procedure: EXCLUIR
1146:  *!
1147:  *!       Called by: SECC()              (function in
PERSONAL.PRG)
1148:  *!                : JUBILAR            (procedure in
PERSONAL.PRG)
1149:  *!
1150:  *!           Calls: COD_PERSONA         (procedure in
PERSONAL.PRG)
1151:  *!                : DESPLEGA           (procedure in
```

```
PERSONAL.PRG)
 1152:  *!
 1153:
*!*************************************************************************
**************
 1154:      PROCEDURE excluir
 1155:          DO cod_persona
 1156:
 1157:         ┌─IF NOT WEXIST('VARIOS') && o definir en el
principal
 1158:         │    DEFINE WINDOW varios FROM 0,0 TO 7,35 TITLE
'ALERTA ..!' SYSTEM
 1159:         └─ENDIF
 1160:           ACTIVATE WINDOW varios
 1161:           MOVE WINDOW varios CENTER
 1162:         ┌─IF dgp.estado = 1
 1163:         │    @ 1,2 SAY 'Desea Excluir del Rol a:
'+dgp.cod_per
 1164:         ├─ELSE
 1165:         │    @ 1,2 SAY 'No se Puede Excluir a: '+dgp.cod_per
 1166:         └─ENDIF
 1167:           STOR 2 TO opc_ac
 1168:           @ 4,5 GET opc_ac PICT '@*H \<ACEPTAR;\<CANCELAR':
 1169:              SIZE 1.308,12.500,0.308 :
 1170:              DEFAULT 1 :
 1171:              FONT "MS Serif", 8 :
 1172:              STYLE "BT"
 1173:           READ MODAL
 1174:           RELE WIND varios
 1175:         ┌─IF opc_ac = 2
 1176:         │    STOR .F. TO vret
 1177:         │    WAIT WIND 'PERSONA NO FUE Excluida ...!'
 1178:  <──────┘    RETURN
 1179:         └─ENDIF
 1180:
 1181:           REPLACE estado WITH 2
 1182:           DO despliega
 1183:           *Select Mens
 1184:           *Locate for codigo = vcod
 1185:           *If found()
 1186:           *   Delete
 1187:           *Endif
 1188:           *Select Dvar
 1189:           *Locate for codigo = vcod
 1190:           *If Found()
 1191:           *   Delete
 1192:           *Endif
 1193:  <──────RETURN
 1194:
 1195:
*!*************************************************************************
**************
 1196:  *!
 1197:  *!       Procedure: CAMBIO_SIS
```

```
 1198:  *!
 1199:  *!       Called by: SECC()              (function in
PERSONAL.PRG)
 1200:  *!
 1201:  *!           Calls: COD_PERSONA         (procedure in
PERSONAL.PRG)
 1202:  *!                : DESPLEGA            (procedure in
PERSONAL.PRG)
 1203:  *!                : VAL_SEC()           (function in
PERSONAL.PRG)
 1204:  *!                : VAL_PUE()           (function in
PERSONAL.PRG)
 1205:  *!
 1206:
*!*************************************************************************
**************
 1207:      PROCEDURE cambio_sis
 1208:          DO cod_persona
 1209:          DO desplega
 1210:          STOR 'enable ' TO vestado
 1211:          @4,41  GET vciu FROM v3 PICTURE "@^" COLOR SCHEME
4
 1212:          READ
 1213:          SELE sst
 1214:          LOCATE FOR ciudad = vciu
 1215:          STOR ALLT(STR(cod_ciu)) + SUBS(vcod,2,5) TO ccod
 1216:          SELE dgp
 1217:          LOCATE FOR cod_per = ccod
 1218:       ┌─IF FOUND()
 1219:       │     WAIT WIND 'CODIGO YA EXISTE'
 1220:       ├─ELSE
 1221:       │     @ 1,41 SAY ccod PICT "######" &vcolor
 1222:       │     @ FGET+6,2,  43 GET vsec FROM v2 PICTURE "@^"
COLOR SCHEME 4  VALID val_sec()
 1223:       │     @ FGET+8,   41 GET vpue FROM v1 PICTURE "@^"
COLOR SCHEME 4  VALID val_pue()
 1224:       │     @ FGET+11, 47 GET vcta DISABLE
 1225:       │     READ
 1226:       │     LOCATE FOR cod_per = vcod
 1227:       │     REPLACE dgp.cod_per WITH ccod
 1228:       │     REPLACE dgp.num_ciu WITH vrciu
 1229:       │     REPLACE dgp.num_sec WITH vcod_sec
 1230:       │     REPLACE dgp.cuent_cont WITH vcta
 1231:       │     REPLACE dgp.num_dep WITH vrdep
 1232:       │     REPLACE dgp.cod_pues WITH vcodpue
 1233:       │     REPLACE dgp.rel_dep  WITH vrel_dep
 1234:       │     REPLACE dgp.cargo    WITH vpue
 1235:       │     SELECT  mens
 1236:       │     LOCATE FOR vcod = codigo
 1237:       │  ┌─IF FOUND()
 1238:       │  │     REPLACE mens.codigo WITH ccod
 1239:       │  │     REPLACE mens.ciudad WITH vrciu
 1240:       │  └─ENDIF
 1241:       │     SELECT  dvar
```

```
1242:           LOCATE FOR vcod = codigo
1243:         IF FOUND()
1244:             REPLACE dvar.sistema     WITH vrciu
1245:             REPLACE dvar.cta_ctble   WITH vcta
1246:             REPLACE dvar.codigo      WITH ccod
1247:             REPLACE dvar.cod_dep     WITH vrdep
1248:             REPLACE dvar.cod_sec     WITH vcod_sec
1249:             REPLACE dvar.jerarquia   WITH vescala
1250:             REPLACE dvar.cta_ctble   WITH vcta
1251:
1252:         ENDIF
1253:       ENDIF
1254: <=====RETURN
1255:
1256:
*!*************************************************************
**************
1257:   *!
1258:   *!       Procedure: BUSCAR
1259:   *!
1260:   *!       Called by: SECC()              (function in
PERSONAL.FRG)
1261:   *!                : _R390JW4G3()        (function in
DSC_GRTR.SPR)
1262:   *!                : _RA81B8ERO()        (function in
ACT_PTMO.SPR)
1263:   *!                : _RA81B8GQD()        (function in
ACT_PTMO.SPR)
1264:   *!                : _R6103FBLN()        (function in
MO_PDLAB.SPR)
1265:   *!                : _R5K0M9Z09()        (function in
DSC_CTA.SPR)
1266:   *!
1267:   *!            Calls: VAL_BUSCAR         (procedure in
AACT_SST.SPR)
1268:   *!                : BUSC_CLV()          (function in
DEP_CARG.SPR)
1269:   *!                : DESPLEGA            (procedure in
PERSONAL.FRG)
1270:   *!                : BUSCARP()           (function in
ACT_PTMO.SPR)
1271:   *!                : REF_SCR             (procedure in
MO_PDLAB.SPR)
1272:   *!                : MENSAJES            (procedure in
MO_PDLAB.SPR)
1273:   *!                : ASIGNAM             (procedure in
MO_PDLAB.SPR)
1274:   *!
1275:
*!*************************************************************
**************
1276:       PROCEDURE buscar
1277:         DO CASE
1278:         CASE modo=1
```

```
 1279:          |          STORE SPACE (6) TO vcod
 1280:          |          STORE SPACE (1) TO vcod1
 1281:          |          STORE 0 TO vcod2
 1282:          |
 1283:          |          @ 1,41 SAY vcod
 1284:          |          @ 1,41 GET vcod1 VALID
INLIST(vcod1, '1', '2', '3', '4', '5')
 1285:          |          @ 1,44 GET vcod2 PICT '999'
 1286:          |          READ
 1287:          |          STOR vcod1 + '00'+IIF(vcod2 <
10, '00',IIF(vcod2<100, '0', ''))+ALLT(STR(vcod2)) TO vcod
 1288:          |       ┌─IF LASTKEY() = 27
 1289: <════════════════RETURN
 1290:          |       └─ENDIF
 1291:          |         SELECT dgp
 1292:          |       ┌─IF modo = 2
 1293:          |       |    LOCATE FOR cod_per = vcod
 1294:          |       ├─ELSE
 1295:          |       |    SEEK vcod
 1296:          |       └─ENDIF
 1297:          |
 1298:          |       ┌─IF FOUND()
 1299:          |       |    DO despliega
 1300:          |       ├─ELSE
 1301:          |       |    WAIT "Registro No Existe....!" WINDOW
TIME 1
 1302:          |       └─ENDIF
 1303:          |    ═══CASE modo = 2
 1304:          |         STORE SPACE (30) TO vnom
 1305:          |         @ 3,41 GET vnom PICT "@!" VALID vnom <>
SPACE(30)
 1306:          |         READ
 1307:          |       ┌─IF LASTKEY() = 27
 1308: <════════════════RETURN
 1309:          |       └─ENDIF
 1310:          |         SELECT dgp
 1311:          |         GO TOP
 1312:          |         LOCATE FOR nombre=ALLT(vnom)
 1313:          |       ┌─IF FOUND()
 1314:          |       |    DO despliega
 1315:          |       ├─ELSE
 1316:          |       |    WAIT "Registro No Existe....!" WINDOW
TIME 1
 1317:          |       └─ENDIF
 1318:          |    ─┘ENDCASE
 1319: <═══════RETURN
 1320:
 1321:
*!*************************************************************************
**************
 1322:  *!
 1323:  *!        Procedure: SIGUIENTE
 1324:  *!
 1325:  *!        Called by: AP_PRMCT.SPR
```

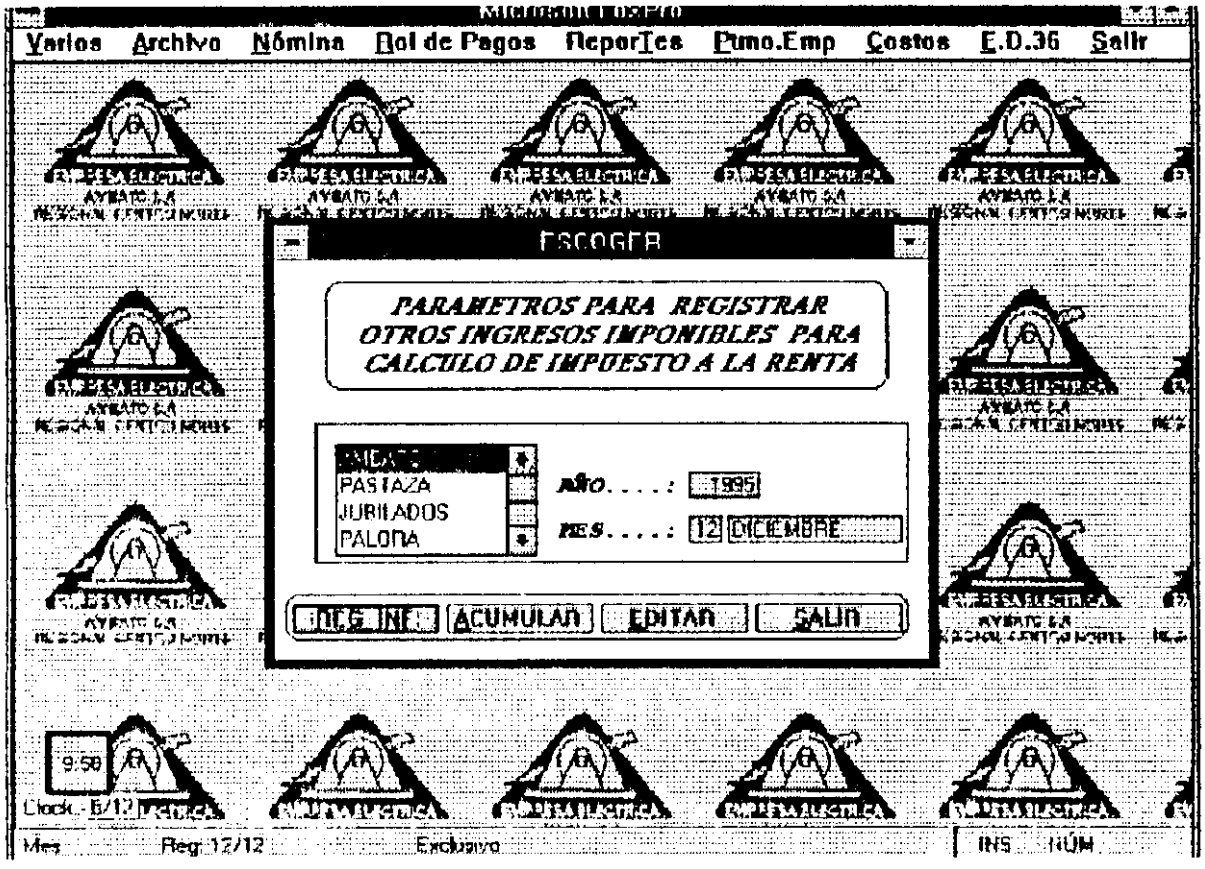
```
1326:  *!                    : AP_PRMCC.SPR
1327:  *!                    : AP_PRMVR.SPR
1328:  *!                    : SECC()              (function in
PERSONAL.PRG)
1329:  *!
1330:  *!             Calls: REFRESH()             (function in
AP_PRMCT.SPR)
1331:  *!                    : DESPLEGA            (procedure in
PERSONAL.PRG)
1332:  *!
1333:
*!*************************************************************************
**************
1334:      PROCEDURE siguiente
1335:          SELECT dgp
1336:          IF .NOT. EOF()
1337:             SKIP
1338:             IF .NOT. EOF()
1339:                DO desplega
1340:             ELSE
1341:                WAIT "Fin de Archivo...!"  WINDOW TIME 1
1342:             ENDIF
1343:          ELSE
1344:             WAIT "Fin de Archivo...!"  WINDOW TIME 1
1345:          ENDIF
1346:  <=======RETURN
1347:
1348:
*!*************************************************************************
**************
1349:  *!
1350:  *!       Procedure: ANTERIOR
1351:  *!
1352:  *!       Called by: AP_PRMCT.SPR
1353:  *!                : AP_PRMCC.SPR
1354:  *!                : AP_PRMVR.SPR
1355:  *!                : SECC()              (function in
PERSONAL.PRG)
1356:  *!
1357:  *!            Calls: REFRESH()          (function in
AP_PRMCT.SPR)
1358:  *!                : DESPLEGA            (procedure in
PERSONAL.PRG)
1359:  *!
1360:
*!*************************************************************************
**************
1361:      PROCEDURE anterior
1362:          SELECT dgp
1363:          IF .NOT. BOF()
1364:             SKIP -1
1365:             IF .NOT. BOF()
1366:                DO desplega
1367:             ELSE
```

```
1368:          |  |     WAIT "Inicio de Archivo...!"  WINDOW TIME 1
1369:          |  ENDIF
1370:          |ELSE
1371:          |    WAIT "Inicio de Archivo...!"  WINDOW TIME 1
1372:          ENDIF
1373:  <=======RETURN
1374:
1375:
1376:        ***DESP
1377:
1378:
*!*************************************************************************
**************
1379:  *!
1380:  *!       Procedure: CAPTURAR
1381:  *!
1382:  *!       Called by: SECC()              (function in
PERSONAL.PRG)
1383:  *!
1384:  *!           Calls: DESPLEGA            (procedure in
PERSONAL.PRG)
1385:  *!
1386:
*!*************************************************************************
**************
1387:      PROCEDURE capturar
1388:          IF vcod=SPACE(6)
1389:          |DO WHILE .T.
1390:          |   @ 1,41 GET vcod &fscampn  PICT 'XXXXXX'
1391:          |   READ
1392:          |   IF LASTKEY()=27 .OR. vcod=SPACE(6)
1393:          |      vcod=SPACE(6)
1394:  <=======|======RETURN
1395:          |   ENDIF
1396:          |   SELE dgp
1397:          |   IF modo = 2
1398:          |      LOCATE FOR cod_per = vcod
1399:          |   ELSE
1400:          |      SEEK vcod
1401:          |   ENDIF
1402:          |   IF FOUND()
1403:          |      DO desplega
1404:          |======EXIT
1405:          |   ELSE
1406:          |      WAIT 'Registro No existe..'WINDOW TIME(1)
1407:          |   ENDIF
1408:          |ENDDO
1409:          ENDIF
1410:          SELE dgp
1411:          STOR ' ' TO A
1412:          STOR pathbmps + ALLTRIM(vcod)+ '.BMP' TO A
1413:          STOR 'N' TO fotsn
1414:          IF .NOT. EMPTY(foto)
1415:             WAIT WIND 'Fotografia Existe. Actualizar <S/N>
```

```
' TO fotsn
 1416:          ENDIF
 1417:          IF EMPTY(foto) OR UPPE(fotsn) = 'S'
 1418:            IF FILE(A)
 1419:               APPEN  GENERAL foto FROM  &a
 1420:            ELSE
 1421:               ?CHR(7)
 1422:               WAIT WIND 'Archivo Bit Map No Existe'
 1423:            ENDIF
 1424:          ENDIF
 1425:          @3.15 SAY foto SIZE 10,13 ISOM
 1426: <=======RETURN
 1427:
 1428:
*!*************************************************************************
**************
 1429:  *!
 1430:  *!       Procedure: CABEZA
 1431:  *!
 1432:  *!       Called by: PERSONAL.PRG
 1433:  *!                : EDITAR                (procedure in
CTRL_RES.SPR)
 1434:  *!                : ING_GRAV.PRG
 1435:  *!                : EGR_VARI.PRG
 1436:  *!
 1437:  *!           Calls: CIUDAD               (procedure in
DSC_GRTR.SPR)
 1438:  *!                : MENSAJE1             (procedure in
DSC_GRTR.SPR)
 1439:  *!
 1440:
*!*************************************************************************
**************
 1111:      PROCEDURE cabesa
 1442:          STOR '  ' TO fsdesc
 1443:          @ 1,30  CLEAR TO 15,78
 1444:          @ 16,22 CLEAR TO 22,78
 1445:          @ 1,30  SAY "CODIGO  :"  &fsdesc
 1446:          *&FSDESC
 1447:          STOR '   'TO fsdesc                                    &&
styl and font of say coment*stor '  FONT "N",11 STYLE "B" ' to
fscampC   && styl and font of say CHARACTER fields
 1448:          @ 3,30  SAY "NOMBRE  : " &fsdesc
 1449:          @ 5,30  SAY "SISTEMA :" &fsdesc
 1450:          @ 7,30 SAY "DEPARTAMENTO:" &fsdesc
 1451:          *  @ 9,30  Say "SECCION :" &FSDESC
 1452:          @ 9,30  SAY "CARGO   :" &fsdesc
 1453:          @ 11,30 SAY "CTA. CONTABLE   : "&fsdesc
 1454:          @ 13,30 SAY "CARGO JEFATURA  : "&fsdesc
 1455:          @ 13,55 SAY "CATEGORIA :"&fsdesc
 1456:          **
 1457:          STOR 'FONT "NBS",8 style "bs"   'TO fsdesc
                 && styl and font of say coment
 1458:          @ 15,14 SAY "FECHA/ING/EMPRESA:"&fsdesc
```

```
1459:          @ 15,50 SAY "FECHA/ING/PUESTO:"&fsdesc
1460:          @ 18,14 SAY "FECHA DE NACIM. :"&fsdesc
1461:          *  @ 18,50 Say "ESTADO CIVIL     :"&FSDESC
1462:          @ 18,50 SAY "TIEMPO/SERVICIO :"&fsdesc
1463:          @ 18,73 SAY "Años"&fsdesc
1464:          @ 18,81 SAY "Meses"&fsdesc
1465:          @ 21,14 SAY "SEXO   :"&fsdesc
1466:          @ 21,28 SAY "ESTADO CIVIL     :"&fsdesc
1467:          @ 21,50 SAY "CARGAS FAMILIAR.:" &fsdesc
1468:          @ 25,14 SAY "CEDULA           :"&fsdesc
1469:          @ 25,50 SAY "CARNET I.E.S.S. :"&fsdesc
1470:          @ 29,14 SAY "BANCO            :"&fsdesc
1471:          @ 27,50 SAY "TIPO/CUENTA   :"&fsdesc
1472:          @ 29,50 SAY "No. DE CUENTA    :"&fsdesc
1473:          @ 33,14 SAY "SUELDO           :"&fsdesc
1474:          @ 33,50 SAY "GASTO REPRESENTACION  :"&fsdesc
1475:          @ 36,14 SAY "GASTO RESPONSABILIDAD :"&fsdesc
1476:          @ 36,50 SAY "TIEMPO  TRABAJADO :"&fsdesc
1477:          @ 39,14 SAY "PERTENECE A :  "&fsdesc
1478:          @ 39,35 SAY "ESTADO     :"&fsdesc
1479:          *  @ 39,55 Say "CATEGORIA :"&FSDESC
1480:          *  @ 41.5,14 Say "GRUPO OCUP."&FSDESC
1481:          @ 41.5,40 SAY "CTA. ABONADO No.:"&fsdesc
1482:          @ 41.5,70 SAY "ZONA:"&fsdesc
1483:  <=======RETURN
1484:
1485:
*!*************************************************************
**************
1486:  *!
1487:  *!        Procedure: TAG
1488:  *!
1489:  *!        Called by: DGP_CARG.SPR
1490:  *!                 : PERSONAL.FRG
1491:  *!
1492:
*!*************************************************************
**************
1493:       PROCEDURE TAG
1494:          FOR i = 1 TO 256
1495:             IF EMPTY(TAG(i))
1496:                i = i - 1
1497:                DIMENSION vartag(i)
1498:  <=========EXIT
1499:             ELSE
1500:                vartag(i) = TAG(i)
1501:             ENDIF
1502:          ENDFOR
1503:  <=======RETURN
1504:          ******************
1505:
*!*************************************************************
**************
1506:  *!
```

```
1507:   *!        Function: VAL_JEF
1508:   *!
1509:   *!       Called by: ING                    (procedure in
PERSONAL.PRG)
1510:   *!
1511:
*!*************************************************************************
*************
1512:       FUNCTION val_jef
1513:           ******************
1514:           PARA vjef
1515:           IF vjef >= 0 AND vjef < 4
1516:               VAR=.T.
1517:               IF vescala != 3
1518:                   SELECT esc1
1519:                   LOCATE FOR escala = vescala
1520:                   STOR vgres_act TO vgrespo
1521:                   STOR vgrep_act TO vgrepre
1522:               ELSE
1523:                   STOR 0 TO vgrespo, vgrepre
1524:               ENDIF
1525:               DO CASE
1526:               CASE vjef=0
1527:                       STOR 0 TO vgrespo,vgrepre
1528:               CASE vjef=1
1529:                       STOR 0 TO vgrespo
1530:               CASE vjef=2
1531:                       STOR 0 TO vgrepre
1532:               ENDCASE
1533:               SHOW GET vgrespo
1534:               SHOW GET vgrepre
1535:               SHOW GET vsueldo
1536:           ELSE
1537:               VAR=.F.
1538:           ENDIF
1539:           RETURN VAR
1540:   *:  EOF: PERSONAL.act
```


Varios Archivo Nómina Rol de Pagos ReporTes Pmo.Emp Costos E.D.36 Salir
ESCOGER
PARAMETROS PARA REGISTRAR
OTROS INGRESOS IMPONIBLES PARA
CALCULO DE IMPUESTO A LA RENTA
PASTAZA
JUBILADOS
PALORA
Año. . . . : 1995
MES. . . . : 12 DICIEMBRE
ACUMULAR
EDITAR
SALIR
9:58
Clock - 6/12
Mes
Reg: 12/12
Exclusivo
INS
NUM

```
    1:
*:*************************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\CTRL_OIG.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995. Sisrol Ambato - Ecuador
    7:  *:  Last modified: 10/04/1996 at 16:06:54
    8:  *:
    9:  *:  Procs & Fncts: _R9HOYJFIT()
   10:  *:                : _R9HOYJFMR()
   11:  *:                : _R9HOYJFR3()
   12:  *:                : ACUMULAR
   13:  *:                : REG_INF
   14:  *:
   15:  *:          Set by: EMPRESA.MPR
   16:  *:
   17:  *:           Calls: _R9HOYJFIT()        (function in
CTRL_OIG.SPR)
   18:  *:                : _R9HOYJFMR()        (function in
CTRL_OIG.SPR)
   19:  *:                : _R9HOYJFR3()        (function in
CTRL_OIG.SPR)
   20:  *:
   21:  *:       Documented 19:14:46
FoxDoc version 3.00a
   22:
*:*************************************************************************
**************
   23:      *
************************************************************
   24:      *        *
   25:      *        * 10/04/1996            CTRL_OIG.SPR
 16:06:52
   26:      *        *
   27:      *
************************************************************
   28:      *        *
   29:      *        * Nombre del autor
   30:      *        *
   31:      *        * Copyright (c) 1996 Nombre de la empresa
   32:      *        * Direcci¢n
   33:      *        * Ciudad.      C.P.
   34:      *        * Pa¢s
   35:      *        *
   36:      *        * Description:
   37:      *        * Este programa lo ha generado autom3ticamente
GENSCRN.
   38:      *        *
   39:      *
************************************************************
   40:
```

```
*
*****************************************************************
*        *
*        *  CTRL_OIG/Windows Código de configuración - SECCION 1
*        *
*
*****************************************************************
*

#REGION 1
**********
REGIONAL opc_ser, opc_rol, opc_qui, mmes, vmes, manio, mquincena, mrol,:
    mopc_qui,mfquin,vciu,vtip
STOR " " TO opc_ser, mmes,opc_rol, fquin
STOR 0 TO opc_qui,mopc_qui,vtip
STOR 1 TO mrol,vciu
STOR .F. TO mquincena
STOR YEAR(DATE()) TO manio
STOR MONTH(DATE()) TO vmes

**********

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
    SET TALK OFF
    m.talkstat = "ON"
ELSE
    m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.rbborder = SET("READBORDER")
SET readborder ON

m.currarea = SELECT()

*
*****************************************************************
*        *
*        *  CTRL_OIG/Windows Bases de datos, índices, relaciones
*        *
*
*****************************************************************
*

IF USED("oig")
```

```
 88:    |    SELECT oig
 89:    |    SET ORDER TO TAG "cod_per"
 90:    |ELSE
 91:    |    SELECT 0
 92:    |    USE (LOCFILE("dbfs\o_ing_gr.dbf","DBF","Buscar
o_ing_gr?"));
 93:    |        AGAIN ALIAS oig ;
 94:    |        ORDER TAG "cod_per"
 95:    |ENDIF
 96:
 97:    |IF USED("mes")
 98:    |    SELECT mes
 99:    |    SET ORDER TO 0
100:    |ELSE
101:    |    SELECT 0
102:    |    USE (LOCFILE("dbfs\meses.org","DBF","Buscar
meses?"));
103:    |        AGAIN ALIAS mes ;
104:    |        ORDER 0
105:    |ENDIF
106:
107:    |IF USED("sst")
108:    |    SELECT sst
109:    |    SET ORDER TO TAG "sist_cod"
110:    |ELSE
111:    |    SELECT 0
112:    |    USE (LOCFILE("dbfs\sistema.dbf","DBF","Buscar
sistema?"));
113:    |        AGAIN ALIAS sst ;
114:    |        ORDER TAG "sist_cod"
115:    |ENDIF
116:
117:    |IF USED("acr")
118:    |    SELECT acr
119:    |    SET ORDER TO TAG "cod_per"
120:    |ELSE
121:    |    SELECT 0
122:    |    USE (LOCFILE("dbfs\ac_imprt.dbf","DBF","Buscar
ac_imprt?"));
123:    |        AGAIN ALIAS acr ;
124:    |        ORDER TAG "cod_per"
125:    |ENDIF
126:
127:    SELECT oig
128:
129:    SET RELATION OFF INTO acr
130:    SET RELATION TO cod_per INTO acr ADDITIVE
131:
132:    SELECT oig
133:
134:    *
************************************************************
135:    *       *
136:    *       *              Windows Definiciones de ventana
```

```
137:      *         *
138:      *
**********************************************************************
139:      *
140:
141:      IF NOT WEXIST("ctrl_par") ;
142:            OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.FJX" ;
143:            OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.SCX" ;
144:            OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.MNX" ;
145:            OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.FRG" ;
146:            OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.FRX" ;
147:            OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.QPR"
148:         DEFINE WINDOW ctrl_par ;
149:            AT  0.000, 0.000  ;
150:            SIZE 17.154,70.200 ;
151:            TITLE "ESCOGER" ;
152:            FONT "MS Sans Serif", 8 ;
153:            FLOAT ;
154:            NOCLOSE ;
155:            MINIMIZE ;
156:            SYSTEM
157:         MOVE WINDOW ctrl_par CENTER
158:      ENDIF
159:
160:
161:      *
**********************************************************************
162:      *         *
163:      *         *  CTRL_OIG/Windows Cdigo de configuracin -
SECCION 2
164:      *         *
165:      *
**********************************************************************
166:      *
167:
168:      #REGION 1
169:      **********
170:      SELE mes
171:      LOCATE FOR mes = vmes
172:      STOR nom_mes TO nmes
173:
174:      #REGION 1
175:      DEFINE POPUP _r9h0yifb9 ;
176:         PROMPT FIELD sst.ciudad ;
177:         SCROLL
178:
179:      *
**********************************************************************
180:      *         *
181:      *         *          CTRL_OIG/Windows Distribucin de
pantalla
182:      *         *
183:      *
**********************************************************************
```

```
184:      *
185:
186:      #REGION 1
187:     ┌IF WVISIBLE("ctrl_par")
188:     │   ACTIVATE WINDOW ctrl_par SAME
189:     ├ELSE
190:     │   ACTIVATE WINDOW ctrl_par NOSHOW
191:     └ENDIF
192:      @ 11.000,30.600 SAY "MES....:" :
193:         FONT "Courier New", 10 :
194:         STYLE "BIT"
195:      @ 9.077,30.600 SAY "AÑO....:" :
196:         FONT "Courier New", 10 :
197:         STYLE "BIT"
198:      @ 14.308,1.200 TO 16.385,68.200 :
199:         PEN 1, 8 :
200:         STYLE "16"
201:      @ 1.385,7.400 SAY "PARAMETROS PARA  REGISTRAR" +
CHR(13) + :
202:         "OTROS INGRESOS IMPONIBLES  PARA" + CHR(13) + :
203:         "CALCULO DE IMPUESTO A LA RENTA" :
204:         SIZE 3.000,40.429, 0.000 :
205:         PICTURE "@I" :
206:         FONT "MS Serif", 10 :
207:         STYLE "BIT"
208:      @ 0.923,5.600 TO 5.615,66.000 :
209:         PEN 1, 8 :
210:         STYLE "16"
211:      @ 7.000,4.400 TO 13.000,68.200 :
212:         PEN 1, 8
213:      @ 7.923,6.600 GET opc_rol :
214:         PICTURE "@&N" :
215:         POPUP _r9h0y,ifb8 :
216:         SIZE 4.615,21.400 :
217:         DEFAULT " " :
218:         FONT "MS Sans Serif", 8 :
219:         STYLE "" :
220:         VALID _r9h0y,ifit()
221:      @ 9.154,45.000 GET manio :
222:         SIZE 1.000,6.600 :
223:         DEFAULT " " :
224:         FONT "MS Sans Serif", 8 :
225:         PICTURE "9999" :
226:         VALID manio > 1990 AND manio < 2999 :
227:         ERROR 'AÑO INVALIDO '
228:      @ 11.077,45.000 GET vmes :
229:         SIZE 1.000,2.600 :
230:         DEFAULT " " :
231:         FONT "MS Sans Serif", 8 :
232:         PICTURE "99" :
233:         VALID _r9h0y,ifmr() :
234:         ERROR 'MES INVALIDO (1..12)'
235:      @ 11.077,49.200 GET mes.nom_mes :
236:         SIZE 1.000,17.800 :
```

```
237:          DEFAULT " " ;
238:          FONT "MS Sans Serif", 8 ;
239:          PICTURE "@K" ;
240:          WHEN .F.
241:       @ 14.769,2.000 GET ope_scr ;
242:          PICTURE "@*HN
\<REG_INF;\<ACUMULAR;\<EDITAR;\<SALIR" ;
243:          SIZE 1.308,13.167,0.667 ;
244:          DEFAULT 1 ;
245:          FONT "MS Sans Serif", 8 ;
246:          STYLE "B" ;
247:          VALID _r9h0yjfr3()
248:
249:       IF NOT WVISIBLE("ctrl_par")
250:          ACTIVATE WINDOW ctrl_par
251:       ENDIF
252:
253:       READ CYCLE ;
254:          VALID .F.
255:
256:       RELEASE WINDOW ctrl_par
257:
258:       *
*************************************************************
259:       *          *
260:       *          *              Windows Cerrando bases de datos
261:       *          *
262:       *
*************************************************************
263:       *
264:
265:       IF USED("oig")
266:          SELECT oig
267:          USE
268:       ENDIF
269:
270:       IF USED("mes")
271:          SELECT mes
272:          USE
273:       ENDIF
274:
275:       IF USED("sst")
276:          SELECT sst
277:          USE
278:       ENDIF
279:
280:       IF USED("acr")
281:          SELECT acr
282:          USE
283:       ENDIF
284:
285:       SELECT (m.currarea)
286:
287:       RELEASE POPUPS _r9h0yjfb9
```

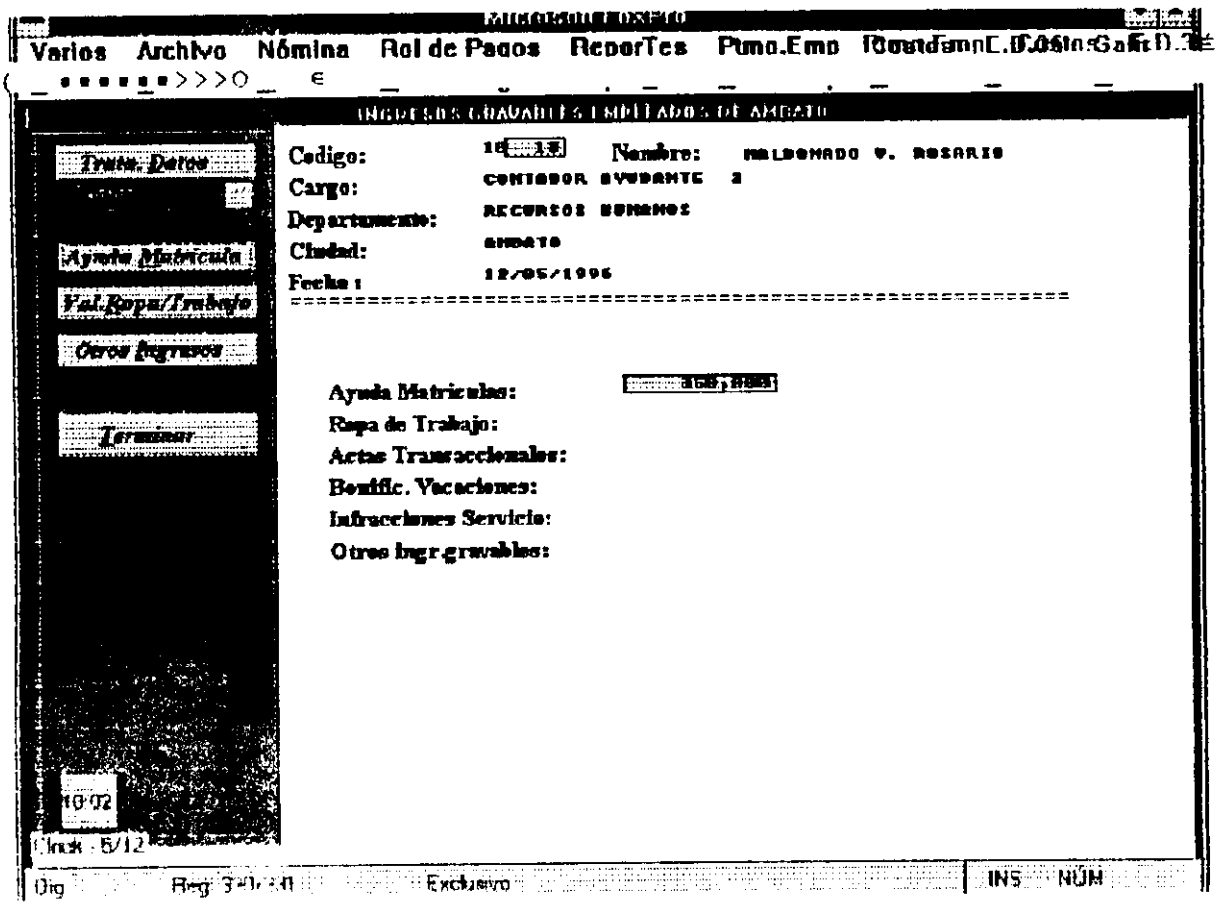
```
288:
289:      #REGION 0
290:
291:      SET readborder &rborder
292:
293:     ┌IF m.talkstat = "ON"
294:     │   SET TALK ON
295:     └ENDIF
296:     ┌IF m.compstat = "ON"
297:     │   SET COMPATIBLE ON
298:     └ENDIF
299:
300:
301:      *
*************************************************************
302:      *        *
303:      *        *            CTRL_OIG/Windows Código de
limpieza
304:      *        *
305:      *
*************************************************************
306:      *
307:
308:      #REGION 1
309:      ********************
310:
311:
312:      *
*************************************************************
313:      *        *
314:      *        * CTRL_OIG/Windows Procedimientos y funciones
de soporte
315:      *        *
316:      *
*************************************************************
317:      *
318:
319:      #REGION 1
320:
*!*************************************************************
**************
321:  *!
322:  *!       Procedure: ACUMULAR
323:  *!
324:  *!            Calls: VALIDAR()          (function in
CTRL_GTO.SPR)
325:  *!
326:
*!*************************************************************
***************
327:      PROCEDURE acumular
328:          ********************
329:          SELECT oig
330:          SET FILTER TO oig.anio = manio AND oig.mes = vmes:
```

```
  331:              AND oig.num_ciu = vciu AND !oig.cierre
  332:          ╔═SCAN
  333:          ╠    REPLACE acr.ac_otroimp WITH acr.ac_otroimp +
oig.ac_otroimp;
  334:          ╠         acr.ac_bonvac WITH acr.ac_bonvac +
oig.ac_bonvac;
  335:          ╠         acr.ac_roptrab WITH acr.ac_roptrab +
oig.ac_roptrab;
  336:          ╠         acr.ac_infserv WITH acr.ac_infserv +
oig.ac_infserv;
  337:          ╠         acr.ac_acttran WITH acr.ac_acttran +
oig.ac_acttran;
  338:          ╠         acr.ac_oting WITH acr.ac_oting +
oig.ac_oting;
  339:          ╠         acr.ac_ayumatr WITH acr.ac_ayumatr +
oig.ac_ayumatr;
  340:          ╠         oig.cierre WITH .T.
  341:          ╚═ENDSCAN
  342:  <═══════RETURN
  343:
  344:          ****************
  345:
*!*************************************************************************
**************
  346:  *!
  347:  *!       Procedure: REG_INF
  348:  *!
  349:  *!            Calls: ING_GRAV.PRG
  350:  *!
  351:
*!*************************************************************************
**************
  352:      PROCEDURE reg_inf
  353:          ****************
  354:          DO ing_grav WITH manio, vmes, vciu
  355:  <═══════RETURN
  356:
  357:          ****************
  358:
*!*************************************************************************
**************
  359:  *!
  360:  *!       Procedure: EDITAR
  361:  *!
  362:  *!       Called by: CREAR              (procedure in
AACT_SST.SPR)
  363:  *!
  364:  *!            Calls: DEAC_KEYS         (procedure in
AACT_SST.SPR)
  365:  *!                 : CABEZA            (procedure in
PERSONAL.PRG)
  366:  *!                 : INGRESO           (procedure in
ACT_PTMO.SPR)
  367:  *!                 : CREA_TMP          (procedure in
```

```
MO_FDLAB.SPR)
  368:  *!                  : MBROW              (procedure in
MO_FDLAB.SPR)
  369:  *!                  : REF_SCR            (procedure in
MO_FDLAB.SPR)
  370:  *!                  : MENSAJES           (procedure in
MO_FDLAB.SPR)
  371:  *!
  372:
*!*************************************************************************
**************
  373:      PROCEDURE editar
  374:          ****************
  375:          _CUROBJ= OBJNUM(opc_rol)
  376: <=======RETURN
  377:
  378:          ***************
  379:
*!*************************************************************************
**************
  380:  *!
  381:  *!       Procedure: SALIR
  382:  *!
  383:  *!       Called by: _RAA1732TG()      (function in
AT_ESCLS.SPR)
  384:  *!                : DA_ASISG.PRG
  385:  *!                : DE_EGRQ1.PRG
  386:  *!                : DE_EGRQ2.PRG
  387:  *!                : DE_EGRMN.PRG
  388:  *!                : DE_IEJUB.PRG
  389:  *!
  390:  *!           Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
  391:  *!
  392:
*!*************************************************************************
**************
  393:      PROCEDURE salir
  394:          ***************
  395:          CLEAR READ
  396: <=======RETURN
  397:
  398:          *
**************************************************
  399:          *        *
  400:          *        * _R9HOYJFIT          opc_rol VALID
  401:          *        *
  402:          *        * Function Origin:
  403:          *        *
  404:          *        * From Platform:      Windows
  405:          *        * From Screen:        CTRL_OIG,
Record Number:   13
  406:          *        * Variable:           opc_rol
  407:          *        * Called By:          VALID Clause
```

```
408:          *         * Object Type:          List
409:          *         * Snippet Number:       1
410:          *         *
411:          *
*************************************************************
412:          *
413:
*!*************************************************************
**************
414:  *!
415:  *!        Function: _R9HOYJFIT
416:  *!
417:  *!       Called by: CTRL_OIG.SPR
418:  *!
419:
*!*************************************************************
**************
420:      FUNCTION _r9h0yjfit      &&  ope_rol VALID
421:          #REGION 1
422:          STOR sst.cod_ciu TO vciu
423:          STOR sst.tipo TO vtip
424:          STOR 2 TO ope_qui
425:          STOR .F. TO mquinceena
426:          IF sst.tipo = 0
427:              STOR 1 TO ope_qui
428:              STOR .T. TO mquinceena
429:          ENDIF
430:
431:          *
*************************************************************
432:          *         *
433:          *         * _R9HOYJFMR            VMES VALID
434:          *         *
435:          *         * Function Origin:
436:          *         *
437:          *         * From Platform:        Windows
438:          *         * From Screen:          CTRL_OIG.
Record Number:   15
439:          *         * Variable:             VMES
440:          *         * Called By:            VALID Clause
441:          *         * Object Type:          Field
442:          *         * Snippet Number:       3
443:          *         *
444:          *
*************************************************************
445:          *
446:
*!*************************************************************
**************
447:  *!
448:  *!        Function: _R9HOYJFMR
449:  *!
450:  *!       Called by: CTRL_OIG.SPR
451:  *!
```

```
452:
*!*************************************************************
**************
453:      FUNCTION _r9h0yjfmr      &&  VMES VALID
454:          #REGION 1
455:          IF vmes > 0 AND vmes < 13
456:              STOR .T. TO vret
457:              SELE mes
458:              LOCATE FOR vmes = mes
459:              STOR nom_mes TO mmes
460:              SHOW GET mes.nom_mes
461:          ELSE
462:              STOR .F. TO vret
463:          ENDIF
464:  <=======RETURN vret
465:
466:          *
*************************************************************
467:          *        *
468:          *        * _R9H0YJFR3            OPC_SCR VALID
469:          *        *
470:          *        * Function Origin:
471:          *        *
472:          *        * From Platform:        Windows
473:          *        * From Screen:          CTRL_OIG.
Record Number:   17
474:          *        * Variable:             OPC_SCR
475:          *        * Called By:            VALID Clause
476:          *        * Object Type:          Push Button
477:          *        * Snippet Number:       3
478:          *        *
479:          *
*************************************************************
480:          *
481:
*!*************************************************************
**************
482:  *!
483:  *!        Function: _R9H0YJFR3
484:  *!
485:  *!      Called by: CTRL OIG.SPR
486:  *!
487:  *!           Calls: &OPC_SCR.PRG
488:  *!
489:
*!*************************************************************
**************
490:      FUNCTION _r9h0yjfr3      &&  OPC_SCR VALID
491:          #REGION 1
492:          DO &opc_scr
493:  *:  EOF: CTRL_OIG.ac1
```


Varios Archivo Nómina Rol de Pagos ReporTes
INGRESOS GRAVABLES EMPLEADOS DE AMBATO
Trans. Datos
Ayuda Matrículas
Val.Ropa/Trabajo
Otros Ingresos
Terminar
Codigo:
Nombre: MALDONADO V. ROSARIO
Cargo: CONTADOR AYUDANTE 2
Departamento: RECURSOS HUMANOS
Ciudad: AMBATO
Fecha : 12/05/1996
Ayuda Matriculas:
Ropa de Trabajo:
Actas Transaccionales:
Bonific. Vacaciones:
Infracciones Servicio:
Otros Ingr.gravables:
Dig
Excluyo
INS
NUM

```
    1:
*:*************************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\ING_GRAV.PRG
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 09/05/1996 at 20:12:12
    8:  *:
    9:  *:  Procs & Fncts: UNO
   10:  *:                : DOS
   11:  *:                : SAL_BARRAS
   12:  *:
   13:  *:          Set by: REG_INF             (procedure in
CTRL_OIG.SPR)
   14:  *:
   15:  *:           Calls: CABEZA              (procedure in
PERSONAL.PRG)
   16:  *:                : SECC()              (function in
PERSONAL.PRG)
   17:  *:
   18:  *:           Uses: PERSONAL.DBF              Alias: DGP
   19:  *:
   20:  *:       CDX files: PERSONAL.CDX
   21:  *:
   22:  *:       Documented 19:15:17
FoxDoc version 3.00a
   23:
*:*************************************************************************
*************
   24:      PARAMETER manio, vmes, veiu
   25:
   26:     IF USED("sec")
   27:        SELECT sec
   28:        SET ORDER TO TAG "secc_cod"
   29:     ELSE
   30:        SELECT 0
   31:        USE (LOCFILE("srol\dbfs\seccion.dbf","DBF","Buscar
seccion?"))
   32:           AGAIN ALIAS sec
   33:           ORDER TAG "secc_cod"
   34:     ENDIF
   35:     *Use Seccion order secc_cod alias Sec in 0
   36:     *Use AC_IMPRTA ORDER COD_PER Alias ACR in 0
   37:     *Use O_ing_gr order cod_per alias oig in 0
   38:     USE personal ORDER cod_per ALIAS dgp IN 0
   39:     SELECT sec
   40:     SET RELATION TO cod_ciu INTO sat
   41:     SELECT dgp
   42:     SET RELATION TO num_sec INTO sec
   43:     SELECT oig
   44:     SET FILTER TO num_ciu = veiu AND anio = manio AND mes
```

```
= vmes
   45:      SET RELATION TO cod_per INTO dgp
   46:
   47:      STOR sst.ciudad TO ciudad, msis
   48:      STOR ' FONT  "N",10 style "b"' TO fsdesc && styl and
font of say coment
   49:      STOR '  ' TO fscampc && styl and font of say CHARACTER
fields
   50:      STOR '  ' TO fscampn && styl and font of say NUMERIC
fields
   51:      DEFINE WINDOW esta FROM 0,0 TO 50,90 PANEL TITLE "
INGRESOS GRAVABLES EMPLEADOS DE " + msis     NOZOOM
   52:      DEFINE WINDOW pregunta FROM 10,58 TO 26, 76 DOUBLE
SHADOW IN WIND esta FLOAT ZOOM
   53:      ACTIVATE WINDOW esta
   54:      REGIONAL modo,salto
   55:      STOR 1 TO modo
   56:      STOR 'enab' TO salto
   57:      STORE SPACE(1) TO
m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14,m15,m16,m17,m18,mc
ont,fij_var
   58:      fil_ciu = 1
   59:      laso = .T.
   60:      DO cabeza
   61:      STOR 'Rubros' TO rtes
   62:      @2.5,3 GET rtes FUNC '^ \<Rubros:\<General:\<Totales':
   63:         SIZE 1.3,20 FONT 'N',8 STYLE "IB":
   64:         DISA
   65:      DO WHILE laso
   66:         @0,0 TO 44,18 DOUBLE COLOR SCHEM 10,4
   67:         @0,0 FILL TO 44,18  COLOR GR+/B
   68:         @ 1,2 GET m18:
   69:            PICTURE "@*N Trata. \<Datos     " ;
   70:            SIZE 1.3,22 FONT 'N',10 STYLE "IB";
   71:            VALID secc(20)
   72:
   73:         @7,2 GET m1;
   74:            PICTURE "@*N Ayuda \<Matricula " ;
   75:            SIZE 1.2,22 FONT 'R',10 STYLE "BI";
   76:            VALID secc(1) WHEN salto = 'enab'
   77:         @ 10,2 GET m2;
   78:            PICTURE "@*N Val.\<Ropa/Trabajo" :
   79:            SIZE 1.2,22 FONT 'R',10 STYLE "BI";
   80:            VALID secc(2) WHEN salto = 'enab'
   81:         @ 13,2 GET m3;
   82:            PICTURE "@*N Otros \<Ingresos    " ;
   83:            SIZE 1.2,22 FONT 'R',10 STYLE "BI";
   84:            VALID secc(3) WHEN salto = 'enab'
   85:
   86:         @ 18,2 GET m13;
   87:            PICTURE "@*N      \<Terminar     " ;
   88:            SIZE 1.5,22 FONT 'R',10 STYLE "BI";
   89:            VALID secc(13) &salto
   90:         READ  CYCLE
```

```
 91:    ENDDO
 92:  RETURN
 93:
 94:
*!*************************************************************************
**************
 95:  *!
 96:  *!        Function: SECC
 97:  *!
 98:  *!      Called by: PERSONAL.PRG
 99:  *!               : MEDIDOR.PRG
100:  *!               : ING_GRAV.PRG
101:  *!               : EGR_VARI.PRG
102:  *!
103:  *!           Calls: INGRESAR          (procedure in
PERSONAL.PRG)
104:  *!                : CAMBIOS           (procedure in
PERSONAL.PRG)
105:  *!                : SIGUIENTE         (procedure in
AP_PRMCT.SPR)
106:  *!                : ANTERIOR          (procedure in
AP_PRMCT.SPR)
107:  *!                : BUSCAR            (procedure in
AACT_SST.SPR)
108:  *!                : ACTIVAR           (procedure in
PERSONAL.PRG)
109:  *!                : EXCLUIR           (procedure in
PERSONAL.PRG)
110:  *!                : CAMBIO_SIS        (procedure in
PERSONAL.PRG)
111:  *!                : CAPTURAR          (procedure in
PERSONAL.PRG)
112:  *!                : JUBILAR           (procedure in
PERSONAL.PRG)
113:  *!                : COPIA_D           (procedure in
MEDIDOR.PRG)
114:  *!                : ACT_MEDIDORES.PRG
115:  *!                : UNO               (procedure in
ING_GRAV.PRG)
116:  *!                : TOTALES           (procedure in
VF_IMPRTA.SPR)
117:  *!
118:
*!*************************************************************************
**************
119:      FUNCTION secc
120:          PARAMETERS K
121:        DO CASE
122:        CASE K= 20
123:             STOR 'Rubros' TO rtes
124:             @2.5,3 GET rtes FUNC '^
\<Rubros:\<General:\<Totales';
125:                  SIZE 1.3,20 FONT 'N',8 STYLE "IB";
126:                  ENAB
```

```
127:                READ
128:              DO CASE
129:           CASE rtes='Rubros'
130:                modo =1
131:                STOR 'enab' TO salto
132:           CASE rtes='General'
133:                modo = 2
134:                STOR 'disa' TO salto
135:                DO uno WITH K
136:           CASE rtes='Totales'
137:                modo = 2
138:                STOR 'disa' TO salto
139:                DO totales
140:              ENDCASE
141:         CASE K = 13
142:              laso = .F.
143:              RELE WINDOW esta
144:              SELE dgp
145:              USE
146:              SELE sec
147:              USE
148: <=======    RETURN
149:         OTHERWISE
150:              DO uno WITH K
151:         ENDCASE
152: <=====   RETURN
153:
154:
155:
*!*************************************************************
**************
156:  *!
157:  *!        Procedure: UNO
158:  *!
159:  *!        Called by: SECC()              (function in
PERSONAL.PRG)
160:  *!
161:  *!            Calls: DESFLEGA            (procedure in
PERSONAL.PRG)
162:  *!
163:
*!*************************************************************
**************
164:      PROCEDURE uno
165:          PARAMETERS k1
166:          STORE SPACE(6) TO varcod
167:          STOR 0 TO varcod1
168:          STOR SPAC(6) TO varcod
169:        DO WHILE LAST()!= 27
170:           DO WHILE .T.
171:              @ 1,37 GET varcod1 PICT '9999' &fscampo
172:              READ
173:              varcod= ALLT(STR(varcod1))
174:              A = LEN(varcod)
```

```
175:              auxvc = ALLT(STR(vciu))
176:              FOR i = A TO 4
177:                  auxvc = auxvc + '0'
178:              ENDFOR
179:              varcod = auxvc +  varcod
180:              IF LASTKEY() = 27
181:                  RETURN
182:              ENDIF
183:              SEEK varcod
184:              IF .NOT. FOUND()
185:                  SELECT dgp
186:                  SEEK varcod
187:                  IF FOUND()
188:                      SELE oig
189:                      APPE BLANK
190:                      REPLACE oig.anio WITH manio;
191:                          oig.mes WITH vmes;
192:                          oig.num_ciu WITH vciu;
193:                          oig.cod_per WITH varcod
194:                  ELSE
195:                      WAIT "Registro No Existe....!" WINDOW
196:                  ENDIF
197:              ENDIF
198:              EXIT
199:          ENDDO
200:          DO desplega WITH k1
201:      ENDDO
202:      laso1 = .T.
203:  RETURN
204:
205:
206:
*!*************************************************************
**************
207:  *!
208:  *!       Procedure: DOS
209:  *!
210:  *!           Calls: DESPLEGA          (procedure in
PERSONAL.PRG)
211:  *!
212:
*!*************************************************************
**************
213:  PROCEDURE DOS
214:      PARAMETERS k1
215:      STORE SPACE(6) TO varcod
216:      STORE SPACE(3) TO varcod1
217:      DEACT WIND pregunta
218:      STOR 0 TO varcod1
219:      STOR SPAC(6) TO varcod
220:      DO WHILE .T.
221:          @ 1,37 GET varcod1 PICT '9999' &fscampo
222:          READ
223:          varcod= ALLT(STR(varcod1))
```

```
  224:          A = LEN(varcod)
  225:          auxvc = ALLT(STR(vciu))
  226:        FOR i = A TO 4
  227:            auxvc = auxvc + '0'
  228:        ENDFOR
  229:          varcod = auxvc +  varcod
  230:        IF LASTKEY() = 27
  231: <=============RETURN
  232:        ENDIF
  233:          SEEK varcod
  234:        IF .NOT. FOUND()
  235:            WAIT "Registro No Existe....!" WINDOW
  236:        ELSE
  237:      v=========EXIT
  238:        ENDIF
  239:      ENDDO
  240:        DEACT WIND pregunta
  241:        DO desplega WITH k1
  242: <=========RETURN
  243:
  244:
*!*************************************************************************
**************
  245:  *!
  246:  *!       Procedure: TOTALES
  247:  *!
  248:  *!       Called by: SECC()              (function in
PERSONAL.PRG)
  249:  *!                : TRAT_DATOS()        (function in
DA_ASISG.PRG)
  250:  *!
  251:  *!           Calls: SIST_TOT            (procedure in
EGR_VARI.PRG)
  252:  *!
  253:
*!*************************************************************************
*************
  254:      PROCEDURE totales
  255:        DO WHILE LAST() != 27
  256:          @ 1,35 SAY SPAC(8)&fsdesc
  257:          @ 1.5,55 SAY SPAC(25)
  258:          @ 1.5,55  SAY 'Totales Generales ' + msis
  259:          @ 3,35 SAY SPAC(50)&fscampc
  260:          @ 5.35  SAY '                              ' &fscampc
  261:          SUM ALL ac_ayumatr TO ayma
  262:          SUM ALL ac_roptrab TO rotr
  263:          SUM ALL ac_acttran TO actr
  264:          SUM ALL ac_bonvac  TO bova
  265:          SUM ALL ac_infserv TO INSE
  266:          SUM ALL ac_oting   TO otin
  267:
  268:          @ 16,46 SAY ayma &fscampn pict '999,999,999'
FUNC '@z'
  269:          @ 18,46 SAY rotr &fscampn pict '999,999,999'
```

```
FUNC '@s'
  270:          @ 20,46 SAY actr &fscampn pict'999,999,999'
FUNC '@s'
  271:          @ 22,46 SAY bova &fscampn pict'999,999,999'
FUNC '@s'
  272:          @ 24,46 SAY INSE &fscampn pict'999,999,999'
FUNC '@s'
  273:          @ 26,46 SAY otin &fscampn pict'999,999,999'
FUNC '@s'
  274:
  275:          @36,25 TO 40,70 DOUBLE
  276:          @ 38,30 GET mcont;
  277:             PICTURE "@* \<Continuar" ;
  278:             SIZE 1.2,20 FONT 'R',10 STYLE "BI";
  279:             MESSA 'Contin·a con otro total'
  280:          @ 38,50 GET mcont;
  281:             PICTURE "@* \<Men·" ;
  282:             SIZE 1.2,20 FONT 'R',10 STYLE "BI";
  283:             MESSA 'Regresa el menu de opciones'
  284:          READ
  285:          @ 1.35 SAY SPAC(8)&fsdesc
  286:          @ 1.5,55 SAY SPAC(25)
  287:          @ 1.5,55  SAY '
  288:          @ 3,35 SAY SPAC(50)&fscampc
  289:          @ 5,35  SAY '                                  ' &fscampc
  290:          @36,25 TO 40,70 CLEAR
  291:          IF mcont = 'Continuar'
  292:             LOOP
  293:          ELSE
  294:             EXIT
  295:          ENDIF
  296:       ENDDO
  297:       RETURN
  298:
  299:          *RFturn
  300:
  301:
*!*************************************************************************
*************
  302:  *!
  303:  *!        Procedure: SAL_BARRAS
  304:  *!
  305:
*!*************************************************************************
*************
  306:     PROCEDURE sal_barras
  307:          DEACT WIND pregunta
  308:          lcsol=.F.
  309:       RETURN
  310:
  311:
*!*************************************************************************
*************
```

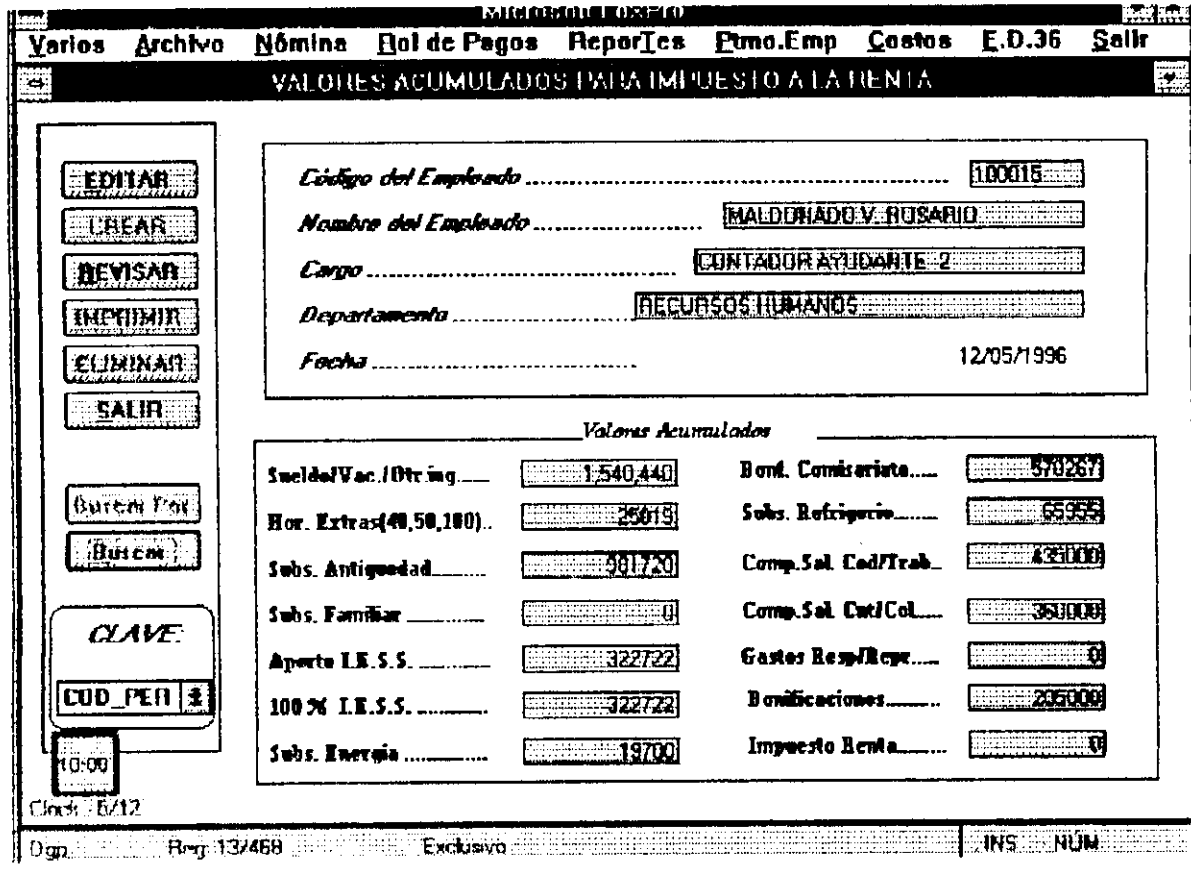
```
  312:  *!
  313:  *!        Procedure: DESPLEGA
  314:  *!
  315:  *!        Called by: BUSCAR              (procedure in
AACT_SST.SPR)
  316:  *!                 : CANCELAR            (procedure in
AACT_SST.SPR)
  317:  *!                 : ANTERIOR            (procedure in
AP_PRMCT.SPR)
  318:  *!                 : SIGUIENTE           (procedure in
AP_PRMCT.SPR)
  319:  *!                 : INGRESAR            (procedure in
PERSONAL.PRG)
  320:  *!                 : COD_PERSONA         (procedure in
PERSONAL.PRG)
  321:  *!                 : CAMBIOS             (procedure in
PERSONAL.PRG)
  322:  *!                 : ACTIVAR             (procedure in
PERSONAL.PRG)
  323:  *!                 : EXCLUIR             (procedure in
PERSONAL.PRG)
  324:  *!                 : CAMBIO_SIS          (procedure in
PERSONAL.PRG)
  325:  *!                 : CAPTURAR            (procedure in
PERSONAL.PRG)
  326:  *!                 : UNO                 (procedure in
ING_GRAV.PRG)
  327:  *!                 : DOS                 (procedure in
ING_GRAV.PRG)
  328:  *!
  329:  *!           Calls: VAL_SEC()           (function in
PERSONAL.PRG)
  330:  *!                 : VAL_FUE()           (function in
PERSONAL.PRG)
  331:  *!
  332:
*!*************************************************************
**************
  333:      PROCEDURE despliega
  334:          PARAMETERS k1
  335:          SEEK varcod
  336:          @ 1,35 SAY SPAC(8)&fsdesc
  337:          @ 1,35 SAY varcod &fscampo
  338:          @ 1,5,55 SAY SPAC(25)
  339:          @ 1,5,55  SAY ALLT(dgp.nombre)
  340:          @ 3,35  SAY dgp.cargo
  341:          *VarDep = Num_Dep
  342:          *VarCiu = Num_Ciu
  343:
  344:          @ 5,35 SAY SPAC(50)&fscampo
  345:          @ 5,35 SAY sec.nombre &fscampo
  346:          @ 7,35  SAY sst.ciudad &fscampo
  347:          @ 9,35  SAY DATE() &fscampo
  348:
```

```
          @ 16.46 SAY ac_ayumatr &fscampn pict '999,999,999' FUNC '@z'
          @ 18.46 SAY ac_roptrab &fscampn pict '999,999,999' FUNC '@z'
          @ 20.46 SAY ac_acttran &fscampn pict '999,999,999' FUNC '@z'
          @ 22.46 SAY ac_bonvac &fscampn pict '999,999,999' FUNC '@z'
          @ 24.46 SAY ac_infserv &fscampn pict '999,999,999' FUNC '@z'
          @ 26.46 SAY ac_oting &fscampn pict '999,999,999' FUNC '@z'

          SET CURSOR ON
          DO CASE
          CASE  modo  =  1
             DO CASE
             CASE k1 = 1
                  @ 16.46  GET ac_ayumatr  &fscampn pict '999,999,999'
             CASE k1 = 2
                  @ 18.46 GET ac_roptrab &fscampn pict '999,999,999'
             CASE k1 = 3
                  @ 20.46 GET ac_acttran &fscampn pict '999,999,999'
                  @ 22.46 GET ac_bonvac &fscampn pict '999,999,999'
                  @ 24.46 GET ac_infserv &fscampn pict '999,999,999'
                  @ 26.46 GET ac_oting &fscampn pict '999,999,999'
             ENDCASE
          CASE modo  = 2
               @ 16.46 GET ac_ayumatr &fscampn pict '999,999,999'
               @ 18.46 GET ac_roptrab &fscampn pict '999,999,999'
               @ 20.46 GET ac_acttran &fscampn pict '999,999,999'
               @ 22.46 GET ac_bonvac &fscampn pict '999,999,999'
               @ 24.46 GET ac_infserv &fscampn pict '999,999,999'
               @ 26.46 GET ac_oting &fscampn pict '999,999,999'
          ENDCASE
          READ
          RETU


*****************************************************************************
          *!
```

```
  333:  *!       Procedure: CABEZA
  384:  *!
  385:  *!       Called by: PERSONAL.PRG
  386:  *!                : EDITAR                (procedure in
CTRL_RES.SPR)
  387:  *!                : ING_GRAV.PRG
  388:  *!                : EGR_VARI.PRG
  389:  *!
  390:  *!           Calls: CIUDAD                (procedure in
DSC_GRTR.SPR)
  391:  *!                : MENSAJE1              (procedure in
DSC_GRTR.SPR)
  392:  *!
  393:
*!*************************************************************************
**************
  394:      PROCEDURE cabeza
  395:          @ 1,20  SAY "Codigo:"&fsdesc
  396:          @ 1,45  SAY "Nombre:"&fsdesc
  397:          @ 3,20  SAY "Cargo:"&fsdesc
  398:          @ 5,20  SAY "Departamento:"&fsdesc
  399:          @ 7,20  SAY "Ciudad:"&fsdesc
  400:          @ 9,20  SAY "Fecha :"&fsdesc
  401:          @ 10,5,20  SAY REPLICATE("=",60)
  402:
  403:          @ 16,23  SAY "Ayuda Matriculas: "&fsdesc
  404:          @ 18,23  SAY "Ropa de Trabajo: "&fsdesc
  405:          @ 20,23  SAY "Actas Transaccionales: "&fsdesc
  406:          @ 22,23  SAY "Bonific. Vacaciones: "&fsdesc
  407:          @ 24,23  SAY "Infracciones Servicio: "&fsdesc
  408:          @ 26,23  SAY "Otros ingr.gravables: "&fsdesc
  409:          RETURN
  410:  *:  EOF: ING_GRAV.act
```


Varios Archivo Nómina Rol de Pagos ReporTes Pmo.Emp Costos E.D.36 Salir
VALORES ACUMULADOS PARA IMPUESTO A LA RENTA
EDITAR
CREAR
REVISAR
IMPRIMIR
CURINAR
SALIR
Buscar Por
Buscar
CLAVE
COD_PER
10:00
Clock : 6/12
Código del Empleado 100015
Nombre del Empleado ..... MALDONADO V. ROSARIO
Cargo ..... CONTADOR AYUDANTE 2
Departamento ..... RECURSOS HUMANOS
Fecha 12/05/1996
Valores Acumulados
Sueldo/Vac./Otr.ing.... 1'540,440
Hor. Extras(40,50,100).. 25019
Subs. Antiguedad..... 581720
Subs. Familiar 0
Aporte I.E.S.S. 322722
100 % I.E.S.S. 322722
Subs. Energia 19700
Bond. Comisariato..... 570267
Subs. Refrigerio..... 65955
Comp.Sal. Cod/Trab.. 420000
Comp.Sal. Cod/Col.... 360000
Gastos Resp/Repr..... 0
Bonificaciones..... 205000
Impuesto Renta..... 0
Dgn Reg: 13/468 Exclusivo INS NUM

```
     1:
*:*************************************************************
**************
     2:  *:
     3:  *: Procedure file: C:\DOC\AC_IMPRT.SPR
     4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
     5:  *:          Author: EL GRUPO FREDY - HUGO - MERCEDES
     6:  *:       Copyright (c) 10-12-1995. Sisrol Ambato - Ecuador
     7:  *:  Last modified: 26/02/1996 at 22:43:34
     8:  *:
     9:  *:  Procs & Fncts: _R891CPJSL()
    10:  *:                : _R891CPJVK()
    11:  *:                : _R891CPJYC()
    12:  *:
    13:  *:          Set by: EMPRESA.MPR
    14:  *:
    15:  *:           Calls: ACT_KEYS            (procedure in
AACT_SST.SPR)
    16:  *:                : _R891CPJSL()        (function in
AC_IMPRT.SPR)
    17:  *:                : _R891CPJVK()        (function in
AC_IMPRT.SPR)
    18:  *:                : _R891CPJYC()        (function in
AC_IMPRT.SPR)
    19:  *:
    20:  *:       Documented 19:14:47
FoxDoc version 3.00a
    21:
*:*************************************************************
**************
    22:      *
************************************************************
    23:      *        *
    24:      *        * 26/02/1996            AC_IMPRT.SPR
 22:43:32
    25:      *        *
    26:      *
************************************************************
    27:      *        *
    28:      *        * Nombre del autor
    29:      *        *
    30:      *        * Copyright (c) 1996 Nombre de la empresa
    31:      *        * Direccisn
    32:      *        * Ciudad.      C.P.
    33:      *        * Pa$s
    34:      *        *
    35:      *        * Description:
    36:      *        * Este programa lo ha generado automßticamente
GENSCRN.
    37:      *        *
    38:      *
************************************************************
    39:
```

```
   40:
   41:     *
*************************************************************
   42:     *        *
   43:     *        *  AC_IMPRT/Windows Código de configuración -
SECCION 1
   44:     *        *
   45:     *
*************************************************************
   46:     *
   47:
   48:     #REGION 1
   49:     **********
   50:     REGIONAL opc_scr, opc_busc, indices, tag_bus,
atrat_dat, trat_dat,opc_ind,;
   51:        editando
   52:     STORE '' TO opc_scr, opc_busc, trat_dat, opc_ind
   53:     STOR .F. TO editando
   54:     **********
   55:
   56:     #REGION 0
   57:     REGIONAL m.currarea, m.talkstat, m.compstat
   58:
   59:   ┌─IF SET("TALK") = "ON"
   60:   │    SET TALK OFF
   61:   │    m.talkstat = "ON"
   62:   ├─ELSE
   63:   │    m.talkstat = "OFF"
   64:   └─ENDIF
   65:     m.compstat = SET("COMPATIBLE")
   66:     SET COMPATIBLE FOXPLUS
   67:
   68:     m.rborder = SET("READBORDER")
   69:     SET readborder ON
   70:
   71:     m.currarea = SELECT()
   72:
   73:
   74:     *
*************************************************************
   75:     *        *
   76:     *        *  AC_IMPRT/Windows Bases de datos, índices,
relaciones
   77:     *        *
   78:     *
*************************************************************
   79:     *
   80:
   81:   ┌─IF USED("dgp")
   82:   │    SELECT dgp
   83:   │    SET ORDER TO TAG "cod_per"
   84:   ├─ELSE
   85:   │    SELECT 0
   86:   │    USE (LOCFILE("dbfs\personal.dbf","DBF","Buscar
```

```
personal?")):
    87:     |       AGAIN ALIAS dgp ;
    88:     |       ORDER TAG "cod_per"
    89:     └─ENDIF
    90:
    91:     ┌─IF USED("sec")
    92:     |   SELECT sec
    93:     |   SET ORDER TO TAG "secc_cod"
    94:     ├─ELSE
    95:     |   SELECT 0
    96:     |   USE (LOCFILE("dbfs\seccion.dbf","DBF","Buscar
seccion?")):
    97:     |       AGAIN ALIAS sec ;
    98:     |       ORDER TAG "secc_cod"
    99:     └─ENDIF
   100:
   101:     ┌─IF USED("pue")
   102:     |   SELECT pue
   103:     |   SET ORDER TO TAG "pues_cod"
   104:     ├─ELSE
   105:     |   SELECT 0
   106:     |   USE (LOCFILE("dbfs\puestos.dbf","DBF","Buscar
puestos?")):
   107:     |       AGAIN ALIAS pue ;
   108:     |       ORDER TAG "pues_cod"
   109:     └─ENDIF
   110:
   111:     ┌─IF USED("acr")
   112:     |   SELECT acr
   113:     |   SET ORDER TO TAG "cod_per"
   114:     ├─ELSE
   115:     |   SELECT 0
   116:     |   USE (LOCFILE("dbfs\ac_imprt.dbf","DBF","Buscar
ac_imprt?")):
   117:     |       AGAIN ALIAS acr ;
   118:     |       ORDER TAG "cod_per"
   119:     └─ENDIF
   120:
   121:     SELECT dgp
   122:
   123:     SET RELATION OFF INTO acr
   124:     SET RELATION TO cod_per INTO acr ADDITIVE
   125:
   126:     SELECT dgp
   127:
   128:     SET RELATION OFF INTO pue
   129:     SET RELATION TO dgp.cod_pues INTO pue ADDITIVE
   130:
   131:     SELECT dgp
   132:
   133:     SET RELATION OFF INTO sec
   134:     SET RELATION TO dgp.num_sec INTO sec ADDITIVE
   135:
   136:     SELECT dgp
```

```
*
*****************************************************************
*       *
*       *                    Windows Definiciones de ventana
*       *
*       *
*****************************************************************
*

IF NOT WEXIST("ac_imprt") ;
      OR UPPER(WTITLE("AC_IMPRT")) == "AC_IMPRT.PJX" ;
      OR UPPER(WTITLE("AC_IMPRT")) == "AC_IMPRT.SCX" ;
      OR UPPER(WTITLE("AC_IMPRT")) == "AC_IMPRT.MNX" ;
      OR UPPER(WTITLE("AC_IMPRT")) == "AC_IMPRT.PRG" ;
      OR UPPER(WTITLE("AC_IMPRT")) == "AC_IMPRT.FRX" ;
      OR UPPER(WTITLE("AC_IMPRT")) == "AC_IMPRT.QPR"
   DEFINE WINDOW ac_imprt ;
      AT  0.000, 0.000  ;
      SIZE 31.154,125.800 ;
      TITLE "VALORES ACUMULADOS PARA IMPUESTO A LA
RENTA" ;
      FONT "MS Sans Serif", 8 ;
      FLOAT ;
      CLOSE ;
      MINIMIZE ;
      SYSTEM
   MOVE WINDOW ac_imprt CENTER
ENDIF


*
*****************************************************************
*       *
*       *  AC_IMPRT/Windows Código de configuración -
SECCION 2
*       *
*
*****************************************************************
*

#REGION 1
**********
DO act_keys
SELE dgp
GOTO TOP
DIMENSION indices(20)
FOR i = 2 TO 20
   IF EMPTY(TAG(i-1))
      i = i - 1
      DIMENSION indices(i-1)
      EXIT
   ELSE
      indices(i-1) = TAG(i-1)
```

```
185:     ||  └─ENDIF
186:     └─ENDFOR
187:        tag_busc  indices(1)
188:        opc_ind = indices(1)
189:        SET ORDER TO &tag_busc
190:        SHOW GETS DISABLE
191:
192:
193:        *
*************************************************************
194:        *           *
195:        *           *          AC_IMPRT/Windows Distribuci¢n de
pantalla
196:        *           *
197:        *
*************************************************************
198:        *
199:
200:        #REGION 1
201:     ┌─IF WVISIBLE("ac_imprt")
202:     │     ACTIVATE WINDOW ac_imprt SAME
203:     ├─ELSE
204:     │     ACTIVATE WINDOW ac_imprt NOSHOW
205:     └─ENDIF
206:        @ 0.923,2.200 TO 28.231,21.400 :
207:           PEN 1, 8
208:        @ 1.346,26.400 TO 13.000,121.400 :
209:           PEN 1, 8
210:        @ 11.154,26.600 TO 11.154,26.600 :
211:           PEN 1, 8
212:        @ 14.769,25.200 TO 29.615,122.200 :
213:           PEN 1, 8
214:        @ 3.615,4.800 GET ope_scr :
215:           PICTURE "@*VN
\<EDITAR:\\CREAR:\<REVISAR:\\IMPRIMIR:\\ELIMINAR:\<SALIR" :
216:           SIZE 1.615,12.167,0.385 :
217:           DEFAULT 1 :
218:           FONT "MS Sans Serif", 8 :
219:           STYLE "B" :
220:           VALID _r891cpjsl()
221:        @ 16.615,4.800 GET opc_busc :
222:           PICTURE "@*VN \\Buscar Por:\<Buscar " :
223:           SIZE 1.692,12.167,0.308 :
224:           DEFAULT 1 :
225:           FONT "MS Sans Serif", 8 :
226:           STYLE "B" :
227:           VALID _r891cpjvk()
228:        @ 25.077,3.800 GET opc_ind :
229:           PICTURE "@^" :
230:           FROM indices :
231:           SIZE 1.538,13.833 :
232:           DEFAULT 1 :
233:           FONT "MS Sans Serif", 8 :
234:           STYLE "B" :
```

```
  235:          VALID _r891cpjvc()
  236:       @ 2.846,29.800 SAY "Código del Empleado
......................................................" +
CHR(13) + :
  237:          "Nombre del Empleado ....................." +
CHR(13) + :
  238:          "Cargo ..................................................."
+ CHR(13) + :
  239:          "Departamento ..........................." + CHR(13)
+ :
  240:          "Fecha ....................................." :
  241:          SIZE 9.000,59.667, 1.000 ;
  242:          FONT "MS Sans Serif", 8 ;
  243:          STYLE "BIT"
  244:       @ 2.846,102.800 GET dgp.cod_per :
  245:          SIZE 1.000,11.200 ;
  246:          DEFAULT " " ;
  247:          FONT "MS Sans Serif", 8 ;
  248:          PICTURE "@K" ;
  249:          WHEN .F.
  250:       @ 4.615,76.400 GET dgp.nombre :
  251:          SIZE 1.000,37.600 ;
  252:          DEFAULT " " ;
  253:          FONT "MS Sans Serif", 8 ;
  254:          PICTURE "@K" ;
  255:          WHEN .F.
  256:       @ 8.538,66.800 GET sec.nombre :
  257:          SIZE 1.000,47.200 ;
  258:          DEFAULT " " ;
  259:          FONT "MS Sans Serif", 8 ;
  260:          PICTURE "@K" ;
  261:          WHEN .F.
  262:       @ 6.615,73.200 GET dgp.cargo :
  263:          SIZE 1.000,40.800 ;
  264:          DEFAULT " " ;
  265:          FONT "MS Sans Serif", 8 ;
  266:          PICTURE "@K" ;
  267:          WHEN .F.
  268:       @ 10.769,101.000 SAY DATE() ;
  269:          SIZE 1.000,13.000 ;
  270:          FONT "MS Sans Serif", 8
  271:       @ 15.769,54.400 GET acr.ac_sueldo :
  272:          SIZE 1.000,15.800 ;
  273:          DEFAULT 0 ;
  274:          FONT "MS Sans Serif", 8 ;
  275:          PICTURE "@KZ 999,999,999" ;
  276:          WHEN editando AND dgp.estado = 1
  277:       @ 17.615,54.400 GET acr.ac_hextras :
  278:          SIZE 1.000,15.800 ;
  279:          DEFAULT 0 ;
  280:          FONT "MS Sans Serif", 8 ;
  281:          PICTURE "@K" ;
  282:          WHEN editando AND dgp.estado = 1
  283:       @ 19.692,54.400 GET acr.ac_subant ;
```

```
284:          SIZE 1.000,15.800 ;
285:          DEFAULT 0 ;
286:          FONT "MS Sans Serif", 8 ;
287:          PICTURE "@K" ;
288:          WHEN editando AND dgp.estado = 1
289:       @ 21.769,54.400 GET acr.ac_subfam ;
290:          SIZE 1.000,15.800 ;
291:          DEFAULT 0 ;
292:          FONT "MS Sans Serif", 8 ;
293:          PICTURE "@K" ;
294:          WHEN editando AND dgp.estado = 1
295:       @ 23.769,54.400 GET acr.ac_apiess ;
296:          SIZE 1.000,15.800 ;
297:          DEFAULT 0 ;
298:          FONT "MS Sans Serif", 8 ;
299:          PICTURE "@K" ;
300:          WHEN editando AND dgp.estado = 1
301:       @ 25.769,54.400 GET acr.ac_100iess ;
302:          SIZE 1.000,15.800 ;
303:          DEFAULT 0 ;
304:          FONT "MS Sans Serif", 8 ;
305:          PICTURE "@K" ;
306:          WHEN editando AND dgp.estado = 1
307:       @ 27.769,54.400 GET acr.ac_subener ;
308:          SIZE 1.000,15.800 ;
309:          DEFAULT 0 ;
310:          FONT "MS Sans Serif", 8 ;
311:          PICTURE "@K" ;
312:          WHEN editando AND dgp.estado = 1
313:       @ 23.692,102.200 GET acr.ac_gastos ;
314:          SIZE 1.000,13.600 ;
315:          DEFAULT 0 ;
316:          FONT "MS Sans Serif", 8 ;
317:          PICTURE "@K" ;
318:          WHEN editando AND dgp.estado = 1
319:       @ 15.615,102.200 GET acr.ac_bcomis ;
320:          SIZE 1.000,13.600 ;
321:          DEFAULT 0 ;
322:          FONT "MS Sans Serif", 8 ;
323:          PICTURE "@K" ;
324:          WHEN editando AND dgp.estado = 1
325:       @ 17.462,102.200 GET acr.ac_subref ;
326:          SIZE 1.000,13.600 ;
327:          DEFAULT 0 ;
328:          FONT "MS Sans Serif", 8 ;
329:          PICTURE "@K" ;
330:          WHEN editando AND dgp.estado = 1
331:       @ 19.385,102.200 GET acr.ac_compct ;
332:          SIZE 1.000,13.600 ;
333:          DEFAULT 0 ;
334:          FONT "MS Sans Serif", 8 ;
335:          PICTURE "@K" ;
336:          WHEN editando AND dgp.estado = 1
337:       @ 21.769,102.200 GET acr.ac_compcc ;
```

```
338:          SIZE 1.000,13.600 :
339:          DEFAULT 0 :
340:          FONT "MS Sans Serif", 8 :
341:          PICTURE "@K" :
342:          WHEN editando AND dgp.estado = 1
343:       @ 27.538,102.200 GET acr.ac_imprent :
344:          SIZE 1.000,13.600 :
345:          DEFAULT 0 :
346:          FONT "MS Sans Serif", 8 :
347:          PICTURE "@K" :
348:          WHEN editando AND dgp.estado = 1
349:       @ 13.846,60.600 SAY 'Valores Acumulados' :
350:          SIZE 0.938,25.200 :
351:          FONT "Futura Md BT", 9 :
352:          STYLE "BI"
353:       @ 15.769,26.400 SAY "Sueldo/Vac./Otr.ing....." +
CHR(13) + :
354:            "" + CHR(13) + :
355:            "Hor. Extras(40,50,100).." + CHR(13) + :
356:            "" + CHR(13) + :
357:            "Subs. Antiguedad....." + CHR(13) + :
358:            "" + CHR(13) + :
359:            "Subs. Familiar ....." + CHR(13) + :
360:            "" + CHR(13) + :
361:            "Aporte I.E.S.S. ....." + CHR(13) + :
362:            "" + CHR(13) + :
363:            "100 %  I.E.S.S. ....." + CHR(13) + :
364:            "" + CHR(13) + :
365:            "Subs. Energia ....." + CHR(13) + :
366:            "" :
367:            SIZE 13.000,21.333, 0.000 :
368:            FONT "MS Serif", 8 :
369:            STYLE "BT"
370:       @ 23.692,77.200 SAY "Gastos Resp/Repr....." :
371:            FONT "MS Serif", 8 :
372:            STYLE "BT"
373:       @ 15.615,76.800 SAY " Bonf. Comisariato....." :
374:            FONT "MS Serif", 8 :
375:            STYLE "BT"
376:       @ 17.462,76.800 SAY " Subs. Refrigerio....." :
377:            FONT "MS Serif", 8 :
378:            STYLE "BT"
379:       @ 19.615,76.800 SAY " Comp.Sal. Cod.Trab.." :
380:            FONT "MS Serif", 8 :
381:            STYLE "BT"
382:       @ 21.692,76.800 SAY " Comp.Sal. Cnt/Col....." :
383:            FONT "MS Serif", 8 :
384:            STYLE "BT"
385:       @ 27.538,78.000 SAY "Impuesto Renta....." :
386:            FONT "MS Serif", 8 :
387:            STYLE "BT"
388:       @ 22.462,6.600 SAY "CLAVE:" :
389:            FONT "MS Sans Serif", 10 :
390:            STYLE "BIT"
```

```
391:      @ 21.846,3.000 TO 27.231,20.800 :
392:         PEN 1, 8 :
393:         STYLE "16"
394:      @ 25.538,102.200 GET acr.ac_bonifi :
395:         SIZE 1.000,13.600 :
396:         DEFAULT 0 :
397:         FONT "MS Sans Serif", 8 :
398:         PICTURE "@K" :
399:         WHEN editando AND dgp.estado = 1
400:      @ 25.538,78.000 SAY "Bonificaciones....." :
401:         FONT "MS Serif", 8 :
402:         STYLE "BT"
403:
404:     IF NOT WVISIBLE("ac_imprt")
405:        ACTIVATE WINDOW ac_imprt
406:     ENDIF
407:
408:      READ CYCLE :
409:         VALID .F.
410:
411:      RELEASE WINDOW ac_imprt
412:
413:      *
**************************************************************
414:      *        *
415:      *        *              Windows Cerrando bases de datos
416:      *        *
417:      *
**************************************************************
418:      *
419:
420:     IF USED("dgp")
421:        SELECT dgp
422:        USE
423:     ENDIF
424:
425:     IF USED("sec")
426:        SELECT sec
427:        USE
428:     ENDIF
429:
430:     IF USED("pue")
431:        SELECT pue
432:        USE
433:     ENDIF
434:
435:     IF USED("acr")
436:        SELECT acr
437:        USE
438:     ENDIF
439:
440:      SELECT (m.currarea)
441:
442:
```

```
  443:      #REGION 0
  444:
  445:      SET readborder &rborder
  446:
  447:     ┌─IF m.talkstat = "ON"
  448:     │     SET TALK ON
  449:     └─ENDIF
  450:     ┌─IF m.compstat = "ON"
  451:     │     SET COMPATIBLE ON
  452:     └─ENDIF
  453:
  454:
  455:      *
**************************************************************
  456:      *        *
  457:      *        *            AC_IMPRT/Windows Código de
limpieza
  458:      *        *
  459:      *
**************************************************************
  460:      *
  461:
  462:      #REGION 1
  463:      ****************
  464:
  465:
  466:      *
**************************************************************
  467:      *        *
  468:      *        * AC_IMPRT/Windows Procedimientos y funciones
de soporte
  469:      *        *
  470:      *
**************************************************************
  471:      *
  472:
  473:      #REGION 1
  474:
*!*************************************************************
**************
  475:  *!
  476:  *!       Procedure: EDITAR
  477:  *!
  478:  *!       Called by: CREAR              (procedure in
AACT_SST.SPR)
  479:  *!
  480:  *!           Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
  481:  *!                : CABEZA             (procedure in
PERSONAL.PRG)
  482:  *!                : INGRESO            (procedure in
ACT_PTMO.SPR)
  483:  *!                : CREA_TMP           (procedure in
MO_PDLAB.SPR)
```

```
  484:  *!                  : MBROW              (procedure in
MO_PDLAB.SPR)
  485:  *!                  : REF_SCR            (procedure in
MO_PDLAB.SPR)
  486:  *!                  : MENSAJES           (procedure in
MO_PDLAB.SPR)
  487:  *!
  488:
*!*************************************************************************
**************
  489:      PROCEDURE editar
  490:          ****************
  491:          STOR .T. TO editando
  492:          SHOW GETS ENABLE
  493:          SHOW GET opc_scr DISABLE
  494:          SHOW GET opc_busc DISABLE
  495:          SHOW GET opc_scr,1 PROMPT "\<CANCELAR" ENABLE
  496:          SHOW GET tag_busc DISABLE
  497:          SHOW GET trat_dat DISABLE
  498:          SHOW GET opc_ind DISABLE
  499:          _CUROBJ=OBJNUM(dgp.ac_sueldo)
  500:          ****************
  501:
*!*************************************************************************
**************
  502:  *!
  503:  *!       Procedure: REVISAR
  504:  *!
  505:  *!           Calls: MSCATTER          (procedure in
AACT_SST.SPR)
  506:  *!                : DEAC_KEYS         (procedure in
AACT_SST.SPR)
  507:  *!                : ACT_KEYS          (procedure in
AACT_SST.SPR)
  508:  *!                : VAL_CAMPO()       (function in
AF_PRMCT.SPR)
  509:  *!                : REFRESH()         (function in
AF_PRMCT.SPR)
  510:  *!                : MBROW             (procedure in
MO_PDLAB.SPR)
  511:  *!
  512:
*!*************************************************************************
**************
  513:      PROCEDURE revisar
  514:          ****************
  515:          SET ORDER TO nombre
  516:          GOTO TOP
  517:          BROW FIELDS acr.cod_per, dgp.nombre,dgp.cargo,;
  518:             acr.ac_sueldo:p='999,999,999',;
  519:             acr.ac_hextras:p='999,999,999',;
  520:             acr.ac_subant:p='999,999,999',;
  521:             acr.ac_subfam:p='999,999,999',;
  522:             acr.ac_apiess:p='999,999,999',;
```

```
  523:              acr.ac_100iess:p='999,999,999',;
  524:              acr.ac_subener:p='999,999,999',;
  525:              acr.ac_gastos:p='999,999,999',;
  526:              acr.ac_bcomis:p='999,999,999',;
  527:              acr.ac_subref:p='999,999,999',;
  528:              acr.ac_compct:p='999,999,999',;
  529:              acr.ac_compcc:p='999,999,999',;
  530:              acr.ac_imprent:p='999,999,999':
  531:              NOEDIT NOMENU
  532:            *                 :
  533:            FOR dgp.estado = 1
  534:            SHOW GETS
  535: <=========RETURN
  536:            *****************
  537:
*!*************************************************************************
**************
  538:  *!
  539:  *!        Procedure: BUSCAR
  540:  *!
  541:  *!        Called by: SECC()              (function in
PERSONAL.PRG)
  542:  *!                 : _R390JW4G3()        (function in
DSC_GRTR.SPR)
  543:  *!                 : _RA81B8ERO()        (function in
ACT_PTMO.SPR)
  544:  *!                 : _RA81B8GQD()        (function in
ACT_PTMO.SPR)
  545:  *!                 : _R6108FBLN()        (function in
MO_PDLAB.SPR)
  546:  *!                 : _R5KOM9Z09()        (function in
DSC_CTA.SPR)
  547:  *!
  548:  *!           Calls: VAL_BUSCAR          (procedure in
AACT_SST.SPR)
  549:  *!                 : BUSC_CLV()          (function in
DGP_CARG.SPR)
  550:  *!                 : DESPLEGA            (procedure in
PERSONAL.PRG)
  551:  *!                 : BUSCARP()           (function in
ACT_PTMO.SPR)
  552:  *!                 : REF_SCR             (procedure in
MO_PDLAB.SPR)
  553:  *!                 : MENSAJES            (procedure in
MO_PDLAB.SPR)
  554:  *!                 : ASIGNAM             (procedure in
MO_PDLAB.SPR)
  555:  *!
  556:
*!*************************************************************************
**************
  557:      PROCEDURE buscar
  558:          ****************
  559:       ┌─IF NOT WEXIST('BUSCAR') && o definir en el
```

```
principal
  560:          |     DEFINE WINDOW buscar FROM 0,0 TO 5,30 TITLE
´BUSCAR´ SYSTEM
  561:          └─ENDIF
  562:            ACTIVATE WINDOW buscar
  563:            MOVE WINDOW buscar CENTER
  564:            reg_ant = RECNO()
  565:            SET ORDER TO cod_per
  566:            mcod_per = SPAC(6)
  567:            STOR ´ ´ TO mcod_per1
  568:            STOR 0   TO mcod_per2
  569:          ┌─DO WHILE .T.
  570:          │     @ 1,1 SAY ´Ingrese el Cєdigo: ´ GET mcod_per1
  571:          │     @ 1,24 GET mcod_per2 PICT ´999´
  572:          │     READ MODAL
  573:          │     STOR mcod_per1 +´00´+IIF(mcod_per2 <
0,´00´,IIF(mcod_per2<100,´0´,´´))+;
  574:          │          ALLT(STR(mcod_per2)) TO mcod_per
  575:          │   ┌─IF LAST() = 27
  576:          │   │     GOTO reg_ant
  577:          │   │     vret = .F.
  578:          │   v═══════EXIT
  579:          │   └─ENDIF
  580:          │     SEEK mcod_per
  581:          │   ┌─IF !FOUND()
  582:          │   │     WAIT WINDOW NOWAIT ´Cєdigo no Existe´
  583:          │   │     GOTO reg_ant
  584:          │   `═══════LOOP
  585:          │   ├─ELSE
  586:          │   │     vret = .T.
  587:          │   v═══════EXIT
  588:          │   └─ENDIF
  589:          └─ENDDO
  590:            DEACTIVATE WINDOW buscar
  591:            SHOW GETS
  592:  <═══════RETURN
  593:            ********************
  594:
*!*************************************************************************
**************
  595:  *!
  596:  *!       Procedure: CANCELAR
  597:  *!
  598:  *!       Called by: ING                (procedure in
PERSONAL.PRG)
  599:  *!                : JUBILAR            (procedure in
PERSONAL.PRG)
  600:  *!
  601:  *!           Calls: MSCATTER           (procedure in
AACT_SST.SPR)
  602:  *!                : ACT_KEYS           (procedure in
AACT_SST.SPR)
  603:  *!                : DESPLEGA           (procedure in
PERSONAL.PRG)
```

```
  604:  *!                  : BORRA_TMP          (procedure in
MO_PDLAB.SPR)
  605:  *!                  : ASIGNAM            (procedure in
MO_PDLAB.SPR)
  606:  *!                  : REF_SCR            (procedure in
MO_PDLAB.SPR)
  607:  *!
  608:
*!*************************************************************************
**************
  609:      PROCEDURE cancelar
  610:          ********************
  611:          STOR .F. TO editando
  612:          SHOW GETS DISABLE
  613:          SHOW GET opc_scr,1 PROMPT "\<EDITAR"
  614:          SHOW GET opc_scr,1  ENABLE
  615:          SHOW GET opc_scr,3  ENABLE
  616:          SHOW GET opc_scr,6  ENABLE
  617:          SHOW GET opc_busc,2 ENABLE
  618:          SHOW GET tag_busc   ENABLE
  619:          SHOW GET trat_dat   ENABLE
  620:          SHOW GET opc_ind    ENABLE
  621: <=======RETURN
  622:          ********************
  623:
*!*************************************************************************
**************
  624:  *!
  625:  *!      Procedure: ACT_KEYS
  626:  *!
  627:  *!      Called by: AACT_SST.SPR
  628:  *!               : AACT_DPT.SPR
  629:  *!               : AACT_SEC.SPR
  630:  *!               : DGP_CARG.SPR
  631:  *!               : AC_IMPRT.SPR
  632:  *!               : DSC_GRTR.SPR
  633:  *!               : ELIMINAR           (procedure in
AACT_SST.SPR)
  634:  *!               : REVISAR            (procedure in
AACT_SST.SPR)
  635:  *!               : CTR_POSP           (procedure in
AACT_SST.SPR)
  636:  *!               : GRABAR             (procedure in
AACT_SST.SPR)
  637:  *!               : CANCELAR           (procedure in
AACT_SST.SPR)
  638:  *!               : ROL_Q1.SPR
  639:  *!               : ROL_Q2.SPR
  640:  *!               : ACT_PTMO.SPR
  641:  *!               : MO_PDLAB.SPR
  642:  *!               : DSC_CTA.SPR
  643:  *!               : _R6108FBI7()       (function in
MO_PDLAB.SPR)
  644:  *!
```

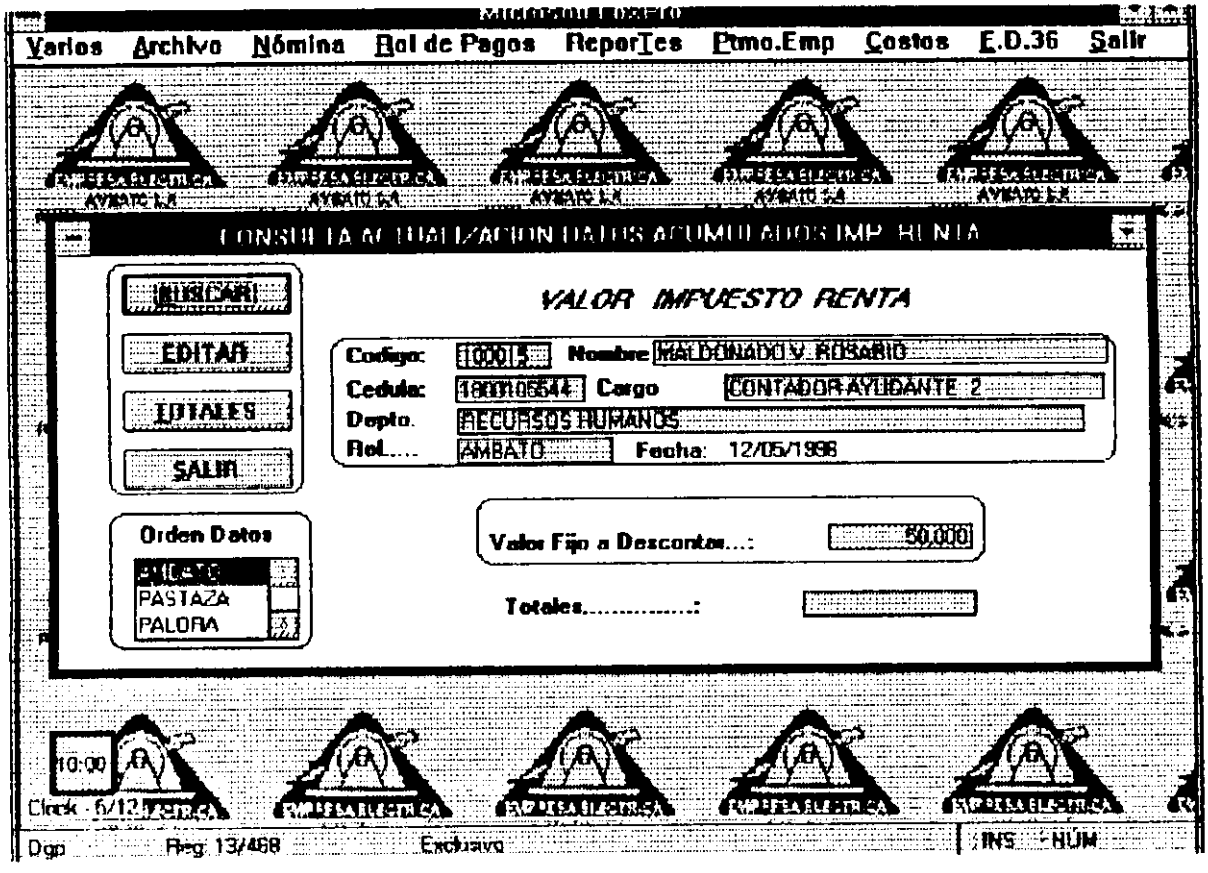
```
  645:  *!          Calls: CTR_POSP          (procedure in
AACT_SST.SPR)
  646:  *!
  647:
*!*********************************************************************
**************
  648:      PROCEDURE act_keys
  649:          ******************
  650:          ON KEY LABEL pgup DO ctr_posp WITH
'ANTERIOR', 'SKIP -1'
  651:          ON KEY LABEL pgdn DO ctr_posp WITH
'SIGUIENTE', 'SKIP'
  652:          ON KEY LABEL home DO ctr_posp WITH 'PRIMERO', 'GOTO
TOP'
  653:          ON KEY LABEL END DO ctr_posp WITH 'ULTIMO', 'GOTO
BOTTOM'
  654: <========RETURN
  655:          **********************
  656:
*!*********************************************************************
**************
  657:  *!
  658:  *!       Procedure: DEAC_KEYS
  659:  *!
  660:  *!       Called by: EDITAR              (procedure in
CTRL_RES.SPR)
  661:  *!                : SALIR               (procedure in
CTRL_RES.SPR)
  662:  *!                : CREAR               (procedure in
AACT_SST.SPR)
  663:  *!                : ELIMINAR            (procedure in
AACT_SST.SPR)
  664:  *!                : REVISAR             (procedure in
AACT_SST.SPR)
  665:  *!                : CTR_POSP            (procedure in
AACT_SST.SPR)
  666:  *!                : _R6103FBI7()        (function in
MO_PDLAB.SPR)
  667:  *!
  668:
*!*********************************************************************
**************
  669:      PROCEDURE deac_keys
  670:          ********************
  671:          ON KEY LABEL pgup
  672:          ON KEY LABEL pgdn
  673:          ON KEY LABEL home
  674:          ON KEY LABEL END
  675: <========RETURN
  676:          ********************
  677:
*!*********************************************************************
**************
  678:  *!
```

```
  679:   *!       Procedure: CTR_POSP
  680:   *!
  681:   *!       Called by: ACT_KEYS          (procedure in
AACT_SST.SPR)
  682:   *!                : ELIMINAR          (procedure in
AACT_SST.SPR)
  683:   *!
  684:   *!           Calls: MSCATTER          (procedure in
AACT_SST.SPR)
  685:   *!                : DEAC_KEYS         (procedure in
AACT_SST.SPR)
  686:   *!                : ASIGNAM           (procedure in
MO_PDLAB.SPR)
  687:   *!                : REF_SCR           (procedure in
MO_PDLAB.SPR)
  688:   *!                : ACT_KEYS          (procedure in
AACT_SST.SPR)
  689:   *!
  690:
*!*************************************************************************
**************
  691:      PROCEDURE ctr_posp
  692:          ********************
  693:          PARAMETER opc_posp, comando
  694:          SELE dgp
  695:          IF BOF()
  696:              GOTO TOP
  697:          ENDIF
  698:          IF EOF()
  699:              GOTO BOTTOM
  700:          ENDIF
  701:          &comando
  702:          DO CASE
  703:          CASE opc_posp = 'PRIMERO'
  704:                WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
  705:          CASE opc_posp = 'SIGUIENTE'
  706:                IF EOF()
  707:                    GOTO BOTTOM
  708:                    WAIT WIND NOWAIT 'FIN DE ARCHIVO'
  709:                ENDIF
  710:          CASE opc_posp = 'ANTERIOR'
  711:                IF BOF()
  712:                    GOTO TOP
  713:                    WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
  714:                ENDIF
  715:          CASE opc_posp = 'ULTIMO'
  716:                WAIT WIND NOWAIT 'FIN DE ARCHIVO'
  717:          ENDCASE
  718:          SHOW GETS
  719:          RETURN
  720:          ****************
  721:
*!*************************************************************************
**************
```

```
722:  *!
723:  *!        Procedure: SALIR
724:  *!
725:  *!        Called by: _RAA1732TG()        (function in
AT_ESCLS.SPR)
726:  *!                 : DA_ASISG.PRG
727:  *!                 : DE_EGRQ1.PRG
728:  *!                 : DE_EGRQ2.PRG
729:  *!                 : DE_EGRMN.PRG
730:  *!                 : DE_IEJUB.PRG
731:  *!
732:  *!           Calls: DEAC_KEYS            (procedure in
AACT_SST.SPR)
733:  *!
734:
*!*************************************************************************
**************
735:      PROCEDURE salir
736:          ***************
737:          DO deac_keys
738:          CLEAR READ
739: <=======RETURN
740:
741:          *
*************************************************************
742:          *       *
743:          *       * _R891CPJSL              opc_scr VALID
744:          *       *
745:          *       * Function Origin:
746:          *       *
747:          *       * From Platform:          Windows
748:          *       * From Screen:            AC_IMPRT.
Record Number:   13
749:          *       * Variable:               opc_scr
750:          *       * Called By:              VALID Clause
751:          *       * Object Type:            Push Button
752:          *       * Snippet Number:         1
753:          *       *
754:          *
*************************************************************
755:          *
756:
*!*************************************************************************
**************
757:  *!
758:  *!        Function: _R891CPJSL
759:  *!
760:  *!      Called by: AC_IMPRT.SPR
761:  *!
762:  *!           Calls: &OPC_SCR.PRG
763:  *!
764:
*!*************************************************************************
**************
```

```
765:      FUNCTION _r891cpjsl      &&  opc_scr VALID
766:          #REGION 1
767:          DO &opc_scr
768:
769:          *
************************************************************
770:          *        *
771:          *        * _R891CPJVK              opc_busc VALID
772:          *        *
773:          *        * Function Origin:
774:          *        *
775:          *        * From Platform:          Windows
776:          *        * From Screen:            AC_IMPRT.
Record Number:   14
777:          *        * Variable:               opc_busc
778:          *        * Called By:              VALID Clause
779:          *        * Object Type:            Push Button
780:          *        * Snippet Number:         2
781:          *        *
782:          *
************************************************************
783:          *
784:
*!*************************************************************************
**************
785:  *!
786:  *!        Function: _R891CPJVK
787:  *!
788:  *!      Called by: AC_IMPRT.SPR
789:  *!
790:  *!           Calls: &OPC_BUSC.PRG
791:  *!
792:
*!*************************************************************************
**************
793:      FUNCTION _r891cpjvk      &&  opc_busc VALID
794:          #REGION 1
795:          DO &opc_busc
796:
797:          *
************************************************************
798:          *        *
799:          *        * _R891CPJYC              OPC_IND VALID
800:          *        *
801:          *        * Function Origin:
802:          *        *
803:          *        * From Platform:          Windows
804:          *        * From Screen:            AC_IMPRT.
Record Number:   15
805:          *        * Variable:               OPC_IND
806:          *        * Called By:              VALID Clause
807:          *        * Object Type:            Popup
808:          *        * Snippet Number:         3
809:          *        *
```

```
*
*************************************************************
*
*!*********************************************************************
**************
*!
*!        Function: _R891CPJYC
*!
*!       Called by: AC_IMPRT.SPR
*!
*!**********************************************************************
**************
     FUNCTION _r891cpjyc      &&  OPC_IND VALID
         #REGION 1
         SET ORDER TO &opc_ind
*:  EOF: AC_IMPRT.ac1
```


Varios Archivo Nómina Rol de Pagos Reportes Ptmo.Emp Costos E.D.36 Salir
CONSULTA ACTUALIZACION DATOS ACUMULADOS IMP. RENTA
BUSCAR
EDITAR
TOTALES
SALIR
VALOR IMPUESTO RENTA
Codigo: 10001 Nombre: MALDONADO V. ROSARIO
Cedula: 1800106544 Cargo CONTADOR AYUDANTE 2
Depto. RECURSOS HUMANOS
Rol.... AMBATO Fecha: 12/05/1998
Orden Datos
AMBATO
PASTAZA
PALORA
Valor Fijo a Descontar...: 50.000
Totales.....:
10:00
Dgp Reg: 13/468 Exclusivo INS NUM

```
    1:
*:*************************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\VF_IMPRTA.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDY - HUGO - MERCEDES
    6:  *:      Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 21/10/1995 at 11:50:28
    8:  *:
    9:  *:  Procs & Fncts: _R4NOPDO8Z()
   10:  *:                : _R4NOPDOG4()
   11:  *:                : _R4NOPDOIW()
   12:  *:                : _R4NOPDON5()
   13:  *:                : TOTALES
   14:  *:
   15:  *:          Set by: EMPRESA.MPR
   16:  *:
   17:  *:           Calls: _R4NOPDO8Z()        (function in
VF_IMPRTA.SPR)
   18:  *:                : _R4NOPDOG4()        (function in
VF_IMPRTA.SPR)
   19:  *:                : _R4NOPDOIW()        (function in
VF_IMPRTA.SPR)
   20:  *:                : _R4NOPDON5()        (function in
VF_IMPRTA.SPR)
   21:  *:
   22:  *:       Documented 19:14:49
FoxDoc version 3.00a
   23:
*:*************************************************************************
**************
   24:      *
************************************************************
   25:      *        *
   26:      *        * 10/21/1995            VF_IMPRT.SPR
 11:50:26
   27:      *        *
   28:      *
************************************************************
   29:      *        *
   30:      *        * Author's Name
   31:      *        *
   32:      *        * Copyright (c) 1995 Company Name
   33:      *        * Address
   34:      *        * City,     Zip
   35:      *        *
   36:      *        * Description:
   37:      *        * This program was automatically generated by
GENSCRN.
   38:      *        *
   39:      *
************************************************************
```

```
 40:
 41:
 42:     *
**********************************************************
 43:     *       *
 44:     *       *          VF_IMPRT/Windows Setup Code -
SECTION 1
 45:     *       *
 46:     *
**********************************************************
 47:     *
 48:
 49:     #REGION 1
 50:     REGIONAL
ope_scr,mcod_per,indices,mindices,editando,TOTAL
 51:     DIMENSION indices(20)
 52:     STOR '' TO ope_scr,mcod_per,mindices
 53:     STOR .F. TO editando
 54:     STOR 0 TO TOTAL
 55:     STOR 1 TO veiu
 56:     **********
 57:
 58:     #REGION 0
 59:     REGIONAL m.currarea, m.talkstat, m.compstat
 60:
 61:     IF SET("TALK") = "ON"
 62:         SET TALK OFF
 63:         m.talkstat = "ON"
 64:     ELSE
 65:         m.talkstat = "OFF"
 66:     ENDIF
 67:     m.compstat = SET("COMPATIBLE")
 68:     SET COMPATIBLE FOXPLUS
 69:
 70:     m.rborder = SET("READBORDER")
 71:     SET readborder ON
 72:
 73:     m.currarea = SELECT()
 74:
 75:
 76:     *
**********************************************************
 77:     *       *
 78:     *       *     VF_IMPRT/Windows Databases, Indexes,
Relations
 79:     *       *
 80:     *
**********************************************************
 81:     *
 82:
 83:     IF USED("mes")
 84:         SELECT mes
 85:         SET ORDER TO 0
 86:     ELSE
```

```
    87:    |     SELECT 0
    88:    |     USE (LOCFILE("dbfs\meses.org","DBF","Where is
meses?"));
    89:    |         AGAIN ALIAS mes ;
    90:    |         ORDER 0
    91:    |ENDIF
    92:
    93:    |IF USED("sst")
    94:    |    SELECT sst
    95:    |    SET ORDER TO TAG "tipo_act"
    96:    |ELSE
    97:    |    SELECT 0
    98:    |    USE (LOCFILE("dbfs\sistema.dbf","DBF","Where is
sistema?"));
    99:    |         AGAIN ALIAS sst ;
   100:    |         ORDER TAG "tipo_act"
   101:    |ENDIF
   102:
   103:    |IF USED("dgp")
   104:    |    SELECT dgp
   105:    |    SET ORDER TO TAG "cod_per"
   106:    |ELSE
   107:    |    SELECT 0
   108:    |    USE (LOCFILE("dbfs\personal.dbf","DBF","Where is
personal?"));
   109:    |         AGAIN ALIAS dgp ;
   110:    |         ORDER TAG "cod_per"
   111:    |ENDIF
   112:
   113:    |IF USED("men")
   114:    |    SELECT men
   115:    |    SET ORDER TO TAG "mens_cod"
   116:    |ELSE
   117:    |    SELECT 0
   118:    |    USE (LOCFILE("dbfs\mensual.dbf","DBF","Where is
mensual?"));
   119:    |         AGAIN ALIAS men ;
   120:    |         ORDER TAG "mens_cod"
   121:    |ENDIF
   122:
   123:    |IF USED("sec")
   124:    |    SELECT sec
   125:    |    SET ORDER TO TAG "secc_cod"
   126:    |ELSE
   127:    |    SELECT 0
   128:    |    USE (LOCFILE("dbfs\seccion.dbf","DBF","Where is
seccion?"));
   129:    |         AGAIN ALIAS sec ;
   130:    |         ORDER TAG "secc_cod"
   131:    |ENDIF
   132:
   133:     SELECT dgp
   134:
   135:     SET RELATION OFF INTO sec
```

```
136:      SET RELATION TO num_sec INTO sec ADDITIVE
137:
138:      SELECT dgp
139:
140:      *
*************************************************************
141:      *        *
142:      *        *                    Windows Window definitions
143:      *        *
144:      *
*************************************************************
145:      *
146:
147:      IF NOT WEXIST("da_imprt") :
148:            OR UPPER(WTITLE("DA_IMPRT")) == "DA_IMPRT.FJX" :
149:            OR UPPER(WTITLE("DA_IMPRT")) == "DA_IMPRT.SCX" :
150:            OR UPPER(WTITLE("DA_IMPRT")) == "DA_IMPRT.MNX" :
151:            OR UPPER(WTITLE("DA_IMPRT")) == "DA_IMPRT.PRG" :
152:            OR UPPER(WTITLE("DA_IMPRT")) == "DA_IMPRT.FRX" :
153:            OR UPPER(WTITLE("DA_IMPRT")) == "DA_IMPRT.QPR"
154:         DEFINE WINDOW da_imprt :
155:            AT  0.000, 0.000  :
156:            SIZE 17.923,117.200 :
157:            TITLE "CONSULTA ACTUALIZACION DATOS ACUMULADOS
IMP. RENTA" :
158:            FONT "MS Sans Serif", 8 :
159:            FLOAT :
160:            NOCLOSE :
161:            MINIMIZE :
162:            SYSTEM
163:         MOVE WINDOW da_imprt CENTER
164:      ENDIF
165:
166:
167:      *
*************************************************************
168:      *        *
169:      *        *          VF_IMPRT Windows Setup Code -
SECTION 2
170:      *        *
171:      *
*************************************************************
172:      *
173:
174:      #REGION 1
175:      **********
176:      SELE sst
177:      STOR ciudad TO msis
178:      STOR cod_ciu TO vciu
179:      SELECT dgp
180:      FOR i = 1 TO 20
181:         IF EMPTY(TAG(i))
182:            i = i - 1
183:            DIMENSION indices(i)
```

```
184:    ╚═══EXIT
185:    ╟   ├─ELSE
186:    ╟   │      indices(i) = TAG(i)
187:    ╟   └─ENDIF
188:    ╚═ENDFOR
189:       STORE indices(1) TO mindices
190:       SET ORDER TO cod_per
191:       SET FILTER TO estado # 2
192:       GOTO TOP
193:       SHOW GETS
194:
195:
196:       #REGION 1
197:       DEFINE POPUP _r4n0pdnxw :
198:          PROMPT FIELD sst.ciudad :
199:          SCROLL
200:
201:       *
*************************************************************
202:       *          *
203:       *          *                VF_IMPRT/Windows Screen Layout
204:       *          *
205:       *
*************************************************************
206:       *
207:
208:       #REGION 1
209:    ┌─IF WVISIBLE("da_imprt")
210:    │     ACTIVATE WINDOW da_imprt SAME
211:    ├─ELSE
212:    │     ACTIVATE WINDOW da_imprt NOSHOW
213:    └─ENDIF
214:       @ 3.846,31.200 SAY "Codigo:"  :
215:          FONT "MS Sans Serif", 8 :
216:          STYLE "BT"
217:       @ 3.846,55.200 SAY "Nombre"  :
218:          FONT "MS Sans Serif", 8 :
219:          STYLE "BT"
220:       @ 5.308,58.200 SAY "Cargo"  :
221:          FONT "MS Sans Serif", 8 :
222:          STYLE "BT"
223:       @ 12.077,46.600 SAY "Valor Fijo a Descontar...:"  :
224:          FONT "MS Sans Serif", 8 :
225:          STYLE "BT"
226:       @ 1.385,51.600 SAY "VALOR  IMPUESTO  RENTA"  :
227:          FONT "MS Sans Serif", 10 :
228:          STYLE "BIT"
229:       @ 6.615,31.200 SAY "Depto."  :
230:          FONT "MS Sans Serif", 8 :
231:          STYLE "BT"
232:       @ 7.923,31.200 SAY "Rol....."  :
233:          FONT "MS Sans Serif", 8 :
234:          STYLE "BT"
235:       @ 8.077,62.400 SAY "Fecha:"  :
```

```
236:          FONT "MS Sans Serif", 8 :
237:          STYLE "BT"
238:       @ 3.615,29.600 TO 9.230,113.800 :
239:          PEN 1, 8 :
240:          STYLE "16"
241:       @ 10.615,45.200 TO 13.538,99.800 :
242:          PEN 1, 8 :
243:          STYLE "16"
244:       @ 0.308,5.600 TO 10.385,26.600 :
245:          PEN 1, 8 :
246:          STYLE "16"
247:       @ 11.615,8.000 SAY " Orden Datos"  :
248:          FONT "MS Sans Serif", 8 :
249:          STYLE "BT"
250:       @ 11.154,5.600 TO 17.154,27.400 :
251:          PEN 1, 8 :
252:          STYLE "16"
253:       @ 5.308,31.200 SAY "Cedula:"  :
254:          FONT "MS Sans Serif", 8 :
255:          STYLE "BT"
256:       @ 14.846,48.000 SAY "Totales.....:"  :
257:          FONT "MS Sans Serif", 8 :
258:          STYLE "BT"
259:       @ 0.769,7.200 GET opc_scr :
260:          PICTURE "@*VN \<BUSCAR;\EDITAR;\<TOTALES;\SALIR"
:
261:          SIZE 1.769,15.167,0.769 :
262:          DEFAULT 1 :
263:          FONT "MS Sans Serif", 8 :
264:          STYLE "B" :
265:          VALID _r4n0pdo8s() 
266:       @ 3.846,64.800 GET dgp.nombre :
267:          SIZE 1.000,47.600 :
268:          DEFAULT " " :
269:          FONT "MS Sans Serif", 8 :
270:          WHEN .F.
271:       @ 3.923,43.600 GET dgp.cod_per :
272:          SIZE 1.000,9.200 :
273:          DEFAULT " " :
274:          FONT "MS Sans Serif", 8 :
275:          WHEN .F.
276:       @ 5.385,43.400 GET dgp.cedula :
277:          SIZE 1.000,13.000 :
278:          DEFAULT " " :
279:          FONT "MS Sans Serif", 8 :
280:          WHEN .F.
281:       @ 5.385,72.400 GET dgp.cargo :
282:          SIZE 1.000,39.600 :
283:          DEFAULT " " :
284:          FONT "MS Sans Serif", 8 :
285:          PICTURE "@K" :
286:          WHEN .F.
287:       @ 6.769,43.600 GET sec.nombre :
288:          SIZE 1.000,66.200 :
```

```
289:          DEFAULT " " :
290:          FONT "MS Sans Serif", 8 :
291:          WHEN .F.
292:       @ 8.077,43.600 GET sst.ciudad :
293:          SIZE 1.000,15.800 :
294:          DEFAULT " " :
295:          FONT "MS Sans Serif", 8 :
296:          WHEN .F.
297:       @ 8.077,72.400 SAY DATE() :
298:          SIZE 1.000,13.000 :
299:          FONT "MS Sans Serif", 8
300:       @ 11.923,83.400 GET dgp.acum_ir :
301:          SIZE 1.000,14.400 :
302:          DEFAULT 0 :
303:          FONT "MS Sans Serif", 8 :
304:          PICTURE "@Z 999,999,999" :
305:          VALID _r4n0pdog4()
306:       @ 13.154,8.200 GET msis :
307:          PICTURE "@&N" :
308:          POPUP _r4n0pdnxw :
309:          SIZE 3.462,17.400 :
310:          DEFAULT " " :
311:          FONT "MS Sans Serif", 8 :
312:          STYLE "" :
313:          VALID _r4n0pdoiw()
314:       @ 14.846,80.600 GET TOTAL :
315:          SIZE 1.000,17.600 :
316:          DEFAULT 0 :
317:          FONT "MS Sans Serif", 8 :
318:          PICTURE "@Z 99,999,999,999" :
319:          WHEN .F.
320:
321:    IF NOT WVISIBLE("da_imprt")
322:       ACTIVATE WINDOW da_imprt
323:    ENDIF
324:
325:    READ CYCLE :
326:       WHEN _r4n0pdon5()
327:
328:    RELEASE WINDOW da_imprt
329:
330:    *
*****************************************************************
331:    *        *
332:    *        *                    Windows Closing Databases
333:    *        *
334:    *
*****************************************************************
335:    *
336:
337:    IF USED("mes")
338:       SELECT mes
339:       USE
340:    ENDIF
```

```
341:
342:     IF USED("sst")
343:         SELECT sst
344:         USE
345:     ENDIF
346:
347:     IF USED("dgp")
348:         SELECT dgp
349:         USE
350:     ENDIF
351:
352:     IF USED("men")
353:         SELECT men
354:         USE
355:     ENDIF
356:
357:     IF USED("sec")
358:         SELECT sec
359:         USE
360:     ENDIF
361:
362:     SELECT (m.currarea)
363:
364:     RELEASE POPUPS _r4n0pdnxw
365:
366:     #REGION 0
367:
368:     SET readborder &rborder
369:
370:     IF m.talkstat = "ON"
371:         SET TALK ON
372:     ENDIF
373:     IF m.compatat = "ON"
374:         SET COMPATIBLE ON
375:     ENDIF
376:
377:
378:     *
*************************************************************
379:     *         *
380:     *         *                VF_IMPRT/Windows Cleanup Code
381:     *         *
382:     *
*************************************************************
383:     *
384:
385:     #REGION 1
386:     ******************
387:
388:
389:     *
*************************************************************
390:     *         *
391:     *         *  VF_IMPRT/Windows Supporting Procedures and
```

```
Funciona
  392:      *        *
  393:      *
*************************************************************
  394:      *
  395:
  396:      #REGION 1
  397:
*!*************************************************************************
**************
  398:  *!
  399:  *!        Procedure: BUSCAR
  400:  *!
  401:  *!        Called by: SECC()              (function in
PERSONAL.PRG)
  402:  *!                 : _R390JW4G3()        (function in
DSC_GRTR.SPR)
  403:  *!                 : _RA81B8ERO()        (function in
ACT_PTMO.SPR)
  404:  *!                 : _RA81B8GQD()        (function in
ACT_PTMO.SPR)
  405:  *!                 : _R6108FBLN()        (function in
MO_PDLAB.SPR)
  406:  *!                 : _R5KOM9Z09()        (function in
DSC_CTA.SPR)
  407:  *!
  408:  *!           Calls: VAL_BUSCAR           (procedure in
AACT_SST.SPR)
  409:  *!                 : BUSC_CLV()          (function in
DGP_CARG.SPR)
  410:  *!                 : DESPLEGA            (procedure in
PERSONAL.PRG)
  411:  *!                 : BUSCARF()           (function in
ACT_PTMO.SPR)
  412:  *!                 : REF_SCR             (procedure in
MO_PDLAB.SPR)
  413:  *!                 : MENSAJES            (procedure in
MO_PDLAB.SPR)
  414:  *!                 : ASIGNAM             (procedure in
MO_PDLAB.SPR)
  415:  *!
  416:
*!*************************************************************************
**************
  417:      PROCEDURE buscar
  418:          *****************
  419:
  420:        IF NOT WEXIST('BUSCAR') && o definir en el
principal
  421:          DEFINE WINDOW buscar FROM 0,0 TO 5,30 TITLE
'BUSCAR' SYSTEM
  422:        ENDIF
  423:          ACTIVATE WINDOW buscar
  424:          MOVE WINDOW buscar CENTER
```

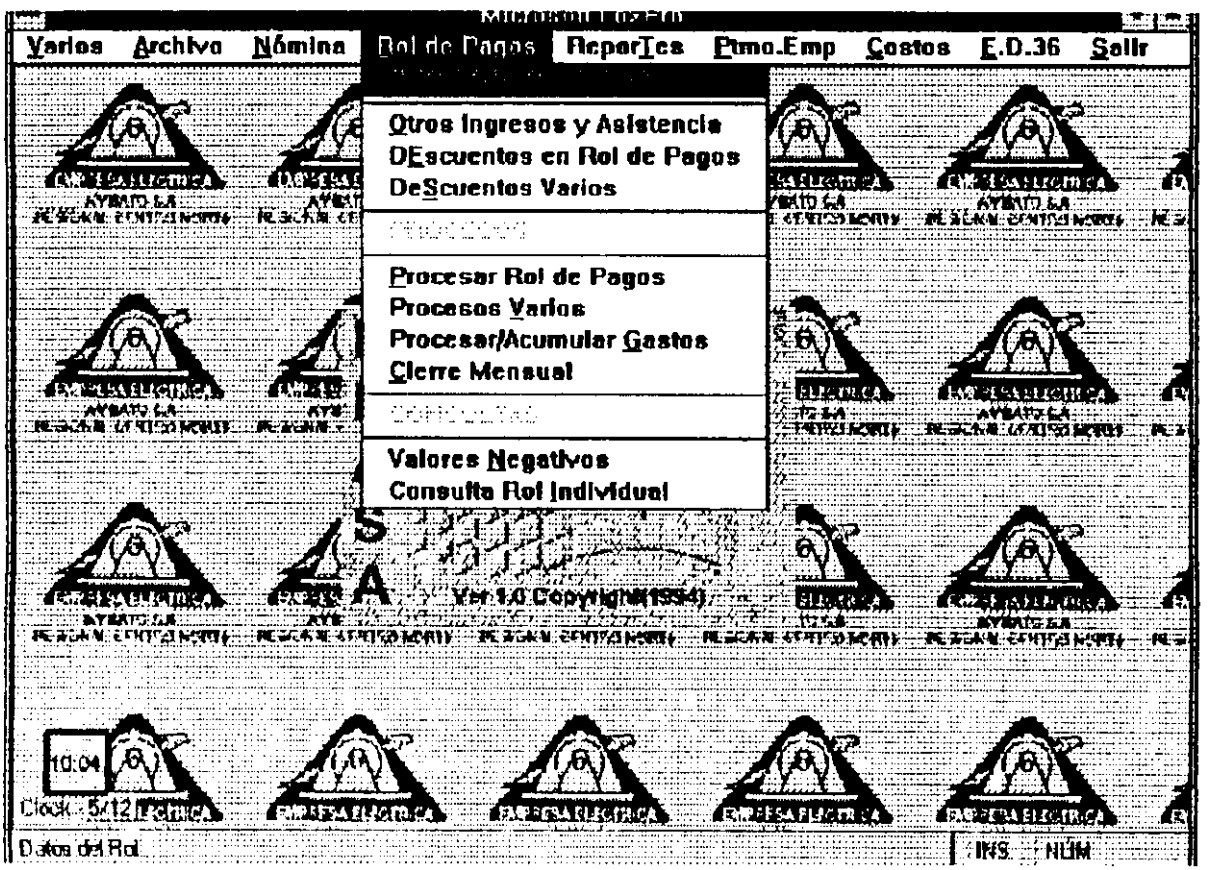
```
  425:          reg_ant = RECNO()
  426:          mcod_per = SPAC(6)
  427:          STOR 0 TO mmcod_per
  428:       ┌─DO WHILE .T.
  429:       │     @ 1,1 SAY 'Ingrese el Cσdigo: ' GET mmcod_per
PICT '999'
  430:       │     READ MODAL
  431:       │     STOR ALLT(STR(mmcod_per)) TO ccod_per
  432:       │     STOR ALLT(STR(veiu)) + '00'+
IIF(mmcod_per<10, '00'+ccod_per,:
  433:       │         IIF(mmcod_per<100, '0'+ccod_per,ccod_per)) TO
mcod_per
  434:       │  ┌─IF LAST() = 27
  435:       │  │     GOTO reg_ant
  436:       │  │     vret = .F.
  437:       │  ├─────EXIT
  438:       │  └─ENDIF
  439:       │    SEEK mcod_per
  440:       │  ┌─IF !FOUND()
  441:       │  │     WAIT WINDOW NOWAIT 'Cσdigo no Existe'
  442:       │  │     GOTO reg_ant
  443:       │  ├─────LOOP
  444:       │  ├─ELSE
  445:       │  │     vret = .T.
  446:       │  ├─────EXIT
  447:       │  └─ENDIF
  448:       └─ENDDO
  449:          DEACTIVATE WINDOW buscar
  450:          SHOW GETS
  451:    ├─────RETURN vret
  452:          ******************
  453:
*!*************************************************************
**************
  454:  *!
  455:  *!       Procedure: EDITAR
  456:  *!
  457:  *!       Called by: CREAR              (procedure in
AACT_SST.SPR)
  458:  *!
  459:  *!           Calls: DEAC KEYS          (procedure in
AACT_SST.SPR)
  460:  *!                : CABECA             (procedure in
PERSONAL.PRG)
  461:  *!                : INGRESO            (procedure in
ACT_FTMO.SPR)
  462:  *!                : CREA_TMP           (procedure in
MO_PDLAB.SPR)
  463:  *!                : MBROW              (procedure in
MO_PDLAB.SPR)
  464:  *!                : REF_SCR            (procedure in
MO_PDLAB.SPR)
  465:  *!                : MENSAJES           (procedure in
MO_PDLAB.SPR)
```

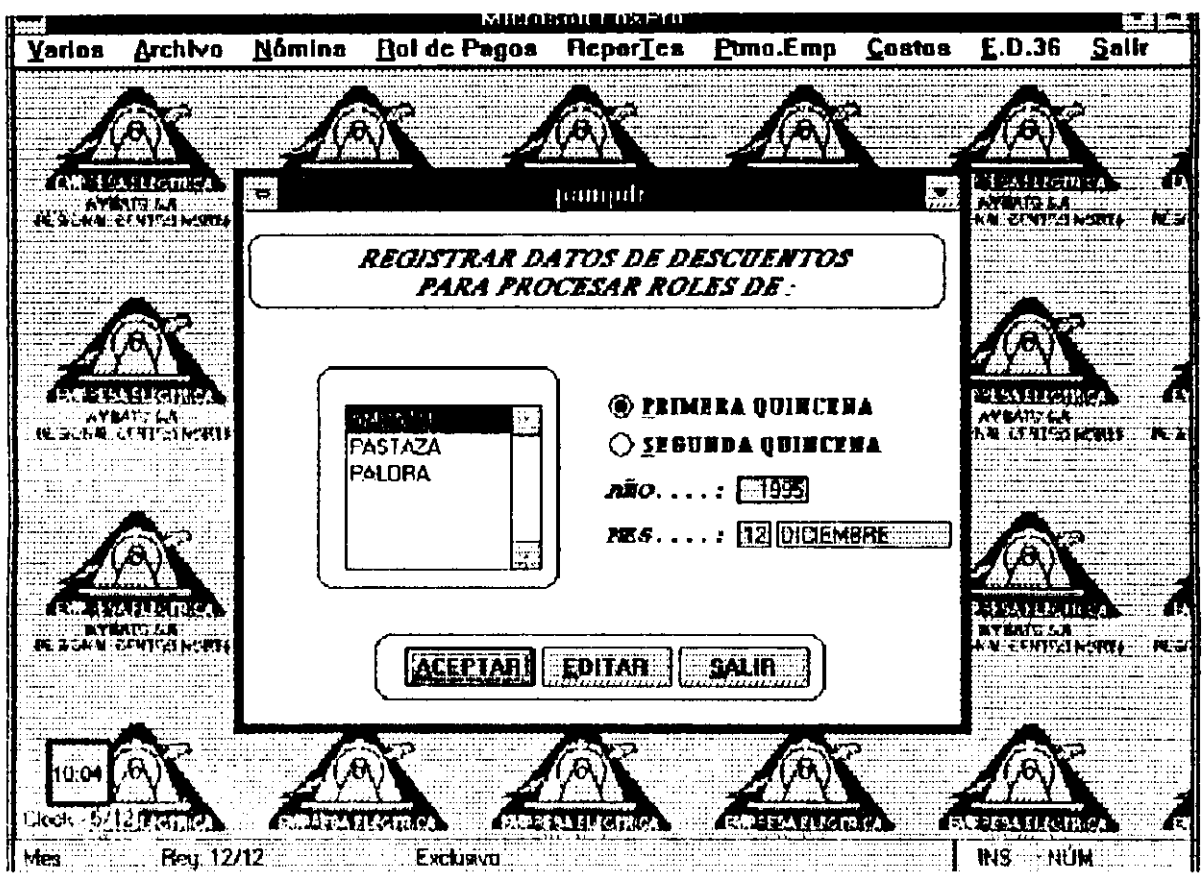
```
  466:  *!
  467:
*!*************************************************************************
**************
  468:      PROCEDURE editar
  469:          *****************
  470:          *show gets
  471:          _CUROBJ= OBJNUM(dgp.acum_ir)
  472:          *show get Dgp.acum_ir enable
  473:          *****************
  474:
*!*************************************************************************
**************
  475:  *!
  476:  *!       Procedure: TOTALES
  477:  *!
  478:  *!       Called by: SECC()              (function in
PERSONAL.PRG)
  479:  *!                : TRAT_DATOS()        (function in
DA_ASISG.PRG)
  480:  *!
  481:  *!          Calls: SIST_TOT             (procedure in
EGR_VARI.PRG)
  482:  *!
  483:
*!*************************************************************************
**************
  484:      PROCEDURE totales
  485:          *****************
  486:          SUM ALL dgp.acum_ir TO TOTAL FOR dgp.num_ciu =
vciu
  487:          SHOW GETS
  488:          RETURN
  489:          *********************
  490:
*!*************************************************************************
**************
  491:  *!
  492:  *!       Procedure: SALIR
  493:  *!
  494:  *!       Called by: _RAA173ZTG()        (function in
AT_ECCLS.SPR)
  495:  *!                : DA_ASISG.PRG
  496:  *!                : DE_EGRQ1.PRG
  497:  *!                : DE_EGRQ2.PRG
  498:  *!                : DE_EGRMN.PRG
  499:  *!                : DE_IEJUF.PRG
  500:  *!
  501:  *!          Calls: DEAC_KEYS            (procedure in
AACT_SST.SPR)
  502:  *!
  503:
*!*************************************************************************
**************
```

```
  504:      PROCEDURE salir
  505:          ***************
  506:          CLEAR READ
  507: <=======RETURN
  508:
  509:          *
*************************************************************
  510:          *        *
  511:          *        * _R4N0PDO8Z              OPC_scr VALID
  512:          *        *
  513:          *        * Function Origin:
  514:          *        *
  515:          *        * From Platform:          Windows
  516:          *        * From Screen:            VF_IMPRT.
Record Number:   23
  517:          *        * Variable:               OPC_scr
  518:          *        * Called By:              VALID Clause
  519:          *        * Object Type:            Push Button
  520:          *        * Snippet Number:         1
  521:          *        *
  522:          *
*************************************************************
  523:          *
  524:
*!*************************************************************************
**************
  525:  *!
  526:  *!        Function:  R4N0PDO8Z
  527:  *!
  528:  *!      Called by: VF_IMPRTA.SPR
  529:  *!
  530:  *!           Calls: &OPC_SCR.PRG
  531:  *!
  532:
*!*************************************************************************
**************
  533:      FUNCTION _r4n0pdo8z      && OPC_scr VALID
  534:          #REGION 1
  535:          DO &opc_scr
  536:
  537:          *
*************************************************************
  538:          *        *
  539:          *        * _R4N0PDOG4              Dgp.acum_ir VALID
  540:          *        *
  541:          *        * Function Origin:
  542:          *        *
  543:          *        * From Platform:          Windows
  544:          *        * From Screen:            VF_IMPRT.
Record Number:   31
  545:          *        * Variable:               Dgp.acum_ir
  546:          *        * Called By:              VALID Clause
  547:          *        * Object Type:            Field
  548:          *        * Snippet Number:         2
```

```
549:          *        *
550:          *
*************************************************************
551:          *
552:
*!*************************************************************
**************
553:   *!
554:   *!        Function: _R4N0PDOG4
555:   *!
556:   *!       Called by: VF_IMPRTA.SPR
557:   *!
558:
*!*************************************************************
**************
559:       FUNCTION _r4n0pdog4      &&  Dgp.acum_ir VALID
560:           #REGION 1
561:           _CUROBJ= OBJNUM(ope_scr)
562:
563:
564:           *
*************************************************************
565:           *        *
566:           *        * _R4N0PDOIW            msis VALID
567:           *        *
568:           *        * Function Origin:
569:           *        *
570:           *        * From Platform:        Windows
571:           *        * From Screen:          VF_IMPRT,
Record Number:   32
572:           *        * Variable:             msis
573:           *        * Called By:            VALID Clause
574:           *        * Object Type:          List
575:           *        * Snippet Number:       3
576:           *        *
577:           *
*************************************************************
578:           *
579:
*!*************************************************************
**************
580:   *!
581:   *!        Function: _R4N0PDOIW
582:   *!
583:   *!       Called by: VF_IMPRTA.SPR
584:   *!
585:
*!*************************************************************
**************
586:       FUNCTION _r4n0pdoiw      &&  msis VALID
587:           #REGION 1
588:           malias = ALIAS()
589:           SELE sst
590:           STOR sst.cod_ciu TO vciu
```

```
591:          SELE &malias
592:          LOCATE FOR dgp.num_ciu = veiu
593:          SHOW GETS
594:
595:
596:          *
*************************************************************
597:          *       *
598:          *       * _R4N0PDON5            Read Level When
599:          *       *
600:          *       * Function Origin:
601:          *       *
602:          *       *
603:          *       * From Platform:        Windows
604:          *       * From Screen:          VF_IMPRT
605:          *       * Called By:            READ Statement
606:          *       * Snippet Number:       4
607:          *       *
608:          *
*************************************************************
609:          *
610:
*!*************************************************************************
**************
611:  *!
612:  *!        Function: _R4N0PDON5
613:  *!
614:  *!       Called by: VF_IMPRTA.SPR
615:  *!
616:
*!*************************************************************************
**************
617:      FUNCTION _r4n0pdon5      && Read Level When
618:          *
619:          * When Code from screen: VF_IMPRT
620:          *
621:          #REGION 1
622:          SHOW GET ope_desp ENABLE
623:          SHOW GET ope_desp,7 DISABLE
624:  *:  EOF: VF_IMPRTA.scl
```


Varios
Archivo
Nómina
Rol de Pagos
ReporTes
Pmo.Emp
Costos
E.D.36
Salir
Otros Ingresos y Asistencia
DEscuentos en Rol de Pagos
DeScuentos Varios
Procesar Rol de Pagos
Procesos Varios
Procesar/Acumular Gastos
Cierre Mensual
Valores Negativos
Consulta Rol Individual
Ver 1.0 Copyright(1994)
10:04
Clock: 5/12
Datos del Rol
INS
NUM


Varios Archivo Nómina Rol de Pagos ReporTes Pmo.Emp Costos E.D.36 Salir
pampdr
REGISTRAR DATOS DE DESCUENTOS
PARA PROCESAR ROLES DE :
PASTAZA
PALORA
PRIMERA QUINCENA
SEGUNDA QUINCENA
AÑO. . . . : 1995
MES. . . . : 12 DICIEMBRE
ACEPTAR
EDITAR
SALIR
10:04
Mes
Reg. 12/12
Exclusivo
INS
NÚM

```
     1:
*:*************************************************************
**************
     2:  *:
     3:  *: Procedure file: C:\DOC\CTRL_ROL.SPR
     4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
     5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
     6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
     7:  *:  Last modified: 23/01/1995 at  4:36:32
     8:  *:
     9:  *:  Procs & Fncts: _R4HOLWAX5()
    10:  *:                : _R4HOLWB14()
    11:  *:                : _R4HOLWB5S()
    12:  *:
    13:  *:          Set by: EMPRESA.MPR
    14:  *:
    15:  *:           Calls: _R4HOLWAX5()        (function in
CTRL_ROL.SPR)
    16:  *:                : _R4HOLWB14()        (function in
CTRL_ROL.SPR)
    17:  *:                : _R4HOLWB5S()        (function in
CTRL_ROL.SPR)
    18:  *:
    19:  *:       Documented 19:14:50
FoxDoc version 3.00a
    20:
*:*************************************************************
**************
    21:      *
************************************************************
    22:      *        *
    23:      *        * 10/15/1995            CTRL_ROL.SPR
 10:12:57
    24:      *        *
    25:      *
************************************************************
    26:      *        *
    27:      *        * Author's Name
    28:      *        *
    29:      *        * Copyright (c) 1995 Company Name
    30:      *        * Address
    31:      *        * City,      Zip
    32:      *        *
    33:      *        * Description:
    34:      *        * This program was automatically generated by
GENSCRN.
    35:      *        *
    36:      *
************************************************************
    37:
    38:      PARAMETERS N
    39:
    40:      *
************************************************************
    41:      *        *
    42:      *        *          CTRL_ROL/Windows Setup Code -
```

```
SECTION 1
   43:      *        *
   44:      *
*****************************************************************
   45:      *
   46:
   47:      #REGION 1
   48:      **********
   49:      REGIONAL ope_ser, ope_rol, ope_qui, mmes, vmes, manio,
mquincena, mrol.:
   50:         mope_qui.mfquin
   51:      STOR ' ' TO ope_ser, mmes,ope_rol, fquin
   52:      STOR 0 TO ope_qui.mope_qui
   53:      STOR 1 TO mrol
   54:      STOR .F. TO mquincena
   55:      STOR YEAR(DATE()) TO manio
   56:      STOR MONTH(DATE()) TO vmes
   57:
   58:      **********
   59:
   60:      #REGION 0
   61:      REGIONAL m.currarea, m.talkstat, m.compstat
   62:
   63:    ┌─IF SET("TALK") = "ON"
   64:    │    SET TALK OFF
   65:    │    m.talkstat = "ON"
   66:    ├─ELSE
   67:    │    m.talkstat = "OFF"
   68:    └─ENDIF
   69:      m.compstat = SET("COMPATIBLE")
   70:      SET COMPATIBLE FOXPLUS
   71:
   72:      m.rborder = SET("READBORDER")
   73:      SET readborder ON
   74:
   75:      m.currarea = SELECT()
   76:
   77:
   78:      *
*****************************************************************
   79:      *        *
   80:      *        *      CTRL_ROL/Windows Databases, Indexes,
Relations
   81:      *        *
   82:      *
*****************************************************************
   83:      *
   84:
   85:    ┌─IF USED("mes")
   86:    │    SELECT mes
   87:    │    SET ORDER TO 0
   88:    ├─ELSE
   89:    │    SELECT 0
   90:    │    USE (LOCFILE("dbfs\meses.org","DBF","Where is
meses?"));
   91:    │       AGAIN ALIAS mes :
   92:    │       ORDER 0
```

```
 93:    ENDIF
 94:
 95:    IF USED("sst")
 96:        SELECT sst
 97:        SET ORDER TO TAG "tipo_act"
 98:    ELSE
 99:        SELECT 0
100:        USE (LOCFILE("dbfs\sistema.dbf","DBF","Where is
sistema?"));
101:            AGAIN ALIAS sst ;
102:            ORDER TAG "tipo_act"
103:    ENDIF
104:
105:    IF USED("dgp")
106:        SELECT dgp
107:        SET ORDER TO TAG "cod_per"
108:    ELSE
109:        SELECT 0
110:        USE (LOCFILE("dbfs\personal.dbf","DBF","Where is
personal?"));
111:            AGAIN ALIAS dgp ;
112:            ORDER TAG "cod_per"
113:    ENDIF
114:
115:    IF USED("men")
116:        SELECT men
117:        SET ORDER TO TAG "mens_cod"
118:    ELSE
119:        SELECT 0
120:        USE (LOCFILE("dbfs\mensual.dbf","DBF","Where is
mensual?"));
121:            AGAIN ALIAS men ;
122:            ORDER TAG "mens_cod"
123:    ENDIF
124:
125:    SELECT act
126:
127:
128:    *
*************************************************************
129:    *       *
130:    *       *                   Windows Window definitions
131:    *       *
132:    *
*************************************************************
133:    *
134:
135:    IF NOT WEXIST("pampdr") ;
136:            OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.DBX" ;
137:            OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.SCX" ;
138:            OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.MNX" ;
139:            OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.PRG" ;
140:            OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.FRX" ;
141:            OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.QPR"
142:        DEFINE WINDOW pampdr ;
143:            AT  0.000, 0.000  ;
144:            SIZE 21.923,77.000 ;
```

```
145:          FONT "MS Sans Serif", 8 :
146:          FLOAT :
147:          NOCLOSE :
148:          MINIMIZE :
149:          SYSTEM
150:       MOVE WINDOW pampdr CENTER
151:    ENDIF
152:
153:
154:    *
*************************************************************
155:    *        *
156:    *        *          CTRL_ROL/Windows Setup Code -
SECTION 2
157:    *        *
158:    *
*************************************************************
159:    *
160:
161:    #REGION 1
162:    **********
163:    SELE mes
164:    LOCATE FOR mes = vmes
165:    STOR nom_mes TO mmes
166:    SELE sst
167:    STOR sst.ciudad TO ope_rol
168:    DO CASE
169:    CASE N = 2
170:       SET ORDER TO tipo_act
171:    CASE N = 3
172:       SET ORDER TO sist_cod
173:    ENDCASE
174:
175:    #REGION 1
176:    DEFINE POPUP _r4h0lwana :
177:       PROMPT FIELD sst.ciudad :
178:       SCROLL
179:
180:    *
*************************************************************
181:    *        *
182:    *        *              CTRL_ROL/Windows Screen Layout
183:    *        *
184:    *
*************************************************************
185:    *
186:
187:    #REGION 1
188:    IF WVISIBLE("pampdr")
189:       ACTIVATE WINDOW pampdr SAME
190:    ELSE
191:       ACTIVATE WINDOW pampdr NOSHOW
192:    ENDIF
193:    @ 13.231,39.200 SAY "MES....:"  :
194:       FONT "Courier New", 10 :
195:       STYLE "BIT"
196:    @ 11.308,39.200 SAY "A_O....:"  :
```

```
197:          FONT "Courier New", 10 ;
198:          STYLE "BIT"
199:       @ 6.692,8.000 TO 16.154,34.000 ;
200:          PEN 1, 8 ;
201:          STYLE "16"
202:       @ 18.154,14.200 TO 21.000,62.600 ;
203:          PEN 1, 8 ;
204:          STYLE "16"
205:       @ 7.692,39.400 GET opc_qui ;
206:          PICTURE "@*RVN \<PRIMERA QUINCENA:\ SEGUNDA
QUINCENA" ;
207:          SIZE 1.308,25.667,0.308 ;
208:          DEFAULT 1 ;
209:          FONT "MS Serif", 8 ;
210:          STYLE "BT" ;
211:          WHEN mquincena
212:       @ 11.385,53.600 GET manio ;
213:          SIZE 1.000,6.600 ;
214:          DEFAULT " " ;
215:          FONT "MS Sans Serif", 8 ;
216:          PICTURE "9999" ;
217:          VALID manio > 1990 AND manio < 2999 ;
218:          ERROR 'AÑO INVALIDO ' ;
219:          DISABLE
220:       @ 13.308,53.600 GET vmes ;
221:          SIZE 1.000,2.600 ;
222:          DEFAULT " " ;
223:          FONT "MS Sans Serif", 8 ;
224:          PICTURE "99" ;
225:          VALID _r4h01wax5() ;
226:          ERROR 'MES INVALIDO (1..12)' ;
227:          DISABLE
228:       @ 18.692,17.400 GET opc_ser ;
229:          PICTURE "@*HN \ ACEPTAR;\<EDITAR;\<SALIR" ;
230:          SIZE 1.769,11.667,0.667 ;
231:          DEFAULT 1 ;
232:          FONT "MS Sans Serif", 8 ;
233:          STYLE "B" ;
234:          VALID _r4h01wb14() ;
235:          DISABLE
236:       @ 1.385,11.200 SAY "REGISTRAR DATOS DE DESCUENTOS" +
CHR(13) + ;
237:          "PARA PROCESAR ROLES DE :"  ;
238:          SIZE 2.000,38.857, 0.000 ;
239:          PICTURE "@I" ;
240:          FONT "MS Serif", 10 ;
241:          STYLE "BIT"
242:       @ 0.923,0.600 TO 4.154,75.600 ;
243:          PEN 1, 8 ;
244:          STYLE "16"
245:       @ 13.308,57.800 GET mes.nom_mes ;
246:          SIZE 1.000,17.800 ;
247:          DEFAULT " " ;
248:          FONT "MS Sans Serif", 8 ;
249:          PICTURE "@K" ;
250:          WHEN .F.
251:       @ 8.308,11.000 GET opc_rol ;
```

```
252:          PICTURE "@&N" :
253:          POPUP _r4h01wapa :
254:          SIZE 6.923,20.800 :
255:          DEFAULT " " :
256:          FONT "MS Sans Serif", 8 :
257:          STYLE "" :
258:          VALID _r4h01wb5s()
259:
260:      IF NOT WVISIBLE("pampdr")
261:          ACTIVATE WINDOW pampdr
262:      ENDIF
263:
264:      READ CYCLE :
265:          DEACTIVATE .F.
266:
267:      RELEASE WINDOW pampdr
268:
269:      *
************************************************************
270:      *          *
271:      *          *                    Windows Closing Databases
272:      *          *
273:      *
************************************************************
274:      *
275:
276:      IF USED("mes")
277:          SELECT mes
278:          USE
279:      ENDIF
280:
281:      IF USED("sst")
282:          SELECT sst
283:          USE
284:      ENDIF
285:
286:      IF USED("dep")
287:          SELECT dep
288:          USE
289:      ENDIF
290:
291:      IF USED("men")
292:          SELECT men
293:          USE
294:      ENDIF
295:
296:      SELECT (m.currarea)
297:
298:      RELEASE POPUPS _r4h01wapa
299:
300:      #REGION 0
301:
302:      SET readborder &rborder
303:
304:      IF m.talkstat = "ON"
305:          SET TALK ON
306:      ENDIF
```

```
  307:     IF m.compstat = "ON"
  308:        SET COMPATIBLE ON
  309:     ENDIF
  310:
  311:
  312:     *
*************************************************************
  313:     *        *
  314:     *        *                CTRL_ROL/Windows Cleanup Code
  315:     *        *
  316:     *
*************************************************************
  317:     *
  318:
  319:     #REGION 1
  320:     ****************
  321:
  322:
  323:     *
*************************************************************
  324:     *        *
  325:     *        *  CTRL_ROL/Windows Supporting Procedures and
Functions
  326:     *        *
  327:     *
*************************************************************
  328:     *
  329:
  330:     #REGION 1
  331:
*!*************************************************************************
**************
  332:  *!
  333:  *!       Procedure: ACEPTAR
  334:  *!
  335:  *!           Calls: VAL_PRO()          (function in
VREG_PTE.SPR)
  336:  *!                : PREPARAR           (procedure in
VREG_PTE.SPR)
  337:  *!                : DA_ASISG.PRG
  338:  *!                : DE_EGRG1.PRG
  339:  *!                : DE_EGRG2.PRG
  340:  *!                : DE_EGRMN.PRG
  341:  *!                : DE_JEJUB.PRG
  342:  *!                : EGR_VAR1.PRG
  343:  *!                : ROL_Q1.SPR
  344:  *!                : ROL_Q2.SPR
  345:  *!                : ROL_MEN.SPR
  346:  *!                : ROL_JUB.SPR
  347:  *!                : ACT_PTMO.SPR
  348:  *!                : MO_PPLAB.SPR
  349:  *!
  350:  *!    Report Forms: VAL_NEGA.FRX
  351:  *!
  352:
*!*************************************************************************
**************
```

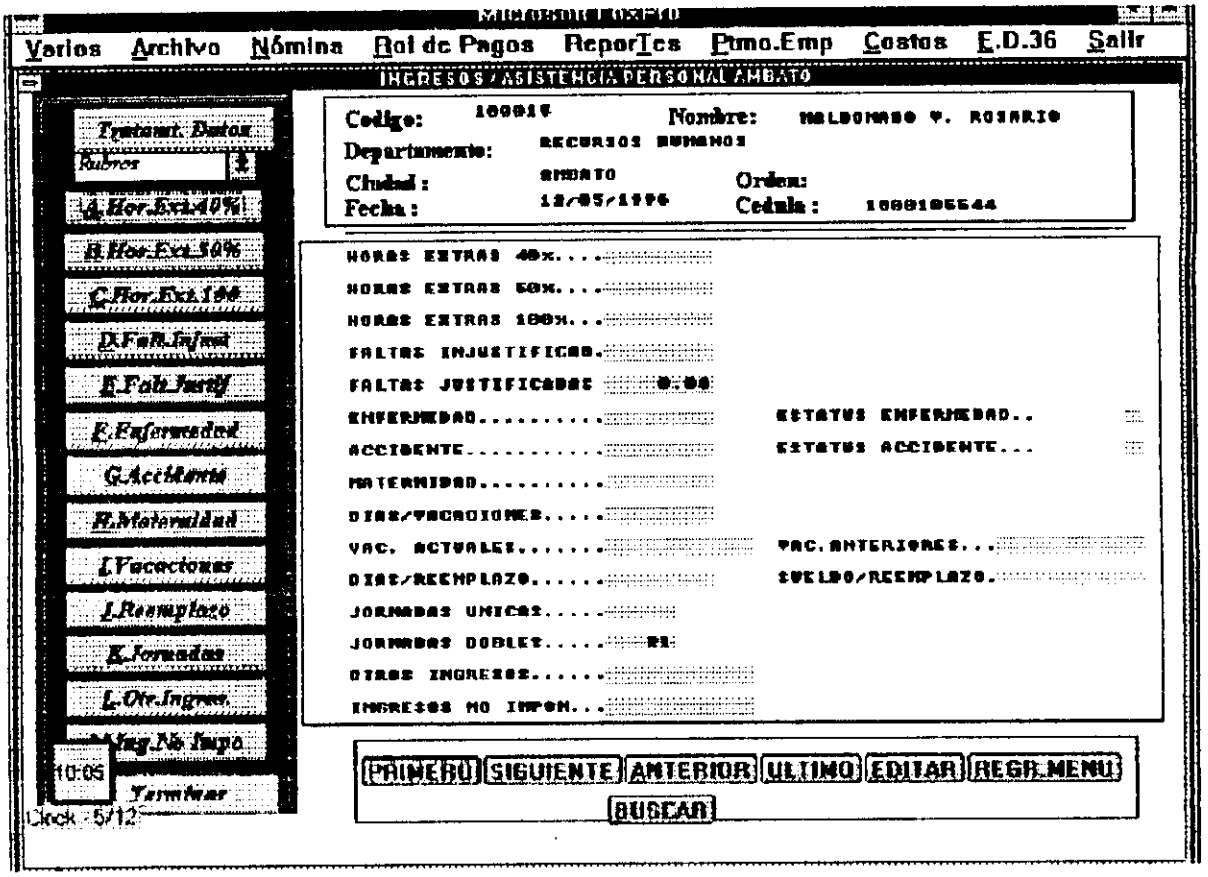
```
  353:      PROCEDURE aceptar
  354:          ******************
  355:          SHOW GETS DISABLE
  356:          SHOW GET opc_ser,2 ENABLE
  357:          SHOW GET opc_ser,3 ENABLE
  358:          SELECT sst
  359:          SEEK mrol
  360:          STOR ' ' TO tip
  361:          STOR .F. TO PROC
  362:          SET DISP TO VGA50
  363:        IF val_pro()
  364:          IF N = 2
  365:             DO da_asisg WITH sst.cod_ciu, sst.ciudad,
PROC, manio, vmes
  366:          ELSE
  367:             DO CASE
  368:             CASE sst.tipo = 0
  369:                DO CASE
  370:                CASE opc_qui = 1
  371:                     STOR 'PRIMERA QUINCENA' TO tip
  372:                     DO de_egrq1 WITH sst.cod_ciu,
sst.ciudad, :
  373:                          opc_qui, tip, PROC
  374:                CASE opc_qui = 2
  375:                     STOR 'SEGUNDA QUINCENA' TO tip
  376:                     DO de_egrq2 WITH sst.cod_ciu,
sst.ciudad, :
  377:                          opc_qui, tip, PROC
  378:                ENDCASE
  379:             CASE sst.tipo = 1
  380:                STOR 3 TO opc_qui
  381:                DO de_egrmn WITH sst.cod_ciu,
sst.ciudad, :
  382:                     opc_qui, PROC
  383:             CASE sst.tipo = 2
  384:                STOR 4 TO opc_qui
  385:                DO de_iejub WITH manio,vmes
  386:             ENDCASE
  387:          ENDIF
  388:        ENDIF
  389:          SELE mon
  390:          SET FILTER TO
  391:          RETURN
  392:
  393:          ****************
  394:
*!*************************************************************
*************
  395:  *!
  396:  *!        Function: VAL_PRO
  397:  *!
  398:  *!       Called by: ACEPTAR              (procedure in
VREG_PTE.SPR)
  399:  *!
  400:
*!*************************************************************
*************
```

```
401:      FUNCTION val_pro
402:          ****************
403:          malias = ALIAS()
404:          SELE men
405:          vret = .T.
406:          SET FILTER TO men.anio = manio AND men.mes = vmes
:
407:             AND men.ciudad = sst.cod_ciu
408:          GO TOP
409:          IF EOF()
410:             IF NOT WEXIST('VARIOS')
411:                DEFINE WINDOW varios FROM 0,0 TO 7,35 TITLE
'ALERTA ..!' SYSTEM
412:             ENDIF
413:             ACTIVATE WINDOW varios
414:             MOVE WINDOW varios CENTER
415:             @ 1,2 SAY 'No Existen Datos del
Rol/'+sst.ciudad
416:             @ 2,3 SAY 'Archivo no ha sido generado...'
417:             STOR 1 TO opc_ac
418:             @ 4,12 GET opc_ac PICT '@*H \<ACEPTAR';
419:                SIZE 1.308,12.500,0.308 ;
420:                DEFAULT 1 ;
421:                FONT "MS Serif", 8 ;
422:                STYLE "BT"
423:             READ MODAL
424:             RELE WIND varios
425:             STOR .F. TO vret
426:          ENDIF
427:          SELECT &malias
428:          RETURN     vret
429:          ****************
430:
*!*************************************************************************
**************
431:  *!
432:  *!       Procedure: EDITAR
433:  *!
434:  *!       Called by: CREAR              (procedure in
AACT_SST.SPR)
435:  *!
436:  *!           Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
437:  *!                : CABEZA             (procedure in
PERSONAL.PRG)
438:  *!                : INGRESO            (procedure in
ACT_FTMO.SPR)
439:  *!                : CREA_TMP           (procedure in
MO_PDLAB.SPR)
440:  *!                : MFROW              (procedure in
MO_PDLAB.SPR)
441:  *!                : REF_SCR            (procedure in
MO_PDLAB.SPR)
442:  *!                : MENSAJES           (procedure in
MO_PDLAB.SPR)
443:  *!
444:
```

```
*!*************************************************************************
**************
  445:      PROCEDURE editar
  446:           *****************
  447:           SHOW GETS DISABLE
  448:           SHOW GET opc_rol ENABLE
  449:  <=======RETURN
  450:           ***************
  451:
*!*************************************************************************
*************
  452:  *!
  453:  *!        Procedure: SALIR
  454:  *!
  455:  *!        Called by: _RAA1732TG()        (function in
AT_ESCLS.SPR)
  456:  *!                 : DA_ASISG.PRG
  457:  *!                 : DE_EGRQ1.PRG
  458:  *!                 : DE_EGRQ2.PRG
  459:  *!                 : DE_EGRMN.PRG
  460:  *!                 : DE_IEJUB.PRG
  461:  *!
  462:  *!           Calls: DEAC_KEYS            (procedure in
AACT_SST.SPR)
  463:  *!
  464:
*!*************************************************************************
**************
  465:      PROCEDURE salir
  466:           ***************
  467:           SET DELE ON
  468:           CLEAR READ
  469:  <=======RETURN
  470:
  471:           *
*************************************************************
  472:           *        *
  473:           *        *  R4HOLWAX5              VMES VALID
  474:           *        *
  475:           *        * Function Origin:
  476:           *        *
  477:           *        * From Platform:         Windows
  478:           *        * From Screen:           CTRL_ROL,
Record Number:   12
  479:           *        * Variable:              VMES
  480:           *        * Called By:             VALID Clause
  481:           *        * Object Type:           Field
  482:           *        * Snippet Number:        1
  483:           *        *
  484:           *
*************************************************************
  485:           *
  486:
*!*************************************************************************
*************
  487:  *!
  488:  *!        Function: _R4HOLWAX5
```

```
489:  *!
490:  *!        Called by: CTRL_ROL.SPR
491:  *!
492:
*!*************************************************************
**************
493:      FUNCTION _r4h01wax5      &&  VMES VALID
494:          #REGION 1
495:          IF vmes > 0 AND vmes < 13
496:              STOR .T. TO vret
497:              SELE mes
498:              LOCATE FOR vmes = mes
499:              STOR nom_mes TO nmes
500:              SHOW GET mes.nom_mes
501:          ELSE
502:              STOR .F. TO vret
503:          ENDIF
504:  <=======RETURN vret
505:
506:          *
*************************************************************
507:          *         *
508:          *         * _R4HOLWB14            OPC_SCR VALID
509:          *         *
510:          *         * Function Origin:
511:          *         *
512:          *         * From Platform:        Windows
513:          *         * From Screen:          CTRL_ROL.
Record Number:   13
514:          *         * Variable:             OPC_SCR
515:          *         * Called By:            VALID Clause
516:          *         * Object Type:          Push Button
517:          *         * Snippet Number:       2
518:          *         *
519:          *
*************************************************************
520:          *
521:
*!*************************************************************
**************
522:  *!
523:  *!        Function: _R4HOLWB14
524:  *!
525:  *!      Called by: CTRL_ROL.SPR
526:  *!
527:  *!           Calls: &OPC_SCR.PRG
528:  *!
529:
*!*************************************************************
**************
530:      FUNCTION  r4h01wb14      &&  OPC_SCR VALID
531:          #REGION 1
532:          DO &ope_scr
533:
534:          *
*************************************************************
535:          *         *
```

```
536:          *        * _R4HOLWB5S            OPC_ROL VALID
537:          *        *
538:          *        * Function Origin:
539:          *        *
540:          *        * From Platform:        Windows
541:          *        * From Screen:          CTRL_ROL.
Record Number:   17
542:          *        * Variable:             OPC_ROL
543:          *        * Called By:            VALID Clause
544:          *        * Object Type:          List
545:          *        * Snippet Number:       3
546:          *        *
547:          *
**************************************************************
548:          *
549:
*!*************************************************************************
**************
550:  *!
551:  *!        Function: _R4HOLWB5S
552:  *!
553:  *!       Called by: CTRL_ROL.SPR
554:  *!
555:
*!*************************************************************************
**************
556:      FUNCTION _r4h01wb5s      &&  OPC_ROL VALID
557:          #REGION 1
558:          STOR sst.cod ciu TO mrol
559:          STOR 0 TO opc_qui
560:          STOR .F. TO mquincena
561:        ┌=DO CASE
562:        ├====CASE opc_rol = 'TODOS'
563:        |       STOR 2 TO opc_qui
564:        |       STOR .F. TO mquincena
565:        ├====CASE sst.tipo = 0
566:        |       SHOW GET opc_qui ENABL
567:        |       STOR 1 TO opc_qui, mopc_qui
568:        |       STOR .T. TO mquincena
569:        └-ENDCASE
570:          SHOW GET opc_qui
571:          SHOW GET manio ENABLE
572:          SHOW GET vmes ENABLE
573:          SHOW GET opc_ser ENABLE
574:  *:  EOF: CTRL_ROL.scl
```


Varios Archivo Nómina Rol de Pagos ReporTes Pmo.Emp Costos E.D.36 Salir
INGRESOS/ASISTENCIA PERSONAL AMBATO
Tratamt. Datos
Rubros
A.Hor.Ext.40%
B.Hor.Ext.50%
C.Hor.Ext.100
D.Falt.Injust
E.Falt.Justif
F.Enfermedad
G.Accidente
H.Maternidad
I.Vacaciones
J.Reemplazo
K.Jornadas
L.Otr.Ingres.
Ing.No Impo
Terminar
10:05
Clock - 5/12
Codigo: 100016
Nombre: MALDONADO V. ROSARIO
Departamento: RECURSOS HUMANOS
Ciudad: AMBATO
Orden:
Fecha: 12/05/1996
Cedula: 1000105644
HORAS EXTRAS 40%....
HORAS EXTRAS 50%....
HORAS EXTRAS 100%...
FALTAS INJUSTIFICAD.
FALTAS JUSTIFICADAS 0.00
ENFERMEDAD.....
ESTATUS ENFERMEDAD..
ACCIDENTE.....
ESTATUS ACCIDENTE...
MATERNIDAD.....
DIAS/VACACIONES.....
VAC. ACTUALES.....
VAC. ANTERIORES...
DIAS/REEMPLAZO.....
SUELDO/REEMPLAZO.
JORNADAS UNICAS.....
JORNADAS DOBLES.....
OTROS INGRESOS.....
INGRESOS NO IMPON...
PRIMERO SIGUIENTE ANTERIOR ULTIMO EDITAR REGR.MENU
BUSCAR

```
    1:
*:*************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\DA_ASISG.PRG
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 05/02/1995 at 20:25:00
    8:  *:
    9:  *:  Procs & Fncts: TRAT_DATOS()
   10:  *:                : OPC_DES()
   11:  *:                : INDICES()
   12:  *:                : RUBROS
   13:  *:                : DESPLAZAR()
   14:  *:
   15:  *:          Set by: ACEPTAR              (procedure in
VREG_PTE.SPR)
   16:  *:
   17:  *:           Calls: TEXTO                (procedure in
GRAFICOS.PRG)
   18:  *:                : TRAT_DATOS()         (function in
DA_ASISG.PRG)
   19:  *:                : OPC_DES()            (function in
DA_ASISG.PRG)
   20:  *:                : SALIR                (procedure in
CTRL_RES.SPR)
   21:  *:
   22:  *:       Documented 19:15:18
FoxDoc version 3.00a
   23:
*:*************************************************************
**************
   24:       ***************************************
   25:       ** REGISTRO DE OTROS INGRESOS Y ASISTENCIA*
   26:       ***************************************
   27:       PARA veiu,msis,PROC,manio,vmes
   28:       STOR veiu TO mcod_ciu
   29:       *~~~~~~~~~~
   30:       *VARIABLES
   31:       *~~~~~~~~~~
   32:       REGIONAL
laso1,modo,salto,m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14,m
15,m16,m17,m18,m19,:
   33:
mcont,trat_dat,opc_bot,opc_des,vartag,mvartag,encontro,fila
   34:       DIMENSION vartag(20)
   35:       STOR "font "N",10 style "b" TO fsdesc && styl and font
of say comment
   36:       STOR " " TO fscampc && styl and font of say CHARACTER
fields
   37:       STOR " " TO fscampn && styl and font of say NUMERIC
fields
   38:       STOR .F. TO encontro
   39:
   40:       STORE SPACE(1) TO
```

```
m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14,m15,m16,m17,m18,m1
9,:
    41:         mcont,mvartag,modo
    42:       STOR 1 TO trat_dat,opc_bot
    43:       laso = .T.
    44:       STOR '' TO opc_des
    45:       STOR 10 TO fila
    46:       *~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
    47:       *BASES DE DATOS (APERTURA)
    48:       *~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
    49:       IF USED("tbpv")
    50:           SELECT tbpv
    51:           SET ORDER TO TAG "PD_EGRQ1"
    52:       ELSE
    53:           SELECT 0
    54:           USE (LOCFILE("dbfs\tb_ppvar.dbf","DBF","Where is
tb_ppvar?"));
    55:                 AGAIN ALIAS tbpv ;
    56:                 ORDER TAG "PD_EGRQ1"
    57:       ENDIF
    58:
    59:       IF USED("sec")
    60:           SELECT sec
    61:           SET ORDER TO TAG "secc_cod"
    62:       ELSE
    63:           SELECT 0
    64:           USE (LOCFILE("dbfs\seccion.dbf","DBF","Where is
Seccion?"));
    65:                 AGAIN ALIAS sec ;
    66:                 ORDER TAG "secc_cod"
    67:       ENDIF
    68:
    69:       IF USED("dgp")
    70:           SELECT dgp
    71:           SET ORDER TO TAG "cod_per"
    72:       ELSE
    73:           SELECT 0
    74:           USE (LOCFILE("dbfs\personal.dbf","DBF","Where is
personal?"));
    75:                 AGAIN ALIAS dgp ;
    76:                 ORDER TAG "cod_per"
    77:       ENDIF
    78:
    79:       SELECT tbpv
    80:       SET UNIQ ON
    81:       SET ORDER TO clabot_as
    82:       COUNT TO A
    83:       DIME arr_bot(A)
    84:       STOR '@*V ' TO mboton
    85:       SCAN
    86:           STOR mboton + ALLT(tbpv.boton)+';' TO mboton
    87:       ENDSCAN
    88:       SET UNIQUE OFF
    89:       SET ORDER TO pd_asist
    90:       COUNT TO A
    91:       DIME aux_arrbot(A,5)
    92:       COPY FIELD campo,detalle,boton,cuando,formato TO ARRAY
```

aux_arrbot

```
        STOR SUBS(mboton,1,LEN(mboton)-1) TO mboton
        STOR 0 TO opc_bot
        *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
        *RELACIONES ENTRE BASES DATOS
        *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
        SELE men
        SET FILTER TO ciudad = vciu AND mes = vmes
        SELE dgp
        SET FILTER TO num_ciu = vciu
        SET RELATION TO num_sec INTO sec ADDIT
        SET RELATION TO num_ciu INTO sst ADDIT
        SET RELATION TO cod_per INTO men ADDIT
        SELE dgp
        FOR i = 1 TO 20
          IF EMPTY(TAG(i))
              i = i - 1
              DIMENSION vartag(i)
              EXIT
          ELSE
              vartag(i) = TAG(i)
          ENDIF
        ENDFOR
        *set order to vartag(1)
        *Stor vartag(1) to mvartag
        SET ORDER TO cod_per
        STOR 'COD_PER' TO mvartag
        GOTO TOP

        *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
        *     DEFINICION DE VENTANAS      *
        *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
        DEFINE WINDOW esta FROM 0.0 TO 49.90 DOUBLE TITLE "
INGRESOS / ASISTENCIA PERSONAL " + msis SHADOW GROW ZOOM
        ACTIVATE WINDOW esta
        *~~~~~~~~~~~~~~~~~~~~~~~~~~~*
        * PROGRAMA PRINCIPAL
        *~~~~~~~~~~~~~~~~~~~~~~~~~~~*
        DO texto
        DO WHILE laso
            @ 0.1 TO 44.20 DOUBLE
            @ 0.1 FILL TO 44,20 COLOR GR+/B
            @ 1.4 GET m19;
                PICTURE "@*N T\<ratamt. Datos" ;
                SIZE 1.5,22 FONT 'R',10 STYLE "BI";
                VALID indices()

            @3.4 GET trat_dat FUNC '^
\ Rubros;\ General;\ Totales';
                SIZE 1.20 FONT 'N',8 STYLE "IB";
                ENAB VALID trat_datos()

            @ 6.3.167 GET opc_bot;
                PICTURE mboton;
                SIZE 1.30,22.00,0.20;
                DEFAULT 1;
```

```
                FONT 'R',10;
                STYLE "BI";
                WHEN trat_dat = 1;
                VALID opc_des()


            @ 42.5,4 GET m12;
                PICTURE "@*N \<Terminar" ;
                SIZE 1.3,22 FONT 'R',10 STYLE "BI";
                VALID salir()
            READ CYCLE MODAL
    ENDDO
<===RETUR
    *~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
    *FIN DE PROGRAMA PRINCIPAL
    *~~~~~~~~~~~~~~~~~~~~~~~~~~~~*

    ****************

*!**************************************************************************
*************
*!
*!      Procedure: SALIR
*!
*!      Called by: _RAA1732TG()          (function in
AT_ESCLS.SPR)
*!                  : DA_ASISG.PRG
*!                  : DE_EGRQ1.PRG
*!                  : DE_EGRQ2.PRG
*!                  : DE_EGRMN.PRG
*!                  : DE_IEJUB.PRG
*!
*!          Calls: DEAC_KEYS             (procedure in
AACT_SST.SPR)
*!

*!**************************************************************************
*************
    FUNCTION salir
        ****************
        SELE dep
        SET RELATION OFF INTO sec ADDIT
        SET RELATION OFF INTO sst ADDIT
        SET RELATION OFF INTO men ADDIT
        laso = .F.
        RELEASE WINDOW esta
<=======RETURN
<=======RETURN

        ****************

*!**************************************************************************
*************
*!
*!      Function: INDICES
*!

```

```
*!*************************************************************************
**************
  196:      FUNCTION indices
  197:          ****************
  198:          @4,7,65 GET mvartag FROM vartag PICTURE "@^" SIZE
1,12
  199:          READ
  200:          SET ORDER TO mvartag
  201:  <=======RETURN
  202:
  203:          ********************
  204:
*!*************************************************************************
**************
  205:  *!
  206:  *!        Function: TRAT_DATOS
  207:  *!
  208:  *!      Called by: DA_ASISG.PRG
  209:  *!               : DE_EGRQ1.PRG
  210:  *!               : DE_EGRQ2.PRG
  211:  *!               : DE_EGRMN.PRG
  212:  *!               : DE_IEJUB.PRG
  213:  *!
  214:  *!          Calls: OPC_DES()          (function in
DA_ASISG.PRG)
  215:  *!               : TOTALES            (procedure in
VF_INFRTA.SPR)
  216:  *!               : RUBROS             (procedure in
DA_ASISG.PRG)
  217:  *!
  218:
*!*************************************************************************
**************
  219:      FUNCTION trat_datos
  220:          ********************
  221:        DO CASE
  222:        ===CASE trat_dat = 1 && 'Rubros'
  223:                *                    =OPC_DES()
  224:        ===CASE trat_dat = 2 && 'General'
  225:              =opc_des()
  226:        ===CASE trat_dat = 3 && 'Totales'
  227:              reg_ant = RECNO()
  228:              DO totales
  229:              GOTO reg_ant
  230:              DO rubros WITH .F.
  231:        ===ENDCASE
  232:          STOR 1 TO trat_dat
  233:          SHOW GET trat_dat
  234:  <=======RETURN
  235:
  236:          ****************
  237:
*!*************************************************************************
**************
  238:  *!
  239:  *!        Procedure: TOTALES
  240:  *!
```

```
  241:  *!      Called by: SECC()              (function in
PERSONAL.PRG)
  242:  *!                : TRAT_DATOS()       (function in
DA_ASISG.PRG)
  243:  *!
  244:  *!          Calls: SIST_TOT            (procedure in
EGR_VARI.PRG)
  245:  *!
  246:
*!*************************************************************************
**************
  247:      PROCEDURE totales
  248:          ******************
  249:          @ 1.35 SAY SPAC(6) &&VarCod &fscampC
  250:          @ 1.5.60 SAY "TOTALES GENERALES"
  251:          @ 3.40  SAY SPAC(35)
  252:          @ 5.40 SAY ALLTRI(sst.ciudad) &fscampc
  253:          @ 6.5.40  SAY DATE() &fscampc
  254:
  255:          SUM ALL &aux_arrbot(01.1)  TO t1 &&he1
  256:          SUM ALL &aux_arrbot(02.1)  TO t2 &&he2
  257:          SUM ALL &aux_arrbot(03.1)  TO t3 &&he3
  258:          SUM ALL &aux_arrbot(04.1)  TO t4  &&fi
  259:          SUM ALL &aux_arrbot(05.1)  TO t5  &&fj
  260:          SUM ALL &aux_arrbot(06.1)  TO t6  &&Enf
  261:          SUM ALL &aux_arrbot(08.1)  TO t7  &&Acc
  262:          SUM ALL &aux_arrbot(10.1)  TO t8  &&Mat
  263:          SUM ALL &aux_arrbot(11.1)  TO t9  &&Vac_t
  264:          SUM ALL &aux_arrbot(12.1)  TO t10 &&Vac_c
  265:          SUM ALL &aux_arrbot(13.1)  TO t11 &&Vac_n
  266:          SUM ALL &aux_arrbot(14.1)  TO t12 &&dr
  267:          SUM ALL &aux_arrbot(15.1)  TO t13 &&Remp
  268:          SUM ALL &aux_arrbot(16.1)  TO t14 &&ju
  269:          SUM ALL &aux_arrbot(17.1)  TO t15 &&jd
  270:          SUM ALL &aux_arrbot(18.1)  TO t16 &&OTING
  271:          SUM ALL &aux_arrbot(19.1)  TO t17 &&INGNIMP
  272:
  273:          @ 10.45 SAY t1 piet "9,999.99" &fscampn  FUNCTION
"@Z"
  274:          @ 12.45 SAY t2 piet "9,999.99" &fscampn  FUNCTION
"@Z"
  275:          @ 14.45 SAY t3 piet "9,999.99" &fscampn  FUNCTION
"@Z"
  276:          @ 16.45 SAY t4 piet "9,999.99" &fscampn  FUNCTION
"@Z"
  277:          @ 18.45 SAY t5 piet "9,999.99" &fscampn  FUNCTION
"@Z"
  278:          @ 20.45 SAY t6 piet "9,999.99" &fscampn  FUNCTION
"@Z"
  279:          @ 20.65 SAY " "
  280:          @ 22.45 SAY t7 piet "9,999.99" &fscampn  FUNCTION
"@Z"
  281:          @ 22.65 SAY " "
  282:          @ 24.45 SAY t8 piet "9,999.99" &fscampn  FUNCTION
"@Z"
  283:          @ 26.45 SAY t9 piet "9,999.99" &fscampn  FUNCTION
"@Z"
```

```
  284:          @ 28,45 SAY t10 &fscampn pict '999,999,999'
FUNCTION '@Z'
  285:          @ 28,75 SAY t11 &fscampn pict '999,999,999'
FUNCTION '@Z'
  286:          @ 30,45 SAY t12 pict '9,999,99'&fscampn FUNCTION
'@Z'
  287:          @ 30,75 SAY t13 &fscampn pict '999,999,999'
FUNCTION '@Z'
  288:          @ 32,45 SAY t14 pict '9,999'&fscampn FUNCTION '@Z'
  289:          @ 34,45 SAY t15 pict '9,999'&fscampn FUNCTION '@Z'
  290:          @ 36,45 SAY t16 pict '999,999,999' &fscampn
FUNCTION '@Z'
  291:          @ 38,45 SAY t17 pict '999,999,999' &fscampn
FUNCTION '@Z'
  292:
  293:          DEFI WIND CONT FROM 41,40 TO 45,58 PANEL
  294:          ACTI WIND CONT
  295:          @ 0,5,1 GET mcont;
  296:             PICTURE "@* \<Aceptar" ;
  297:             SIZE 1,20,20 FONT 'R',10 STYLE "BI"
  298:          READ
  299:          RELE WIND CONT
  300:          @3,4 GET trat_dat FUNC ''
\<Rubros;\<General;\<Totales' ;
  301:             SIZE 1,20 FONT 'N',8 STYLE "IB" DISABLE
  302: <=======RETURN
  303:
  304:          ****************
  305:
*!*************************************************************************
**************
  306:  *!
  307:  *!       Procedure: RUBROS
  308:  *!
  309:  *!       Called by: BUSC_CLV()          (function in
DGP_CARG.SPR)
  310:  *!                : TRAT_DATOS()        (function in
DA_ASISG.PRG)
  311:  *!                : DESPLAZAR()         (function in
DA_ASISG.PRG)
  312:  *!
  313:  *!          Calls: VAL_MED()           (function in
DE_EGRG1.PRG)
  314:  *!
  315:
*!*************************************************************************
**************
  316:      PROCEDURE rubros
  317:          ****************
  318:          PARAMETER vmodo
  319:          SELE men
  320:          IF !EOF()
  321:             STOR .F. TO encontro
  322:             @ 1,  35  SAY dgp.cod_per &&VarCod &fscampC
  323:             @ 1.5,60  SAY SUBS(dgp.nombre ,1,25)
  324:             @ 3,  40  SAY sec.nombre &fscampc
  325:             @ 5,  40  SAY ALLTRI(sst.ciudad) &fscampc
```

```
  326:          |     @ 6,5,40  SAY DATE() &fscampc
  327:          |     @ 7,  65  SAY dgp.cedula &fscampc
  328:          |     @ 10,45 GET &aux_arrbot(01,1) WHEN
&aux_arrbot(01,4) PICT &aux_arrbot(01,5) FUNC '@z' &fscampn
&&Hex1
  329:          |     @ 12,45 GET &aux_arrbot(02,1) WHEN
&aux_arrbot(02,4) PICT &aux_arrbot(02,5) FUNC '@z' && Hex2
&fscampN pict '99.99'
  330:          |     @ 14,45 GET &aux_arrbot(03,1) WHEN
&aux_arrbot(03,4) PICT &aux_arrbot(03,5) FUNC '@z' && Hex3
&fscampN pict '99.99'
  331:          |
  332:          |     @ 16,45 GET &aux_arrbot(04,1) WHEN
&aux_arrbot(04,4) PICT &aux_arrbot(04,5) FUNC '@z' && Faltas1
&fscampN pict '99.99'
  333:          |     @ 18,45 GET &aux_arrbot(05,1) WHEN
&aux_arrbot(05,4) PICT &aux_arrbot(05,5) FUNC '@z' && Faltas2
&fscampN pict '99.99'
  334:          |     @ 20,45 GET &aux_arrbot(06,1) WHEN
&aux_arrbot(06,4) PICT &aux_arrbot(06,5) FUNC '@z' && Enferm
&fscampN pict '99.99'
  335:          |     @ 20,85 GET &aux_arrbot(07,1) WHEN
&aux_arrbot(07,4) PICT &aux_arrbot(07,5) VALID
INLIST(men.stsacc,0,1,2,3) ERROR "Dato no valido. Ingrese (0 -
3)" FUNC '@z' && Stsenf &fscampN
  336:          |     @ 22,45 GET &aux_arrbot(08,1) WHEN
&aux_arrbot(08,4) PICT &aux_arrbot(08,5) FUNC '@z' && Accide
&fscampN pict '99.99'
  337:          |     @ 22,85 GET &aux_arrbot(09,1) WHEN
&aux_arrbot(09,4) PICT &aux_arrbot(09,5) VALID
INLIST(men.stsacc,0,1,2,3) ERROR "Dato no valido. Ingrese (0 -
3)" FUNC '@z' && Stsacc &fscampN
  338:          |     @ 24,45 GET &aux_arrbot(10,1) WHEN
&aux_arrbot(10,4) PICT &aux_arrbot(10,5) FUNC '@z' && Matern
&fscampN pict '99.99'
  339:          |     @ 26,45 GET &aux_arrbot(11,1) WHEN
&aux_arrbot(11,4) PICT &aux_arrbot(11,5) FUNC '@z' && Vac_Tie
&fscampN pict '99.99'
  340:          |     @ 28,45 GET &aux_arrbot(12,1) WHEN
&aux_arrbot(12,4) PICT &aux_arrbot(12,5) FUNC '@z' && Vac_act
&fscampN
  341:          |     @ 28,75 GET &aux_arrbot(13,1) WHEN
&aux_arrbot(13,4) PICT &aux_arrbot(13,5) FUNC '@z' && Vac_ant
&fscampN
  342:          |     @ 30,45 GET &aux_arrbot(14,1) WHEN
&aux_arrbot(14,4) PICT &aux_arrbot(14,5) FUNC '@z' && diastrempl
&fscampN  pict '99.99'
  343:          |     @ 30,75 GET &aux_arrbot(15,1) WHEN
&aux_arrbot(15,4) PICT &aux_arrbot(15,5) FUNC '@z' && reemplazos
&fscampN
  344:          |     @ 32,45 GET &aux_arrbot(16,1) WHEN
&aux_arrbot(16,4) PICT &aux_arrbot(16,5) FUNC '@z' && Jor_Uni
&fscampN pict '99'
  345:          |     @ 34,45 GET &aux_arrbot(17,1) WHEN
&aux_arrbot(17,4) PICT &aux_arrbot(17,5) FUNC '@z' && Jor_Dob
&fscampN pict '99'
  346:          |     @ 36,45 GET &aux_arrbot(18,1) WHEN
```

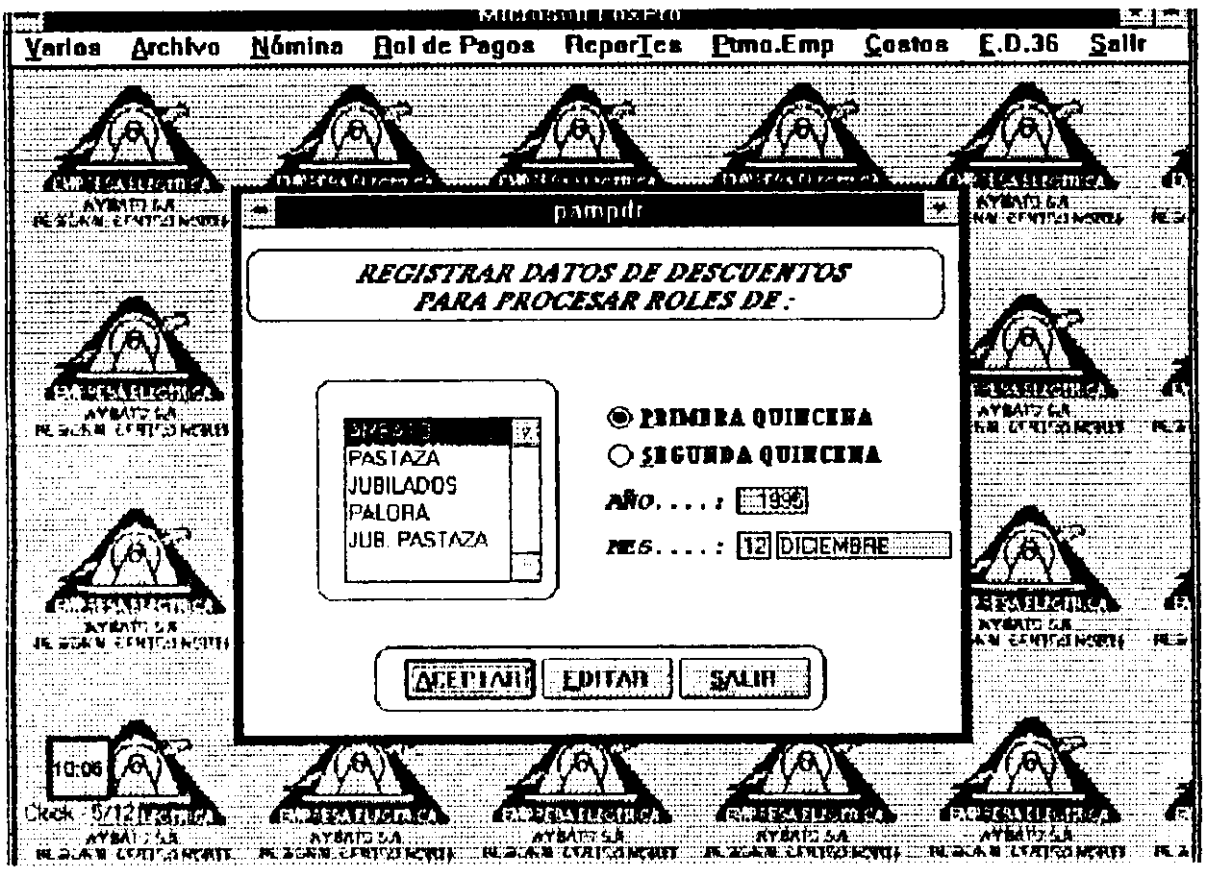
```
&aux_arrbot(18,4) PICT &aux_arrbot(18,5) FUNC '@z' && OTR_ING
&fscampN
  347:          @ 38,45 GET &aux_arrbot(19,1) WHEN
&aux_arrbot(19,4) PICT &aux_arrbot(19,5) FUNC '@z' && ING_NOIMP
&fscampN
  348:          IF vmodo
  349:             READ
  350:             FLUSH
  351:          ENDIF
  352:       ENDIF
  353:       SELE dep
  354:       RETURN
  355:       ******************
  356:
*!*************************************************************************
**************
  357:  *!
  358:  *!       Procedure: TEXTO
  359:  *!
  360:  *!       Called by: DA_ASISG.PRG
  361:  *!                : DE_EGRQ1.PRG
  362:  *!                : DE_EGRQ2.PRG
  363:  *!                : DE_EGRMN.PRG
  364:  *!                : DE_IEJUB.PRG
  365:  *!
  366:
*!*************************************************************************
**************
  367:    PROCEDURE texto
  368:       ******************
  369:       @0,5,23 TO 8,5,85
  370:
  371:       @ 1,25  SAY "Codigo:" &fsdesc
  372:       @ 1,50  SAY "Nombre:"&fsdesc
  373:       @ 3,25  SAY "Departamento:"&fsdesc
  374:       @ 5,25  SAY "Ciudad :"&fsdesc
  375:       @ 5,55  SAY "Orden: " &fsdesc
  376:       @ 6,5,35 SAY "Fecha :"&fsdesc
  377:       @ 6,5,35 SAY "Cedula :"&fsdesc
  378:
  379:
  380:       @ 8,5,25 SAY REPLICATE("=",60)
  381:       @fila-.5,21,5 TO fila + 29,5 ,87
  382:       @ 10,25 SAY aux_arrbot(1,2) &&"H. Extras
40%:"&fsdesc
  383:       @ 12,25 SAY aux_arrbot(2,2) && "H. Extras
50%:"&fsdesc
  384:       @ 14,25 SAY aux_arrbot(3,2) && "H. Extars
100%:"&fsdesc
  385:       @ 16,25 SAY aux_arrbot(4,2) && "Faltas
Injustificadas:"&fsdesc
  386:       @ 18,25 SAY aux_arrbot(5,2) && "Faltas
Justificadas  :"&fsdesc
  387:       @ 20,25 SAY aux_arrbot(6,2) && "Enfermedad
:"&fsdesc
  388:       @ 20,58 SAY aux_arrbot(7,2) && "Estatus :"&fsdesc
  389:       @ 22,25 SAY aux_arrbot(8,2) && "Accidentes
```

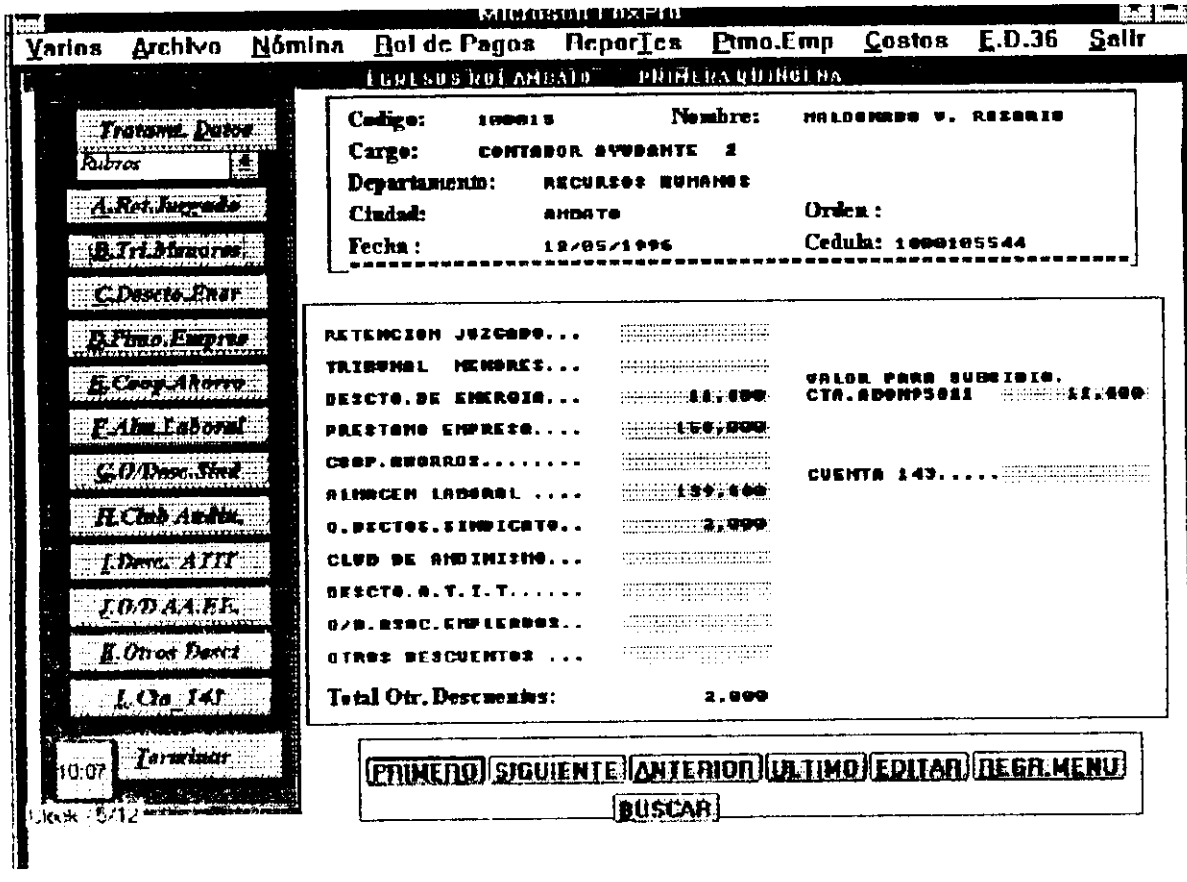
```
:"&fsdesc
  390:          @ 22,58 SAY aux_arrbot(9,2) && "Estatus :"&fsdesc
  391:          @ 24,25 SAY aux_arrbot(10,2) && "Maternidad
:"&fsdesc
  392:          @ 26,25 SAY aux_arrbot(11,2) && "Vacaciones
Dias:"&fsdesc
  393:          @ 28,25 SAY aux_arrbot(12,2) && "Vac. Act.(Val)
:"&fsdesc
  394:          @ 28,58 SAY aux_arrbot(13,2) && "Anteriores
(Val):"&fsdesc
  395:          @ 30,25 SAY aux_arrbot(14,2) && "Dias Reemplazo
:"&fsdesc
  396:          @ 30,58 SAY aux_arrbot(15,2) && "Sueldo
:"&fsdesc
  397:          @ 32,25 SAY aux_arrbot(16,2) && "Jornada
Unica:"&fsdesc
  398:          @ 34,25 SAY aux_arrbot(17,2) && "Jornada
Doble:"&fsdesc
  399:          @ 36,25 SAY aux_arrbot(18,2) && "Otros
ingresos:"&fsdesc
  400:          @ 38,25 SAY aux_arrbot(19,2) && "Ingresos no
imponibles:"&fsdesc
  401: <=======RETURN
  402:
  403:          ****************
  404:
*!*************************************************************
**************
  405:  *!
  406:  *!        Function: OPC_DES
  407:  *!
  408:  *!      Called by: DA_ASISG.PRG
  409:  *!               : DE_EGRQ1.PRG
  410:  *!               : DE_EGRQ2.PRG
  411:  *!               : DE_EGRMN.PRG
  412:  *!               : DE_IEJUB.PRG
  413:  *!               : TRAT_DATOS()        (function in
DA_ASISG.PRG)
  414:  *!
  415:  *!         Calls: DESPLAZAR()          (function in
DA_ASISG.PRG)
  416:  *!              : BUSC_CLV()           (function in
DGE_CARG.SPR)
  417:  *!
  418:
*!*************************************************************
**************
  419:      FUNCTION opc_des
  420:          ***************
  421:          @40,25 TO 45,5,85
  422:          STOR .T. TO vopc_des
  423:          STOR .F. TO vmodo
  424:          DO WHILE vopc_des
  425:              @ 41,5,26 GET opc_des :
  426:                  PICTURE '@*H
\<PRIMERO:\<SIGUIENTE:\<ANTERIOR:\<ULTIMO:\<EDITAR:\<REGR.MENU':
  427:                  SIZE 1,2,7,0,25:
```

```
428:                DEFAULT 1;
429:                FONT "MS Sans Serif", 5;
430:                STYLE "BT";
431:                VALID desplazar(opc_des)
432:             @ 44.45 GET opc_busc ;
433:                PICTURE "@*H \<BUSCAR";
434:                SIZE 1.2,7.0.25;
435:                DEFAULT 1;
436:                FONT "MS Sans Serif", 5;
437:                STYLE "BT";
438:                VALID busc_clv()
439:             READ CYCLE MODAL
440:          ENDDO
441:          @40.25 TO 46.85 CLEAR
442:       RETURN
443:
444:          ****************
445:
*!*************************************************************************
**************
446:    *!
447:    *!          Function: DESPLAZAR
448:    *!
449:    *!          Called by: OPC_DES()          (function in
DA_ASISG.PRG)
450:    *!
451:    *!          Calls: RUBROS                 (procedure in
DA_ASISG.PRG)
452:    *!
453:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!
454:    FUNCTION desplazar
455:             !!!!!!!!!!!!!!!!
456:             PARAMETER modo
457:             SET ORDER TO mvartag
458:             STOR .F. TO vmodo
459:          DO CASE
460:          CASE modo = 'PRIMERO'
461:             GOTO TOP
462:             WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
463:          CASE modo = 'SIGUIENTE'
464:             IF NOT EOF()
465:                SKIP
466:             ENDIF
467:             IF EOF()
468:                GO BOTTOM
469:                WAIT WIND NOWAIT 'FIN DE ARCHIVO'
470:             ENDIF
471:          CASE modo = 'ANTERIOR'
472:             IF NOT BOF()
473:                SKIP -1
474:             ENDIF
475:             IF BOF()
476:                GO TOP
477:                WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
478:             ENDIF
```

```
479:          ├──CASE modo = 'ULTIMO'
480:          │      GOTO BOTTOM
481:          │      WAIT WIND NOWAIT 'FIN DE ARCHIVO'
482:          ├──CASE modo = 'EDITAR' AND NOT PROC
483:          │      STOR .T. TO vmodo
484:          ├──CASE modo  = 'CANC.EDITAR'
485:          │      SHOW GET ope_des,5 PROMPT '\<EDITAR' ENABLE
486:          │      SHOW GET ope_des,6 ENABLE
487:          ├──CASE modo = 'REGR.MENU'
488:          │      STOR .F. TO vope_des
489:          └─ENDCASE
490:            SET ORDER TO mvartag
491:            DO rubros WITH vmodo
492:  <═══════RETURN
493:
494:            ********************
495:
*!*************************************************************************
**************
496:  *!
497:  *!        Function: BUSC_CLV
498:  *!
499:  *!       Called by: CREAR                (procedure in
AACT_SST.SPR)
500:  *!                : BUSCAR               (procedure in
AACT_SST.SPR)
501:  *!                : OPC_DES()            (function in
DA_ASISG.PRG)
502:  *!                : INGRESO              (procedure in
ACT_PTMO.SPR)
503:  *!
504:  *!           Calls: RUBROS               (procedure in
DA_ASISG.PRG)
505:  *!                : AYU_DPTO             (procedure in
MO_PDLAB.SPR)
506:  *!                : BUSC_GRUPO()         (function in
MO_PDLAB.SPR)
507:  *!
508:
*!*************************************************************************
**************
509:       FUNCTION busc_clv
510:            ********************
511:          IF NOT WEXIST('BUSCAR') && o definir en el
principal
512:             DEFINE WINDOW buscar FROM 0,0 TO 5,30 TITLE
'BUSCAR' SYSTEM
513:          ENDIF
514:            ACTIVATE WINDOW buscar
515:            MOVE WINDOW buscar CENTER
516:            reg_ant = RECNO()
517:            mcod_per = SPAC(6)
518:            SET ORDER TO vartag(1)
519:            mcod_per = SPAC(6)
520:            STOR 0 TO mmcod_per
521:          DO WHILE .T.
522:             @ 1,1 SAY 'Ingrese el Cdigo: ' GET mmcod_per
```

```
PICT '999'
  523:          ||     READ MODAL
  524:          ||     STOR ALLT(STR(mmcod_per)) TO ccod_per
  525:          ||     STOR ALLT(STR(veiu)) + '00'+
IIF(mmcod_per<10, '00'+ccod_per,:
  526:          ||         IIF(mmcod_per<100, '0'+ccod_per,ccod_per)) TO
mcod_per
  527:          ||   ┌─IF LAST() = 27
  528:          ||   │     GOTO reg_ant
  529:          ||   │     vret = .F.
  530:          ||   v═══════EXIT
  531:          ||   └─ENDIF
  532:          ||     SEEK mcod_per
  533:          ||   ┌─IF !FOUND()
  534:          ||   │     WAIT WINDOW NOWAIT 'Código no Existe'
  535:          ||   │     GOTO reg_ant
  536:          ||   ^═══════LOOP
  537:          ||   ├─ELSE
  538:          ||   │     vret = .T.
  539:          ||   v═══════EXIT
  540:          ||   └─ENDIF
  541:          ||└─ENDDO
  542:          ||   DEACTIVATE WINDOW buscar
  543:          ||   SET ORDER TO mvartag
  544:          ||   DO rubros WITH .F.
  545: <═══════RETURN vret
  546:  *:  EOF: DA_ASISG.act
```


Varios Archivo Nómina Rol de Pagos ReporTes Pmo.Emp Costos E.D.36 Salir
pampdr
REGISTRAR DATOS DE DESCUENTOS
PARA PROCESAR ROLES DE :
PASTAZA
JUBILADOS
PALORA
JUB. PASTAZA
PRIMERA QUINCENA
SEGUNDA QUINCENA
AÑO. . . . : 1996
MES. . . . : 12 DICIEMBRE
ACEPTAR
EDITAR
SALIR
10:06


Varios Archivo Nómina Rol de Pagos Reportes Ptmo.Emp Costos E.D.36 Salir
INGRESOS ROL AMBATO PRIMERA QUINCENA
Tratamt. Datos
Filtros
A.Ret.Juzgado
B.Tri.Menores
C.Descto.Pmar
D.Ptmo.Empres
E.Coop.Ahorro
F.Alm.Laboral
G.D/Desc.Sind
H.Club Andin.
I.Desc. AIT
J.O.D.A.A.E.
K.Otros Descr
L.Cta 141
10:07
Terminar
Codigo: 100015
Nombre: MALDONADO V. ROSARIO
Cargo: CONTADOR AYUDANTE 2
Departamento: RECURSOS HUMANOS
Ciudad: AMBATO
Orden :
Fecha : 12/05/1996
Ceduka: 1000105544
RETENCION JUZGADO...
TRIBUNAL MENORES...
DESCTO.DE ENERGIA... 11.000
PRESTAMO EMPRESA.... 150.000
COOP.AHORROS.....
ALMACEN LABORAL 139.600
O.DECTOS.SINDICATO.. 2.000
CLUB DE ANDINISMO...
DESCTO.A.T.I.T.....
O/D.ASOC.EMPLEADOS..
OTROS DESCUENTOS ...
Total Otr.Descuentos: 2.000
VALOR PARA SUBSIDIO.
CTA.ADONP5011 11.400
CUENTA 143.....
PRIMERO SIGUIENTE ANTERIOR ULTIMO EDITAR NEGR.MENU
BUSCAR

```
    1:
*:**********************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\DE_EGRQ1.PRG
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 05/02/1995 at 20:26:26
    8:  *:
    9:  *:  Procs & Fncts: VAL_MED()
   10:  *:
   11:  *:          Set by: ACEPTAR              (procedure in
VREG_FTE.SPR)
   12:  *:
   13:  *:           Calls: TEXTO                (procedure in
GRAFICOS.PRG)
   14:  *:                : TRAT_DATOS()         (function in
DA_ASISG.PRG)
   15:  *:                : OFC_DES()            (function in
DA_ASISG.PRG)
   16:  *:                : SALIR                (procedure in
CTRL_RES.SPR)
   17:  *:
   18:  *:         Indexes: CLABOT_E1                (tag in
TB_FTVAR.CDX)
   19:  *:
   20:  *:       Documented 19:15:20
FoxPro version 3.00a
   21:
*:**********************************************************************
**************
   22:       ************************************************
   23:       ** REGISTRO DE DESCUENTOS PARA LA PRIMERA QUINCENA*
   24:       ************************************************
   25:       PARAMETER veiu,msis,vtip,tip,PROC
   26:       *----------*
   27:       *VARIABLES
   28:       *----------*
   29:       REGIONAL
modo,salto,m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14,m15,:
   30:
m16,m17,m18,m19,meont,fijo,var,vartag,mvartag,trat_dat,ope_bot,mod
o,ope_des,fijo,:
   31:          vfijo, fila
   32:       DIMENSION vartag(20)
   33:       STOR msis TO ciudad
   34:       STOR ' FONT  "N",10 style "b" TO fsdesc && styl and
font of say comment
   35:       STOR '  ' TO fscampc && styl and font of say CHARACTER
fields
   36:       STOR '  ' TO fscampn && styl and font of say NUMERIC
fields
   37:       STOR 0 TO fijo
   38:       STOR 1 TO trat_dat
   39:       STOR 'enab' TO salto
```

```
 40:      STORE SPACE(1) TO
m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14,m15,m16,m17,m18,mc
ont,fi,i_var,m19,:
 41:         modo,opc_bot,opc_des,mvartag
 42:      fil_ciu = 1
 43:      STOR .T. TO laso
 44:      STOR .F. TO vfijo
 45:      STOR 13 TO fila
 46:
 47:      *~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
 48:      *BASES DE DATOS (APERTURA)
 49:      *~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
 50:    IF USED("tbpv")
 51:       SELECT tbpv
 52:       SET ORDER TO TAG "PD_EGRQ1"
 53:    ELSE
 54:       SELECT 0
 55:       USE (LOCFILE("dbfs\tb_ppvar.dbf","DBF","Where is
tb_ppvar?"));
 56:            AGAIN ALIAS tbpv :
 57:            ORDER TAG "PD_EGRQ1"
 58:    ENDIF
 59:
 60:    IF USED("sec")
 61:       SELECT sec
 62:       SET ORDER TO TAG "secc_cod"
 63:    ELSE
 64:       SELECT 0
 65:       USE (LOCFILE("dbfs\seccion.dbf","DBF","Where is
Seccion?"));
 66:            AGAIN ALIAS sec :
 67:            ORDER TAG "secc_cod"
 68:    ENDIF
 69:
 70:    IF USED("dep")
 71:       SELECT dep
 72:       SET ORDER TO TAG "cod_per"
 73:    ELSE
 74:       SELECT 0
 75:       USE (LOCFILE("dbfs\personal.dbf","DBF","Where is
personal?"));
 76:            AGAIN ALIAS dep :
 77:            ORDER TAG "cod_per"
 78:    ENDIF
 79:    IF USED("lee")
 80:       SELECT lee
 81:       SET ORDER TO TAG 0
 82:    ELSE
 83:       SELECT 0
 84:       USE (LOCFILE("dbfs\ley_cont.dbf","DBF","Where is
ley_cont?"));
 85:            AGAIN ALIAS lee :
 86:            ORDER TAG 0
 87:    ENDIF
 88:      SELE lee
 89:      SET FILTER TO anio = manio AND mes = vmes
 90:    IF USED("let")
```

```
        SELECT lct
        SET ORDER TO TAG 0
    ELSE
        SELECT 0
        USE (LOCFILE("dbfs\ley_labo.dbf","DBF","Where is ley_labo?"));
            AGAIN ALIAS lct ;
            ORDER TAG 0
    ENDIF
    SELE lct
    SET FILTER TO anio = manio AND mes = vmes

    SELE dgp
    SET FILTER TO num_ciu = vciu
    SET ORDER TO cod_per
    mvartag = 'cod_per'
    SELECT tbpv
    INDEX ON clave+boton TAG clabot_e1 FOR clave = 'EQ1' UNIQUE

    SET ORDER TO clabot_e1
    COUNT TO A
    DIME arr_bot(A)
    STOR '@*V ' TO mboton
    SCAN
       STOR mboton + ALLT(tbpv.boton)+';' TO mboton
    ENDSCAN
    SET ORDER TO pd_egrq1
    COUNT TO A
    DIME aux_arrbot(A,5)
    COPY FIELD campo,detalle,boton,cuando,formato TO ARRAY aux_arrbot
    STOR SUBS(mboton,1,LEN(mboton)-1) TO mboton
    STOR 0 TO opc_bot
    *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
    *RELACIONES ENTRE BASES DATOS
    *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
    SELE dgr
    SET RELATION TO num_sec INTO sec ADDITIVE
    SET RELATION TO num_ciu INTO sat ADDITIVE
    SET RELATION TO cod_per INTO men ADDITIVE
    SELE dgr
    FOR i = 1 TO 20
       IF EMPTY(TAG(i))
          i = i - 1
          DIMENSION vartag(i)
          EXIT
       ELSE
          vartag(i) = TAG(i)
       ENDIF
    ENDFOR
    SET ORDER TO vartag(1)
    GOTO TOP
    *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
    *      DEFINICION DE VENTANAS         *
    *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
    DEFINE WINDOW esta FROM 0,0 TO 50,90 PANEL TITLE "
```

```
EGRESOS ROL " + msis + "   " + tip NOZOOM
        DEFINE WINDOW pregunta FROM 10.58 TO 26. 76 DOUBLE
SHADOW IN WIND esta FLOAT ZOOM
        ACTIVATE WINDOW esta
        *~~~~~~~~~~~~~~~~~~~~~~~~*
        * PROGRAMA PRINCIPAL
        *~~~~~~~~~~~~~~~~~~~~~~~~*
        DO texto
        DO WHILE laso
           @ 0,1 TO 44,20 DOUBLE
           @ 0,1 FILL TO 44,20 COLOR GR+/B
           @ 1,4 GET m19;
              PICTURE "@*N Tratamt. \<Datos" ;
              SIZE 1.5,22 FONT 'R',10 STYLE "BI";
              VALID indices()

           @3,4 GET trat_dat FUNC '^
\<Rubros;\<General;\<Valor Fijo;\<Totales';
              SIZE 1.20 FONT 'N',8 STYLE "IB";
              ENAB VALID trat_datos()

           @ 6,3.167 GET opc_bot;
              PICTURE mboton;
              SIZE 1.30,22.00,0.20;
              DEFAULT 1;
              FONT 'R',10;
              STYLE "BI";
              WHEN trat_dat = 1 OR trat_dat = 3;
              VALID opc_des()

           @ 40,4 GET m12;
              PICTURE "@*N \<Terminar" ;
              SIZE 1.5,22 FONT 'R',10 STYLE "BI";
              VALID salir()
           READ CYCLE MODAL
        ENDDO
        RETUR
        *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
        *FIN DE PROGRAMA PRINCIPAL
        *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*


        ****************

*!*************************************************************
**************
     *!
     *!      Procedure: SALIR
     *!
     *!      Called by: _RAA1732TG()        (function in
AT_ESCLS.SPR)
     *!                  : DA_ASISG.PRG
     *!                  : DE_EGRQ1.PRG
     *!                  : DE_EGRQ2.PRG
     *!                  : DE_EGRMN.PRG
     *!                  : DE_IEJUB.PRG
     *!
```

```
  196:  *!          Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
  197:  *!
  198:
*!*************************************************************************
**************
  199:      FUNCTION salir
  200:          ****************
  201:          SELE dgp
  202:          SET RELATION OFF INTO sec
  203:          SET RELATION OFF INTO sst
  204:          SET RELATION OFF INTO men
  205:          laso = .F.
  206:          RELEASE WINDOW esta
  207: <-------RETURN
  208:
  209:          ****************
  210:
*!*************************************************************************
**************
  211:  *!
  212:  *!        Function: INDICES
  213:  *!
  214:
*!*************************************************************************
**************
  215:      FUNCTION indices
  216:          ****************
  217:          @6.7.67 GET mvartag FROM vartag PICTURE "@^" SIZE
1,12
  218:          READ
  219:          SET ORDER TO mvartag
  220: <-------RETURN
  221:
  222:          ****************
  223:
*!*************************************************************************
**************
  224:  *!
  225:  *!        Function: TRAT_DATOS
  226:  *!
  227:  *!       Called by: DA_ASISG.PRG
  228:  *!                : DE_EGR01.PRG
  229:  *!                : DE_EGR02.PRG
  230:  *!                : DE_EGRMN.PRG
  231:  *!                : DE_IEJUE.PRG
  232:  *!
  233:  *!          Calls: CPC_DES()          (function in
DA_ASISG.PRG)
  234:  *!                 : TOTALES          (procedure in
VE_IMPETA.SPR)
  235:  *!                 : RUBROS           (procedure in
DA_ASISG.PRG)
  236:  *!
  237:
*!*************************************************************************
**************
```

```
238:      FUNCTION trat_datos
239:          ********************
240:          STOR .F. TO vfijo
241:          DO CASE
242:          CASE trat_dat = 1 && 'Rubros'
243:               *                    =OPC_DES()
244:          CASE trat_dat = 2 && 'General'
245:               =opc_des()
246:          CASE trat_dat = 3 && 'valor fijo'
247:               *                    =OPC_DES()
248:               STOR .T. TO vfijo
249:               STOR 0 TO fijo
250:               DEFI WIND fijo FROM 10,10 TO 15,50
251:               ACTI WIND fijo
252:               @1,1 SAY 'Ingrese el Valor: '
253:               @1,20 GET fijo PICT '999,999,999'
254:               READ
255:               RELE WIND fijo
256:          CASE trat_dat = 4 && 'Totales'
257:               reg_ant = RECNO()
258:               DO totales
259:               GOTO reg_ant
260:               DO rubros WITH .F.
261:          ENDCASE
262:          STOR 1 TO trat_dat
263:          SHOW GET trat_dat
264:          RETURN
265:
266:          ******************
267:
*!*************************************************************************
**************
268:  *!
269:  *!       Procedure: TOTALES
270:  *!
271:  *!       Called by: SECC()              (function in
PERSONAL.PRG)
272:  *!                : TRAT_DATOS()        (function in
IVA_ASISG.PRG)
273:  *!
274:  *!            Calls: SIST_TOT           (procedure in
EGR_VARI.PRG)
275:  *!
276:
*!*************************************************************************
**************
277:      PROCEDURE totales
278:          ******************
279:          @ 1.5, 35  SAY SPAC(6) &&VarCod &fscampC
280:          @ 1.5, 60  SAY 'TOTALES GENERALES'
281:          @ 3.5, 40  SAY SPAC(35)
282:          @ 5.5, 40  SAY SPAC(35)
283:          @ 7.5, 40  SAY ALLTRI(sst.ciudad) &fscampc
284:          @ 9.5, 40  SAY DATE() &fscampc
285:          @ 9.5, 67  SAY '          '
286:
287:          SUM ALL &aux_arrbot(01,1)  TO t1  &&Ret_Juz1 TO
```

```
RJ1
  288:          SUM ALL &aux_arrbot(02,1)  TO t2  &&Tri_Men1 To
tm1
  289:          SUM ALL &aux_arrbot(03,1)  TO t3  &&Pag_Ene  To Pe
  290:          SUM ALL &aux_arrbot(04,1)  TO t4  &&Medidor  To Me
  291:          SUM ALL &aux_arrbot(06,1)  TO t6  &&Coop_Ah1 To
Cah1
  292:          SUM ALL &aux_arrbot(07,1)  TO t7  &&Pre_Emp1 To
Pe1
  293:          SUM ALL &aux_arrbot(08,1)  TO t8  &&Alm_Lab1 To
A11
  294:          SUM ALL &aux_arrbot(09,1)  TO t9  &&desc_sind1 To
ds1
  295:          SUM ALL &aux_arrbot(10,1)  TO t10 &&club_and11 To
Ca1
  296:          SUM ALL &aux_arrbot(11,1)  TO t11 &&Atit1      To
At1
  297:          SUM ALL &aux_arrbot(12,1)  TO t12 &&aso_emple1 To
Ae1
  298:          SUM ALL &aux_arrbot(13,1)  TO t13 &&VOtrosDes1 To
vod1
  299:          SUM ALL &aux_arrbot(14,1)  TO t14 &&VOtrosDes1 To
vod1
  300:
  301:          @ fila+2, 46 SAY t1  &fscampn PICT '999,999,999'
FUNC '@z' &&RJ1
  302:          @ fila+4, 46 SAY t2  &fscampn PICT '999,999,999'
FUNC '@z' &&TM1
  303:          @ fila+6, 46 SAY t3  &fscampn PICT '999,999,999'
FUNC '@z' &&PE
  304:          @ fila+6, 68 SAY 'xxxxx'
  305:          @ fila+6, 75 SAY t4  &fscampn PICT '999,999,999'
FUNC '@z' &&Me
  306:          @ fila+8, 46 SAY t6  &fscampn PICT '999,999,999'
FUNC '@z' &&CAh1
  307:          @ fila+10,46 SAY t7  &fscampn PICT '999,999,999'
FUNC '@z' &&PE1
  308:          @ fila+12,46 SAY t8  &fscampn PICT '999,999,999'
FUNC '@z' &&A11
  309:          @ fila+14,46 SAY t9  &fscampn PICT '999,999,999'
FUNC '@z' &&ds1
  310:          @ fila+16,46 SAY t10 &fscampn PICT '999,999,999'
FUNC '@z' &&At1
  311:          @ fila+18,46 SAY t11 &fscampn PICT '999,999,999'
FUNC '@z' &&CA1
  312:          @ fila+20,46 SAY t12 &fscampn PICT '999,999,999'
FUNC '@z' &&AE1
  313:          @ fila+22,46 SAY t13 &fscampn PICT '999,999,999'
FUNC '@z' &&VOM1
  314:          @ fila+24.5,46 SAY (t9 + t10 + t11 + t12 + t13)
PICT '999,999,999';
  315:            FUNC '@z' &&vOD1+ca1+ds1+ae1+at1 &fscampN
pict '999,999,999'
  316:          @ fila+11, 75 SAY t14  &fscampn PICT '999,999,999'
FUNC '@z' &&cta143
  317:
  318:          DEFI WIND CONT FROM 41,40 TO 45,58 PANEL
```

```
  319:          ACTI WIND CONT
  320:          @ 0.5,1 GET mcont;
  321:             PICTURE "@* \<Aceptar" ;
  322:             SIZE 1.2,20 FONT 'R',10 STYLE "BI"
  323:          READ
  324:          RELE WIND CONT
  325:          *   @3,4 get trat_dat func ''
\<Rubros:\<General:\<Totales' ;
  326:          SIZE 1,20 FONT 'N',8 STYLE "IB" DISABLE
  327: <=======RETURN
  328:
  329:
  330:          ****************
  331:
*!*************************************************************************
**************
  332:  *!
  333:  *!       Procedure: RUBROS
  334:  *!
  335:  *!       Called by: BUSC_CLV()          (function in
DGE_CARG.SPR)
  336:  *!                : TRAT_DATOS()        (function in
DA_ASISG.PRG)
  337:  *!                : DESPLAZAR()         (function in
DA_ASISG.PRG)
  338:  *!
  339:  *!           Calls: VAL_MED()           (function in
DE_EGRG1.PRG)
  340:  *!
  341:
*!*************************************************************************
**************
  342:      PROCEDURE rubros
  343:          ****************
  344:          PARAMETER vmodo
  345:          SELE men
  346:          IF !EOF()
  347:             STOR .F. TO encontro
  348:             @ 1.5,  35 SAY dep.cod_per &&VarCod &fscampC
  349:             @ 1.5,60 SAY SUBS(dep.nombre ,1,25)
  350:             @ 3.5,35 SAY dep.cargo
  351:             @ 5.5,40 SAY sec.nombre &fscampe
  352:             @ 7.5,40 SAY ALLTRI(ast.ciudad) &fscampe
  353:             @ 9.5,40 SAY DATE() &fscampe
  354:             @ 9.5,67 SAY dep.cedula
  355:             IF vfijo
  356:                FOR i = 1 TO A
  357:                   IF &aux_arrbot(i,4)
  358:                      REPLACE &aux_arrbot(i,1) WITH fijo
  359:                      STOR .T. TO vmodo
  360:                   ENDIF
  361:                ENDFOR
  362:             ENDIF
  363:             @ fila+2,46 GET &aux_arrbot(01,1) WHEN
&aux_arrbot(01,4);
  364:                    PICT &aux_arrbot(01,5) FUNC '@z' &&fscampN
piet '999,999,999'
```

```
   365:
   366:          @ fila+4,46 GET &aux_arrbot(02,1) WHEN
&aux_arrbot(02,4);
   367:                PICT &aux_arrbot(02,5) FUNC '@z' &&Tri_Men1
fscampN pict '999,999,999'
   368:
   369:          @ fila+6,46 GET &aux_arrbot(03,1) WHEN
&aux_arrbot(03,4) VALID val_med();
   370:                PICT &aux_arrbot(03,5) FUNC '@z' &&Pag_Ene
fscampN pict '999'
   371:
   372:          @ fila+6,68 SAY &aux_arrbot(05,1) &&Dgp.medidor
   373:
   374:          @ fila+6,75 GET &aux_arrbot(04,1) WHEN
&aux_arrbot(04,4);
   375:                PICT &aux_arrbot(04,5) FUNC '@z' &&Medidor
&fscampN pict '999,999,999'
   376:
   377:          @ fila+8,46 GET &aux_arrbot(06,1) WHEN
&aux_arrbot(06,4);
   378:                PICT &aux_arrbot(06,5) FUNC '@z' &&Pre_Emp1
&fscampN pict '999,999,999'
   379:
   380:          @ fila+10,46 GET &aux_arrbot(07,1) WHEN
&aux_arrbot(07,4);
   381:                PICT &aux_arrbot(07,5) FUNC '@z' &&Coop_Ah1
&fscampN pict '999,999,999'
   382:
   383:          @ fila+12,46 GET &aux_arrbot(08,1) WHEN
&aux_arrbot(08,4);
   384:                PICT &aux_arrbot(08,5) FUNC '@z' &&Alm_Lab1
&fscampN pict '999,999,999'
   385:
   386:          @ fila+14,46 GET &aux_arrbot(09,1) WHEN
&aux_arrbot(09,4);
   387:                PICT &aux_arrbot(09,5) FUNC '@z'
&&dese_ind1 &fscampN pict '999,999,999'
   388:
   389:          @ fila+16,46 GET &aux_arrbot(10,1) WHEN
&aux_arrbot(10,4);
   390:                PICT &aux_arrbot(10,5) FUNC '@z'
&&club_and11 &fscampN pict '999,999,999'
   391:
   392:          @ fila+18,46 GET &aux_arrbot(11,1) WHEN
&aux_arrbot(11,4);
   393:                PICT &aux_arrbot(11,5) FUNC '@z' &&atit1
&fscampN pict '999,999,999'
   394:
   395:          @ fila+20,46 GET &aux_arrbot(12,1) WHEN
&aux_arrbot(12,4);
   396:                PICT &aux_arrbot(12,5) FUNC '@z'
&&aso_emple1 &fscampN pict '999,999,999'
   397:
   398:          @ fila+22,46 GET &aux_arrbot(13,1) WHEN
&aux_arrbot(13,4);
   399:                PICT &aux_arrbot(13,5) FUNC '@z'
&&vOtrosDes1 &fscampN pict '999,999,999'
```

```
  400:
  401:          @ fila+11,75 GET &aux_arrbot(14,1) WHEN
&aux_arrbot(14,4);
  402:              PICT &aux_arrbot(14,5)
  403:          IF vmodo
  404:              READ MODAL
  405:              FLUSH
  406:          ENDIF
  407:          @ fila+24.5,46 SAY &aux_arrbot(9,1) +
&aux_arrbot(10,1) + &aux_arrbot(11,1)+;
  408:              &aux_arrbot(12,1)+ &aux_arrbot(13,1) PICT
'999,999,999';
  409:              FUNC '@z'
&&vOtrosDes1+club_andi1+desc_sind1+atit1+aso_emple1 &fscampN
pict '999,999,999'
  410:          ENDIF
  411:          SELE dgp
  412: <=======RETURN
  413:
  414:          ****************
  415:
*!*************************************************************
**************
  416:  *!
  417:  *!        Function: OPC_DES
  418:  *!
  419:  *!      Called by: DA_ASISG.PRG
  420:  *!               : DE_EGRQ1.PRG
  421:  *!               : DE_EGRQ2.PRG
  422:  *!               : DE_EGRMN.PRG
  423:  *!               : DE_IEJUB.PRG
  424:  *!               : TRAT_DATOS()        (function in
DA_ASISG.PRG)
  425:  *!
  426:  *!          Calls: DESPLAZAR()         (function in
DA_ASISG.PRG)
  427:  *!               : BUSC_CLV()          (function in
DGP_CARG.SPR)
  428:  *!
  429:
*!*************************************************************
**************
  430:      FUNCTION opc_des
  431:          ****************
  432:          @40,25 TO 45,55,65
  433:          STOR .T. TO vopc_des
  434:          STOR .F. TO vmodo
  435:          DO WHILE vopc_des
  436:              @ 41,5,26 GET opc_des ;
  437:                  PICTURE '@*H
\<PRIMERO;\<SIGUIENTE;\<ANTERIOR;\<ULTIMO;\<EDITAR;\<REGR.MENU':
  438:                  SIZE 1.2,7,0.25;
  439:                  DEFAULT 1;
  440:                  FONT "MS Sans Serif", 5;
  441:                  STYLE "BT";
  442:                  VALID desplazar(opc_des)
  443:              @ 44,45 GET opc_busc ;
```

```
444:                PICTURE '@*H \<BUSCAR';
445:                SIZE 1.2,7.0.25;
446:                DEFAULT 1;
447:                FONT "MS Sans Serif", 5;
448:                STYLE "BT";
449:                VALID busc_clv()
450:             READ CYCLE MODAL
451:          ENDDO
452:          @40.25 TO 46.85 CLEAR
453:       RETURN
454:
455:          ****************
456:
*!*************************************************************************
**************
457:    *!
458:    *!        Function: DESPLAZAR
459:    *!
460:    *!      Called by: OPC_DES()          (function in
DA_ASISG.PRG)
461:    *!
462:    *!          Calls: RUBROS             (procedure in
DA_ASISG.PRG)
463:    *!
464:
*!*************************************************************************
**************
465:       FUNCTION desplazar
466:          ****************
467:          PARAMETER modo
468:          SET ORDER TO mvartag
469:          STOR .F. TO vmodo
470:          DO CASE
471:             CASE modo = 'PRIMERO'
472:                GOTO TOP
473:                WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
474:             CASE modo = 'SIGUIENTE'
475:                IF NOT EOF()
476:                   SKIP
477:                ENDIF
478:                IF EOF()
479:                   GO BOTTOM
480:                   WAIT WIND NOWAIT 'FIN DE ARCHIVO'
481:                ENDIF
482:             CASE modo = 'ANTERIOR'
483:                IF NOT BOF()
484:                   SKIP -1
485:                ENDIF
486:                IF BOF()
487:                   GO TOP
488:                   WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
489:                ENDIF
490:             CASE modo = 'ULTIMO'
491:                GOTO BOTTOM
492:                WAIT WIND NOWAIT 'FIN DE ARCHIVO'
493:             CASE modo = 'EDITAR' AND NOT PROC
494:                STOR .T. TO vmodo
```

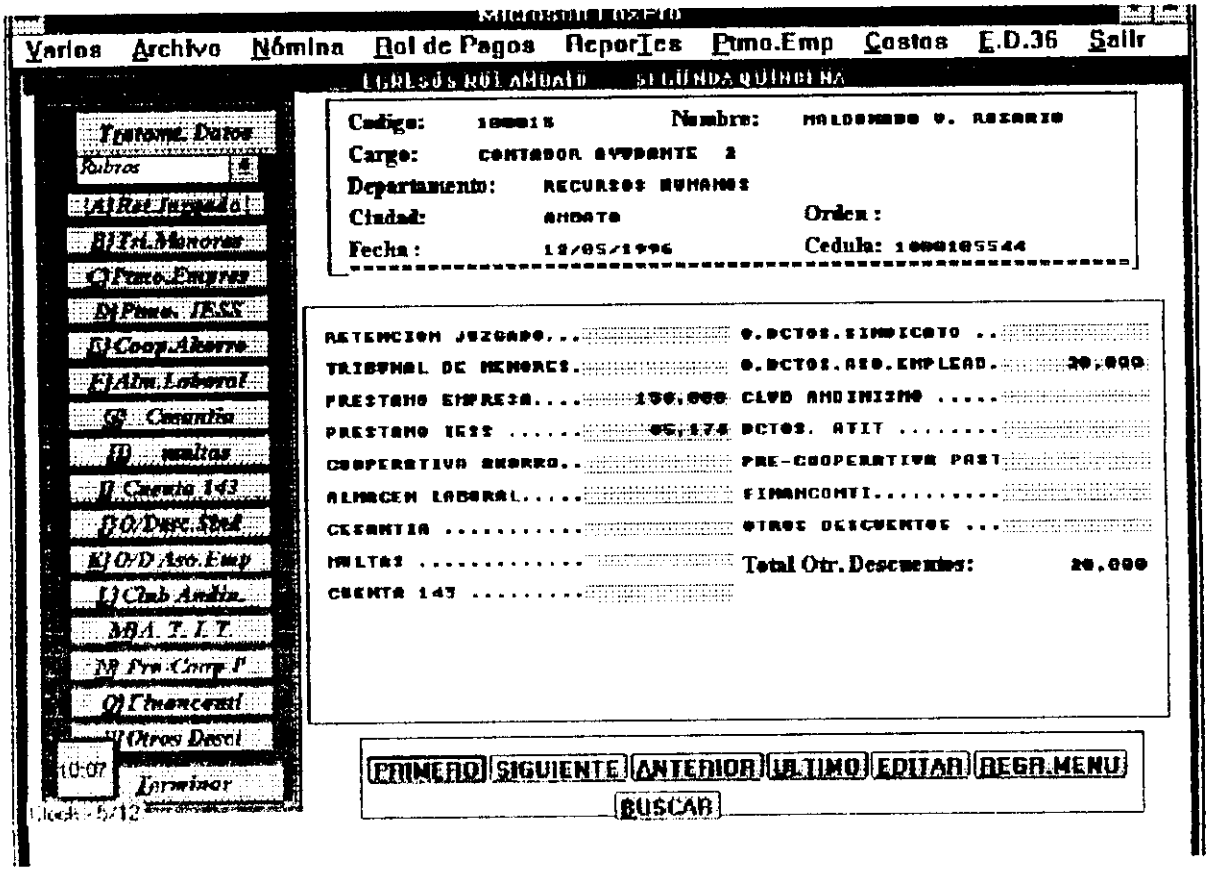
```
  495:          ╠═══CASE modo  = 'CANC.EDITAR'
  496:          ║       SHOW GET ope_des,5 PROMPT '\<EDITAR' ENABLE
  497:          ║       SHOW GET ope_des,6 ENABLE
  498:          ╠═══CASE modo = 'REGR.MENU'
  499:          ║       STOR .F. TO vope_des
  500:          ╚═ENDCASE
  501:            DO rubros WITH vmodo
  502:            SET ORDER TO mvartag
  503:  <═══════RETURN
  504:
  505:            ********************
  506:
*!*************************************************************************
**************
  507:  *!
  508:  *!        Function: BUSC_CLV
  509:  *!
  510:  *!      Called by: CREAR              (procedure in
AACT_SST.SPR)
  511:  *!                : BUSCAR            (procedure in
AACT_SST.SPR)
  512:  *!                : OPC_DES()         (function in
DA_ASISG.PRG)
  513:  *!                : INGRESO           (procedure in
ACT_PIMO.SPR)
  514:  *!
  515:  *!           Calls: RUBROS            (procedure in
DA_ASISG.PRG)
  516:  *!                : AYU_DFTO          (procedure in
MO_PDLAB.SPR)
  517:  *!                : BUSC_GRUPO()      (function in
MO_PDLAB.SPR)
  518:  *!
  519:
*!*************************************************************************
**************
  520:      FUNCTION busc_clv
  521:          ********************
  522:        ┌─IF NOT WEXIST('BUSCAR') && o definir en el
principal
  523:        │     DEFINE WINDOW buscar FROM 0,0 TO 5,30 TITLE
'BUSCAR' SYSTEM
  524:        └─ENDIF
  525:          ACTIVATE WINDOW buscar
  526:          MOVE WINDOW buscar CENTER
  527:          reg_ant = RECNO()
  528:          SET ORDER TO vartag(1)
  529:          mcod_per = SPAC(6)
  530:          STOR 0 TO mmcod_per
  531:        ┌─DO WHILE .T.
  532:        │     @ 1,1 SAY 'Ingrese el Código: ' GET mmcod_per
PICT '999'
  533:        │     READ MODAL
  534:        │     STOR ALLT(STR(mmcod_per)) TO ccod_per
  535:        │     STOR ALLT(STR(veiu)) +'00'+
IIF(mmcod_per<10,'00'+ccod_per,:
  536:        │          IIF(mmcod_per<100,'0'+ccod_per,ccod_per)) TO
```

```
mcod_per
  537:          IF LAST() = 27
  538:              GOTO reg_ant
  539:              vret = .F.
  540:          EXIT
  541:          ENDIF
  542:          SEEK mcod_per
  543:          IF !FOUND()
  544:              WAIT WINDOW NOWAIT 'Cśdigo no Existe'
  545:              GOTO reg_ant
  546:          LOOP
  547:          ELSE
  548:              vret = .T.
  549:          EXIT
  550:          ENDIF
  551:      ENDDO
  552:      DEACTIVATE WINDOW buscar
  553:      SET ORDER TO mvartag
  554:      DO rubros WITH .F.
  555:  RETURN vret
  556:
  557:      ***************
  558:
*!*************************************************************************
**************
  559:  *!
  560:  *!       Procedure: TEXTO
  561:  *!
  562:  *!       Called by: DA_ASISG.PRG
  563:  *!                : DE_EGRG1.PRG
  564:  *!                : DE_EGRG2.PRG
  565:  *!                : DE_EGRMN.PRG
  566:  *!                : DE_IEJUB.PRG
  567:  *!
  568:
*!*************************************************************************
**************
  569:  PROCEDURE texto
  570:      ***************
  571:      @0.5,23 TO 11,65
  572:      @ 1,25    SAY "Codigo:"&fsdesc
  573:      @ 1,50    SAY "Nombre:"&fsdesc
  574:      @ 3,25    SAY "Cargo:"&fsdesc
  575:      @ 5,25    SAY "Departamento:"&fsdesc
  576:      @ 7,25    SAY "Ciudad:"&fsdesc
  577:      @ 7,60    SAY "Orden :"&fsdesc
  578:      @ 9,25    SAY "Fecha :"&fsdesc
  579:      @ 9,60    SAY "Cedula:"&fsdesc
  580:      @ 10,5,25 SAY REPLICATE("-",60)
  581:
  582:      @fila,21,5 TO fila + 26,5 ,37
  583:      @ fila+2,23 SAY aux_arrbot(1,2) && "Retencion
Juzgado:"&fsdesc
  584:      @ fila+4,23 SAY aux_arrbot(2,2) &&"Tribunal
Menores:"&fsdesc
  585:      @ fila+6,23 SAY aux_arrbot(3,2) &&"Fago
Energia:"&fsdesc
```

```
  586:          @ fila+5,60 SAY aux_arrbot(4,2)
&&"Descto.Energia:"&fsdesc
  587:          @ fila+6,60 SAY aux_arrbot(5,2) &&"No."
  588:          @ fila+8,23 SAY aux_arrbot(6,2) &&"Cooperativa
ahorros:"&fsdesc
  589:          @ fila+10,23 SAY aux_arrbot(7,2) &&"Prestamo
Empresa:"&fsdesc
  590:          @ fila+12,23 SAY aux_arrbot(8,2) &&"Almacen
Laboral:"&fsdesc
  591:          @ fila+14,23 SAY aux_arrbot(9,2)
&&"Sindicato:"&fsdesc
  592:          @ fila+16,23 SAY aux_arrbot(10,2) &&"Club
Andinismo:"&fsdesc
  593:          @ fila+18,23 SAY aux_arrbot(11,2) &&"A.T.I.T.
:"&fsdesc
  594:          @ fila+20,23 SAY aux_arrbot(12,2) &&"Asociacion
Emple:"&fsdesc
  595:          @ fila+22,23 SAY aux_arrbot(13,2) &&"Otros
Descuentos:" &fsdesc
  596:          @ fila+24,23 SAY "Total Otr. Descuentos:" &fsdesc
  597:          @ fila+11 ,60 SAY aux_arrbot(14,2) &&"cta 143"
  598: <=======RETURN
  599:          ****************
  600:
*!*************************************************************
**************
  601:  *!
  602:  *!        Function: VAL_MED
  603:  *!
  604:  *!      Called by: RUBROS              (procedure in
DA_ASISG.PRG)
  605:  *!
  606:
*!*************************************************************
**************
  607:      FUNCTION val_med
  608:          ****************
  609:          REPLACE &aux_arrbot(04,1) WITH ;
  610:              IIF(!EMPTY(dsp.medidor),&aux_arrbot(03,1),0)
  611:          IF dep.zona = 1
  612:            DO CASE
  613:            ======CASE (men.medidor > lec.cc_suben)  AND
(men.medidor    lec.cc_subemur)
  614:                    IF (men.medidor > lec.cc_suben)
  615:                        vsubener = men.medidor
  616:                    ELSE
  617:                        vsubener = lec.cc_suben
  618:                    ENDIF
  619:            ======CASE (men.medidor > lec.cc_subemur)
  620:                        vsubener = lec.cc_subemur
  621:            ======OTHERWISE
  622:                        vsubener = men.medidor
  623:            ====ENDCASE
  624:              REPLACE men.medidor WITH vsubener
  625:          ELSE
  626:            IF (men.medidor>lec.cc_suben)
  627:                vsubener = lec.cc_suben
```

```
628:            ELSE
629:                vsubener = men.medidor
630:            ENDIF
631:            REPLACE men.medidor WITH vsubener
632:        ENDIF
633:    RETURN .T.
634:  *:  EOF: DE_EGRQ1.act
```


Varios Archivo Nómina Bol de Pagos ReporTes Ptmo.Emp Costos E.D.36 Salir
REGISTRAR DATOS DE DESCUENTOS
PARA PROCESAR ROLES DE :
OPERATIVO
PASTAZA
JUBILADOS
PALORA
JUB. PASTAZA
PRIMERA QUINCENA
SEGUNDA QUINCENA
AÑO. . . . : 1996
MES. . . . : 12
ACEPTAR
EDITAR
SALIR
10:07


Varios Archivo Nómina Rol de Pagos ReporTes Pmo.Emp Costos E.D.36 Salir
Codigo: 100015 Nombre: MALDONADO V. ROSARIO
Cargo: CONTADOR AYUDANTE 2
Departamento: RECURSOS HUMANOS
Ciudad: AMBATO Orden:
Fecha: 12/05/1996 Cedula: 1804185544
RETENCION JUZGADO...
TRIBUNAL DE MENORES.
PRESTAMO EMPRESA.... 150,000
PRESTAMO IESS 85,174
COOPERATIVA AHORRO..
ALMACEN LABORAL.....
CESANTIA .....
MULTAS .....
CUENTA 143 .....
0.DCTOS.SINDICATO ..
0.DCTOS.ASO.EMPLEAD. 20,000
CLUB ANDINISMO .....
DCTOS. ATIT .....
PRE-COOPERATIVA PAST
FINANCOMTI.....
OTROS DESCUENTOS ...
Total Otr. Descuentos: 20.000
PRIMERO SIGUIENTE ANTERIOR ULTIMO EDITAR REGR.MENU
BUSCAR

```
    1:
*:*************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\DE_EGR02.PRG
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 05/02/1995 at 20:27:24
    8:  *:
    9:  *:          Set by: ACEPTAR              (procedure in
VREG_PTE.SPR)
   10:  *:
   11:  *:           Calls: TEXTO                (procedure in
GRAFICOS.PRG)
   12:  *:                : TRAT_DATOS()         (function in
DA_ASISG.PRG)
   13:  *:                : OPC_DES()            (function in
DA_ASISG.PRG)
   14:  *:                : SALIR                (procedure in
CTRL_RES.SPR)
   15:  *:
   16:  *:         Indexes: CLABOT_E2                (tag in
TE_PPVAR.CDX)
   17:  *:
   18:  *:      Documented 19:15:21
FoxDoc version 3.00a
   19:
*:*************************************************************
**************
   20:      ************************************************
   21:      ** REGISTRO DE DESCUENTOS PARA LA SEGUNDA QUINCENA*
   22:      ************************************************
   23:
   24:      PARAMETER veiu,msis,vtip,tip,PROC
   25:      *~~~~~~~~~~~~~~~~
   26:      *VARIABLES
   27:      *~~~~~~~~~~~~~~~~
   28:      REGIONAL
mode,salto,m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14,m15,;
   29:
m16,m17,m18,m19,mcont,fijo,var,vartag,mvartag,trat_dat,opc_bot,mod
o,opc_des,fijo,;
   30:          vfijo, fila
   31:      DIMENSION vartag(20)
   32:      STOR msis TO ciudad
   33:      STOR ' FONT  "N",10 style "b" TO fsdesc && styl and
font of say comment
   34:      STOR '  ' TO fscamc && styl and font of say CHARACTER
fields
   35:      STOR '  ' TO fscampn && styl and font of say NUMERIC
fields
   36:      STOR 0 TO fijo
   37:      STOR 1 TO trat_dat
   38:      STOR 'enab' TO salto
   39:      STORE SPACE(1) TO
```

```
m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14,m15,m16,m17,m18,mc
ont,fij_var,m19,:
   40:          modo,opc_bot,opc_des,mvartag
   41:       fil_ciu = 1
   42:       STOR .T. TO laso
   43:       STOR .F. TO vfijo
   44:       STOR 13 TO fila
   45:
   46:       *~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
   47:       *BASES DE DATOS (APERTURA)
   48:       *~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
   49:       IF USED("tbpv")
   50:          SELECT tbpv
   51:          SET ORDER TO TAG "PD_EGRQ1"
   52:       ELSE
   53:          SELECT 0
   54:          USE (LOCFILE("dbfs\tb_ppvar.dbf","DBF","Where is
tb_ppvar?"));
   55:                AGAIN ALIAS tbpv :
   56:                ORDER TAG "PD_EGRQ1"
   57:       ENDIF
   58:
   59:       IF USED("sec")
   60:          SELECT sec
   61:          SET ORDER TO TAG "secc_cod"
   62:       ELSE
   63:          SELECT 0
   64:          USE (LOCFILE("dbfs\seccion.dbf","DBF","Where is
Seccion?"));
   65:                AGAIN ALIAS sec :
   66:                ORDER TAG "secc_cod"
   67:       ENDIF
   68:
   69:       IF USED("dep")
   70:          SELECT dep
   71:          SET ORDER TO TAG "cod_per"
   72:       ELSE
   73:          SELECT 0
   74:          USE (LOCFILE("dbfs\personal.dbf","DBF","Where is
personal?"));
   75:                AGAIN ALIAS dep :
   76:                ORDER TAG "cod_per"
   77:       ENDIF
   78:       SELE dep
   79:       SET FILTER TO num_ciu = vciu
   80:       SET ORDER TO cod_per
   81:       mvartag = 'cod_per'
   82:
   83:       *Select A
   84:       *    Use Sistema alias sat
   85:       *    set order to sist_cod
   86:
   87:       *Select B
   88:       *    Use tb_ppvar alias tbpv
   89:       *    set order to PD_EGRQ1
   90:
   91:       *Select C
```

```
    *  Use Seccion Alias Secc
    *  set order to Secc_cod

*Select D
*  Use Mensual alias Mens
*  Set filter to ciudad = vciu
*  set order to MENS_COD

*Select E
*  Use Personal alias pers
*  Set filter to num_ciu = vciu
*  set order to Cod_per

SELECT tbpv
INDEX ON clave+boton TAG clabot_e2 FOR clave = 'EQ2' UNIQUE

SET ORDER TO clabot_e2
COUNT TO A
DIME arr_bot(A)
STOR '@*V ' TO mboton
SCAN
    STOR mboton + ALLT(tbpv.boton)+';' TO mboton
ENDSCAN
SET ORDER TO pd_egrq2
COUNT TO A
DIME aux_arrbot(A.5)
COFY FIELD campo.detalle.boton.cuando.formato TO ARRAY aux_arrbot
STOR SUBS(mboton.1.LEN(mboton)-1) TO mboton
STOR 0 TO ope_bot
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
*RELACIONES ENTRE BASES'DATOS
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
SELE dgr
SET RELATION TO num_sec INTO sec ADDITIVE
SET RELATION TO num_ciu INTO sst ADDITIVE
SET RELATION TO cod_per INTO men ADDITIVE
SELE dgr
FOR i = 1 TO 20
  IF EMPTY(TAG(i))
      i = i - 1
      DIMENSION vartag(i)
      EXIT
  ELSE
      vartag(i) = TAG(i)
  ENDIF
ENDFOR
SET ORDER TO vartag(1)
GOTO TOP
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
*    DEFINICION DE VENTANAS    *
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
DEFINE WINDOW esta FROM 0.0 TO 50.90 PANEL TITLE " EGRESOS ROL " + msis + "  " + tip NOZOOM
DEFINE WINDOW pregunta FROM 10.58 TO 26. 76 DOUBLE SHADOW IN WIND esta FLOAT ZOOM
```

```
      ACTIVATE WINDOW esta
      *~~~~~~~~~~~~~~~~~~~~~~~~*
      * PROGRAMA PRINCIPAL
      *~~~~~~~~~~~~~~~~~~~~~~~~*
      DO texto
      DO WHILE laso
         @ 0.1 TO 44.20 DOUBLE
         @ 0.1 FILL TO 44.20 COLOR GR+/B
         @ 1.4 GET m19;
            PICTURE "@*N T\<ratamt. Datos" ;
            SIZE 1.5,22 FONT 'R',10 STYLE "BI";
            VALID indices()

         @3.4 GET trat_dat PICTURE '@^ \<Rubros;\<General;\<Valor Fijo;\<Totales';
            SIZE 1.20 FONT 'N',8 STYLE "IB";
            ENAB VALID trat_datos()

         @ 6,3.167 GET opc_bot;
            PICTURE mboton;
            SIZE 1.22.00,0.20;
            DEFAULT 1;
            FONT 'R',10;
            STYLE "BI";
            WHEN trat_dat = 1 OR trat_dat = 3;
            VALID opc_des()

         @ 42.4 GET m12;
            PICTURE "@*N \<Terminar" ;
            SIZE 1.3,22 FONT 'R',10 STYLE "BI";
            VALID salir()
         READ CYCLE MODAL
      ENDDO
   RETURN
      *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
      *FIN DE PROGRAMA PRINCIPAL
      *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*


      ****************

*!****************************************************************************************

   *!
   *!       Procedure: SALIR
   *!
   *!       Called by: _RAA1732TG()       (function in AT_ESCLS.SPR)
   *!                : DA_ASISG.PRG
   *!                : DE_EGRQ1.PRG
   *!                : DE_EGRQ2.PRG
   *!                : DE_EGRMN.PRG
   *!                : DE_IEJUB.PRG
   *!
   *!           Calls: DEAC KEYS          (procedure in AACT_SST.SPR)
   *!
```

```
 197:
*!*************************************************************
**************
 198:      FUNCTION salir
 199:          ****************
 200:          SELE dgp
 201:          SET RELATION OFF INTO sec
 202:          SET RELATION OFF INTO sat
 203:          SET RELATION OFF INTO men
 204:          laso = .F.
 205:          RELEASE WINDOW esta
 206: <========RETURN
 207:
 208:          ****************
 209:
*!*************************************************************
**************
 210:  *!
 211:  *!        Function: INDICES
 212:  *!
 213:
*!*************************************************************
**************
 214:      FUNCTION indices
 215:          ****************
 216:          @6,7,67 GET mvartag FROM vartag PICTURE "@^" SIZE
1,12
 217:          READ
 218:          SET ORDER TO mvartag
 219: <========RETURN
 220:
 221:          ****************
 222:
*!*************************************************************
**************
 223:  *!
 224:  *!        Function: TRAT_DATOS
 225:  *!
 226:  *!       Called by: PA_ASISG.PRG
 227:  *!                : PE_EGRQ1.PRG
 228:  *!                : PE_EGRQC.PRG
 229:  *!                : PE_EGRMN.PRG
 230:  *!                : PE_IEJUB.PRG
 231:  *!
 232:  *!          Calls: OPC_DES()          (function in
PA_ASISG.PRG)
 233:  *!                 : TOTALES          (procedure in
VF_IMPRTA.SPR)
 234:  *!                 : RUBROS           (procedure in
PA_ASISG.PRG)
 235:  *!
 236:
*!*************************************************************
**************
 237:      FUNCTION trat_datos
 238:          ****************
 239:          STOR .F. TO vfijo
```

```
240:          DO CASE
241:          CASE trat_dat = 1 && 'Rubros'
242:               *                    =OPC_DES()
243:          CASE trat_dat = 2 && 'General'
244:               =opc_des()
245:          CASE trat_dat = 3 && 'valor fijo'
246:               *                    =OPC_DES()
247:               STOR .T. TO vfijo
248:               STOR 0 TO fijo
249:               DEFI WIND fijo FROM 10,10 TO 15,50
250:               ACTI WIND fijo
251:               @1,1 SAY 'Ingrese el Valor: '
252:               @1,20 GET fijo PICT '999,999,999'
253:               READ
254:               RELE WIND fijo
255:          CASE trat_dat = 4 && 'Totales'
256:               reg_ant = RECNO()
257:               DO totales
258:               GOTO reg_ant
259:               DO rubros WITH .F.
260:          ENDCASE
261:          STOR 1 TO trat_dat
262:          SHOW GET trat_dat
263:          RETURN
264:
265:          ******************
266:
*!*************************************************************************
**************
267:  *!
268:  *!       Procedure: TOTALES
269:  *!
270:  *!       Called by: SECC()              (function in
PERSONAL.PRG)
271:  *!                : TRAT_DATOS()        (function in
DA_ASISG.PRG)
272:  *!
273:  *!           Calls: SIST_TOT            (procedure in
EGR_VARI.PRG)
274:  *!
275:
*!*************************************************************************
**************
276:      PROCEDURE totales
277:          ******************
278:          @ 1.5, 35  SAY SPAC(6) &&VarCod &fscampC
279:          @ 1.5, 60  SAY 'TOTALES GENERALES'
280:          @ 3.5, 40  SAY SPAC(35)
281:          @ 5.5, 40  SAY SPAC(35)
282:          @ 7.5, 40  SAY ALLTRI(sst.ciudad) &fscampc
283:          @ 9.5, 40  SAY DATE() &fscampc
284:          @ 9.5, 67  SAY '          '
285:
286:          SUM ALL &aux_arrbot(01,1) TO t1 &&Ret_Juz2 TO RJ2
287:          SUM ALL &aux_arrbot(02,1) TO t2 &&Tri_Men2 To tm2
288:          SUM ALL &aux_arrbot(03,1) TO t3 &&Pre_emp2  To Pe
289:          SUM ALL &aux_arrbot(04,1) TO t4 &&Pre_iess  To Me
```

```
 290:          SUM ALL &aux_arrbot(05,1) TO t5 &&Coop_Ah2 To Cah2
 291:          SUM ALL &aux_arrbot(06,1) TO t6 &&Alm_Lab2 To Al2
 292:          SUM ALL &aux_arrbot(07,1) TO t7 &&Cesantia  To Ces
 293:          SUM ALL &aux_arrbot(08,1) TO t8 &&Multas    To Mul
 294:          SUM ALL &aux_arrbot(09,1) TO t9 &&cta_143   To
ct143
 295:
 296:          SUM ALL &aux_arrbot(10,1) TO t10 &&desc_sind2 To
ds2
 297:          SUM ALL &aux_arrbot(11,1) TO t11 &&club_andi2 To
Ca2
 298:          SUM ALL &aux_arrbot(12,1) TO t12 &&aso_emple2 To
OAe2
 299:          SUM ALL &aux_arrbot(13,1) TO t13 &&Atit        To
At
 300:          SUM ALL &aux_arrbot(14,1) TO t14 &&Pre_coop    To
PC
 301:          SUM ALL &aux_arrbot(15,1) TO t15 &&Financonti  to
fin
 302:          SUM ALL &aux_arrbot(16,1) TO t16 &&VOtrosDes2 To
vod2
 303:
 304:          @ fila+2,  43 SAY t1  &fscampn PICT '999,999,999'
FUNC '@z' &&RJ2
 305:          @ fila+4,  43 SAY t2  &fscampn PICT '999,999,999'
FUNC '@z' &&TM2
 306:          @ fila+6,  43 SAY t3  &fscampn PICT '999,999,999'
FUNC '@z' &&FE
 307:          @ fila+8,  43 SAY t4  &fscampn PICT '999,999,999'
FUNC '@z' &&CAh2
 308:          @ fila+10, 43 SAY t5  &fscampn PICT '999,999,999'
FUNC '@z'
 309:          @ fila+12, 43 SAY t6  &fscampn PICT '999,999,999'
FUNC '@z'
 310:          @ fila+14, 43 SAY t7  &fscampn PICT '999,999,999'
FUNC '@z'
 311:          @ fila+16, 43 SAY t8  &fscampn PICT '999,999,999'
FUNC '@z'
 312:          @ fila+18, 43 SAY t9  &fscampn PICT '999,999,999'
FUNC '@z'
 313:
 314:          @ fila+2,  75 SAY t10  &fscampn PICT '999,999,999'
FUNC '@z'
 315:          @ fila+4,  75 SAY t11  &fscampn PICT '999,999,999'
FUNC '@z'
 316:          @ fila+6,  75 SAY t12  &fscampn PICT '999,999,999'
FUNC '@z'
 317:          @ fila+8,  75 SAY t13  &fscampn PICT '999,999,999'
FUNC '@z'
 318:          @ fila+10, 75 SAY t14  &fscampn PICT '999,999,999'
FUNC '@z'
 319:          @ fila+12, 75 SAY t15  &fscampn PICT '999,999,999'
FUNC '@z'
 320:          @ fila+14, 75 SAY t16  &fscampn PICT '999,999,999'
FUNC '@z'
 321:          @ fila+16.5,75 SAY (t10 + t11 + t12 + t13 + t14 +
t15 + t16 ) PICT '999,999,999';
```

```
  322:              FUNC '@z' &&vOD1+ca1+ds1+ae1+at1 &fscampN
pict '999,999,999'
  323:
  324:           DEFI WIND CONT FROM 41,40 TO 45,58 PANEL
  325:           ACTI WIND CONT
  326:           @ 0,5,1 GET mcont;
  327:              PICTURE "@* \<Aceptar" ;
  328:              SIZE 1,2,20 FONT 'R',10 STYLE "BI"
  329:           READ
  330:           RELE WIND CONT
  331: <=======RETURN
  332:
  333:
  334:           ****************
  335:
*!*************************************************************************
**************
  336:  *!
  337:  *!       Procedure: RUBROS
  338:  *!
  339:  *!       Called by: BUSC_CLV()          (function in
DGP_CARG.SPR)
  340:  *!                : TRAT_DATOS()        (function in
DA_ASISG.PRG)
  341:  *!                : DESPLAZAR()         (function in
DA_ASISG.PRG)
  342:  *!
  343:  *!          Calls: VAL_MED()           (function in
DE_EGRQ1.PRG)
  344:  *!
  345:
*!*************************************************************************
**************
  346:      PROCEDURE rubros
  347:          ***************
  348:          PARAMETER vmodo
  349:          SELE men
  350:        IF !EOF()
  351:          |  STOR .F. TO encontro
  352:          |  @ 1,5,  35 SAY dgp.cod_per &&VarCod &fscampC
  353:          |  @ 1,5,60 SAY SUBS(dgp.nombre ,1,25)
  354:          |  @ 3,5,35 SAY dgp.cargo
  355:          |  @ 5,5,40 SAY sec.nombre &fscampc
  356:          |  @ 7,5,40 SAY ALLTRI(sst.ciudad) &fscampc
  357:          |  @ 9,5,40 SAY DATE() &fscampc
  358:          |  @ 9,5,67 SAY dgp.cedula
  359:          |  IF vfijo
  360:          |  |  FOR i = 1 TO A
  361:          |  |  |  IF &aux_arrbot(i,4)
  362:          |  |  |  |  REPLACE &aux_arrbot(i,1) WITH fijo
  363:          |  |  |  |  STOR .T. TO vmodo
  364:          |  |  |  ENDIF
  365:          |  |  ENDFOR
  366:          |  ENDIF
  367:          |  @ fila+2,43 GET &aux_arrbot(01,1) WHEN
&aux_arrbot(01,4);
  368:          |        PICT &aux_arrbot(01,5) FUNC '@z' &&fscampN
```

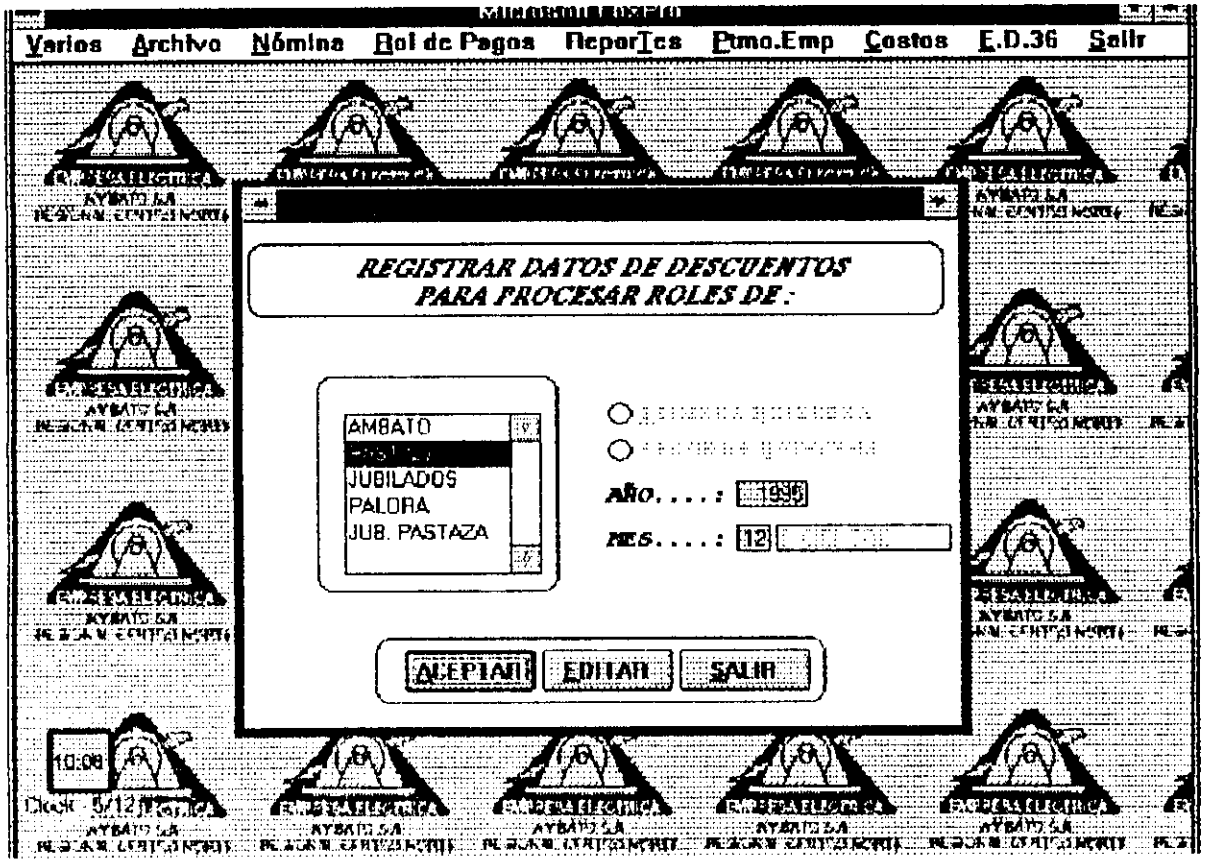
```
pict '999,999,999'
  369:
  370:          @ fila+4,43 GET &aux_arrbot(02,1) WHEN
&aux_arrbot(02,4):
  371:                PICT &aux_arrbot(02,5) FUNC '@z' &&Tri_Men1
fscampN pict '999,999,999'
  372:
  373:          @ fila+6,43 GET &aux_arrbot(03,1) WHEN
&aux_arrbot(03,4):
  374:                PICT &aux_arrbot(03,5) FUNC '@z' &&Pre_Emp2
&fscampN pict '999,999,999'
  375:
  376:
  377:          @ fila+8,43 GET &aux_arrbot(04,1) WHEN
&aux_arrbot(04,4):
  378:                PICT &aux_arrbot(04,5) FUNC '@z' &&Pre_less
&fscampN pict '999,999,999'
  379:
  380:          @ fila+10,43 GET &aux_arrbot(05,1) WHEN
&aux_arrbot(05,4):
  381:                PICT &aux_arrbot(05,5)  FUNC '@z' &&Coop_Ah2
&fscampN pict '999,999,999'
  382:
  383:          @ fila+12,43 GET &aux_arrbot(06,1) WHEN
&aux_arrbot(06,4):
  384:                PICT &aux_arrbot(06,5) FUNC '@z' &&Alm_Lab2
&fscampN pict '999,999,999'
  385:
  386:          @ fila+14,43 GET &aux_arrbot(07,1) WHEN
&aux_arrbot(07,4):
  387:                PICT &aux_arrbot(07,5) FUNC '@z' &&Cesantia
&fscampN pict '999,999,999'
  388:
  389:          @ fila+16,43 GET &aux_arrbot(08,1) WHEN
&aux_arrbot(08,4):
  390:                PICT &aux_arrbot(08,5) FUNC '@z' && Multas
&fscampN pict '999,999,999'
  391:
  392:          @ fila+18,43 GET &aux_arrbot(09,1) WHEN
&aux_arrbot(09,4):
  393:                PICT &aux_arrbot(09,5) FUNC '@z' &&CTA_143
&fscampN pict '999,999,999'
  394:
  395:          @ fila+2,75 GET &aux_arrbot(10,1) WHEN
&aux_arrbot(10,4):
  396:                PICT &aux_arrbot(10,5) FUNC '@z'
&&desc_sind2 &fscampN pict '999,999,999'
  397:
  398:          @ fila+4,75 GET &aux_arrbot(11,1) WHEN
&aux_arrbot(11,4):
  399:                PICT &aux_arrbot(11,5) FUNC '@z'
&&aso_emple2 &fscampN pict '999,999,999'
  400:
  401:          @ fila+6 ,75 GET &aux_arrbot(12,1) WHEN
&aux_arrbot(12,4):
  402:                PICT &aux_arrbot(12,5) FUNC '@z'
&&club_andi2 &fscampN pict '999,999,999'
```

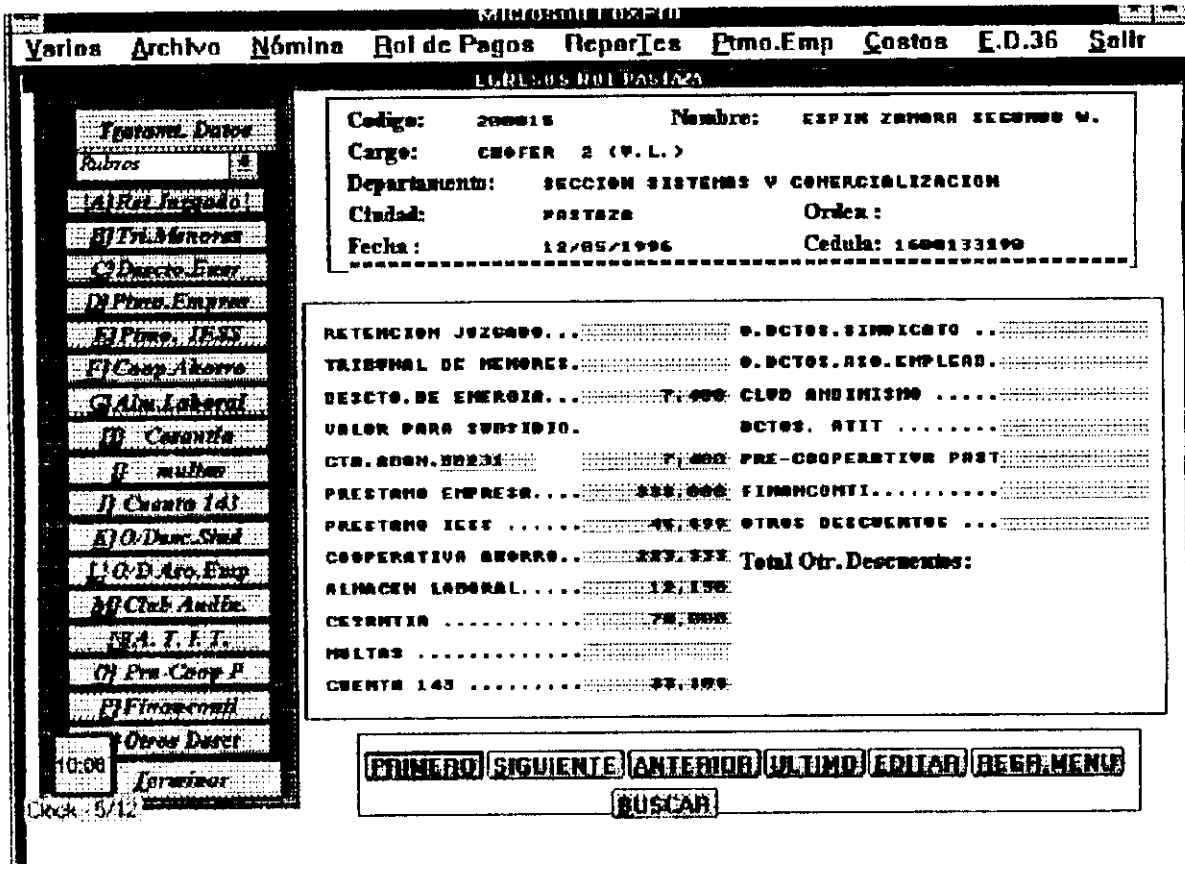
```
DGP_CARG.SPR)
  443:  *!
  444:
*!*************************************************************************
**************
  445:      FUNCTION opc_des
  446:          ****************
  447:          @40,25 TO 45,5,85
  448:          STOR .T. TO vopc_des
  449:          STOR .F. TO vmodo
  450:        ==DO WHILE vopc_des
  451:            @ 41,5,26 GET opc_des ;
  452:               PICTURE '@*H
\<PRIMERO;\<SIGUIENTE;\<ANTERIOR;\<ULTIMO;\<EDITAR;\<REGR.MENU':
  453:               SIZE 1,2,7,0,25;
  454:               DEFAULT 1;
  455:               FONT "MS Sans Serif", 5;
  456:               STYLE "BT";
  457:               VALID desplazar(opc_des)
  458:            @ 44,45 GET opc_busc ;
  459:               PICTURE '@*H \<BUSCAR';
  460:               SIZE 1,2,7,0,25;
  461:               DEFAULT 1;
  462:               FONT "MS Sans Serif", 5;
  463:               STYLE "BT";
  464:               VALID busc_clv()
  465:            READ CYCLE MODAL
  466:        ==ENDDO
  467:          @40,25 TO 46,85 CLEAR
  468: <========RETURN
  469:
  470:          ****************
  471:
*!*************************************************************************
**************
  472:  *!
  473:  *!        Function: DESPLAZAR
  474:  *!
  475:  *!      Called by: OPC_DES()          (function in
DA_ASISG.PRG)
  476:  *!
  477:  *!          Calls: RUBROS             (procedure in
DA_ASISG.PRG)
  478:  *!
  479:
*!*************************************************************************
**************
  480:      FUNCTION desplazar
  481:          ****************
  482:          PARAMETER modo
  483:          SET ORDER TO mvartag
  484:          STOR .F. TO vmodo
  485:        ==DO CASE
  486:        ====CASE modo = 'PRIMERO'
  487:               GOTO TOP
  488:               WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
  489:        ====CASE modo = 'SIGUIENTE'
```

```
490:            IF NOT EOF()
491:               SKIP
492:            ENDIF
493:            IF EOF()
494:               GO BOTTOM
495:               WAIT WIND NOWAIT 'FIN DE ARCHIVO'
496:            ENDIF
497:         CASE modo = 'ANTERIOR'
498:            IF NOT BOF()
499:               SKIP -1
500:            ENDIF
501:            IF BOF()
502:               GO TOP
503:               WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
504:            ENDIF
505:         CASE modo = 'ULTIMO'
506:            GOTO BOTTOM
507:            WAIT WIND NOWAIT 'FIN DE ARCHIVO'
508:         CASE modo = 'EDITAR' AND NOT PROC
509:            STOR .T. TO vmodo
510:         CASE modo  = 'CANC.EDITAR'
511:            SHOW GET opc_des,5 PROMPT '\<EDITAR' ENABLE
512:            SHOW GET opc_des,6 ENABLE
513:         CASE modo = 'REGR.MENU'
514:            STOR .F. TO vopc_des
515:         ENDCASE
516:         SET ORDER TO mvartag
517:         IF modo # 'REGR.MENU'
518:            DO rubros WITH vmodo
519:         ENDIF
520: <-------RETURN
521:
522:         ********************
523:
*!*************************************************************
**************
524:  *!
525:  *!        Function: BUSC_CLV
526:  *!
527:  *!      Called by: CREAR              (procedure in
AACT_SST.SPR)
528:  *!              : BUSCAR              (procedure in
AACT_SST.SPR)
529:  *!              : CDC_DES()           (function in
DA_ASISG.PRG)
530:  *!              : INGRESO             (procedure in
ACT_PTMO.SPR)
531:  *!
532:  *!         Calls: RUBROS              (procedure in
DA_ASISG.PRG)
533:  *!              : AYU_PPTO            (procedure in
MO_PDLAB.SPR)
534:  *!              : BUSC_GRUPO()        (function in
MO_PDLAB.SPR)
535:  *!
536:
*!*************************************************************
```

```
**************
  537:      FUNCTION busc_clv
  538:          ******************
  539:        ┌─IF NOT WEXIST("BUSCAR") && o definir en el
principal
  540:        │     DEFINE WINDOW buscar FROM 0,0 TO 5,30 TITLE
"BUSCAR" SYSTEM
  541:        └─ENDIF
  542:          ACTIVATE WINDOW buscar
  543:          MOVE WINDOW buscar CENTER
  544:          reg_ant = RECNO()
  545:          SET ORDER TO vartag(1)
  546:          mcod_per = SPAC(6)
  547:          STOR 0 TO mmcod_per
  548:        ┌─DO WHILE .T.
  549:        │     @ 1,1 SAY "Ingrese el Cσdigo: " GET mmcod_per
PICT "999"
  550:        │     READ MODAL
  551:        │     STOR ALLT(STR(mmcod_per)) TO ccod_per
  552:        │     STOR ALLT(STR(vciu)) +"00"+
IIF(mmcod_per<10."00"+ccod_per,:
  553:        │         IIF(mmcod_per<100,"0"+ccod_per,ccod_per)) TO
mcod_per
  554:        │   ┌─IF LAST() = 27
  555:        │   │     GOTO reg_ant
  556:        │   │     vret = .F.
  557:        │   ╞═══EXIT
  558:        │   └─ENDIF
  559:        │     SEEK mcod_per
  560:        │   ┌─IF !FOUND()
  561:        │   │     WAIT WINDOW NOWAIT "Cσdigo no Existe"
  562:        │   │     GOTO reg_ant
  563:        │   ╞═══LOOP
  564:        │   ├─ELSE
  565:        │   │     vret = .T.
  566:        │   ╞═══EXIT
  567:        │   └─ENDIF
  568:        └─ENDDO
  569:          DEACTIVATE WINDOW buscar
  570:          SET ORDER TO mvartag
  571:          DO rubros WITH .F.
  572:  <═══════RETURN vret
  573:
  574:
  575:          ***************
  576:
*!*************************************************************
**************
  577:  *!
  578:  *!       Procedure: TEXTO
  579:  *!
  580:  *!       Called by: DA_ASISG.PRG
  581:  *!                : DE_EGRQ1.PRG
  582:  *!                : DE_EGRQ2.PRG
  583:  *!                : DE_EGRMN.PRG
  584:  *!                : DE_IEJUB.PRG
  585:  *!
```

```
 586:
*!*************************************************************************
**************
 587:      PROCEDURE texto
 588:           ***************
 589:           @0.5,23 TO 11,85
 590:           @ 1,25     SAY "Codigo:"&fsdesc
 591:           @ 1,50     SAY "Nombre:"&fsdesc
 592:           @ 3,25     SAY "Cargo:"&fsdesc
 593:           @ 5,25     SAY "Departamento:"&fsdesc
 594:           @ 7,25     SAY "Ciudad:"&fsdesc
 595:           @ 7,60     SAY "Orden :"&fsdesc
 596:           @ 9,25     SAY "Fecha :"&fsdesc
 597:           @ 9,60     SAY "Cedula:"&fsdesc
 598:           @ 10.5,25 SAY REPLICATE("=",60)
 599:
 600:           @fila,21.5 TO fila + 26.5 ,87
 601:
 602:           @ fila+2,23  SAY aux_arrbot(1,2) &&"Retencion
Juzgado:"&fsdesc
 603:           @ fila+4,23  SAY aux_arrbot(2,2) &&"Tribunal
Menores:"&fsdesc
 604:           @ fila+6,23  SAY aux_arrbot(3,2) &&"Prestamo
Empresa:"&fsdesc
 605:           @ fila+8,23  SAY aux_arrbot(4,2) &&"Prestamo
Iess:"&fsdesc
 606:           @ fila+10,23 SAY aux_arrbot(5,2) &&"Cooperativa
ahorros:"&fsdesc
 607:           @ fila+12,23 SAY aux_arrbot(6,2) &&"Almacen
Laboral:"&fsdesc
 608:           @ fila+14,23 SAY aux_arrbot(7,2)
&&"Cesantia:"&fsdesc
 609:           @ fila+16,23 SAY aux_arrbot(8,2)
&&"Multas:"&fsdesc
 610:           @ fila+18,23 SAY aux_arrbot(9,2) &&"Cta
143:"&fsdesc
 611:
 612:           @ fila+2, 55 SAY aux_arrbot(10,2)
&&"Sindicato:"&fsdesc
 613:           @ fila+4, 55 SAY aux_arrbot(11,2) &&"Asociacion
Emple:"&fsdesc
 614:           @ fila+6, 55 SAY aux_arrbot(12,2) &&"Club
Andinismo"&fsdesc
 615:           @ fila+8, 55 SAY aux_arrbot(13,2) &&"A.T.I.T.
:"&fsdesc
 616:           @ fila+10,55 SAY aux_arrbot(14,2)
&&"Pre-Cooperativa :"&fsdesc
 617:           @ fila+12,55 SAY aux_arrbot(15,2) &&"Financonti
:"&fsdesc
 618:           @ fila+14,55 SAY aux_arrbot(16,2) &&"Otros
Descuentos:" &fsdesc
 619:
 620:           @ fila+16,55 SAY "Total Otr. Descuentos:" &fsdesc
 621: <========RETURN
 622:
 623:  *:  EOF: DE_EGRQ2.act
```


Varios Archivo Nómina Rol de Pagos ReporTes Ptmo.Emp Costos E.D.36 Salir
REGISTRAR DATOS DE DESCUENTOS
PARA PROCESAR ROLES DE :
AMBATO
JUBILADOS
PALORA
JUB. PASTAZA
Año. . . . : 1996
MES. . . . : 12
ACEPTAR
EDITAR
SALIR
10:08


Varios Archivo Nómina Rol de Pagos ReporTes Prmo.Emp Costos E.D.36 Salir
Codigo:
Nombre:
Cargo: CHOFER 2 (V.L.)
Departamento: SECCION SISTEMAS Y COMERCIALIZACION
Ciudad: PASTAZA
Orden:
Fecha: 12/05/1996
Cedula:
RETENCION JUZGADO...
TRIBUNAL DE MENORES.
DESCTO.DE EMERGIA...
VALOR PARA SUBSIDIO.
CTA.AHOR.
PRESTAMO EMPRESA....
PRESTAMO IESS .....
COOPERATIVA AHORRO..
ALMACEN LABORAL.....
CETAMTIA .....
MULTAS .....
CUENTA 143 .....
O.DCTOS.SINDICATO ..
O.DCTOS.ASO.EMPLEAD.
CLUB ANDINISMO .....
DCTOS. ATIT .....
PRE-COOPERATIVA PAST
FINANCOMTI.....
OTROS DESCUENTOS ...
Total Otr.Descuentos:
PRIMERO SIGUIENTE ANTERIOR ULTIMO EDITAR REGR.MENU
BUSCAR

```
    1:
*:*************************************************************
*************
    2:  *:
    3:  *: Procedure file: C:\DOC\DE_EGRMN.PRG
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-05-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 23/10/1995 at  4:35:12
    8:  *:
    9:  *:          Set by: ACEPTAR             (procedure in
VREG_PTE.SPR)
   10:  *:
   11:  *:           Calls: TEXTO               (procedure in
GRAFICOS.PRG)
   12:  *:                : TRAT_DATOS()        (function in
DA_ASISG.PRG)
   13:  *:                : OPC_DES()           (function in
DA_ASISG.PRG)
   14:  *:                : SALIR               (procedure in
CTRL_RES.SPR)
   15:  *:
   16:  *:         Indexes: CLABOT_EM                (tag in
TB_PFVAR.CDX)
   17:  *:
   18:  *:       Documented 19:15:23
FoxDoc version 3.00a
   19:
*:*************************************************************
*************
   20:      ************************************************
   21:      * Programa: Actualización de Egresos e Ingresos *
   22:      *           del personal durante el  mes        *
   23:      ************************************************
   24:      PARAMETERS vciu,msis,vtip,PROC
   25:      *PARAMETER VCIU,MSIS,VTIP,TIP,PROC
   26:      *~~~~~~~~~~*
   27:      *VARIABLES
   28:      *~~~~~~~~~~*
   29:      REGIONAL
modo,salto,m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14,m15,;
   30:
m16,m17,m18,m19,mcont,fij_var,vartag,mvartag,trat_dat,opc_bot,mod
o,opc_des,fijo,;
   31:          vfijo, fila
   32:      DIMENSION vartag(20)
   33:      STOR msis TO ciudad
   34:      STOR ' FONT  "N",10 style "b" ' TO fsdesc && styl and
font of say coment
   35:      STOR '   ' TO fscampc && styl and font of say CHARACTER
fields
   36:      STOR '   ' TO fscampn && styl and font of say NUMERIC
fields
   37:      STOR 0 TO fijo
   38:      STOR 1 TO trat_dat
   39:      STOR 'enab' TO salto
```

```
3    40:      STORE SPACE(1) TO
m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14,m15,m16,m17,m18,mc
ont,fij_var,m19,:
    41:         modo,opc_bot,opc_des,mvartag
    42:       fil_ciu = 1
    43:       STOR .T. TO laso
    44:       STOR .F. TO vfijo
    45:       STOR 13 TO fila
    46:
    47:       *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
    48:       *BASES DE DATOS (APERTURA)
    49:       *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
    50:       *Select A
    51:       *   Use Sistema alias sst
    52:       *   set order to sist_cod
    53:
    54:       *Select B
    55:       *   Use tb_ppvar alias tbpv
    56:       *   set order to FD_EGRQ1
    57:
    58:       *Select C
    59:       *  Use Seccion Alias Secc
    60:       *  set order to Secc_cod
    61:
    62:       *Select D
    63:       *  Use Mensual alias Mens
    64:       *  Set filter to ciudad = vciu
    65:       *  set order to MENS_COD
    66:
    67:       *Select E
    68:       *  Use Personal alias pers
    69:       *  Set filter to num_ciu = vciu
    70:       *  set order to Cod_per
    71:
    72:       IF USED("tbpv")
    73:           SELECT tbpv
    74:           SET ORDER TO TAG "FD_EGRQ1"
    75:       ELSE
    76:           SELECT 0
    77:           USE (LOCFILE("dbfs\tb_ppvar.dbf","DBF","Where is
tb_ppvar?"));
    78:                AGAIN ALIAS tbpv :
    79:                ORDER TAG "FD_EGRQ1"
    80:       ENDIF
    81:
    82:       IF USED("sec")
    83:           SELECT sec
    84:           SET ORDER TO TAG "secc_cod"
    85:       ELSE
    86:           SELECT 0
    87:           USE (LOCFILE("dbfs\seccion.dbf","DBF","Where is
Seccion?"));
    88:                AGAIN ALIAS sec :
    89:                ORDER TAG "secc_cod"
    90:       ENDIF
    91:
    92:       IF USED("dep")
```

```
 93:    |     SELECT dgp
 94:    |    SET ORDER TO TAG "cod_per"
 95:    |-ELSE
 96:    |    SELECT 0
 97:    |    USE (LOCFILE("dbfs\personal.dbf","DBF","Where is
personal?"));
 98:    |        AGAIN ALIAS dgp :
 99:    |        ORDER TAG "cod_per"
100:    |-ENDIF
101:    SELE dgp
102:    SET FILTER TO num_ciu = vciu
103:    SET ORDER TO cod_per
104:    mvartag = 'COD_PER'
105:
106:    |-IF USED("lcc")
107:    |    SELECT lcc
108:    |    SET ORDER TO TAG 0
109:    |-ELSE
110:    |    SELECT 0
111:    |    USE (LOCFILE("dbfs\ley_cont.dbf","DBF","Where is
ley_cont?"));
112:    |        AGAIN ALIAS lcc ;
113:    |        ORDER TAG 0
114:    |-ENDIF
115:    SELE lcc
116:    SET FILTER TO anio = manio AND mes = vmes
117:    |-IF USED("lct")
118:    |    SELECT lct
119:    |    SET ORDER TO TAG 0
120:    |-ELSE
121:    |    SELECT 0
122:    |    USE (LOCFILE("dbfs\ley_labo.dbf","DBF","Where is
ley_labo?"));
123:    |        AGAIN ALIAS lct ;
124:    |        ORDER TAG 0
125:    |-ENDIF
126:    SELE lct
127:    SET FILTER TO anio = manio AND mes = vmes
128:
129:    SELECT tbpv
130:    INDEX ON clave+boton TAG clabot_em FOR clave = 'EMN'
UNIQUE
131:
132:    SET ORDER TO clabot_em
133:    COUNT TO A
134:    DIME arr_bot(A)
135:    STOR '@W ' TO mboton
136:    |-SCAN
137:    |    STOR mboton + ALLT(tbpv.boton)+ ';' TO mboton
138:    |-ENDSCAN
139:    SET ORDER TO pd_egrmn
140:    COUNT TO A
141:    DIME aux_arrbot(A,5)
142:    COPY FIELD campo,detalle,boton,cuando,formato TO ARRAY
aux_arrbot
143:    STOR SUBS(mboton,1,LEN(mboton)-1) TO mboton
144:    STOR 0 TO opc_bot
```

```
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
*RELACIONES ENTRE BASES DATOS
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
SELE dgp
SET RELATION TO num_sec INTO sec ADDITIVE
SET RELATION TO num_ciu INTO sst ADDITIVE
SET RELATION TO cod_per INTO men ADDITIVE
SELE dgp
FOR i = 1 TO 20
  IF EMPTY(TAG(i))
     i = i - 1
     DIMENSION vartag(i)
     EXIT
  ELSE
     vartag(i) = TAG(i)
  ENDIF
ENDFOR
*set order to vartag(1)
SET ORDER TO cod_per
STOR 'COD_PER' TO mvartag
GOTO TOP
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
*    DEFINICION DE VENTANAS    *
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
DEFINE WINDOW esta FROM 0.0 TO 50.90 PANEL TITLE " EGRESOS ROL " + msis  NOZOOM
DEFINE WINDOW pregunta FROM 10.58 TO 26. 76 DOUBLE SHADOW IN WIND esta FLOAT ZOOM
ACTIVATE WINDOW esta
*~~~~~~~~~~~~~~~~~~~~~~~~*
* PROGRAMA PRINCIPAL
*~~~~~~~~~~~~~~~~~~~~~~~~*
DO texto
DO WHILE laso
    @ 0.1 TO 44,20 DOUBLE
    @ 0.1 FILL TO 44,20 COLOR GR+/B
    @ 1.4 GET m19:
        PICTURE "@*N T\<ratamt. Datos" ;
        SIZE 1.5.22 FONT 'R',10 STYLE "BI";
        VALID indices()

    @3.4 GET trat_dat PICTURE '@' \<Rubros:\<General:\<Valor Fijo:\<Totales':
        SIZE 1.20 FONT 'N',8 STYLE "IB";
        ENAB VALID trat_datos()

    @ 6.3.167 GET opc_bot:
        PICTURE mboton:
        SIZE 1,22.00.0.1;
        DEFAULT 1:
        FONT 'R',10:
        STYLE "BI";
        WHEN trat_dat = 1 OR trat_dat = 3:
        VALID opc_des()

    @ 42.4 GET m12:
        PICTURE "@*N \<Terminar" ;
```

```
    ||          SIZE 1.3.22 FONT 'R',10 STYLE "BI";
    ||          VALID salir()
    ||        READ CYCLE MODAL
    └─ENDDO
<===RETUR
       *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
       *FIN DE PROGRAMA PRINCIPAL
       *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*


       ****************

*!****************************************************************************
*!
*!        Procedure: SALIR
*!
*!        Called by: _RAA1732TG()          (function in AT_ESCLS.SPR)
*!                 : DA_ASISG.PRG
*!                 : DE_EGRQ1.PRG
*!                 : DE_EGRQ2.PRG
*!                 : DE_EGRMN.PRG
*!                 : DE_1EJUB.PRG
*!
*!            Calls: DEAC_KEYS             (procedure in AACT_SST.SPR)
*!

*!****************************************************************************
      FUNCTION salir
           ****************
           SELE dsp
           SET RELATION OFF INTO sec
           SET RELATION OFF INTO set
           SET RELATION OFF INTO men
           laso = .F.
           RELEASE WINDOW esta
<==========RETURN

           ****************

*!****************************************************************************
*!
*!        Function: INDICES
*!

*!****************************************************************************
      FUNCTION indices
           ****************
           @6.7.67 GET mvartag FROM vartag PICTURE "@^" SIZE 1.12
           READ
           SET ORDER TO mvartag
<==========RETURN
```

```
246:
247:          ********************
248:
*!*************************************************************************
**************
249:  *!
250:  *!          Function: TRAT_DATOS
251:  *!
252:  *!          Called by: DA_ASISG.PRG
253:  *!                   : DE_EGRQ1.PRG
254:  *!                   : DE_EGRQ2.PRG
255:  *!                   : DE_EGRMN.PRG
256:  *!                   : DE_IEJUB.PRG
257:  *!
258:  *!              Calls: OPC_DES()          (function in
DA_ASISG.PRG)
259:  *!                   : TOTALES            (procedure in
VF_IMPRTA.SPR)
260:  *!                   : RUBROS             (procedure in
DA_ASISG.PRG)
261:  *!
262:
*!*************************************************************************
**************
263:      FUNCTION trat_datos
264:          ********************
265:          STOR .F. TO vfijo
266:          DO CASE
267:          CASE trat_dat = 1 && 'Rubros'
268:               *                    =OPC_DES()
269:          CASE trat_dat = 2 && 'General'
270:              =opc_des()
271:          CASE trat_dat = 3 && 'valor fijo'
272:               *                    =OPC_DES()
273:               STOR .T. TO vfijo
274:               STOR 0 TO fijo
275:               DEFI WIND fijo FROM 10,10 TO 15,50
276:               ACTI WIND fijo
277:               @1,1 SAY 'Ingrese el Valor: '
278:               @1,20 GET fijo PICT '999,999,999'
279:               READ
280:               RELE WIND fijo
281:          CASE trat_dat = 4 && 'Totales'
282:               reg_ant = RECNO()
283:               DO totales
284:               GOTO reg_ant
285:               DO rubros WITH .F.
286:          ENDCASE
287:          STOR 1 TO trat_dat
288:          SHOW GET trat_dat
289:          RETURN
290:
291:          ******************
292:
*!*************************************************************************
**************
293:  *!
294:  *!          Procedure: TOTALES
```

```
  295:  *!
  296:  *!        Called by: SECC()                (function in
PERSONAL.PRG)
  297:  *!                 : TRAT_DATOS()          (function in
DA_ASISG.PRG)
  298:  *!
  299:  *!            Calls: SIST_TOT              (procedure in
EGR_VARI.PRG)
  300:  *!
  301:
*!*************************************************************
**************
  302:      PROCEDURE totales
  303:          ****************
  304:          @ 1.5, 35  SAY SPAC(6) &&VarCod &fscampC
  305:          @ 1.5, 60  SAY 'TOTALES GENERALES'
  306:          @ 3.5, 40  SAY SPAC(35)
  307:          @ 5.5, 40  SAY SPAC(35)
  308:          @ 7.5, 40  SAY ALLTRI(sst.ciudad) &fscampc
  309:          @ 9.5, 40  SAY DATE() &fscampc
  310:          @ 9.5, 67  SAY '
  311:
  312:          SUM ALL &aux_arrbot(01,1) TO t1  &&Ret_Juz2 TO RJ2
  313:          SUM ALL &aux_arrbot(02,1) TO t2  &&Tri_Men2 To tm2
  314:          SUM ALL &aux_arrbot(03,1) TO t3  &&pag_ene  To Pe
  315:          SUM ALL &aux_arrbot(04,1) TO t5  &&medidor valor
para subsidio
  316:          SUM ALL &aux_arrbot(06,1) TO t6  &&Pre_emp2  To Pe
  317:          SUM ALL &aux_arrbot(07,1) TO t7  &&Pre_iess  To Me
  318:          SUM ALL &aux_arrbot(08,1) TO t8  &&Coop_Ah2 To
Cah2
  319:          SUM ALL &aux_arrbot(09,1) TO t9  &&Alm_Lab2 To A12
  320:          SUM ALL &aux_arrbot(10,1) TO t10 &&Cesantia  To
Ces
  321:          SUM ALL &aux_arrbot(11,1) TO t11 &&Multas    To Mul
  322:          SUM ALL &aux_arrbot(12,1) TO t12 &&eta_143   To
et143
  323:
  324:          SUM ALL &aux_arrbot(13,1) TO t13 &&desc_sind2 To
Asc
  325:          SUM ALL &aux_arrbot(14,1) TO t14 &&club_and12 To
Cla
  326:          SUM ALL &aux_arrbot(15,1) TO t15 &&aso_emple2 To
Ase
  327:          SUM ALL &aux_arrbot(16,1) TO t16 &&Atit          To
At
  328:          SUM ALL &aux_arrbot(17,1) TO t17 &&Pre_coop    To
PC
  329:          SUM ALL &aux_arrbot(18,1) TO t18 &&Financonti  to
fin
  330:          SUM ALL &aux_arrbot(19,1) TO t19 &&VOtrosDes2 To
vod2
  331:
  332:          @ fila+2,  43 SAY t1  &fscampn PICT '999,999,999'
FUNC '@Z'
  333:          @ fila+4,  43 SAY t2  &fscampn PICT '999,999,999'
FUNC '@Z'
  334:          @ fila+6,  43 SAY t3  &fscampn PICT '999,999,999'
```

```
FUNC '@z'
  335:          @ fila+10, 43 SAY t5  &fscampn PICT '999,999,999'
FUNC '@z'
  336:          @ fila+12, 43 SAY t6  &fscampn PICT '999,999,999'
FUNC '@z'
  337:          @ fila+14, 43 SAY t7  &fscampn PICT '999,999,999'
FUNC '@z'
  338:          @ fila+16, 43 SAY t8  &fscampn PICT '999,999,999'
FUNC '@z'
  339:          @ fila+18, 43 SAY t9  &fscampn PICT '999,999,999'
FUNC '@z'
  340:          @ fila+20, 43 SAY t10  &fscampn PICT '999,999,999'
FUNC '@z'
  341:          @ fila+22, 43 SAY t11  &fscampn PICT '999,999,999'
FUNC '@z'
  342:          @ fila+24, 43 SAY t12  &fscampn PICT '999,999,999'
FUNC '@z'
  343:
  344:          @ fila+2,  75 SAY t13  &fscampn PICT '999,999,999'
FUNC '@z'
  345:          @ fila+4,  75 SAY t14  &fscampn PICT '999,999,999'
FUNC '@z'
  346:          @ fila+6,  75 SAY t15  &fscampn PICT '999,999,999'
FUNC '@z'
  347:          @ fila+8,  75 SAY t16  &fscampn PICT '999,999,999'
FUNC '@z'
  348:          @ fila+10, 75 SAY t17  &fscampn PICT '999,999,999'
FUNC '@z'
  349:          @ fila+12, 75 SAY t18  &fscampn PICT '999,999,999'
FUNC '@z'
  350:          @ fila+14, 75 SAY t19  &fscampn PICT '999,999,999'
FUNC '@z'
  351:          @ fila+16.5,75 SAY (t13 + t14 + t15 + t16 + t17 +
t18 + t19 );
  352:              PICT '999,999,999' FUNC '@z'
  353:
  354:          DEFI WIND CONT FROM 41,40 TO 45,58 PANEL
  355:          ACTI WIND CONT
  356:          @ 0,5,1 GET mcont;
  357:              PICTURE "@* \<Aceptar" ;
  358:              SIZE 1.2,20 FONT 'R',10 STYLE "BI"
  359:          READ
  360:          RELE WIND CONT
  361:          RETURN
  362:
  363:
  364:          ****************
  365:
*!*************************************************************
**************
  366:  *!
  367:  *!       Procedure: RUBROS
  368:  *!
  369:  *!       Called by: BUSC_CLV()          (function in
DGP_CARG.SPR)
  370:  *!                : TRAT_DATOS()        (function in
DA_ASISG.PRG)
  371:  *!                : DESPLAZAR()         (function in
```

```
DA_ASISG.PRG)
  372:  *!
  373:  *!          Calls: VAL_MED()          (function in
DE_EGRQ1.PRG)
  374:  *!
  375:
*!*************************************************************
**************
  376:      PROCEDURE rubros
  377:          ****************
  378:          PARAMETER vmodo
  379:          SELE men
  380:          IF !EOF()
  381:              STOR .F. TO encontro
  382:              @ 1.5,  35 SAY dgp.cod_per &&VarCod &fscampC
  383:              @ 1.5,60 SAY SUBS(dgp.nombre ,1,25)
  384:              @ 3.5,35 SAY dgp.cargo
  385:              @ 5.5,40 SAY sec.nombre &fscampc
  386:              @ 7.5,40 SAY ALLTRI(sst.ciudad) &fscampc
  387:              @ 9.5,40 SAY DATE() &fscampc
  388:              @ 9.5,67 SAY dgp.cedula
  389:              IF vfijo
  390:                  FOR i = 1 TO A
  391:                      IF &aux_arrbot(i,4)
  392:                          REPLACE &aux_arrbot(i,1) WITH fijo
  393:                          STOR .T. TO vmodo
  394:                      ENDIF
  395:                  ENDFOR
  396:              ENDIF
  397:              @ fila+2,43 GET &aux_arrbot(01,1) WHEN
&aux_arrbot(01,4);
  398:                  PICT &aux_arrbot(01,5) FUNC '@z' &&fscampN
piet '999.999.999'
  399:
  400:              @ fila+4,43 GET &aux_arrbot(02,1) WHEN
&aux_arrbot(02,4);
  401:                  PICT &aux_arrbot(02,5) FUNC '@z' &&Tri_Men1
fscampN piet '999.999.999'
  402:
  403:              @ fila+6,43 GET &aux_arrbot(03,1) WHEN
&aux_arrbot(03,4) VALID val_med();
  404:                  PICT &aux_arrbot(03,5) FUNC '@z' &&Pre_Emp2
&fscampN piet '999.999.999'
  405:
  406:
  407:              @ fila+10,43 GET &aux_arrbot(04,1) WHEN
&aux_arrbot(04,4) ;
  408:                  PICT &aux_arrbot(04,5) FUNC '@z' &&Pre_Iess
&fscampN piet '999.999.999'
  409:
  410:              @ fila+10,32 GET &aux_arrbot(05,1) WHEN
&aux_arrbot(05,4);
  411:                  PICT &aux_arrbot(05,5) FUNC '@z' &&Coop_Ah2
&fscampN piet '999.999.999'
  412:
  413:              @ fila+12,43 GET &aux_arrbot(06,1) WHEN
&aux_arrbot(06,4);
  414:                  PICT &aux_arrbot(06,5) FUNC '@z' &&Alm_Lab2
```

```
&fscampN pict '999,999,999'
  415:          |
  416:          |     @ fila+14,43 GET &aux_arrbot(07,1) WHEN
&aux_arrbot(07,4):
  417:          |          PICT &aux_arrbot(07,5) FUNC '@z' &&Cesantia
&fscampN pict '999,999,999'
  418:          |
  419:          |     @ fila+16,43 GET &aux_arrbot(08,1) WHEN
&aux_arrbot(08,4):
  420:          |          PICT &aux_arrbot(08,5) FUNC '@z' && Multas
&fscampN pict '999,999,999'
  421:          |
  422:          |     @ fila+18,43 GET &aux_arrbot(09,1) WHEN
&aux_arrbot(09,4):
  423:          |          PICT &aux_arrbot(09,5) FUNC '@z' &&CTA_143
&fscampN pict '999,999,999'
  424:          |
  425:          |     @ fila+20,43 GET &aux_arrbot(10,1) WHEN
&aux_arrbot(10,4):
  426:          |          PICT &aux_arrbot(10,5) FUNC '@z'
&&desc_sind2 &fscampN pict '999,999,999'
  427:          |
  428:          |     @ fila+22,43 GET &aux_arrbot(11,1) WHEN
&aux_arrbot(11,4):
  429:          |          PICT &aux_arrbot(11,5) FUNC '@z'
&&aso_emple2 &fscampN pict '999,999,999'
  430:          |
  431:          |     @ fila+24,43 GET &aux_arrbot(12,1) WHEN
&aux_arrbot(12,4):
  432:          |          PICT &aux_arrbot(12,5) FUNC '@z'
&&club_andi2 &fscampN pict '999,999,999'
  433:          |     *#########*
  434:          |
  435:          |     @ fila+02,75 GET &aux_arrbot(13,1) WHEN
&aux_arrbot(13,4):
  436:          |          PICT &aux_arrbot(13,5) FUNC '@z' &&atit
&fscampN pict '999,999,999'
  437:          |
  438:          |     @ fila+04,75 GET &aux_arrbot(14,1) WHEN
&aux_arrbot(14,4):
  439:          |          PICT &aux_arrbot(14,5) FUNC '@z' &&Fre_coop
&fscampN pict '999,999,999'
  440:          |
  441:          |     @ fila+06,75 GET &aux_arrbot(15,1) WHEN
&aux_arrbot(15,4):
  442:          |          PICT &aux_arrbot(15,5) FUNC '@z'
&&Financonti &fscampN pict '999,999,999'
  443:          |
  444:          |     @ fila+08,75 GET &aux_arrbot(16,1) WHEN
&aux_arrbot(16,4):
  445:          |          PICT &aux_arrbot(16,5) FUNC '@z'
&&vOtrosDes2 &fscampN pict '999,999,999'
  446:          |
  447:          |     @ fila+10,75 GET &aux_arrbot(17,1) WHEN
&aux_arrbot(17,4):
  448:          |          PICT &aux_arrbot(17,5) FUNC '@z'
&&vOtrosDes2 &fscampN pict '999,999,999'
  449:          |
```

```
   450:          |     @ fila+12,75 GET &aux_arrbot(18,1) WHEN
&aux_arrbot(18,4);
   451:          |          PICT &aux_arrbot(18,5) FUNC '@z'
&&vOtrosDes2 &fscampN pict '999,999,999'
   452:          |
   453:          |     @ fila+14,75 GET &aux_arrbot(19,1) WHEN
&aux_arrbot(19,4);
   454:          |          PICT &aux_arrbot(19,5) FUNC '@z'
&&vOtrosDes2 &fscampN pict '999,999,999'
   455:          |
   456:          |    IF vmodo
   457:          |       READ MODAL
   458:          |       FLUSH
   459:          |    ENDIF
   460:          |     @ fila+16.5,75 SAY &aux_arrbot(13,1) +
&aux_arrbot(14,1)+;
   461:          |          &aux_arrbot(15,1)+ &aux_arrbot(16,1) +
&aux_arrbot(17,1)+;
   462:          |          &aux_arrbot(18,1) + &aux_arrbot(19,1)  PICT
'999,999,999';
   463:          |          FUNC '@z' &&vOtrosDes2+ atit+ financonti +
dom + pre_coop +;
   464:          |          club_andi2+desc_sind2 + aso_emple2 &fscampn
pict '999,999,999'
   465:          |
   466:          |    ENDIF
   467:            SELE dgp
   468:    <=====RETURN
   469:
   470:          ****************
   471:
*!*************************************************************************
*************
   472:   *!
   473:   *!         Function: OPC_DES
   474:   *!
   475:   *!       Called by: DA_ASISG.PRG
   476:   *!                : DE_EGRQ1.PRG
   477:   *!                : DE_EGRQ2.PRG
   478:   *!                : DE_EGRMN.PRG
   479:   *!                : DE_IEJUB.PRG
   480:   *!                : TRAT_DATOS()        (function in
DA_ASISG.PRG)
   481:   *!
   482:   *!           Calls: DESPLAZAR()         (function in
DA_ASISG.PRG)
   483:   *!                : BUSC_CLV()          (function in
DGP_CARG.SPR)
   484:   *!
   485:
*!*************************************************************************
*************
   486:       FUNCTION opc_des
   487:          ****************
   488:          @40,25 TO 45,5,85
   489:          STOR .T. TO vopc_des
   490:          STOR .F. TO vmodo
   491:        ╔=DO WHILE vopc_des
```

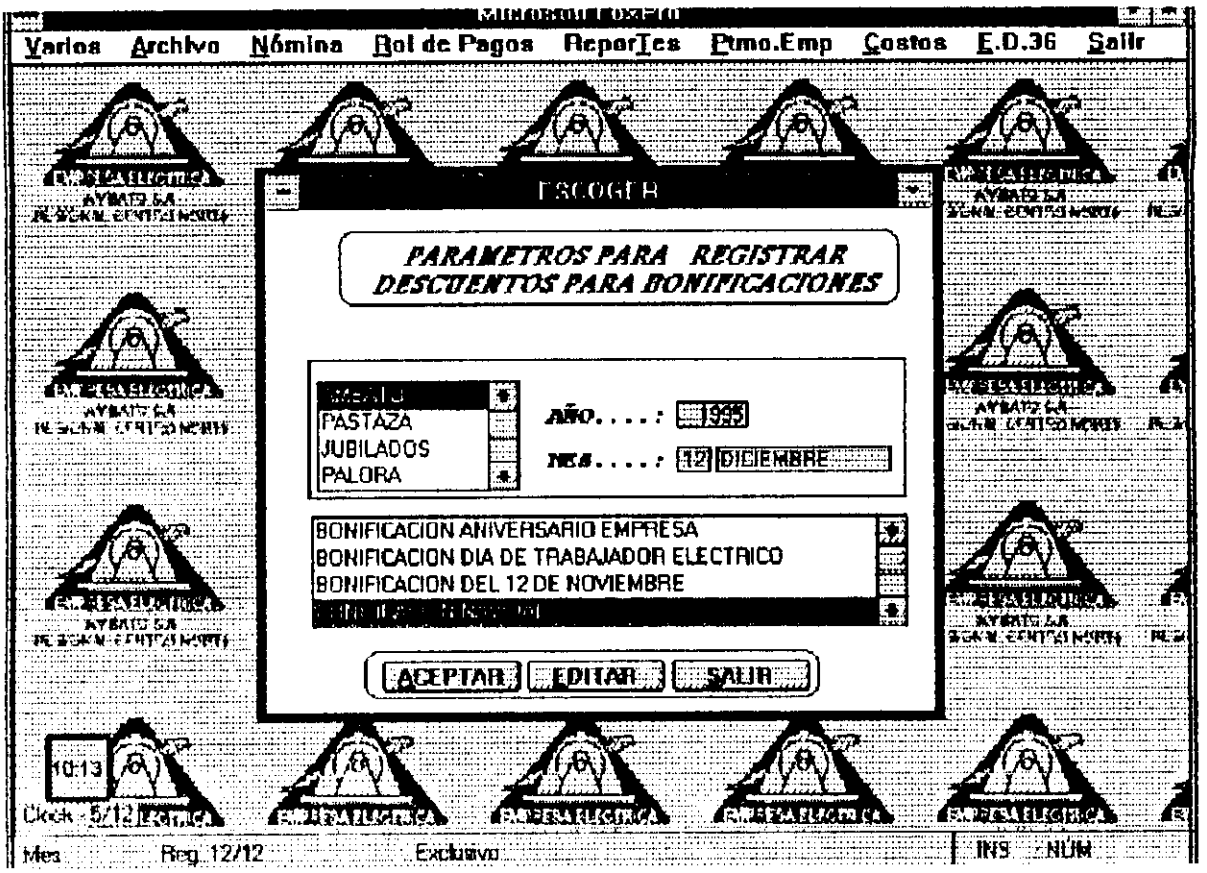
```
492:              @ 41.5,26 GET opc_des :
493:                 PICTURE '@*H
\<PRIMERO:\<SIGUIENTE:\<ANTERIOR:\<ULTIMO:\<EDITAR:\<REGR.MENU':
494:                 SIZE 1.2,7,0.25;
495:                 DEFAULT 1;
496:                 FONT "MS Sans Serif", 5;
497:                 STYLE "BT";
498:                 VALID desplazar(opc_des)
499:              @ 44.45 GET opc_busc :
500:                 PICTURE '@*H \<BUSCAR';
501:                 SIZE 1.2,7,0.25;
502:                 DEFAULT 1;
503:                 FONT "MS Sans Serif", 5;
504:                 STYLE "BT";
505:                 VALID busc_clv()
506:              READ CYCLE MODAL
507:           ENDDO
508:           @40,25 TO 46,85 CLEAR
509:  <=======RETURN
510:
511:           ****************
512:
*!*************************************************************************
**************
513:  *!
514:  *!        Function: DESPLAZAR
515:  *!
516:  *!      Called by: OPC_DES()          (function in
DA_ASISG.PRG)
517:  *!
518:  *!          Calls: RUBROS             (procedure in
DA_ASISG.PRG)
519:  *!
520:
*!*************************************************************************
**************
521:     FUNCTION desplazar
522:           ****************
523:           PARAMETER modo
524:           STOR .F. TO vmodo
525:        DO CASE
526:  ------CASE modo = 'PRIMERO'
527:              GOTO TOP
528:              WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
529:  ------CASE modo = 'SIGUIENTE'
530:              IF NOT EOF()
531:                 SKIP
532:              ENDIF
533:              IF EOF()
534:                 GO BOTTOM
535:                 WAIT WIND NOWAIT 'FIN DE ARCHIVO'
536:              ENDIF
537:  ======CASE modo = 'ANTERIOR'
538:              IF NOT BOF()
539:                 SKIP -1
540:              ENDIF
541:              IF BOF()
542:                 GO TOP
```

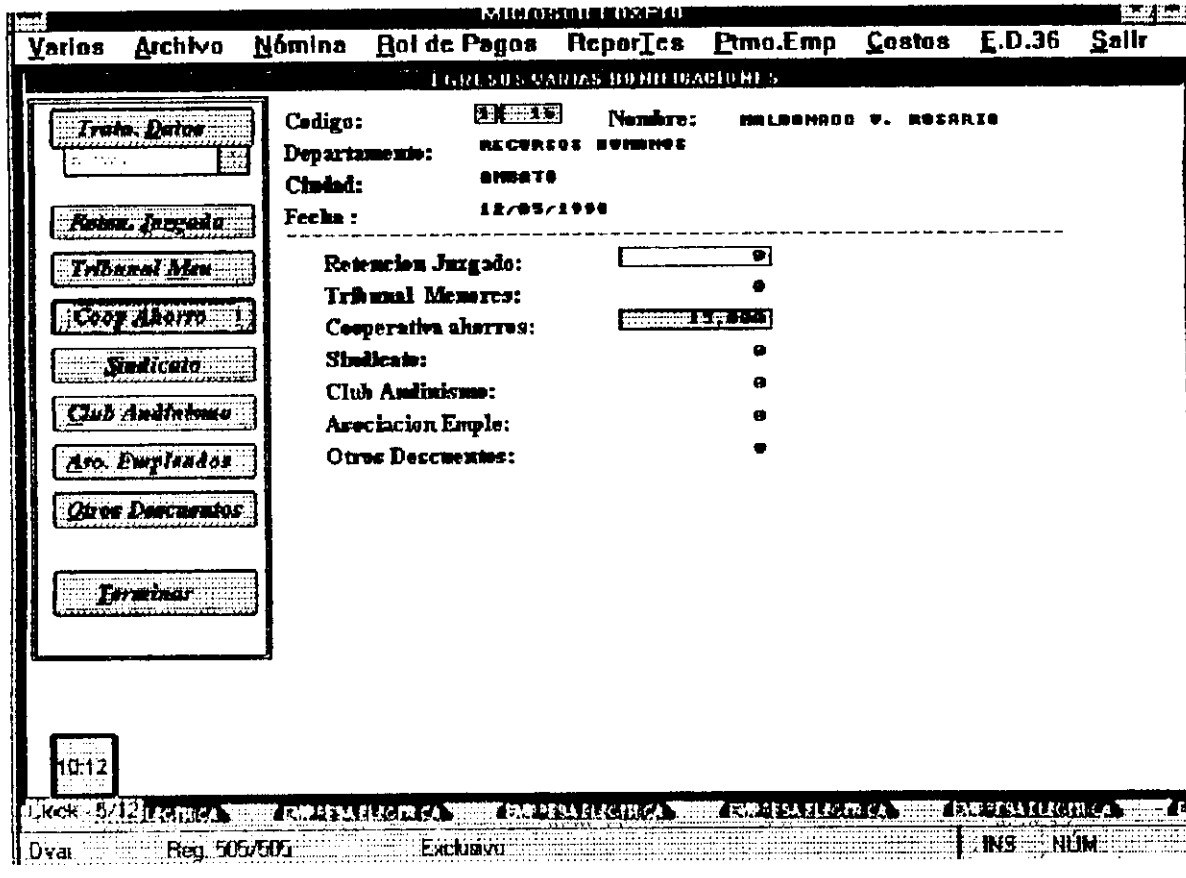
```
543:                WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
544:             ENDIF
545:          CASE modo = 'ULTIMO'
546:             GOTO BOTTOM
547:             WAIT WIND NOWAIT 'FIN DE ARCHIVO'
548:          CASE modo = 'EDITAR' AND NOT PROC
549:             STOR .T. TO vmodo
550:          CASE modo  = 'CANC.EDITAR'
551:             SHOW GET opc_des,5 PROMPT '\<EDITAR' ENABLE
552:             SHOW GET opc_des,6 ENABLE
553:          CASE modo = 'REGR.MENU'
554:             STOR .F. TO vopc_des
555:       ENDCASE
556:       IF modo # 'REGR.MENU'
557:          DO rubros WITH vmodo
558:       ENDIF
559: <     RETURN
560:
561:          ********************
562:
*!*************************************************************************
**************
563:  *!
564:  *!        Function: BUSC_CLV
565:  *!
566:  *!      Called by: CREAR                (procedure in
AACT_SST.SPR)
567:  *!               : BUSCAR               (procedure in
AACT_SST.SPR)
568:  *!               : OPC_DES()            (function in
DA_ASISG.PRG)
569:  *!               : INGRESO              (procedure in
ACT_PTMO.SPR)
570:  *!
571:  *!          Calls: RUBROS               (procedure in
DA_ASISG.PRG)
572:  *!               : AYU_DPTO             (procedure in
MO_PDLAB.SPR)
573:  *!               : BUSC_GRUPO()         (function in
MO_PDLAB.SPR)
574:  *!
575:
*!*************************************************************************
*************
576:      FUNCTION busc_clv
577:          ******************
578:       IF NOT WEXIST('BUSCAR') && o definir en el
principal
579:          DEFINE WINDOW buscar FROM 0,0 TO 5,30 TITLE
'BUSCAR' SYSTEM
580:       ENDIF
581:          ACTIVATE WINDOW buscar
582:          MOVE WINDOW buscar CENTER
583:          reg_ant = RECNO()
584:          SET ORDER TO vartag(1)
585:          mcod_per = SPAC(6)
586:          STOR 0 TO mmcod_per
587:       DO WHILE .T.
```

```
588:          ||     @ 1,1 SAY 'Ingrese el Cśdigo: ' GET mmcod_per
PICT '999'
589:          ||     READ MODAL
590:          ||     STOR ALLT(STR(mmcod_per)) TO ccod_per
591:          ||     STOR ALLT(STR(vciu)) + '00'+
IIF(mmcod_per<10, '00'+ccod_per,:
592:          ||         IIF(mmcod_per<100, '0'+ccod_per,ccod_per)) TO
mcod_per
593:          ||   ┌─IF LAST() = 27
594:          ||   │     GOTO reg_ant
595:          ||   │     vret = .F.
596:          ||   v═══════EXIT
597:          ||   └─ENDIF
598:          ||     SEEK mcod_per
599:          ||   ┌─IF !FOUND()
600:          ||   │     WAIT WINDOW NOWAIT 'Cśdigo no Existe'
601:          ||   │     GOTO reg_ant
602:          ||   `═══════LOOP
603:          ||   ├─ELSE
604:          ||   │     vret = .T.
605:          ||   v═══════EXIT
606:          ||   └─ENDIF
607:          ||└─ENDDO
608:          ||  DEACTIVATE WINDOW buscar
609:          ||  DO rubros WITH .F.
610:  <═══════RETURN vret
611:
612:
613:          ***************
614:
*!*************************************************************************
**************
615:  *!
616:  *!       Procedure: TEXTO
617:  *!
618:  *!       Called by: DA_ASISG.PRG
619:  *!                : DE_EGRQ1.PRG
620:  *!                : DE_EGRQ2.PRG
621:  *!                : DE_EGRMN.PRG
622:  *!                : DE_IEJUB.PRG
623:  *!
624:
*!*************************************************************************
**************
625:      PROCEDURE texto
626:          ***************
627:          @0.5,23 TO 11,85
628:          @ 1,25     SAY "Codigo:"&fsdesc
629:          @ 1,50     SAY "Nombre:"&fsdesc
630:          @ 3,25     SAY "Cargo:"&fsdesc
631:          @ 5,25     SAY "Departamento:"&fsdesc
632:          @ 7,25     SAY "Ciudad:"&fsdesc
633:          @ 7,60     SAY "Orden :"&fsdesc
634:          @ 9,25     SAY "Fecha :"&fsdesc
635:          @ 9,60     SAY "Cedula:"&fsdesc
636:          @ 10,5,25 SAY REPLICATE("=",60)
637:
638:          @fila,21,5 TO fila + 26,5 ,67
```

```
   639:
   640:          @ fila+2,23  SAY aux_arrbot(1,2) &&"Retencion
Juzgado:"&fsdesc
   641:          @ fila+4,23  SAY aux_arrbot(2,2) &&"Tribunal
Menores:"&fsdesc
   642:          @ fila+6,23  SAY aux_arrbot(3,2) &&"Descuento
Energia:"&fsdesc
   643:          @ fila+8,23  SAY aux_arrbot(4,2) &&"valor para
subsidio:"&fsdesc
   644:          @ fila+10,23 SAY aux_arrbot(5,2) &&"No.Cta
Abonado:"&fsdesc
   645:          @ fila+12,23 SAY aux_arrbot(6,2) &&"Prestamo
Empresa:"&fsdesc
   646:          @ fila+14,23 SAY aux_arrbot(7,2) &&"Prestamo
Iess:"&fsdesc
   647:          @ fila+16,23 SAY aux_arrbot(8,2) &&"Cooperativa
ahorros:"&fsdesc
   648:          @ fila+18,23 SAY aux_arrbot(9,2)  &&"Almacen
Laboral:"&fsdesc
   649:          @ fila+20,23 SAY aux_arrbot(10,2)
&&"Cesantia:"&fsdesc
   650:          @ fila+22, 23 SAY aux_arrbot(11,2)
&&"Multas:"&fsdesc
   651:          @ fila+24, 23 SAY aux_arrbot(12,2) &&"Cta
143:"&fsdesc
   652:
   653:          @ fila+2, 55 SAY aux_arrbot(13,2)
&&"Sindicato:"&fsdesc
   654:          @ fila+4,55 SAY aux_arrbot(14,2)  &&"Asociacion
Emple:"&fsdesc
   655:          @ fila+6,55 SAY aux_arrbot(15,2)  &&"Club
Andinismo"&fsdesc
   656:          @ fila+8,55 SAY aux_arrbot(16,2)  &&"A.T.I.T.
:"&fsdesc
   657:          @ fila+10,55 SAY aux_arrbot(17,2)
&&"Pre-Cooperativa :"&fsdesc
   658:          @ fila+12,55 SAY aux_arrbot(18,2) &&"Financonti
:"&fsdesc
   659:          @ fila+14,55 SAY aux_arrbot(19,2) &&"Otros
Descuentos:" &fsdesc
   660:
   661:          @ fila+16,55 SAY "Total Otr. Descuentos: " &fsdesc
   662:  *------RETURN
   663:
   664:
   665:
   666:          ****************
   667:
*!*************************************************************************
*************
   668:  *!
   669:  *!         Function: VAL_MED
   670:  *!
   671:  *!       Called by: RUBROS                (procedure in
DA_ASISG.PRG)
   672:  *!
   673:
```

```
**************
  674:      FUNCTION val_med
  675:          ****************
  676:          REPLACE &aux_arrbot(04.1) WITH :
  677:              IIF(!EMPTY(dgp.medidor),&aux_arrbot(03.1),0)
  678:          IF dgp.zona = 1
  679:             DO CASE
  680:             CASE (men.medidor > lcc.cc_suben)  AND
(men.medidor < lcc.cc_subenur)
  681:                    IF (men.medidor > lcc.cc_suben)
  682:                        vsubener = men.medidor
  683:                    ELSE
  684:                        vsubener = lcc.cc_suben
  685:                    ENDIF
  686:             CASE (men.medidor > lcc.cc_subenur)
  687:                    vsubener = lcc.cc_subenur
  688:             OTHERWISE
  689:                    vsubener = men.medidor
  690:             ENDCASE
  691:             REPLACE men.medidor WITH vsubener
  692:          ELSE
  693:             IF (men.medidor>lcc.cc_suben)
  694:                 vsubener = lcc.cc_suben
  695:             ELSE
  696:                 vsubener = men.medidor
  697:             ENDIF
  698:             REPLACE men.medidor WITH vsubener
  699:          ENDIF
  700:  <-------RETURN .T.
  701:  *:  EOF: DE_EGRMN.act
```


Varios Archivo Nómina Bol de Pagos ReporTes Ptmo.Emp Costos E.D.36 Salir
ESCOGER
PARAMETROS PARA REGISTRAR
DESCUENTOS PARA BONIFICACIONES
MENU
PASTAZA
JUBILADOS
PALORA
AÑO. . . . : 1995
MES. . . . : 12 DICIEMBRE
BONIFICACION ANIVERSARIO EMPRESA
BONIFICACION DIA DE TRABAJADOR ELECTRICO
BONIFICACION DEL 12 DE NOVIEMBRE
ACEPTAR
EDITAR
SALIR
10:13
Mes
Reg. 12/12
Exclusivo
INS
NUM


Varios Archivo Nómina Rol de Pagos Reportes Ptmo.Emp Costos E.D.36 Salir
Trans. Datos
Retenc. Juzgado
Tribunal Men
Coop Ahorro
Sindicato
Club Andinismo
Asc. Empleados
Otros Descuentos
Terminar
Codigo:
Nombre:
Departamento:
RECURSOS HUMANOS
Ciudad:
AMBATO
Fecha :
12/05/1998
Retencion Juzgado:
Tribunal Menores:
Cooperativa ahorros:
Sindicato:
Club Andinismo:
Asociacion Emple:
Otros Descuentos:
10:12
Dval
Reg. 505/505
Exclusivo
INS
NUM

```
    1:
*:*************************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\CTRL_DVB.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 08/02/1995 at 17:03:16
    8:  *:
    9:  *:  Procs & Fncts: _R7R10JY6J()
   10:  *:                : _R7R10JYAH()
   11:  *:                : _R7R10JYER()
   12:  *:                : _R7R10JYHF()
   13:  *:
   14:  *:          Set by: EMPRESA.MPR
   15:  *:
   16:  *:           Calls: _R7R10JY6J()        (function in
CTRL_DVB.SPR)
   17:  *:                : _R7R10JYAH()        (function in
CTRL_DVB.SPR)
   18:  *:                : _R7R10JYER()        (function in
CTRL_DVB.SPR)
   19:  *:                : _R7R10JYHF()        (function in
CTRL_DVB.SPR)
   20:  *:
   21:  *:       Documented 19:14:52
FoxDoc version 3.00a
   22:
*:*************************************************************************
**************
   23:      *
************************************************************
   24:      *        *
   25:      *        * 08/02/1995            CTRL_DVB.SPR
 17:03:15
   26:      *        *
   27:      *
************************************************************
   28:      *        *
   29:      *        * Nombre del autor
   30:      *        *
   31:      *        * Copyright (c) 1995 Nombre de la empresa
   32:      *        * Direccion
   33:      *        * Ciudad.      C.P.
   34:      *        * Fono
   35:      *        *
   36:      *        * Description:
   37:      *        * Este programa lo ha generado automaticamente
GENSCRN.
   38:      *        *
   39:      *
************************************************************
   40:
   41:      PARAMETERS mproceso
   42:
   43:      *
```

```
*************************************************************
   44:      *        *
   45:      *        *  CTRL_DVB/Windows Código de configuración -
SECCION 1
   46:      *        *
   47:      *
*************************************************************
   48:      *
   49:
   50:      #REGION 1
   51:      **********
   52:      REGIONAL opc_ser, opc_rol, opc_qui, mmes, vmes, manio,
mquincena, mrol.:
   53:          mopc_qui,mfquin,vciu,vtip,opc_bon,mclave
   54:      STOR ' ' TO opc_ser, mmes,opc_rol, fquin,
opc_bon,mclave
   55:      STOR 0 TO opc_qui,mopc_qui,vtip
   56:      STOR 1 TO mrol,vciu
   57:      STOR .F. TO mquincena
   58:      STOR YEAR(DATE()) TO manio
   59:      STOR MONTH(DATE()) TO vmes
   60:
   61:      **********
   62:
   63:      #REGION 0
   64:      REGIONAL m.currarea, m.talkstat, m.compstat
   65:
   66:     -IF SET("TALK") = "ON"
   67:          SET TALK OFF
   68:          m.talkstat = "ON"
   69:     -ELSE
   70:          m.talkstat = "OFF"
   71:     -ENDIF
   72:      m.compstat = SET("COMPATIBLE")
   73:      SET COMPATIBLE FOXPLUS
   74:
   75:      m.rborder = SET("READBORDER")
   76:      SET readborder ON
   77:
   78:      m.currarea = SELECT()
   79:
   80:
   81:      *
*************************************************************
   82:      *        *
   83:      *        *  CTRL_DVB/Windows Bases de datos, índices,
relaciones
   84:      *        *
   85:      *
*************************************************************
   86:      *
   87:
   88:     -IF USED("var")
   89:          SELECT VAR
   90:          SET ORDER TO 0
   91:     -ELSE
   92:          SELECT 0
   93:          USE (LOCFILE("dbfs\varios.dbf","DBF","Buscar
```

```
varios?")):
   94:     |        AGAIN ALIAS VAR ;
   95:     |        ORDER 0
   96:     |    ENDIF
   97:
   98:     |    IF USED("mes")
   99:     |        SELECT mes
  100:     |        SET ORDER TO 0
  101:     |    ELSE
  102:     |        SELECT 0
  103:     |        USE (LOCFILE("dbfs\meses.org","DBF","Buscar
meses?")):
  104:     |        AGAIN ALIAS mes ;
  105:     |        ORDER 0
  106:     |    ENDIF
  107:
  108:     |    IF USED("sst")
  109:     |        SELECT sst
  110:     |        SET ORDER TO TAG "sist_cod"
  111:     |    ELSE
  112:     |        SELECT 0
  113:     |        USE (LOCFILE("dbfs\sistema.dbf","DBF","Buscar
sistema?")):
  114:     |        AGAIN ALIAS sst ;
  115:     |        ORDER TAG "sist_cod"
  116:     |    ENDIF
  117:
  118:       SELECT VAR
  119:
  120:
  121:     *
*************************************************************
  122:     *       *
  123:     *       *              Windows Definiciones de ventana
  124:     *       *
  125:     *
*************************************************************
  126:     *
  127:
  128:     |    IF NOT WEXIST("ctrl_par") ;
  129:     |        OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.PJX" ;
  130:     |        OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.SCX" ;
  131:     |        OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.MNX" ;
  132:     |        OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.PRG" ;
  133:     |        OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.FRX" ;
  134:     |        OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.QPR"
  135:     |        DEFINE WINDOW ctrl_par ;
  136:     |        AT  0.000, 0.000  ;
  137:     |        SIZE 21.615,71.800 ;
  138:     |        TITLE "ESCOGER" ;
  139:     |        FONT "MS Sans Serif", 8 ;
  140:     |        FLOAT ;
  141:     |        NOCLOSE ;
  142:     |        MINIMIZE ;
  143:     |        SYSTEM
  144:     |        MOVE WINDOW ctrl_par CENTER
  145:     |    ENDIF
  146:
```

```
  147:
  148:      *
*************************************************************
  149:      *        *
  150:      *        *  CTRL_DVB/Windows Cšdigo de configuracišn -
SECCION 2
  151:      *        *
  152:      *
*************************************************************
  153:      *
  154:
  155:      #REGION 1
  156:      **********
  157:      SELE mes
  158:      LOCATE FOR mes = vmes
  159:      STOR nom_mes TO mmes
  160:      SELE sst
  161:      STOR sst.ciudad TO opc_rol
  162:      SELE VAR
  163:      STOR var.descrip TO opc_bon
  164:
  165:      #REGION 1
  166:      DEFINE POPUP _r7r10jxyr :
  167:         PROMPT FIELD sst.ciudad :
  168:         SCROLL
  169:      DEFINE POPUP _r7r10jxyv :
  170:         PROMPT FIELD var.descrip :
  171:         SCROLL
  172:
  173:      *
*************************************************************
  174:      *        *
  175:      *        *        CTRL_DVB/Windows Distribucišn de
pantalla
  176:      *        *
  177:      *
*************************************************************
  178:      *
  179:
  180:      #REGION 1
  181:      IF WVISIBLE("ctrl_par")
  182:         ACTIVATE WINDOW ctrl_par SAME
  183:      ELSE
  184:         ACTIVATE WINDOW ctrl_par NOSHOW
  185:      ENDIF
  186:      @ 10.231,30.400 SAY "MES....:" :
  187:         FONT "Courier New", 10 :
  188:         STYLE "BIT"
  189:      @ 8.308,30.400 SAY "AÑO....:" :
  190:         FONT "Courier New", 10 :
  191:         STYLE "BIT"
  192:      @ 19.000,10.600 TO 21.077,59.600 :
  193:         PEN 1, 8 :
  194:         STYLE "16"
  195:      @ 1.385,10.000 SAY "PARAMETROS PARA   REGISTRAR" +
CHR(13) + :
  196:         "DESCUENTOS PARA BONIFICACIONES" :
  197:         SIZE 2.000,40.714, 0.000 :
```

```
198:          PICTURE "@I" :
199:          FONT "MS Serif", 10 :
200:          STYLE "BIT"
201:       @ 0.923,8.000 TO 4.154,68.400 :
202:          PEN 1, 8 :
203:          STYLE "16"
204:       @ 6.462,4.600 TO 12.462,69.000 :
205:          PEN 1, 8
206:       @ 7.385,5.600 GET ope_rol :
207:          PICTURE "@&N" :
208:          POPUP _r7r10jxyr :
209:          SIZE 4.615,21.400 :
210:          DEFAULT " " :
211:          FONT "MS Sans Serif", 8 :
212:          STYLE "" :
213:          VALID _r7r10jy6j()
214:       @ 8.385,44.800 GET manio :
215:          SIZE 1.000,6.600 :
216:          DEFAULT " " :
217:          FONT "MS Sans Serif", 8 :
218:          PICTURE "9999" :
219:          VALID manio > 1990 AND manio < 2999 :
220:          ERROR 'AÑO INVALIDO '
221:       @ 10.308,44.800 GET vmes :
222:          SIZE 1.000,2.600 :
223:          DEFAULT " " :
224:          FONT "MS Sans Serif", 8 :
225:          PICTURE "99" :
226:          VALID _r7r10jyah() :
227:          ERROR 'MES INVALIDO (1..12)'
228:       @ 10.308,49.000 GET mes.nom_mes :
229:          SIZE 1.000,17.800 :
230:          DEFAULT " " :
231:          FONT "MS Sans Serif", 8 :
232:          PICTURE "@K" :
233:          WHEN .F.
234:       @ 13.231,5.000 GET ope_ben :
235:          PICTURE "@&N" :
236:          POPUP _r7r10jxvv :
237:          SIZE 4.615,63.800 :
238:          DEFAULT " " :
239:          FONT "MS Sans Serif", 8 :
240:          STYLE "" :
241:          VALID _r7r10jyer()
242:       @ 19.462,13.600 GET ope_ser :
243:          PICTURE "@!HN \<ACEPTAR:\<EDITAR:\<SALIR" :
244:          SIZE 1.308,12.333,0.667 :
245:          DEFAULT 1 :
246:          FONT "MS Sans Serif", 8 :
247:          STYLE "B" :
248:          VALID _r7r10jyhf()
249:
250:    IF NOT WVISIBLE("ctrl_par")
251:       ACTIVATE WINDOW ctrl_par
252:    ENDIF
253:
254:    READ CYCLE :
255:       VALID .F.
```

```
256:
257:      RELEASE WINDOW ctrl_par
258:
259:      *
**********************************************************
260:      *         *
261:      *         *              Windows Cerrando bases de datos
262:      *         *
263:      *
**********************************************************
264:      *
265:
266:     IF USED("var")
267:        SELECT VAR
268:        USE
269:     ENDIF
270:
271:     IF USED("mes")
272:        SELECT mes
273:        USE
274:     ENDIF
275:
276:     IF USED("sst")
277:        SELECT sst
278:        USE
279:     ENDIF
280:
281:     SELECT (m.currarea)
282:
283:     RELEASE POPUPS _r7r10jxyr, _r7r10jxyv
284:
285:     #REGION 0
286:
287:     SET readborder &rborder
288:
289:     IF m.talkstat = "ON"
290:        SET TALK ON
291:     ENDIF
292:     IF m.compstat = "ON"
293:        SET COMPATIBLE ON
294:     ENDIF
295:
296:
297:      *
**********************************************************
298:      *         *
299:      *         *              CTRL.PRG Windows Cadena de
Timeout
300:      *         *
301:      *
**********************************************************
302:      *
303:
304:      #REGION 1
305:      ****************
306:
307:
308:      *
```

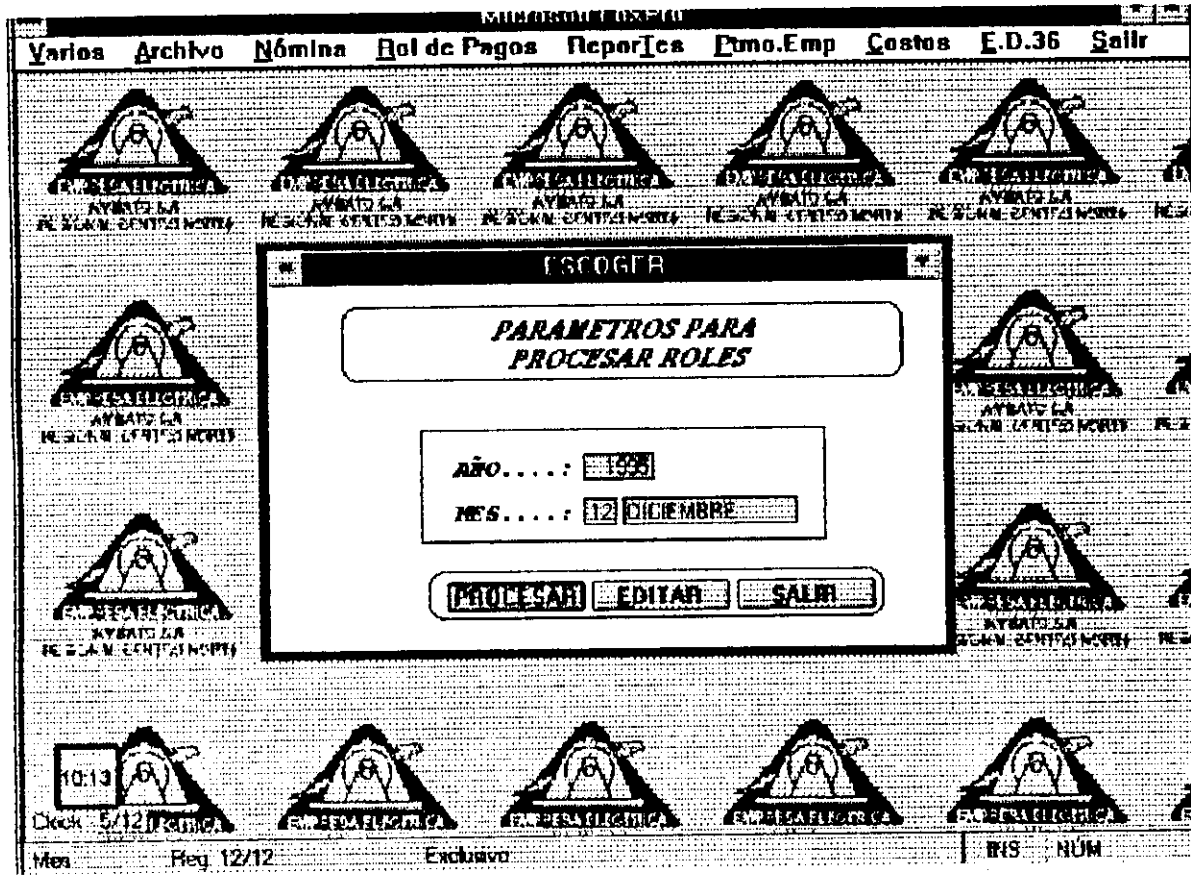
```
************************************************************
  309:      *        *
  310:      *        * CTRL_DVB/Windows Procedimientos y funciones
de soporte
  311:      *        *
  312:      *
************************************************************
  313:      *
  314:
  315:      #REGION 1
  316:
*!*************************************************************************
**************
  317:  *!
  318:  *!       Procedure: ACEPTAR
  319:  *!
  320:  *!           Calls: VAL_PRO()          (function in
VREG_PTE.SPR)
  321:  *!                : PREPARAR           (procedure in
VREG_PTE.SPR)
  322:  *!                : DA_ASISG.PRG
  323:  *!                : DE_EGRQ1.PRG
  324:  *!                : DE_EGRQ2.PRG
  325:  *!                : DE_EGRMN.PRG
  326:  *!                : DE_IEJUB.PRG
  327:  *!                : EGR_VARI.PRG
  328:  *!                : ROL_Q1.SPR
  329:  *!                : ROL_Q2.SPR
  330:  *!                : ROL_MEN.SPR
  331:  *!                : ROL_JUB.SPR
  332:  *!                : ACT_PTMO.SPR
  333:  *!                : MO_PDLAB.SPR
  334:  *!
  335:  *!    Report Forms: VAL_NEGA.FRX
  336:  *!
  337:
*!*************************************************************************
**************
  338:      PROCEDURE aceptar
  339:          ****************
  340:          DO egr_vari WITH mclave.manio
  341:          CLEAR READ
  342:          *&MIPROCESO WITH MCLAVE.MANIO
  343:          *show gets
  344:          RETURN
  345:          ****************
  346:
*!*************************************************************************
**************
  347:  *!
  348:  *!       Procedure: EDITAR
  349:  *!
  350:  *!       Called by: CREAR             (procedure in
AACT_SST.SPR)
  351:  *!
  352:  *!           Calls: DEAC_KEYS         (procedure in
AACT_SST.SPR)
  353:  *!                : CABEZA            (procedure in
```

```
PERSONAL.PRG)
  354:  *!                  : INGRESO              (procedure in
ACT_PTMO.SPR)
  355:  *!                  : CREA_TMP             (procedure in
MO_PDLAB.SPR)
  356:  *!                  : MBROW                (procedure in
MO_PDLAB.SPR)
  357:  *!                  : REF_SCR              (procedure in
MO_PDLAB.SPR)
  358:  *!                  : MENSAJES             (procedure in
MO_PDLAB.SPR)
  359:  *!
  360:
*!*************************************************************************
**************
  361:      PROCEDURE editar
  362:          ****************
  363:          _CUROBJ= OBJNUM(ope_rol)
  364: <=======RETURN
  365:          ****************
  366:
*!*************************************************************************
**************
  367:  *!
  368:  *!       Procedure: SALIR
  369:  *!
  370:  *!       Called by: _RAA1732TG()        (function in
AT_FSCLS.SPR)
  371:  *!                : DA_ASISG.PRG
  372:  *!                : DE_EGRQ1.PRG
  373:  *!                : DE_EGRQ2.PRG
  374:  *!                : DE_EGRMN.PRG
  375:  *!                : DE_IEJUB.PRG
  376:  *!
  377:  *!           Calls: DEAC_KEYS            (procedure in
AACT_SST.SPR)
  378:  *!
  379:
*!*************************************************************************
**************
  380:      PROCEDURE salir
  381:          ***************
  382:          CLEAR READ
  383: <=======RETURN
  384:
  385:          *
*************************************************************
  386:          *       *
  387:          *       * _R7E10YGJ            ope_rol VALID
  388:          *       *
  389:          *       * Function Origin:
  390:          *       *
  391:          *       * From Platform:       Windows
  392:          *       * From Screen:         CTRL_DVE.
Record Number:   11
  393:          *       * Variable:            ope_rol
  394:          *       * Called By:           VALID Clause
  395:          *       * Object Type:         List
```

```
396:          *         * Snippet Number:       1
397:          *         *
398:          *
*************************************************************
399:          *
400:
*!*************************************************************
**************
401:    *!
402:    *!        Function: _R7R10JY6J
403:    *!
404:    *!       Called by: CTRL_DVB.SPR
405:    *!
406:
*!*************************************************************
**************
407:      FUNCTION _r7r10jy6j      &&  opc_rol VALID
408:          #REGION 1
409:          STOR sst.cod_ciu TO veiu
410:          STOR sst.tipo TO vtip
411:
412:
413:          *
*************************************************************
414:          *         *
415:          *         * _R7R10JYAH              VMES VALID
416:          *         *
417:          *         * Function Origin:
418:          *         *
419:          *         * From Platform:          Windows
420:          *         * From Screen:            CTRL_DVB,
Record Number:   13
421:          *         * Variable:               VMES
422:          *         * Called By:              VALID Clause
423:          *         * Object Type:            Field
424:          *         * Snippet Number:         2
425:          *         *
426:          *
*************************************************************
427:          *
428:
*!*************************************************************
**************
429:    *!
430:    *!        Function: _R7R10JYAH
431:    *!
432:    *!       Called by: CTRL_DVB.SPR
433:    *!
434:
*!*************************************************************
**************
435:      FUNCTION _r7r10jyah      &&  VMES VALID
436:          #REGION 1
437:          IF vmes > 0 AND vmes < 13
438:              STOR .T. TO vret
439:              SELE mes
440:              LOCATE FOR vmes = mes
441:              STOR nom_mes TO mmes
```

```
442:            SHOW GET mes.nom_mes
443:        ELSE
444:            STOR .F. TO vret
445:        ENDIF
446:  <=====RETURN vret
447:
448:        *
*************************************************************
449:        *       *
450:        *       * _R7R10JYER            ope_BON VALID
451:        *       *
452:        *       * Function Origin:
453:        *       *
454:        *       * From Platform:        Windows
455:        *       * From Screen:          CTRL_DVB.
Record Number:   15
456:        *       * Variable:             ope_BON
457:        *       * Called By:            VALID Clause
458:        *       * Object Type:          List
459:        *       * Snippet Number:       3
460:        *       *
461:        *
*************************************************************
462:        *
463:
*!*************************************************************
**************
464:  *!
465:  *!        Function: _R7R10JYER
466:  *!
467:  *!      Called by: CTRL_DVB.SPR
468:  *!
469:
*!*************************************************************
**************
470:      FUNCTION _r7r10jyer      &&  ope_BON VALID
471:          #REGION 1
472:          STOR var_tipo TO molave
473:
474:          *
*************************************************************
475:          *       *
476:          *       * _R7R10JYHF            OPC_SCR VALID
477:          *       *
478:          *       * Function Origin:
479:          *       *
480:          *       * From Platform:        Windows
481:          *       * From Screen:          CTRL_DVB.
Record Number:   16
482:          *       * Variable:             OPC_SCR
483:          *       * Called By:            VALID Clause
484:          *       * Object Type:          Push Button
485:          *       * Snippet Number:       4
486:          *       *
487:          *
*************************************************************
488:          *
489:
```

```
*!*************************************************************************
**************
  490:  *!
  491:  *!          Function: _R7R10JYHF
  492:  *!
  493:  *!        Called by: CTRL_DVR.SPR
  494:  *!
  495:  *!             Calls: &OPC_SCR.PRG
  496:  *!
  497:
*!*************************************************************************
**************
  498:       FUNCTION _r7r10jyhf      &&  OPC_SCR VALID
  499:            #REGION 1
  500:            DO &opc_scr
  501:  *:  EOF: CTRL_DVB.ac1
```


Varios Archivo Nómina Rol de Pagos Reportes Emp.Emp Costos E.D.36 Salir
ESCOGER
PARAMETROS PARA
PROCESAR ROLES
AÑO.... : 1998
MES.... : 12 DICIEMBRE
PROCESAR EDITAR SALIR
10:13
Mes
Reg. 12/12
Exclusivo
INS
NUM

```
    1:
*:*************************************************************************
********
    2:  *:
    3:  *: Procedure file: C:\DOC\CTRL_PRO.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *: Last modified: 08/12/1995 at  0:17:04
    8:  *:
    9:  *: Procs & Fncts: _R5Z1G1S7Z()
   10:  *:                : _R5Z1G1SC3()
   11:  *:
   12:  *:          Set by: EMPRESA.MPR
   13:  *:
   14:  *:           Calls: _R5Z1G1S7Z()        (function in
CTRL_PRO.SPR)
   15:  *:                  : _R5Z1G1SC3()        (function in
CTRL_PRO.SPR)
   16:  *:
   17:  *:       Documented 19:14:53
FoxDoc version 3.00a
   18:
*:*************************************************************************
********
   19:      *
************************************************************
   20:      *         *
   21:      *         * 12/08/1995            CTRL_PRO.SPR
00:17:02
   22:      *         *
   23:      *
************************************************************
   24:      *         *
   25:      *         * Nombre del autor
   26:      *         *
   27:      *         * Copyright (c) 1995 Nombre de la empresa
   28:      *         * Direccion
   29:      *         * Ciudad.       C.P.
   30:      *         * Telefono
   31:      *         *
   32:      *         * Description:
   33:      *         * Este programa lo ha generado automBticamente
GENSCRN.
   34:      *         *
   35:      *
************************************************************
   36:
   37:
   38:      *
************************************************************
   39:      *         *
   40:      *         *  CTRL_PRO/Windows Cdigo de configuracion -
SECCION 1
   41:      *         *
   42:      *
************************************************************
   43:      *
```

```
 44:
 45:     #REGION 1
 46:     **********
 47:     REGIONAL opc_scr, opc_rol, opc_qui, mmes, vmes, manio,
mquincena, mrol,;
 48:        mopc_qui,mfquin
 49:     STOR " " TO opc_scr, mmes,opc_rol, fquin
 50:     STOR 0 TO opc_qui.mopc_qui
 51:     STOR 1 TO mrol
 52:     STOR .F. TO mquincena
 53:     STOR YEAR(DATE()) TO manio
 54:     STOR MONTH(DATE()) TO vmes
 55:
 56:     **********
 57:
 58:     #REGION 0
 59:     REGIONAL m.currarea, m.talkstat, m.compstat
 60:
 61:  ┌─IF SET("TALK") = "ON"
 62:  │     SET TALK OFF
 63:  │     m.talkstat = "ON"
 64:  ├─ELSE
 65:  │     m.talkstat = "OFF"
 66:  └─ENDIF
 67:     m.compstat = SET("COMPATIBLE")
 68:     SET COMPATIBLE FOXPLUS
 69:
 70:     m.rborder = SET("READBORDER")
 71:     SET readborder ON
 72:
 73:     m.currarea = SELECT()
 74:
 75:
 76:     *
*************************************************************
 77:     *        *
 78:     *        *  CTRL_PRO/Windows Bases de datos, índices,
relaciones
 79:     *        *
 80:     *
*************************************************************
 81:     *
 82:
 83:  ┌─IF USED("ctr")
 84:  │     SELECT ctr
 85:  │     SET ORDER TO TAG "clave"
 86:  ├─ELSE
 87:  │     SELECT 0
 88:  │     USE (LOCFILE("dbfs\ctrl_rol.dbf","DBF","Buscar
ctrl rol?"));
 89:  │           AGAIN ALIAS ctr :
 90:  │           ORDER TAG "clave"
 91:  └─ENDIF
 92:
 93:  ┌─IF USED("mes")
 94:  │     SELECT mes
 95:  │     SET ORDER TO 0
 96:  ├─ELSE
```

```
 97:     |     SELECT 0
 98:     |    USE (LOCFILE("dbfs\meses.org","DBF","Buscar  meses?"));
 99:     |        AGAIN ALIAS mes ;
100:     |        ORDER 0
101:     |ENDIF
102:
103:     |IF USED("sst")
104:     |    SELECT sst
105:     |    SET ORDER TO TAG "sist_cod"
106:     |ELSE
107:     |    SELECT 0
108:     |    USE (LOCFILE("dbfs\sistema.dbf","DBF","Buscar
sistema?"));
109:     |        AGAIN ALIAS sst ;
110:     |        ORDER TAG "sist_cod"
111:     |ENDIF
112:
113:     SELECT ctr
114:
115:
116:     *
*************************************************************
117:     *       *
118:     *       *              Windows Definiciones de ventana
119:     *       *
120:     *
*************************************************************
121:     *
122:
123:     |IF NOT WEXIST("ctrl_par") ;
124:     |       OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.PJX" ;
125:     |       OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.SCX" ;
126:     |       OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.MNX" ;
127:     |       OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.PRG" ;
128:     |       OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.FRX" ;
129:     |       OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.QPR"
130:     |    DEFINE WINDOW ctrl_par ;
131:     |       AT  0.000, 0.000  ;
132:     |       SIZE 15.615,72.600 ;
133:     |       TITLE "ESCOGER" ;
134:     |       FONT "MS Sans Serif", 8 ;
135:     |       FLOAT ;
136:     |       NOCLOSE ;
137:     |       MINIMIZE ;
138:     |       SYSTEM
139:     |    MOVE WINDOW ctrl_par CENTER
140:     |ENDIF
141:
142:
143:     *
*************************************************************
144:     *       *
145:     *       *  CTRL_PRO/Windows Código de configuración -
SECCION 2
146:     *       *
147:     *
*************************************************************
148:     *
```

```
149:
150:      #REGION 1
151:      **********
152:      SELE mes
153:      LOCATE FOR mes = vmes
154:      STOR nom_mes TO mmes
155:
156:      *
*****************************************************************
157:      *        *
158:      *        *          CTRL_PRO/Windows Distribuci≤n de pantalla
159:      *        *
160:      *
*****************************************************************
161:      *
162:
163:      #REGION 1
164:     ┌IF WVISIBLE("ctrl_par")
165:     │   ACTIVATE WINDOW ctrl_par SAME
166:     ├ELSE
167:     │   ACTIVATE WINDOW ctrl_par NOSHOW
168:     └ENDIF
169:      @ 9.231,20.000 SAY "MES....:" :
170:         FONT "Courier New", 10 :
171:         STYLE "BIT"
172:      @ 7.308,20.000 SAY "A±O....:" :
173:         FONT "Courier New", 10 :
174:         STYLE "BIT"
175:      @ 12.308,17.200 TO 14.385,66.200 :
176:         PEN 1, 8 :
177:         STYLE "16"
178:      @ 7.385,34.400 GET manio :
179:         SIZE 1.000,6.600 :
180:         DEFAULT " " :
181:         FONT "MS Sans Serif", 8 :
182:         PICTURE "9999" :
183:         VALID manio > 1990 AND manio < 2999 :
184:         ERROR 'A±O INVALIDO '
185:      @ 9.308,34.400 GET vmes :
186:         SIZE 1.000,2.600 :
187:         DEFAULT " " :
188:         FONT "MS Sans Serif", 8 :
189:         PICTURE "99" :
190:         VALID _r5z1g1s7z() :
191:         ERROR 'MES INVALIDO (1..12)'
192:      @ 12.769,19.200 GET ope_scr :
193:         PICTURE "@*HN \<PROCESAR;\<EDITAR;\<SALIR" :
194:         SIZE 1.308,12.333,0.667 :
195:         DEFAULT 1 :
196:         FONT "MS Sans Serif", 8 :
197:         STYLE "B" :
198:         VALID _r5z1g1sc3()
199:      @ 1.385,22.200 SAY "PARAMETROS PARA    " + CHR(13) + :
200:         "PROCESAR ROLES" :
201:         SIZE 2.000,23.143, 0.000 :
202:         PICTURE "@I" :
203:         FONT "MS Serif", 10 :
204:         STYLE "BIT"
```

```
205:      @ 0.923,8.000 TO 4.154,68.400 :
206:         PEN 1, 8 :
207:         STYLE "16"
208:      @ 9.308,38.600 GET mes.nom_mes :
209:         SIZE 1.000,17.800 :
210:         DEFAULT " " :
211:         FONT "MS Sans Serif", 8 :
212:         PICTURE "@K" :
213:         WHEN .F.
214:      @ 6.231,16.400 TO 11.077,60.000 :
215:         PEN 1, 8
216:
217:   IF NOT WVISIBLE("ctrl_par")
218:      ACTIVATE WINDOW ctrl_par
219:   ENDIF
220:
221:      READ CYCLE :
222:         VALID .F.
223:
224:      RELEASE WINDOW ctrl_par
225:
226:      *
*************************************************************
227:      *        *
228:      *        *              Windows Cerrando bases de datos
229:      *        *
230:      *
*************************************************************
231:      *
232:
233:   IF USED("ctr")
234:      SELECT ctr
235:      USE
236:   ENDIF
237:
238:   IF USED("mes")
239:      SELECT mes
240:      USE
241:   ENDIF
242:
243:   IF USED("sst")
244:      SELECT sst
245:      USE
246:   ENDIF
247:
248:      SELECT (m.currarea)
249:
250:
251:      #REGION 0
252:
253:      SET readborder &rborder
254:
255:   IF m.talkstat = "ON"
256:      SET TALK ON
257:   ENDIF
258:   IF m.compstat = "ON"
259:      SET COMPATIBLE ON
260:   ENDIF
```

```
  261:
  262:
  263:      *
************************************************************
  264:      *        *
  265:      *        *            CTRL_PRO/Windows CSdigo de limpieza
  266:      *        *
  267:      *
************************************************************
  268:      *
  269:
  270:      #REGION 1
  271:      ******************
  272:
  273:
  274:      *
************************************************************
  275:      *        *
  276:      *        * CTRL_PRO/Windows Procedimientos y funciones de
soporte
  277:      *        *
  278:      *
************************************************************
  279:      *
  280:
  281:      #REGION 1
  282:
*!*************************************************************************
********
  283:  *!
  284:  *!       Procedure: PROCESAR
  285:  *!
  286:  *!           Calls: MEDIDOR.PRG
  287:  *!                : CALCULO.PRG
  288:  *!                : VALIDAR()          (function in
CTRL_GTO.SPR)
  289:  *!                : ABRIR_ARCH         (procedure in
PROC_DMO.SPR)
  290:  *!                : GN_MPL             (procedure in
PROC_DMO.SPR)
  291:  *!                : GN_PDMO            (procedure in
PROC_DMO.SPR)
  292:  *!                : GN_CTA_RUB         (procedure in
PROC_DMO.SPR)
  293:  *!                : GN_ASI_CON         (procedure in
PROC_DMO.SPR)
  294:  *!
  295:  *!          Uses: SOBRETIE.DBF
  296:  *!                : &MARCHI                  Alias: CP
  297:  *!                : DSCTOSVA.DBF             Alias: DVAR
  298:  *!                : PERSONAL.DBF             Alias: DGP
  299:  *!                : LISTBANC.DBF             Alias: BNC
  300:  *!                : TOT_ROL.DBF              Alias: TROL
  301:  *!                : PTMO_EMP.DBF
  302:  *!                : CTAS_CPE.DBF
  303:  *!                : TMP_ACPE.DBF             Alias: TMPPE
  304:  *!                : TMP_PE2.DBF
  305:  *!                : ROL_IBS.DBF
```

```
306:  *!                 : ROL_IBST.DBF
307:  *!                 : DBF.DBF
308:  *!
309:  *!        Indexes: CODIGO                    (tag in PUESTOS.CDX)
310:  *!                 : CODIGO.CDX
311:  *!                 : &TMP_MPL..CDX
312:  *!
313:  *!      CDX files: DSCTOSVA.CDX
314:  *!                 : PERSONAL.CDX
315:  *!                 : LISTBANC.CDX
316:  *!                 : FTMO_EMP.CDX
317:  *!                 : ROL_IBS.CDX
318:  *!
319:
*!*************************************************************************
********
320:      PROCEDURE procesar
321:          ******************
322:          STOR 2 TO ing_egr
323:          *close data
324:          *Sele 24
325:          *Use sistema alias sst
326:          SELE sst
327:          COUNT TO A
328:          DIME sis[a]
329:          GO TOP
330:          FOR i = 1 TO A
331:             STOR ciudad TO sis[i]
332:             IF NOT EOF()
333:                SKIP
334:             ENDIF
335:          ENDF
336:          STOR sis[1] TO msis
337:          STOR 0 TO vciu.vtip
338:          DEFI WIND escoge FROM 18.30 TO 32.70 MDI SYST TITLE
'Escoja el Rol a Procesar'
339:          ACTI WIND escoge
340:          @2.2 GET msis FROM sis PICT "@^" :
341:             SIZE 1.20 :
342:             DEFAULT 1 :
343:             COLOR SCHEME 10, 2
344:          READ
345:          GO TOP
346:          LOCA FOR ALLT(ciudad) = ALLT(msis)
347:          STOR cod_ciu TO vciu
348:
349:          DO CASE
350:          CASE tipo = 0
351:                STOR 'Primera Quincena' TO tip
352:                @6.2 GET tip PICT '@' Primera Quincena;Segunda
Quincena'
353:                READ
354:                IF tip = 'Primera Quincena'
355:                   STOR 1 TO vtip
356:                ELSE
357:                   STOR 2 TO vtip
358:                ENDIF
359:          CASE tipo = 1
```

```
360:          |        STOR 3 TO vtip
361:          |=====CASE tipo = 2
362:          |        STOR 4 TO vtip
363:          |    ENDCASE
364:          |    DEAC WIND escoge
365:          |    malias = ALIAS()
366:          |    SELE ctr
367:          |    SEEK
VAL(ALLT(STR(manio))+ALLT(STR(vmes))+ALLT(STR(veiu))+ALLT(STR(vtip)))
368:          |-IF FOUND() AND control = 'C'
369:          |  |-IF NOT WEXIST('VARIOS') && o definir en el principal
370:          |  |    DEFINE WINDOW varios FROM 0,0 TO 7,35 TITLE
'ALERTA ..!' SYSTEM
371:          |  |  ENDIF
372:          |  |  ACTIVATE WINDOW varios
373:          |  |  MOVE WINDOW varios CENTER
374:          |  |  @ 1,2 SAY 'El Proceso ya ha sido cerrado '
375:          |  |  @ 2,6 SAY 'Proceso No Permitido'
376:          |  |  STOR 1 TO ope_ac
377:          |  |  @ 4,10 GET ope_ac PICT '@*H \<ACEPTAR';
378:          |  |     SIZE 1.308,12.500,0.308 :
379:          |  |     DEFAULT 1 ;
380:          |  |     FONT "MS Serif", 8 ;
381:          |  |     STYLE "BT"
382:          |  |  READ MODAL
383:          |  |  RELE WIND varios
384:          |  |  SELE &malias
385:          |-ELSE
386:          |     DO calculo WITH veiu,vtip,manio,vmes
387:          |  ENDIF
388:          |-IF USED("mes")
389:          |     SELECT mes
390:          |     SET ORDER TO 0
391:          |  ELSE
392:          |     SELECT 0
393:          |     USE (LOCFILE("dbfs\meses.org","DBF","Where is
meses?"));
394:          |        AGAIN ALIAS mes ;
395:          |        ORDER 0
396:          |  ENDIF
397:
398:          |-IF USED("sst")
399:          |     SELECT sst
400:          |     SET ORDER TO TAG "sist_cod"
401:          |  ELSE
402:          |     SELECT 0
403:          |     USE (LOCFILE("dbfs\sistema.dbf","DBF","Where is
sistema?"));
404:          |        AGAIN ALIAS sst ;
405:          |        ORDER TAG "sist_cod"
406:          |  ENDIF
407:
408: <=======RETURN
409:          ****************
410:
*!*************************************************************************
********
411:  *!
```

```
  412:  *!       Procedure: EDITAR
  413:  *!
  414:  *!       Called by: CREAR              (procedure in
AACT_SST.SPR)
  415:  *!
  416:  *!           Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
  417:  *!                : CABEZA             (procedure in
PERSONAL.PRG)
  418:  *!                : INGRESO            (procedure in
ACT_PTMO.SPR)
  419:  *!                : CREA_TMP           (procedure in
MO_PDLAB.SPR)
  420:  *!                : MBROW              (procedure in
MO_PDLAB.SPR)
  421:  *!                : REF_SCR            (procedure in
MO_PDLAB.SPR)
  422:  *!                : MENSAJES           (procedure in
MO_PDLAB.SPR)
  423:  *!
  424:
*!*************************************************************************
********
  425:      PROCEDURE editar
  426:          ****************
  427:          *SHOW GETS DISABLE
  428:          _CUROBJ= OBJNUM(manio)
  429: <=======RETURN
  430:          ****************
  431:
*!*************************************************************************
********
  432:  *!
  433:  *!       Procedure: SALIR
  434:  *!
  435:  *!       Called by: _RAA1732TG()       (function in
AT_ESCLS.SPR)
  436:  *!                : DA_ASISG.PRG
  437:  *!                : DE_EGRQ1.PRG
  438:  *!                : DE_EGRQ2.PRG
  439:  *!                : DE_EGRMN.PRG
  440:  *!                : DE_IEJUB.PRG
  441:  *!
  442:  *!           Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
  443:  *!
  444:
*!*************************************************************************
********
  445:      PROCEDURE salir
  446:          ***************
  447:          CLEAR READ
  448: <=======RETURN
  449:
  450:          *
*************************************************************
  451:          *       *
  452:          *       * _R5Z1G1S7Z          VMES VALID
```

```
453:          *        *
454:          *        * Function Origin:
455:          *        *
456:          *        * From Platform:        Windows
457:          *        * From Screen:          CTRL_PRO.      Record
Number:    9
458:          *        * Variable:             VMES
459:          *        * Called By:            VALID Clause
460:          *        * Object Type:          Field
461:          *        * Snippet Number:       1
462:          *        *
463:          *
*************************************************************
464:          *
465:
*!*************************************************************************
********
466:  *!
467:  *!        Function: _R5Z1G1S7Z
468:  *!
469:  *!      Called by: CTRL_PRO.SPR
470:  *!
471:
*!*************************************************************************
********
472:      FUNCTION _r5z1g1s7z     &&  VMES VALID
473:          #REGION 1
474:          IF vmes > 0 AND vmes < 13
475:              STOR .T. TO vret
476:              SELE mes
477:              LOCATE FOR vmes = mes
478:              STOR nom_mes TO mmes
479:              SHOW GET mes.nom_mes
480:          ELSE
481:              STOR .F. TO vret
482:          ENDIF
483:  <=======RETURN vret
484:
485:          *
*************************************************************
486:          *        *
487:          *        * _R5Z1G1SC3            OPC_SCR VALID
488:          *        *
489:          *        * Function Origin:
490:          *        *
491:          *        * From Platform:        Windows
492:          *        * From Screen:          CTRL_PRO.      Record
Number:   10
493:          *        * Variable:             OPC_SCR
494:          *        * Called By:            VALID Clause
495:          *        * Object Type:          Push Button
496:          *        * Snippet Number:       2
497:          *        *
498:          *
*************************************************************
499:          *
500:
*!*************************************************************************
```

```
********
  501:  *!
  502:  *!        Function: _R5Z1G1SC3
  503:  *!
  504:  *!      Called by: CTRL_PRO.SPR
  505:  *!
  506:  *!           Calls: &OPC_SCR.PRG
  507:  *!
  508:
*!*************************************************************************
********
  509:      FUNCTION _r5z1g1sc3      &&  OPC_SCR VALID
  510:          #REGION 1
  511:          DO &opc_scr
  512:  *:  EOF: CTRL_PRO.ac1
```

```
    1:
*:*********************************************************************
********
    2:  *:
    3:  *: Procedure file: C:\DOC\CALCULO.PRG
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 10/04/1995 at 17:32:20
    8:  *:
    9:  *:  Procs & Fncts: INICIALIZAR
   10:  *:                : SELECCIONAR
   11:  *:
   12:  *:         Set by: RESPALDAR          (procedure in
CTRL_RES.SPR)
   13:  *:                : PROCESAR          (procedure in
VREG_ENG.SPR)
   14:  *:
   15:  *:          Calls: INICIALIZAR        (procedure in
CALCULO.PRG)
   16:  *:                : CAL_DIAS()        (function in
GRAFICOS.PRG)
   17:  *:                : SUB_ANT()         (function in
GRAFICOS.PRG)
   18:  *:                : H_EXT_AME()       (function in
GRAFICOS.PRG)
   19:  *:                : H_EXT_AME75()     (function in
GRAFICOS.PRG)
   20:  *:                : SUB_REF()         (function in
GRAFICOS.PRG)
   21:  *:                : C_S1()            (function in
GRAFICOS.PRG)
   22:  *:                : C_S2()            (function in
GRAFICOS.PRG)
   23:  *:                : BONI_COMPLE()     (function in
GRAFICOS.PRG)
   24:  *:                : SUB_TRANS()       (function in
GRAFICOS.PRG)
   25:  *:                : DEC_SEX()         (function in
GRAFICOS.PRG)
   26:  *:                : PROVISION()       (function in
GRAFICOS.PRG)
   27:  *:                : BASETERCER()      (function in
GRAFICOS.PRG)
   28:  *:                : BASEIESS()        (function in
GRAFICOS.PRG)
   29:  *:                : FON_RESE()        (function in
GRAFICOS.PRG)
   30:  *:                : IMP_RENTA()       (function in
GRAFICOS.PRG)
   31:  *:                : PING_Q1.PRG
   32:  *:                : PEGR_Q1.PRG
   33:  *:                : PING_Q2.PRG
   34:  *:                : PEGR_Q2.PRG
   35:  *:                : PING_MEN.PRG
   36:  *:                : PEGR_MEN.PRG
   37:  *:                : ROL_JUBI.PRG
   38:  *:                : SELECCIONAR       (procedure in
```

```
CALCULO.PRG)
   39:  *:
   40:  *:            Uses: PERSONAL.DBF              Alias: DGP
   41:  *:                : LEY_LABO.DBF              Alias: LEYL
   42:  *:                : LEY_CONT.DBF              Alias: LCC
   43:  *:                : MENSUAL.DBF               Alias: MEN
   44:  *:                : ROL.DBF                   Alias: ROL
   45:  *:                : DEPARTAM.DBF              Alias: DPTO
   46:  *:                : SECCION.DBF               Alias: SEC
   47:  *:                : PARAMVAR.DBF              Alias: PARA
   48:  *:                : IMPRENTA.DBF
   49:  *:                : MEN.DBF                   Alias: MEN
   50:  *:                : LISTBANC.DBF              Alias: BNC
   51:  *:                : AC_IMPRTA.DBF             Alias: ACR
   52:  *:                : MESES.ORG                 Alias: MES
   53:  *:
   54:  *:         Indexes: DEP                       (tag in DEPARTAM.CDX)
   55:  *:                : CODIGO                    (tag in PUESTOS.CDX)
   56:  *:                : SEC_REP                   (tag in SECCION.CDX)
   57:  *:                : GRU_CTA                   (tag in PERSONAL.CDX)
   58:  *:
   59:  *:       CDX files: PERSONAL.CDX
   60:  *:                : LEY_CONT.CDX
   61:  *:                : MENSUAL.CDX
   62:  *:                : ROL.CDX
   63:  *:                : DEPARTAM.CDX
   64:  *:                : SECCION.CDX
   65:  *:                : LISTBANC.CDX
   66:  *:                : AC_IMPRTA.CDX
   67:  *:                : MESES.CDX
   68:  *:
   69:  *:       Documented 19:15:13
FoxDoc version 3.00a
   70:
*:*********************************************************************************
********
   71:      ********************************
   72:      ** REALIZA PROCESOS ROL DE PAGOS*
   73:      ********************************
   74:      PARAMETERS vciu, vtip, manio, vmes
   75:      CLOSE DATA
   76:      SET ESCA ON
   77:      SET STATUS BAR ON
   78:      SET EXAC ON
   79:      SET PROCEDURE TO graficos
   80:      PUBLIC vcodrol,vcodrol1,ep,vvaca,dvact
   81:      SET readborder ON
   82:      SELECT A
   83:      USE personal ALIAS dgp
   84:      SET ORDER TO cod_per
   85:
   86:      SELECT B
   87:      USE ley_labo ALIAS leyl
   88:      SET FILTER TO anio = manio AND mes = vmes
   89:
   90:      SELECT C
   91:      USE ley_cont ALIAS leyc
   92:      SET FILTER TO anio = manio AND mes = vmes
```

```
 93:
 94:      SELECT D
 95:      USE mensual ALIAS mens
 96:      SET ORDER TO mens_cod
 97:      SELECT * FROM mensual mens INTO DBF dbfs\men:
 98:         WHERE mes = vmes AND anio = manio  ORDER BY codigo
 99:      *and ciudad = vciu
100:      *set filter to mes = vmes and anio = manio
101:      SELE men
102:      USE
103:
104:      SELECT E
105:      USE rol ALIAS rol
106:      SELECT * FROM rol rol INTO DBF dbfs\roltmp:
107:         WHERE !(VAL(mes) = vmes AND VAL(anio) = manio)
108:      SELE roltmp
109:      USE
110:      SELE rol
111:      DELE FOR !(VAL(mes) = vmes AND VAL(anio) = manio)
112:      PACK
113:      SET ORDER TO cod_per
114:
115:      SELECT F
116:      USE departam ALIAS dpto
117:      *SET ORDER TO COD_DEP
118:      INDEX ON cod_dep TAG dep
119:      SET ORDER TO dep
120:
121:      SELECT G
122:      USE seccion ALIAS secc
123:      SET ORDER TO secc_cod
124:
125:      SELE i
126:      USE paramvar ALIAS PARA
127:
128:      SELE 20
129:      USE imprenta
130:      USE men ALIAS men IN 0
131:      USE listbanc ALIAS bnc IN 0
132:      USE ac_imprta ORDER cod_per ALIAS acr IN 0
133:
134:      SELE men
135:      INDEX ON codigo TAG codigo
136:
137:      DEFI WIND waiting FROM 15,35 TO 20,70  PANEL
138:      *mdi system
139:      ACTI WIND waiting
140:      @ 0,1 SAY " Realizando Calculos de Rol...!"
141:      PUBLIC
dsueldo,hsueldo,msueldo,ndias,varex1,varex2,varex3,vsuban1,vsuban2,vtot
,vcodrol,manios:
142:
vhx,sueldo_1,dsa,da,pa1,pa2,de,pe1,pe2,dm,pm1,pm2,vvaca,dv
143:      DO inicializar
144:      @ 0,1 SAY " Realizando Calculos de Rol...!"
145:      SELE rol
146:      SET FILTER TO cod_rol = vcodrol
147:      GO TOP
```

```
653:      DEAC WIND waiting
654:      SELE A
655:      USE
656:      SELE B
657:      USE
658:      SELE D
659:      USE
660:      DO seleccionar
661:      ACTI WIND waiting
662:      ?CHR(7)
663:      @ 0,1 SAY " Proceso de Rol Terminado  ...!"
664:      STOR 1 TO mcont
665:      @ 2,13 GET mcont;
666:         PICTURE "@*h \<Aceptar" ;
667:         SIZE 1.15 FONT 'R',10 STYLE "BI"
668:      READ MODAL
669:      DEACT WIND waiting
670: <===RETURN
671:
672:
*!*************************************************************************
********
673:  *!
674:  *!       Procedure: INICIALIZAR
675:  *!
676:  *!       Called by: CALCULO.PRG
677:  *!
678:  *!           Calls: SELECCIONAR         (procedure in
CALCULO.PRG)
679:  *!
680:
*!*************************************************************************
********
681:      PROCEDURE inicializar
682:          SELECT rol
683:          STORE SPACE(6) TO veodrol,veodrol1
684:          veodrol =  ALLTRIM(STR(veiu)) +;
685:             ALLTRIM(IIF(vmes >= 10, STR(vmes), '0'+
ALLTRIM(STR(vmes)))) +;
686:             SUBSTR(ALLTRIM(STR(manio)),3,2) +;
687:             ALLTRIM(STR(vtip))
688:          veodrol1 =  ALLTRIM(STR(veiu)) +;
689:             ALLTRIM(IIF(vmes >= 10, STR(vmes), '0'+
ALLTRIM(STR(vmes)))) +;
690:             SUBSTR(ALLTRIM(STR(manio)),3,2) +;
691:             ALLTRIM(STR(1))
692:
693:          GO TOP
694:          LOCATE FOR cod_rol = veodrol
695:          STOR 0 TO mcont
696:          IF FOUND()
697:             IF NOT WEXIST('waiting')
698:                DEFI WIND waiting FROM 15,35 TO 20,70 PANEL
699:             ENDIF
700:             ACTI WIND waiting
701:             @0,0 SAY " Rol Ya Ha Sido Generado....!"
702:             STOR 1 TO mcont
703:             ?CHR(7)
```

```
704:          @ 2,8 GET mcont;
705:             PICTURE "@*h c\<Aceptar;\<Continuar" ;
706:             SIZE 1,10 FONT 'R',10 STYLE "BI"
707:          READ MODAL
708:          CLEAR
709:          IF mcont = 1
710:             DEAC WIND waiting
711:             CLOSE DATA
712:             *       Return to empresa
713:             ********* REGRESA A PANTALLA INICIAL/PARAMETROS
******
714:             DO seleccionar
715: <============RETURN TO ctrl_pro.spr
716:          ELSE
717:             DELE FOR cod_rol = vcodrol
718:             PACK
719:          ENDIF
720:       ENDIF
721:       IF NOT FOUND() OR mcont = 2
722:          SELECT dgp
723:          IF vtip = 4
724:             STOR 'Estado = 3' TO cond2
725:          ELSE
726:             STOR  'Num_Ciu = Vciu .and. estado = 1'  TO cond2
727:          ENDIF
728:          SCAN FOR &cond2
729:             SELECT rol
730:             APPEND BLANK
731:             REPLACE cod_rol   WITH vcodrol,;
732:                anio       WITH ALLT(STR(manio)),;
733:                mes        WITH STR(vmes,2),;
734:                fecha      WITH DATE(),;
735:                ciudad     WITH vciu,;
736:                tipo       WITH vtip,;
737:                cod_per    WITH dgp.cod_per,;
738:                nombres    WITH dgp.nombre,;
739:                cargo      WITH
IIF(vtip#4,dgp.cargo,ALLT(dgp.cargo))+' '+DTOC(dgp.fec_jub)),;
740:                cta_ctble WITH dgp.cuent_cont,;
741:                num_sec    WITH dgp.num_sec,;
742:                cod_dpto   WITH dgp.num_dep;
743:                sueldo_nom WITH dgp.sueldo
744:             SELE F
745:             GO TOP
746:             LOCATE FOR cod_dep = dgp.num_dep
747:             SELE G
748:             GO TOP
749:             LOCATE FOR cod_sec = dgp.num_sec
750:             SELE rol
751:             REPLACE departam WITH f.nombre
752:             REPLACE seccion WITH g.nombre
753:             REPLACE grupo WITH g.grupotra
754:             REPLACE jerarquia WITH ABS(18 - dgp.categor )
755:             REPLACE persind WITH dgp.sindicato
756:             IF vtip = 2
757:                regis= RECNO()
758:                STOR cod_per TO cp
759:                LOCATE FOR cod_per = cp AND cod_rol = vcodrol1
```

```
760:                    STOR 0 TO AS
761:                    IF FOUN()
762:                        STOR ant_sueldo TO AS
763:                        GO regis
764:                        REPLACE ant_sueldo WITH AS
765:                    ENDIF
766:                ENDIF
767:            ENDSCAN
768:            SELECT rol
769:            REINDEX
770:        ENDIF
771:  <=====RETURN
772:        **********************
773:
*!*************************************************************************
********
774:  *!
775:  *!      Procedure: SELECCIONAR
776:  *!
777:  *!      Called by: CALCULO.PRG
778:  *!               : INICIALIZAR        (procedure in
CALCULO.PRG)
779:  *!
780:  *!           Uses: ROL.DBF                Alias: ROL
781:  *!               : ROLTMP.DBF
782:  *!
783:  *!      CDX files: ROL.CDX
784:  *!
785:
*!*************************************************************************
********
786:      PROCEDURE seleccionar
787:          **********************
788:          USE rol
789:          APPEN FROM dbfs\roltmp
790:          USE
791:          DELE FILE(dbfs\roltmp.dbf)
792:          IF USED("mes")
793:              SELECT mes
794:              SET ORDER TO 0
795:          ELSE
796:              SELECT 0
797:              USE (LOCFILE("dbfs\meses.org","DBF","Where is
meses?"));
798:                  AGAIN ALIAS mes ;
799:                  ORDER 0
800:          ENDIF
801:
802:          IF USED("sst")
803:              SELECT sst
804:              SET ORDER TO TAG "sist_cod"
805:          ELSE
806:              SELECT 0
807:              USE (LOCFILE("dbfs\sistema.dbf","DBF","Where is
sistema?"));
808:                  AGAIN ALIAS sst ;
809:                  ORDER TAG "sist_cod"
810:          ENDIF
```

```
    1:
*:*************************************************************************
********
    2:  *:
    3:  *: Procedure file: C:\DOC\GRAFICOS.PRG
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995. Sisrol Ambato - Ecuador
    7:  *:  Last modified: 12/05/1995 at 18:34:14
    8:  *:
    9:  *:  Procs & Fncts: CTRLERR
   10:  *:                : ACERCA
   11:  *:                : TAG
   12:  *:                : ESTILOS
   13:  *:                : BMS
   14:  *:                : BMS1
   15:  *:                : TEXTO
   16:  *:                : SALI
   17:  *:                : LOGO
   18:  *:                : ESC_SIST
   19:  *:                : SISTEMA_F
   20:  *:                : CAL_DIAS()
   21:  *:                : CAL_SUEL()
   22:  *:                : CAL_SALA()
   23:  *:                : SUB_ANT()
   24:  *:                : SUB_TRANS()
   25:  *:                : SUB_REF()
   26:  *:                : BASEIMP()
   27:  *:                : BASEIESS()
   28:  *:                : BASETERCER()
   29:  *:                : C_S1()
   30:  *:                : C_S2()
   31:  *:                : BONI_COMPLE()
   32:  *:                : DEC_SEX5555()
   33:  *:                : DEC_SEX()
   34:  *:                : FON_RESE()
   35:  *:                : PROVISION()
   36:  *:                : IMC_RENTA()
   37:  *:                : H_EXT_AME()
   38:  *:                : H_EXT_AME75()
   39:  *:
   40:  *:         Set by: RESTALDAR          (procedure in
CTRL_RES.SPR)
   41:  *:
   42:  *:      Documented 19:14:09
FoxDoc version 3.00a
   43:
*:*************************************************************************
********
   44:      *****************
   45:
*!*************************************************************************
********
   46:  *!
   47:  *!      Procedure: CTRLERR
   48:  *!
   49:  *!      Called by: RESPALDAR          (procedure in
CTRL_RES.SPR)
```

```
 50:  *!
 51:  *!          Calls: OFF.PRG
 52:  *!               : SET.PRG
 53:  *!
 54:
*!*************************************************************************
********
 55:      PROCEDURE ctrlerr
 56:          ****************
 57:          PARAMETER merror, mens, mens1, mprog, mn mlφn
 58:        ┌─DO CASE
 59:        ├────CASE merror = 56
 60:        │       WAIT WIND 'INSERTE OTRO DISKETTE'
 61:        │       RETRY
 62:        ├────OTHERWISE
 63:        │     ┌─IF NOT WEXIST("ctrlerr")
 64:        │     │   DEFINE WINDOW ctrlerr ;
 65:        │     │      AT  0.000, 0.000  ;
 66:        │     │      SIZE 21.923,77.000 ;
 67:        │     │      FONT "MS Sans Serif", 8 ;
 68:        │     │      FLOAT ;
 69:        │     │      NOCLOSE ;
 70:        │     │      MINIMIZE ;
 71:        │     │      SYSTEM
 72:        │     │   MOVE WINDOW ctrlerr CENTER
 73:        │     └─ENDIF
 74:        │       ACTIVATE WINDOW  ctrlerr
 75:        │       ?CHR(7)
 76:        │       ? 'Anote el n mero y tipo de error y comunφquese
con su soporte tΘcnico'
 77:        │       ?
 78:        │       ? 'N mero de error: ' + LTRIM(STR(merror))
 79:        │       ? 'Mensaje de error: ' + mens
 80:        │       ? 'Iφnea de cΣdigo con error: ' + mens1
 81:        │       ? 'N mero de lφnea de error: ' +
LTRIM(STR(mn mlφn))
 82:        │       ? 'Programa con error: ' + mprog
 83:        │       ?
 84:        │       ? 'Programa reiniciarß automßticamente'
 85:        │       ?CHR(7)
 86:        │       STOR 1 TO mcont
 87:        │       @ 2,13 GET mcont;
 88:        │          PICTURE "@*h \<Aceptar" ;
 89:        │          SIZE 1.15 FONT 'R',10 STYLE "BI"
 90:        │       READ MODAL
 91:        ├──────CANCEL
 92:        │       DO OFF
 93:        │       DO SET
 94:        └─ENDCASE
 95:      ────RETURN
 96:
 97:
 98:          ****************
 99:
*!*************************************************************************
********
100:  *!
101:  *!        Procedure: ACERCA
```

```
102:  *!
103:  *!        Called by: EMPRESA.MPR
104:  *!
105:
*!*************************************************************************
********
106:      PROCEDURE acerca
107:          ****************
108:          DEFINE WIND LOGO FROM 0.0 TO 22.36 SYSTEM MDI TITLE
'PRESENTACION'
109:          ACTI WIND LOGO
110:          MOVE WIND LOGO CENTER
111:          *@0.0 SAY pathbmps+'EMPRESA.bmp' BITMAP ISOM SIZE 35,35
112:          @0.0 SAY 'EMPRESA.bmp' BITMAP ISOM SIZE 35,35
113:          STOR '' TO mcont
114:          @ 12,12 GET mcont:
115:             PICTURE "@* \<Aceptar" :
116:             SIZE 1.2,20 FONT 'R',10 STYLE "BI"
117:          READ MODAL
118:          RELE WIND LOGO
119: <========RETURN
120:
121:          **************
122:
*!*************************************************************************
********
123:  *!
124:  *!        Procedure: TAG
125:  *!
126:  *!        Called by: DGP_CARG.SPR
127:  *!                 : PERSONAL.PRG
128:  *!
129:
*!*************************************************************************
********
130:      PROCEDURE TAG
131:          **************
132:          FOR i = 1 TO 256
133:             IF EMPTY(TAG(i))
134:                i = i - 1
135:                DIMENSION vartag(i)
136:             <=========EXIT
137:             ELSE
138:                vartag(i) = TAG(i)
139:             ENDIF
140:          ENDFOR
141: <========RETURN
142:
143:
*!*************************************************************************
********
144:  *!
145:  *!        Procedure: ESTILOS
146:  *!
147:  *!        Called by: PERSONAL.PRG
148:  *!
149:
*!*************************************************************************
```

```
********
  150:      PROCEDURE estilos
  151:           STOR ' FONT  "N",12 style "b" 'TO fsdesc && styl and font
of say coment
  152:           *stor ' FONT "N",10 STYLE "B"' to fscampC && styl and
font of say CHARACTER fields
  153:           STOR '  ' TO fscampc && styl and font of say CHARACTER
fields
  154:           *stor ' FONT "N",10 STYLE "B"' to fscampN && styl and
font of say NUMERIC fields
  155:           STOR ' ' TO fscampn && styl and font of say NUMERIC
fields
  156:           STOR '  SIZE 1.5,22 FONT "R",10 STYLE "BI" ' TO
fsbotones
  157:           STOR '  SIZE 1.5,12 FONT "R",10 STYLE "BI" ' TO
fsboton1
  158:
  159: <=======RETURN
  160:
  161:
*!*************************************************************************
********
  162:  *!
  163:  *!       Procedure: BMS
  164:  *!
  165:
*!*************************************************************************
********
  166:      PROCEDURE bms
  167:           STOR '' TO pathbmps
  168:           @ 1,1  SAY pathbmps+'Top.bmp' BITMAP ISOM SIZE 3,3
  169:           @ 4,1  SAY pathbmps+'Bottom.bmp' BITMAP ISOM SIZE 3,3
  170:           @ 7,1 SAY pathbmps+'Exchange.bmp' BITMAP ISOM SIZE 3,3
  171:           @ 10,1 SAY pathbmps+'Exit.bmp' BITMAP ISOM SIZE 3,3
  172: <=======RETURN
  173:
  174:
*!*************************************************************************
********
  175:  *!
  176:  *!       Procedure: BMS1
  177:  *!
  178:
*!*************************************************************************
********
  179:      PROCEDURE bms1
  180:           STOR '' TO pathbmps
  181:           @ 1,17  SAY pathbmps+'Memo.bmp' BITMAP ISOM SIZE 4,3
  182:           @ 5,17 SAY pathbmps+'snippets.bmp' BITMAP ISOM SIZE 4,3
  183:           @ 9,17 SAY pathbmps+'erase01.bmp' BITMAP ISOM SIZE 4,3
  184:           @ 13,17 SAY pathbmps+'Top.bmp' BITMAP ISOM SIZE 4,3
  185:           @ 17,17  SAY pathbmps+'Bottom.bmp' BITMAP ISOM SIZE 4,3
  186:           @ 21,17 SAY pathbmps+'Exchange.bmp' BITMAP ISOM SIZE 4,3
  187:           @ 25,17 SAY pathbmps+'Exit.bmp' BITMAP ISOM SIZE 4,3
  188: <=======RETURN
  189:
  190:
*!*************************************************************************
```

```
********
  191:  *!
  192:  *!       Procedure: TEXTO
  193:  *!
  194:  *!       Called by: DA_ASISG.PRG
  195:  *!                : DE_EGRQ1.PRG
  196:  *!                : DE_EGRQ2.PRG
  197:  *!                : DE_EGRMN.PRG
  198:  *!                : DE_IEJUB.PRG
  199:  *!
  200:
*!*************************************************************************
********
  201:      PROCEDURE texto
  202:          STOR ' FONT  "N",16 style "b"' TO fsdesc && styl and
font of say coment
  203:          @ 15,15 SAY "Creaci$n y Actualizaci$n de Departamentos
por Secciones"&fsdesc
  204:          @ 18,15 SAY "            Presione [ ENTER ] para
Empezar...      "&fsdesc
  205:          READ
  206:          @ 15,15 SAY "
                                                          "&fsdesc
  207:          @ 18,15 SAY "
                                                "&fsdesc
  208: <=======RETURN
  209:
  210:
*!*************************************************************************
********
  211:  *!
  212:  *!       Procedure: SALI
  213:  *!
  214:
*!*************************************************************************
********
  215:      PROCEDURE sali
  216:          PARA rf1,rco1
  217:          STOR '' TO pathbmps
  218:          @ rf1+4,rco1+2 SAY pathbmps+'Exit.bmp' BITMAP ISOM SIZE
4,4
  219: <=======RETURN
  220:
  221:
*!*************************************************************************
********
  222:  *!
  223:  *!       Procedure: LOGO
  224:  *!
  225:  *!       Called by: EMPRESA.MPR
  226:  *!
  227:
*!*************************************************************************
********
  228:      PROCEDURE LOGO
  229:          STOR '' TO pathbmps
  230:          DEFINE WIND hola FROM 0,0 TO 52,92 FILL FILE
pathbmps+'EEASA.bmp'
```

```
231:          ACTI WIND hola
232:          @15.25 SAY pathbmps+'EMPRESA.bmp' BITMAP ISOM SIZE 35.35
233:  <=======RETURN
234:
235:          **** ESCOJE EL SISTEMA UNO ****
236:
*!*************************************************************************
********
237:  *!
238:  *!       Procedure: ESC_SIST
239:  *!
240:  *!            Uses: SISTEMA.DBF              Alias: SST
241:  *!
242:  *!       CDX files: SISTEMA.CDX
243:  *!
244:
*!*************************************************************************
********
245:      PROCEDURE esc_sist
246:          SELE A
247:          USE sistema
248:          COUNT TO A
249:          DIME sis[a]
250:          GO TOP
251:          FOR i = 1 TO A
252:              STOR ciudad TO sis[i]
253:              IF NOT EOF()
254:                  SKIP
255:              ENDIF
256:          ENDFOR
257:          STOR sis[1] TO sist
258:          STOR 0 TO veiu,vtip
259:          DEFI WIND escoge FROM 08,40 TO 20,70  TITL 'Escoja el
 Sistema:' MDI SYSTEM
260:          ACTI WIND escoge
261:          @2,2 GET sist FROM sis PICT "@^" :
262:              DEFAULT 1 :
263:              COLOR SCHEME 10, 2
264:          READ
265:          DEAC WIND escoge
266:          LOCATE FOR ciudad=sist
267:          STOR cod_ciu TO veiu
268:          STOR tipo TO vtip
269:          USE
270:  <=======RETURN
271:          ***** escoje sistema filtrando *****
272:
*!*************************************************************************
********
273:  *!
274:  *!       Procedure: SISTEMA_F
275:  *!
276:  *!            Uses: SISTEMA.DBF              Alias: SST
277:  *!
278:  *!       CDX files: SISTEMA.CDX
279:  *!
280:
*!*************************************************************************
```

```
********
  281:      PROC sistema_f
  282:           SELE 24
  283:           USE sistema ALIAS sist
  284:           SET ORDER TO tipo_act
  285:           COUNT TO A
  286:           DIME sis[a]
  287:           GO TOP
  288:           FOR i = 1 TO A
  289:              STOR ciudad TO sis[i]
  290:              IF NOT EOF()
  291:                 SKIP
  292:              ENDIF
  293:           ENDF
  294:           STOR sis[1] TO msis
  295:           STOR 0 TO vciu.vtip
  296:           DEFI WIND escoge FROM 18,30 TO 32,70 MDI SYST TITLE
'Escoja el Sistema: '
  297:           ACTI WIND escoge
  298:           @2,2 GET msis FROM sis PICT "@^" ;
  299:              SIZE 1,20 ;
  300:              DEFAULT 1 ;
  301:              COLOR SCHEME 10, 2
  302:           READ
  303:           GO TOP
  304:           LOCA FOR ALLT(ciudad) = ALLT(msis)
  305:           STOR cod_ciu TO vciu
  306:           DEAC WIND escoge
  307:           USE
  308:           * with vciu,Vtip
  309:           RETURN
  310:
  311:           ************************************************
  312:           *  Programa de Funciones para el cálculo del rol *
  313:           *  Autor: Mercedes Vánez, Santiago Guevara       *
  314:           ************************************************
  315:
  316:           * -- Calculo de dias que tiene el mes --
  317:
*!*************************************************************************
********
  318:   *!
  319:   *!        Function: CAL_DIAS
  320:   *!
  321:   *!      Called by: CALCULO.FRG
  322:   *!
  323:   *!           Uses: MESES.ORG              Alias: MES
  324:   *!
  325:   *!      CDX files: MESES.CDX
  326:   *!
  327:
*!*************************************************************************
********
  328:      FUNCTION cal_dias
  329:           PARAMETER vmes
  330:           SELECT 25
  331:           USE meses.org
  332:           *VarMes  = Month(Date())
```

```
  varmes = vmes
  SELECT 25
  LOCATE FOR mes = varmes
  vdias = dias
  IF MOD(YEAR(DATE()),4) = 0 .AND. mes = 2
     vdias = 29
  ENDIF
  RETUR(vdias)

  *--- Calculo de sueldo ------

*!*************************************************************************
********
*!
*!        Function: CAL_SUEL
*!
*!*************************************************************************
********
FUNCTION cal_suel
     PARAMETERS sue, dia
     RETURN((sue/dia))

     *----- Salarios -------

*!*************************************************************************
********
*!
*!        Function: CAL_SALA
*!
*!*************************************************************************
********
FUNCTION cal_sala
     PARAMETERS vtip
     IF vtip = 1
          vr = dep.sueldo * (para.por_antsue/100)
     ELSE
          vr = dep.sueldo
     ENDIF
     RETURN(vr)

     *--- Calculo Subsidio Antiguedad---

*!*************************************************************************
********
*!
*!        Function: SUB_ANT
*!
*!        Called by: CALCULO.PRG
*!
*!*************************************************************************
********
FUNCTION sub_ant
     PARAMETERS nanios
     v1= (c.cc_sapsb*(c.cc_sasbt/100))+((c.cc_sapsv/100) *
b.salminvit)
```

```
  378:          v1 = ROUN(v1,0)
  379:          v1 = v1 * nanios
  380:          RETURN(v1)
  381:
  382:          *---- Calculo Subsidio TRansporte ---
  383:
*!*************************************************************************
********
  384:  *!
  385:  *!        Function: SUB_TRANS
  386:  *!
  387:  *!       Called by: CALCULO.PRG
  388:  *!
  389:
*!*************************************************************************
********
  390:      FUNCTION sub_trans
  391:          PARAMETERS vt
  392:          vt1 = b.ot_trans * vt * para.jor_subtra
  393:          RETURN(vt1)
  394:
  395:          *------Subsidio Refrigerio -------
  396:
*!*************************************************************************
********
  397:  *!
  398:  *!        Function: SUB_REF
  399:  *!
  400:  *!       Called by: CALCULO.PRG
  401:  *!
  402:
*!*************************************************************************
********
  403:      FUNCTION sub_ref
  404:          PARAMETERS j1,j2
  405:          R = (j1*c.cc_refju)+(j2*c.cc_refid)
  406:          RETURN(R)
  407:
  408:          *------ Base Inmonible------
  409:
*!*************************************************************************
********
  410:  *!
  411:  *!        Function: BASEIMP
  412:  *!
  413:  *!       Called by: IMP_RENTA()        (function in
GRAFICOS.PRG)
  414:  *!
  415:
*!*************************************************************************
********
  416:      FUNCTION baseimp
  417:          PARAMETER vmes
  418:          *mmes = month(date())
  419:          mmes = vmes
  420:          STOR 0 TO actu1,actu2, acum
  421:          SELE rol
  422:          IF vtip = 2
```

```
  423:          SET FILTER TO rol.cod_rol = veodrol1
  424:          LOCATE FOR dgp.cod_per = rol.cod_per
  425:        IF FOUND()
  426:            actu1 = rol.sub_famil + rol.sub_energ +
rol.bonf_comis
  427:        ENDIF
  428:      ENDIF
  429:      SET FILT TO rol.cod_rol = veodrol
  430:      LOCATE FOR dgp.cod_per = rol.cod_per
  431:    IF vtip = 3
  432:        actu1 = rol.sub_famil + rol.sub_energ +
rol.bonf_comis
  433:        *             ?dgp.cod_per
  434:        *             ? 'actu1',Rol.Sub_Famil,Rol.sub_energ,
Rol.Bonf_comis
  435:        *             ?actu1
  436:    ENDIF
  437:      actu2 = rol.sueldo + rol.otros_imp + rol.vac_ant +
rol.vac_act +;
  438:          rol.hextras +  rol.sub_antig +   rol.comp_salcc +
rol.sub_refr;
  439:          - rol.apor_iess + rol.iess_100
  440:
  441:      *?'actu2', Rol.sueldo , Rol.otros_imp , Rol.Vac_ant,
Rol.Vac_act ,;
  442:      rol.hextras ,  rol.sub_antig ,   rol.comp_salcc ,
rol.sub_refr;
  443:      , rol.apor_iess , rol.iess_100
  444:      *?actu2
  445:      acum = acr.ac_sueldo  +  acr.ac_hextras + acr.ac_subfam
+ acr.ac_subant + acr.ac_subref + ;
  446:      acr.ac_bcomis - acr.ac_apiess +  acr.ac_100iess +
acr.ac_subener + acr.ac_ayumatr +;
  447:      acr.ac_oting + acr.ac_gastos + acr.ac_compcc
  448:      *?'acum' , Acr.ac_sueldo  ,  Acr.ac_Hextras ,
Acr.ac_subfam , Acr.ac_subant , Acr.ac_subref , ;
  449:      acr.ac_bcomis , acr.ac_apiess ,  acr.ac_100iess ,
acr.ac_subener , acr.ac_ayumatr ,;
  450:      acr.ac_oting , acr.ac_gastos , acr.ac_compcc
  451:      *unit
  452:      imponi = acum + actu1+ actu2
  453:      RETURN(imponi)
  454:
  455:      *------- Base Imponible para aportes y fondos de reserva
****
  456:
*!*************************************************************************
********
  457:  *!
  458:  *!        Function: BASEIESS
  459:  *!
  460:  *!      Called by: FUN_RESE()          (function in
GRAFICOS.PRG)
  461:  *!                : CALCULO.PRG
  462:  *!
  463:
*!*************************************************************************
********
```

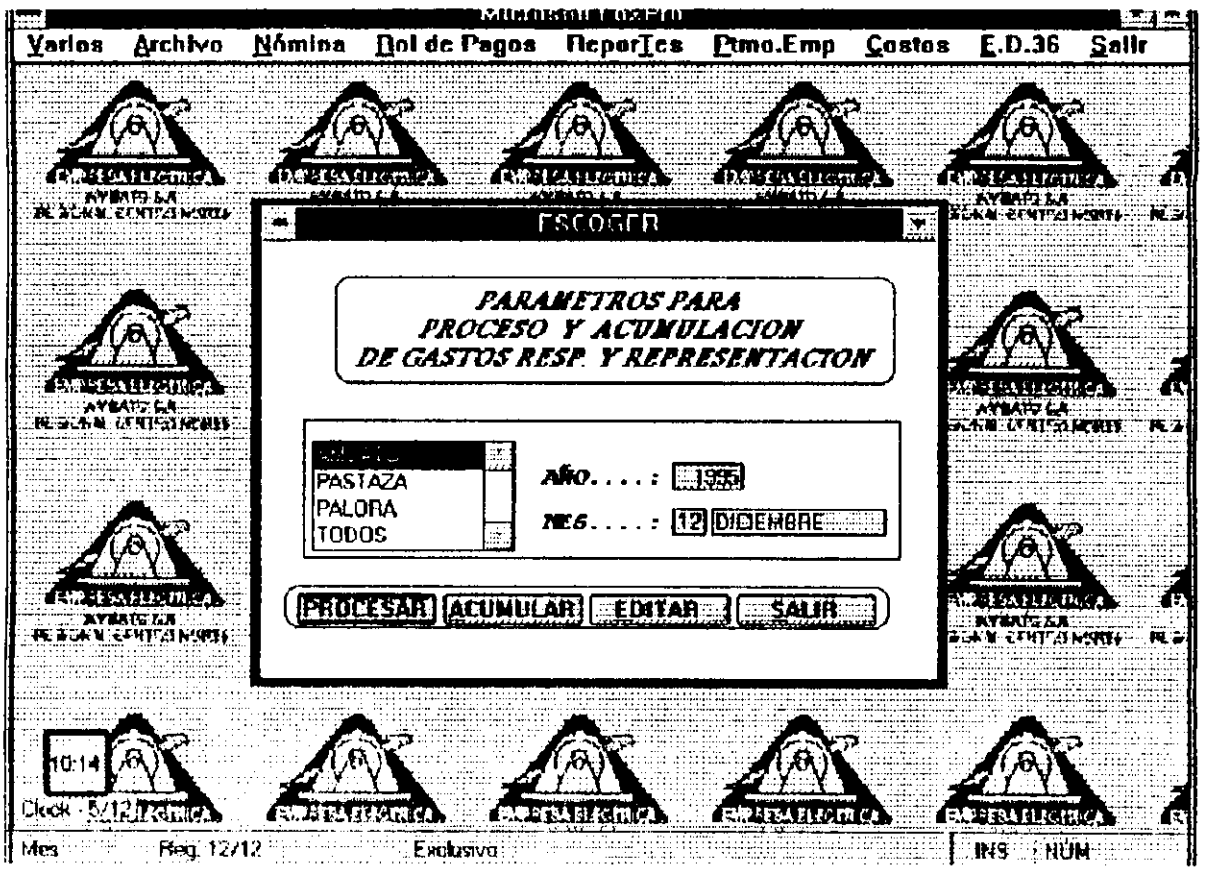
```
  464:      FUNCTION baseiess
  465:          *Impone = Rol.sueldo + Rol.Sub_Antig + Rol.hextras +:
  466:          rol.otros_imp + rol.vac_act + rol.vac_ant
  467:          ** Rol.Materni75
  468:
  469:          *Impone = Rol.sueldo + Rol.Sub_Antig + Rol.hextras +:
  470:          rol.otros_imp + rol.vac_act + rol.vac_ant +:
  471:          IIF(dgp.tiempo_ser >= 12, rol.sub_refr, 0)
  472:
  473:          impone = rol.sueldo + rol.sub_antig + rol.hextras +:
  474:          rol.otros_imp + rol.vac_act + rol.vac_ant +:
  475:          rol.sub_refr
  476:
  477:          RETURN(impone)
  478:
  479:
  480:          ***-------- base imponible para decimo tercer sueldo
********
  481:
*!*************************************************************************
********
  482:  *!
  483:  *!        Function: BASETERCER
  484:  *!
  485:  *!       Called by: CALCULO.PRG
  486:  *!
  487:
*!*************************************************************************
********
  488:      FUNCTION basetercer
  489:          malias = ALIAS()
  490:          reg_act = RECNO()
  491:          STOR rol.cod_per TO cp
  492:          SELE rol
  493:          reg_ant = RECNO()
  494:          SET FILT TO
  495:          LOCATE FOR cod_per = cp AND cod_rol = vcodrol1
  496:          STOR bonf_comis TO mbonf_comis
  497:          SET FILTER TO cod_rol = vcodrol
  498:          GO reg_ant
  499:          SELE &malias
  500:          GO reg_act
  501:          impo13 = ((rol.sueldo + rol.sub_antig + rol.sub_ant75  +
rol.hextras + rol.otros_imp +:
  502:              rol.iess_100 + dgp.gast_rep + dgp.gast_res +
rol.otros_ini + mbonf_comis+:
  503:              rol.comp_salec))
  504:          RETURN(impo13)
  505:
  506:          *------ Compensacion Salarial ---
  507:
*!*************************************************************************
********
  508:  *!
  509:  *!        Function: C_S1
  510:  *!
  511:  *!       Called by: CALCULO.PRG
  512:  *!
```

```
513:
*!*************************************************************************
********
514:      FUNCTION c_s1
515:          comsa = IIF(dgp.sueldo > b.ct_topecv.:
516:             b.ct_valtope,b.ct_boncv)
517:          RETURN(ROUND(comsa*dgp.tiempo,0))
518:          *****
519:
*!*************************************************************************
********
520:  *!
521:  *!        Function: C_S2
522:  *!
523:  *!       Called by: CALCULO.PRG
524:  *!
525:
*!*************************************************************************
********
526:      FUNCTION c_s2
527:          xcs = (IIF(dgp.tiempo_ser >= para.n_anios_cs.
c.cc_boncv, 0))
528:          RETURN(ROUND(xcs*dgp.tiempo,0))
529:
530:          *------ Bonificacion Complementaria -----
531:
*!*************************************************************************
********
532:  *!
533:  *!        Function: BONI_COMPLE
534:  *!
535:  *!       Called by: CALCULO.PRG
536:  *!
537:
*!*************************************************************************
********
538:      FUNCTION boni_comple
539:          PARAMETER vmes
540:          SELECT 25
541:          LOCATE FOR mes = vmes
542:          *Month(Date())
543:          boni = IIF((mes = para.ex_bncomp1 .OR. mes =
para.ex_bncomp2);
544:             .0 ,b.ct_bcomp)
545:          *Return(Boni)
546:          RETURN(ROUND(boni*dgp.tiempo,0))
547:
548:          *-- ----- Decimo Sexto ------
549:
*!*************************************************************************
********
550:  *!
551:  *!        Function: DEC_SEX5555
552:  *!
553:
*!*************************************************************************
********
554:      FUNCTION dec_sex5555
```

```
  555:          sexto= IIF((dgp.sueldo > 150000 AND dgp.sueldo <
600001),dgp.sueldo / 8, b.salminvit)
  556:          sexto = ROUND(sexto,0)
  557:          RETURN(ROUND(sexto*dgp.tiempo,0))
  558:
  559:
  560:
*!*************************************************************************
********
  561:  *!
  562:  *!        Function: DEC_SEX
  563:  *!
  564:  *!       Called by: CALCULO.PRG
  565:  *!
  566:
*!*************************************************************************
********
  567:      FUNCTION dec_sex
  568:          dec = (para.par_xvi_1*(b.salminvit)/para.par_xvi_2)
  569:          dec1 = dec * para.par_xvi_3
  570:          IF dgp.sueldo <=  dec1
  571:             sexto = dec
  572:          ELSE
  573:             IF dgp.sueldo <=  (b.salminvit * para.par_xvi_3)
  574:                sexto = dgp.sueldo / para.par_xvi_3
  575:             ELSE
  576:                sexto = b.salminvit
  577:             ENDIF
  578:          ENDIF
  579:          sexto = ROUND(sexto,0)
  580:          RETURN(ROUND(sexto*dgp.tiempo,0))
  581:
  582:          *------ Fondos de Reserva -------
  583:
*!*************************************************************************
********
  584:  *!
  585:  *!        Function: FON_RESE
  586:  *!
  587:  *!       Called by: CALCULO.PRG
  588:  *!
  589:  *!           Calls: BASEIESS()          (function in
GRAFICOS.PRG)
  590:  *!
  591:
*!*************************************************************************
********
  592:      FUNCTION fon_rese
  593:          *  Fon = Iif(Dgp.TIEMPO_SER + iif((day(Dgp.fec_ing) < 11
and month(Dgp.fec_ing) = 8),1,0) > PARA.N_ANIOS_FR, Baseiess()/12,0)
  594:          fon = (baseiess()/12)
  595:          RETURN(fon)
  596:
  597:          *------ Bonificaciones Varias -----
  598:
*!*************************************************************************
********
  599:  *!
```

```
600:  *!        Function: PROVISION
601:  *!
602:  *!       Called by: CALCULO.PRG
603:  *!
604:
*!*************************************************************************
********
605:      FUNCTION provision
606:          PARAMETERS F
607:          R = F/12
608:  <=======RETURN (R)
609:
610:          *----- Impuesto a la renta ----*
611:
*!*************************************************************************
********
612:  *!
613:  *!        Function: IMP_RENTA
614:  *!
615:  *!       Called by: CALCULO.PRG
616:  *!
617:  *!           Calls: BASEIMP()          (function in
GRAFICOS.PRG)
618:  *!
619:
*!*************************************************************************
********
620:      FUNCTION imp_renta
621:          m = 12
622:          x = ((baseimp(vmes)/vmes) * m)
623:          SELE 20
624:          GO TOP
625:          SCAN
626:              IF x >= bas1 .AND. x < exe1
627:                  res = vfb1+ ROUN( ( (x-bas1) * (pexe1/100) ),2 )
628:                  GO BOTT
629:              ELSE
630:                  res = 0
631:              ENDIF
632:          ENDSCAN
633:          res = res - acre.ac_imprent
634:          res = IIF(res < 0, 0, (res / (12 - vmes+1)))
635:          SELE A
636:          RETURN(res)
637:
638:          *    horas extras -------*
639:          *--- 25 %
640:
*!*************************************************************************
********
641:  *!
642:  *!        Function: H_EXT_AME
643:  *!
644:  *!       Called by: CALCULO.PRG
645:  *!
646:
*!*************************************************************************
********
```

```
 647:      FUNCTION h_ext_ame
 648:          PARA vhe, ndd, indice
 649:          *a =  round(((Vhe/30)*(30-ndd)),0)
 650:          *b =  round(((Vhe/30)*(ndd)* indice),0)
 651:          A =  ROUND(((vhe/ndias)*(ndias-ndd)),0)
 652:          B =  ROUND(((vhe/ndias)*(ndd)* indice),0)
 653:          RETURN(A+B)
 654:          *//// 75 %
 655:
*!*************************************************************************
********
 656:  *!
 657:  *!        Function: H_EXT_AME75
 658:  *!
 659:  *!       Called by: CALCULO.PRG
 660:  *!
 661:
*!*************************************************************************
********
 662:      FUNCTION h_ext_ame75
 663:          PARA vhe, ndd, indice
 664:          *b =  round(((Vhe/30)*(ndd)* indice),0)
 665:          B =  ROUND(((vhe/ndias)*(ndd)* indice),0)
 666:          RETURN(B)
 667:
 668:
 669:
 670:
 671:  *:  EOF: GRAFICOS.act
```


Varios Archivo Nómina Rol de Pagos ReporTes Pmo.Emp Costos E.D.36 Salir
ESCOGER
PARAMETROS PARA
PROCESO Y ACUMULACION
DE GASTOS RESP. Y REPRESENTACION
PASTAZA
PALORA
TODOS
Año. . . . : 1995
MES. . . . : 12 DICIEMBRE
PROCESAR
ACUMULAR
EDITAR
SALIR
10:14
Clock - 5/12
Mes
Reg. 12/12
Exclusivo
INS
NUM

```
    1:
*:*************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\CTRL_GTO.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 04/03/1995 at 21:59:50
    8:  *:
    9:  *:  Procs & Fncts: _R891D6I5V()
   10:  *:                : _R891D6IA4()
   11:  *:                : _R891D6IEI()
   12:  *:                : VALIDAR()
   13:  *:
   14:  *:          Set by: EMPRESA.MPR
   15:  *:
   16:  *:           Calls: _R891D6I5V()        (function in
CTRL_GTO.SPR)
   17:  *:                : _R891D6IA4()        (function in
CTRL_GTO.SPR)
   18:  *:                : _R891D6IEI()        (function in
CTRL_GTO.SPR)
   19:  *:
   20:  *:       Documented 19:14:55
FoxDoc version 3.00a
   21:
*:*************************************************************
**************
   22:      *
************************************************************
   23:      *        *
   24:      *        * 26/02/1995            CTRL_GTO.SPR
 22:56:42
   25:      *        *
   26:      *
************************************************************
   27:      *        *
   28:      *        * Nombre del autor
   29:      *        *
   30:      *        * Copyright (c) 1995 Nombre de la empresa
   31:      *        * Direcci¥n
   32:      *        * Ciudad.      C.P.
   33:      *        * Pa¥s
   34:      *        *
   35:      *        * Description:
   36:      *        * Este programa lo ha generado automáticamente
GENSCRN.
   37:      *        *
   38:      *
************************************************************
   39:
   40:
   41:      *
************************************************************
   42:      *        *
   43:      *        *  CTRL_GTO/Windows C¥digo de configuraci¥n -
```

```
SECCION 1
   44:      *        *
   45:      *
************************************************************
   46:      *
   47:
   48:      #REGION 1
   49:      **********
   50:      REGIONAL opc_scr, opc_rol, mmes, vmes, manio, mrol,
mcod_ciu, mfiltro, vciu
   51:      STOR ' ' TO opc_scr, mmes,opc_rol, mfiltro
   52:      STOR 1 TO mrol,mcod_ciu, vciu
   53:      STOR YEAR(DATE()) TO manio
   54:      STOR MONTH(DATE()) TO vmes
   55:
   56:      **********
   57:
   58:      #REGION 0
   59:      REGIONAL m.currarea, m.talkstat, m.compstat
   60:
   61:    ┌─IF SET("TALK") = "ON"
   62:    │    SET TALK OFF
   63:    │    m.talkstat = "ON"
   64:    ├─ELSE
   65:    │    m.talkstat = "OFF"
   66:    └─ENDIF
   67:      m.compstat = SET("COMPATIBLE")
   68:      SET COMPATIBLE FOXPLUS
   69:
   70:      m.rborder = SET("READBORDER")
   71:      SET readborder ON
   72:
   73:      m.currarea = SELECT()
   74:
   75:
   76:      *
************************************************************
   77:      *        *
   78:      *        *  CTRL_GTO Windows Bases de datos, Yndices,
relaciones
   79:      *        *
   80:      *
************************************************************
   81:      *
   82:
   83:    ┌─IF USED("acr")
   84:    │    SELECT acr
   85:    │    SET ORDER TO TAG "cod_per"
   86:    ├─ELSE
   87:    │    SELECT 0
   88:    │    USE (LOCFILE("dbfs\ac_imprt.dbf","DBF","Buscar
ac_imprt")) ;
   89:    │          AGAIN ALIAS acr :
   90:    │          ORDER TAG "cod_per"
   91:    └─ENDIF
   92:
   93:    ┌─IF USED("mes")
   94:    │    SELECT mes
```

```
   95:         SET ORDER TO 0
   96:      ELSE
   97:         SELECT 0
   98:         USE (LOCFILE("dbfs\meses.org","DBF","Buscar
meses?"));
   99:             AGAIN ALIAS mes :
  100:             ORDER 0
  101:      ENDIF
  102:
  103:      IF USED("sst")
  104:         SELECT sst
  105:         SET ORDER TO TAG "todos_act"
  106:      ELSE
  107:         SELECT 0
  108:         USE (LOCFILE("dbfs\sistema.dbf","DBF","Buscar
sistema?"));
  109:             AGAIN ALIAS sst :
  110:             ORDER TAG "todos_act"
  111:      ENDIF
  112:
  113:      IF USED("ctr")
  114:         SELECT ctr
  115:         SET ORDER TO 0
  116:      ELSE
  117:         SELECT 0
  118:         USE (LOCFILE("dbfs\ctrl_rol.dbf","DBF","Buscar
ctrl_rol?"));
  119:             AGAIN ALIAS ctr :
  120:             ORDER 0
  121:      ENDIF
  122:
  123:      IF USED("gto")
  124:         SELECT gto
  125:         SET ORDER TO 0
  126:      ELSE
  127:         SELECT 0
  128:         USE (LOCFILE("dbfs\gastos.dbf","DBF","Buscar
gastos?"));
  129:             AGAIN ALIAS gto :
  130:             ORDER 0
  131:      ENDIF
  132:
  133:      SELECT sst
  134:
  135:      SELECT gto
  136:      SELECT gto
  137:
  138:      SET RELATION OFF INTO acr
  139:      SET RELATION TO codigo INTO acr ADDITIVE
  140:
  141:      SELECT sst
  142:
  143:      *
************************************************************
  144:      *        *
  145:      *        *              Windows Definiciones de ventana
  146:      *        *
  147:      *
```

```
************************************************************
  148:      *
  149:
  150:     IF NOT WEXIST("ctrl_par") ;
  151:           OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.PJX" ;
  152:           OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.SCX" ;
  153:           OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.MNX" ;
  154:           OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.PRG" ;
  155:           OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.FRX" ;
  156:           OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.QPR"
  157:        DEFINE WINDOW ctrl_par ;
  158:           AT  0.000, 0.000  ;
  159:           SIZE 18.846,72.800 ;
  160:           TITLE "ESCOGER" ;
  161:           FONT "MS Sans Serif", 8 ;
  162:           FLOAT ;
  163:           NOCLOSE ;
  164:           MINIMIZE ;
  165:           SYSTEM
  166:        MOVE WINDOW ctrl_par CENTER
  167:     ENDIF
  168:
  169:
  170:      *
************************************************************
  171:      *        *
  172:      *        *  CTRL_GTO/Windows Código de configuración -
SECCION 2
  173:      *        *
  174:      *
************************************************************
  175:      *
  176:
  177:      #REGION 1
  178:      **********
  179:      SELE mes
  180:      LOCATE FOR mes = vmes
  181:      STOR nom_mes TO mmes
  182:
  183:      #REGION 1
  184:      DEFINE POPUP _r891d6hwp :
  185:         PROMPT FIELD sst.ciudad :
  186:         SCROLL
  187:
  188:      *
************************************************************
  189:      *        *
  190:      *        *        CTRL_GTO/Windows Distribución de
pantalla
  191:      *        *
  192:      *
************************************************************
  193:      *
  194:
  195:      #REGION 1
  196:     IF WVISIBLE("ctrl_par")
  197:        ACTIVATE WINDOW ctrl_par SAME
  198:     ELSE
```

```
199:        ACTIVATE WINDOW ctrl_par NOSHOW
200:     ENDIF
201:     @ 11.538,30.400 SAY "MES....:" :
202:        FONT "Courier New", 10 :
203:        STYLE "BIT"
204:     @ 9.615,30.400 SAY "ADO....:" :
205:        FONT "Courier New", 10 :
206:        STYLE "BIT"
207:     @ 14.923,2.600 TO 16.769,67.800 :
208:        PEN 1, 8 :
209:        STYLE "16"
210:     @ 2.000,8.400 SAY "PARAMETROS PARA " + CHR(13) + :
211:        "PROCESO  Y  ACUMULACION " + CHR(13) + :
212:        "DE GASTOS RESP. Y REPRESENTACION" :
213:        SIZE 3.000,41.429, 0.000 :
214:        PICTURE "@I" :
215:        FONT "MS Serif", 10 :
216:        STYLE "BIT"
217:     @ 1.538,8.000 TO 6.230,68.400 :
218:        PEN 1, 8 :
219:        STYLE "16"
220:     @ 7.769,4.600 TO 13.769,69.000 :
221:        PEN 1, 8
222:     @ 8.692,5.600 GET opc_rol :
223:        PICTURE "@&N" :
224:        POPUP _r891d6hwp :
225:        SIZE 4.615,21.400 :
226:        DEFAULT " " :
227:        FONT "MS Sans Serif", 8 :
228:        STYLE "" :
229:        VALID _r891d6i5v()
230:     @ 9.692,44.800 GET manio :
231:        SIZE 1.000,6.600 :
232:        DEFAULT " " :
233:        FONT "MS Sans Serif", 8 :
234:        PICTURE "9999" :
235:        VALID manio > 1990 AND manio < 2999 :
236:        ERROR 'ADO INVALIDO '
237:     @ 11.615,44.800 GET vmes :
238:        SIZE 1.000,2.600 :
239:        DEFAULT " " :
240:        FONT "MS Sans Serif", 8 :
241:        PICTURE "99" :
242:        VALID _r891d6i94() :
243:        ERROR 'MES INVALIDO (1..12)'
244:     @ 11.615,49.000 GET mes.nom_mes :
245:        SIZE 1.000,17.200 :
246:        DEFAULT " " :
247:        FONT "MS Sans Serif", 8 :
248:        PICTURE "@K" :
249:        WHEN .F.
250:     @ 15.308,4.000 GET opc_scr :
251:        PICTURE "@*HN
|<PROCESAR;|<ACUMULAR;|<EDITAR;|<SALIR" :
252:        SIZE 1.308,12.500,0.667 :
253:        DEFAULT 1 :
254:        FONT "MS Sans Serif", 8 :
255:        STYLE "B" :
```

```
256:          VALID _r891d6iei()
257:
258:      IF NOT WVISIBLE("ctrl_par")
259:          ACTIVATE WINDOW ctrl_par
260:      ENDIF
261:
262:        READ CYCLE ;
263:          VALID .F.
264:
265:        RELEASE WINDOW ctrl_par
266:
267:        *
************************************************************
268:        *         *
269:        *         *              Windows Cerrando bases de datos
270:        *         *
271:        *
************************************************************
272:        *
273:
274:      IF USED("acr")
275:          SELECT acr
276:          USE
277:      ENDIF
278:
279:      IF USED("mes")
280:          SELECT mes
281:          USE
282:      ENDIF
283:
284:      IF USED("sst")
285:          SELECT sst
286:          USE
287:      ENDIF
288:
289:      IF USED("ctr")
290:          SELECT ctr
291:          USE
292:      ENDIF
293:
294:      IF USED("gtc")
295:          SELECT gto
296:          USE
297:      ENDIF
298:
299:        SELECT (m.currarea)
300:
301:        RELEASE POPUPS _r891d6hwp
302:
303:        #REGION 0
304:
305:        SET readborder &rborder
306:
307:      IF m.talkstat = "ON"
308:          SET TALK ON
309:      ENDIF
310:      IF m.compstat = "ON"
311:          SET COMPATIBLE ON
```

```
  312:     ENDIF
  313:
  314:
  315:     *
************************************************************
  316:     *        *
  317:     *        *            CTRL_GTO/Windows Código de
limpieza
  318:     *        *
  319:     *
************************************************************
  320:     *
  321:
  322:     #REGION 1
  323:     ******************
  324:
  325:
  326:     *
************************************************************
  327:     *        *
  328:     *        * CTRL_GTO/Windows Procedimientos y funciones
de soporte
  329:     *        *
  330:     *
************************************************************
  331:     *
  332:
  333:     #REGION 1
  334:
*!*************************************************************************
**************
  335:  *!
  336:  *!       Procedure: PROCESAR
  337:  *!
  338:  *!           Calls: MEDIDOR.PRG
  339:  *!                : CALCULO.PRG
  340:  *!                : VALIDAR()          (function in
CTRL_GTO.SPR)
  341:  *!                : ABRIR_ARCH         (procedure in
PROC_DMO.SPR)
  342:  *!                : GN_MPL             (procedure in
PROC_DMO.SPR)
  343:  *!                : GN_PDMO            (procedure in
PROC_DMO.SPR)
  344:  *!                : GN_CTA_RUB         (procedure in
PROC_DMO.SPR)
  345:  *!                : GN_ASI_CON         (procedure in
PROC_DMO.SPR)
  346:  *!
  347:  *!            Uses: SOBRETIE.DBF
  348:  *!                : ?MAESTRI                  Alias: CP
  349:  *!                : DSCTOSVA.DBF              Alias: DVAR
  350:  *!                : PERSONAL.DBF              Alias: DGP
  351:  *!                : LISTBANC.DBF              Alias: BNC
  352:  *!                : TOT_ROL.DBF               Alias: TROL
  353:  *!                : PTMO_EMP.DBF
  354:  *!                : CTAS_CPE.DBF
  355:  *!                : TMP_ACPE.DBF              Alias: TMPCPE
```

```
  356:  *!                  : TMP_PE2.DBF
  357:  *!                  : ROL_IBS.DBF
  358:  *!                  : ROL_IBST.DBF
  359:  *!                  : DBF.DBF
  360:  *!
  361:  *!          Indexes: CODIGO                    (tag in
PUESTOS.CDX)
  362:  *!                  : CODIGO.CDX
  363:  *!                  : &TMP_MPL..CDX
  364:  *!
  365:  *!        CDX files: DSCTOSVA.CDX
  366:  *!                  : PERSONAL.CDX
  367:  *!                  : LISTBANC.CDX
  368:  *!                  : PTMO_EMP.CDX
  369:  *!                  : ROL_IBS.CDX
  370:  *!
  371:
*!*************************************************************************
**************
  372:      PROCEDURE procesar
  373:          ******************
  374:          IF validar(1)
  375:              USE dsctosva ORDER var_dsc ALIAS dscv IN 0
  376:              SELE dscv
  377:              SET FILTER TO clave = 'GAST'
  378:              USE personal ORDER cod_per ALIAS dgp IN 0
  379:              USE listbanc ALIAS lbc IN 0
  380:              SELE lbc
  381:              DELE ALL FOR &mfiltro lbc.clave = 'GAST' AND
lbc.anio = manio AND lbc.mes = vmes
  382:              PACK
  383:              SELE dgp
  384:              *SET RELATION TO CATEGOR INTO ESC
  385:              SET RELATION TO cod_per INTO dscv ADDITIVE
  386:              * SCAN FOR &MFILTRO (dgp.GAST_REP +
dgp.GAST_RES) > 0 AND ESTADO = 1
  387:              SCAN FOR (dgp.gast_rep + dgp.gast_res) > 0 AND
estado = 1
  388:                  SELE lbc
  389:                  APPEN BLANK
  390:                  REPLACE lbc.anio      WITH manio.;
  391:                      lbc.mes       WITH vmes.;
  392:                      lbc.clave     WITH 'GAST'.;
  393:                      lbc.sistema   WITH dgp.num_ciu.;
  394:                      lbc.codigo    WITH dgp.cod_per.;
  395:                      lbc.nombres   WITH dgp.nombre.;
  396:                      lbc.tipo      WITH dgp.tipo_cts.;
  397:                      lbc.nu_cuent  WITH dgp.num_cts.;
  398:                      lbc.tota_r    WITH
(dgp.gast_res+dgp.gast_rep)-dscv.otr_dscts1
  399:                  SELE gto
  400:                  APPEN BLANK
  401:                  REPLACE gto.anio          WITH manio.;
  402:                      gto.mes           WITH vmes.;
  403:                      gto.clave         WITH 'GAST'.;
  404:                      gto.sistema       WITH dgp.num_ciu.;
  405:                      gto.codigo        WITH dgp.cod_per.;
  406:                      gto.gtos_res      WITH dgp.gast_res.;
```

```
 407:          ||        gto.gtos_rep     WITH dgp.gast_rep.;
 408:          ||        gto.ingreso      WITH
(dgp.gast_res+dgp.gast_rep),;
 409:          ||        gto.otr_dsctsl   WITH dscv.otr_dsctsl
 410:          ||     SELE dgp
 411:          ||     *** ACUMULA LOS GASTOS PARA EL IMPUESTO A LA
RENTA ****
 412:          ||     *   SELE PERS
 413:          ||     *   REPLACE PERS.AC_GASTOS WITH
(PERS.AC_GASTOS+ (PERS.GAST_REP+PERS.GAST_RES))
 414:          ||ENDSCAN
 415:          | SELE dgp
 416:          | USE
 417:          | SELE lbc
 418:          | USE
 419:          | SELE dscv
 420:          | USE
 421:          |ENDIF
 422: <========RETURN
 423:
 424:        ******************
 425:
*!*************************************************************************
**************
 426:  *!
 427:  *!       Procedure: ACUMULAR
 428:  *!
 429:  *!            Calls: VALIDAR()          (function in
CTRL_GTO.SPR)
 430:  *!
 431:
*!*************************************************************************
**************
 432:     PROCEDURE acumular
 433:        ******************
 434:        IF validar(2)
 435:          SCAN
 436:             REPLACE acr.ac_gastos  WITH acr.ac_gastos +
gto.ingreso,;
 437:                     acr.ac_gtosres WITH acr.ac_gtosres +
gto.gtos_res,;
 438:                     acr.ac_gtosrep WITH acr.ac_gtosrep +
gto.gtos_rep,;
 439:                     gto.cierre WITH .T.
 440:          ENDSCAN
 441:        ENDIF
 442: <======RETURN
 443:        ****************
 444:
*!*************************************************************************
**************
 445:  *!
 446:  *!       Procedure: EDITAR
 447:  *!
 448:  *!       Called by: CREAR              (procedure in
AACT_SST.SPR)
 449:  *!
 450:  *!            Calls: DEAC_KEYS         (procedure in
```

```
AACT_SST.SPR)
  451:  *!                  : CABEZA              (procedure in
PERSONAL.PRG)
  452:  *!                  : INGRESO             (procedure in
ACT_PTMO.SPR)
  453:  *!                  : CREA_TMP            (procedure in
MO_PDLAB.SPR)
  454:  *!                  : MBROW               (procedure in
MO_PDLAB.SPR)
  455:  *!                  : REF_SCR             (procedure in
MO_PDLAB.SPR)
  456:  *!                  : MENSAJES            (procedure in
MO_PDLAB.SPR)
  457:  *!
  458:
*!*********************************************************************
**************
  459:      PROCEDURE editar
  460:          ****************
  461:          _CUROBJ= OBJNUM(opc_rol)
  462: <=======RETURN
  463:          ****************
  464:
*!*********************************************************************
**************
  465:  *!
  466:  *!       Procedure: SALIR
  467:  *!
  468:  *!       Called by: _RAA1732TG()       (function in
AT_ESCLS.SPR)
  469:  *!                : DA_ASISG.PRG
  470:  *!                : DE_EGR01.PRG
  471:  *!                : DE_EGR02.PRG
  472:  *!                : DE_EGRMN.PRG
  473:  *!                : DE_TEJUB.PRG
  474:  *!
  475:  *!           Calls: DEMO_KEYS           (procedure in
AACT_SST.SPR)
  476:  *!
  477:
*!*********************************************************************
**************
  478:      PROCEDURE salir
  479:          ***************
  480:          CLEAR READ
  481:  ========RETURN
  482:          ******************
  483:
*!*********************************************************************
**************
  484:  *!
  485:  *!        Function: VALIDAR
  486:  *!
  487:  *!       Called by: PROCESAR           (procedure in
VREG_ENG.SPR)
  488:  *!                : ACUMULAR           (procedure in
CTRL_OIG.SPR)
  489:  *!
```

```
490:
*!*************************************************************
**************
491:      FUNCTION  validar
492:           *****************
493:           PARAMETER mopc
494:           SELECT gto
495:           SET FILTER TO sistema = vciu AND anio = manio AND
mes = vmes AND !cierre
496:           GO TOP
497:           IF ((!EOF() AND mopc = 1) OR (EOF() AND mopc = 2))
498:              IF NOT WEXIST("Mproc")
499:                 DEFINE WINDOW mproc ;
500:                    FROM  0.000, 0.000  ;
501:                    TO 7,40;
502:                    PANEL;
503:                    FONT "MS Sans Serif", 8 ;
504:                    NOCLOSE
505:                 MOVE WINDOW mproc CENTER
506:              ENDIF
507:              ACTI WIND mproc
508:              ?CHR(7)
509:              @0,5 SAY 'No se puede realizar el proceso ya
que '
510:              @1,1 SAY 'no existe Informaci¾n o yß se realizó
el PROCESO'
511:              @2,15 SAY "Proceso Terminado ... "
512:              STOR '' TO macepta
513:              @ 3,17 GET macepta :
514:                 PICTURE "@* \<ACEPTAR" :
515:                 DEFAULT 1
516:              READ MODAL
517:              DEAC WIND mproc
518:              RETURN .F.
519:           ENDIF
520:           RETURN .T.
521:
522:
523:      *
*************************************************************
524:      *        *
525:      *        * _R991D615V            opc_rol VALID
526:      *        *
527:      *        * Function Origin:
528:      *        *
529:      *        * From Platform:        Windows
530:      *        * From Screen:          CTRL_GTO,
Record Number:   14
531:      *        * Variable:             opc_rol
532:      *        * Called By:            VALID Clause
533:      *        * Object Type:          List
534:      *        * Snippet Number:       1
535:      *        *
536:      *
*************************************************************
537:      *
538:
*!*************************************************************
```

```
**************
  539:  *!
  540:  *!        Function: _R891D615V
  541:  *!
  542:  *!       Called by: CTRL_GTO.SPR
  543:  *!
  544:
*!*****************************************************************
**************
  545:      FUNCTION _r891d615v      &&  opc_rol VALID
  546:          #REGION 1
  547:          STOR sst.cod_ciu TO mcod_ciu, vciu
  548:          STOR 'MCOD_CIU = GTO.SISTEMA AND '  TO mfiltro
  549:          IF sst.tipo = 9
  550:             STOR ' ' TO mfiltro
  551:          ENDIF
  552:
  553:          *
*************************************************************
  554:          *        *
  555:          *        * _R891D61A4            VMES VALID
  556:          *        *
  557:          *        * Function Origin:
  558:          *        *
  559:          *        * From Platform:        Windows
  560:          *        * From Screen:          CTRL_GTO.
Record Number:   16
  561:          *        * Variable:             VMES
  562:          *        * Called By:            VALID Clause
  563:          *        * Object Type:          Field
  564:          *        * Snippet Number:       2
  565:          *        *
  566:          *
 ***********************************************************
  567:          *
  568:
*!*****************************************************************
**************
  569:  *!
  570:  *!        Function: _R891D61A4
  571:  *!
  572:  *!       Called by: CTRL_GTO.SPR
  573:  *!
  574:
*!*****************************************************************
**************
  575:      FUNCTION _r891d61a4      &&  VMES VALID
  576:          #REGION 1
  577:          IF vmes > 0 AND vmes < 13
  578:             STOR .T. TO vret
  579:             SELE mes
  580:             LOCATE FOR vmes = mes
  581:             STOR nom_mes TO mmes
  582:             SHOW GET mes.nom_mes
  583:          ELSE
  584:             STOR .F. TO vret
  585:          ENDIF
  586:  <=======RETURN vret
```

```
  587:
  588:          *
*************************************************************
  589:          *        *
  590:          *        * _R891D6IEI            OPC_SCR VALID
  591:          *        *
  592:          *        * Function Origin:
  593:          *        *
  594:          *        * From Platform:        Windows
  595:          *        * From Screen:          CTRL_GTO,
Record Number:   18
  596:          *        * Variable:             OPC_SCR
  597:          *        * Called By:            VALID Clause
  598:          *        * Object Type:          Push Button
  599:          *        * Snippet Number:       3
  600:          *        *
  601:          *
*************************************************************
  602:          *
  603:
*!*****************************************************************************
**************
  604:  *!
  605:  *!        Function: _R891D6IEI
  606:  *!
  607:  *!       Called by: CTRL_GTO.SPR
  608:  *!
  609:  *!           Calls: &OPC_SCR.PRG
  610:  *!
  611:
*!*****************************************************************************
**************
  612:      FUNCTION _r891d6iei      && OPC_SCR VALID
  613:          #REGION 1
  614:          DO &opc_scr
  615:  *:  EOF: CTRL_GTO.ac1
```

```
    1:
*:*************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\CTRL_CIE.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 10/03/1995 at 21:16:22
    8:  *:
    9:  *:  Procs & Fncts: _R8M19LEJL()
   10:  *:                : _R8M19LEOB()
   11:  *:                : _R8M19LESO()
   12:  *:
   13:  *:          Set by: EMPRESA.MPR
   14:  *:
   15:  *:           Calls: _R8M19LEJL()        (function in
CTRL_CIE.SPR)
   16:  *:                : _R8M19LEOB()        (function in
CTRL_CIE.SPR)
   17:  *:                : _R8M19LESO()        (function in
CTRL_CIE.SPR)
   18:  *:
   19:  *:       Documented 19:14:56
FoxDoc version 3.00a
   20:
*:*************************************************************
**************
   21:      *
************************************************************
   22:      *        *
   23:      *        * 10/03/1995          CTRL_CIE.SPR
 21:16:19
   24:      *        *
   25:      *
************************************************************
   26:      *        *
   27:      *        * Nombre del autor
   28:      *        *
   29:      *        * Copyright (c) 1995 Nombre de la empresa
   30:      *        * Direcci¥n
   31:      *        * Ciudad,      C.P.
   32:      *        * Pa¡s
   33:      *        *
   34:      *        * Description:
   35:      *        * Este programa lo ha generado automßticamente
 SENSCRN.
   36:      *        *
   37:      *
************************************************************
   38:
   39:
   40:      *
************************************************************
   41:      *        *
   42:      *        *  CTRL_CIE/Windows C¢digo de configuraci¢n -
SECCION 1
```

```
   43:      *        *
   44:      *
**********************************************************
   45:      *
   46:
   47:      #REGION 1
   48:      **********
   49:      REGIONAL opc_scr, opc_rol, opc_qui, mmes, vmes, manio,
mquincena, mrol,:
   50:         mopc_qui,mfquin,vciu,vtip
   51:      STOR ' ' TO opc_scr, mmes,opc_rol, fquin
   52:      STOR 0 TO opc_qui,mopc_qui,vtip
   53:      STOR 1 TO mrol,vciu
   54:      STOR .F. TO mquincena
   55:      STOR YEAR(DATE()) TO manio
   56:      STOR MONTH(DATE()) TO vmes
   57:
   58:      **********
   59:
   60:      #REGION 0
   61:      REGIONAL m.currarea, m.talkstat, m.compstat
   62:
   63:     ┌IF SET("TALK") = "ON"
   64:     │   SET TALK OFF
   65:     │   m.talkstat = "ON"
   66:     ├ELSE
   67:     │   m.talkstat = "OFF"
   68:     └ENDIF
   69:      m.compstat = SET("COMPATIBLE")
   70:      SET COMPATIBLE FOXPLUS
   71:
   72:      m.rborder = SET("READBORDER")
   73:      SET readborder ON
   74:
   75:      m.currarea = SELECT()
   76:
   77:
   78:      *
**********************************************************
   79:      *        *
   80:      *        *  CTRL_CIE/Windows Bases de datos, Yndices,
relaciones
   81:      *        *
   82:      *
**********************************************************
   83:      *
   84:
   85:     ┌IF USED("rol")
   86:     │   SELECT rol
   87:     │   SET ORDER TO TAG "cod_per"
   88:     ├ELSE
   89:     │   SELECT 0
   90:     │   USE (LOCFILE("dbfs\rol.dbf","DBF","Buscar  rol?"));
   91:     │      AGAIN ALIAS rol :
   92:     │      ORDER TAG "cod_per"
   93:     └ENDIF
   94:
   95:     ┌IF USED("mes")
```

```
    96:     |       SELECT mes
    97:     |       SET ORDER TO 0
    98:     |   ELSE
    99:     |       SELECT 0
   100:     |       USE (LOCFILE("dbfs\meses.org","DBF","Buscar
meses?"));
   101:     |           AGAIN ALIAS mes :
   102:     |           ORDER 0
   103:     |   ENDIF
   104:
   105:     |   IF USED("sst")
   106:     |       SELECT sst
   107:     |       SET ORDER TO TAG "sist_cod"
   108:     |   ELSE
   109:     |       SELECT 0
   110:     |       USE (LOCFILE("dbfs\sistema.dbf","DBF","Buscar
sistema?"));
   111:     |           AGAIN ALIAS sst ;
   112:     |           ORDER TAG "sist_cod"
   113:     |   ENDIF
   114:
   115:     |   IF USED("dgp")
   116:     |       SELECT dgp
   117:     |       SET ORDER TO TAG "cod_per"
   118:     |   ELSE
   119:     |       SELECT 0
   120:     |       USE (LOCFILE("dbfs\personal.dbf","DBF","Buscar
personal?"));
   121:     |           AGAIN ALIAS dgp ;
   122:     |           ORDER TAG "cod_per"
   123:     |   ENDIF
   124:
   125:     |   IF USED("men")
   126:     |       SELECT men
   127:     |       SET ORDER TO 0
   128:     |   ELSE
   129:     |       SELECT 0
   130:     |       USE (LOCFILE("dbfs\mensual.dbf","DBF","Buscar
mensual?"));
   131:     |           AGAIN ALIAS men ;
   132:     |           ORDER 0
   133:     |   ENDIF
   134:
   135:     |   IF USED("ctr")
   136:     |       SELECT ctr
   137:     |       SET ORDER TO 0
   138:     |   ELSE
   139:     |       SELECT 0
   140:     |       USE (LOCFILE("dbfs\ctrl_rol.dbf","DBF","Buscar
ctrl_rol?"));
   141:     |           AGAIN ALIAS ctr ;
   142:     |           ORDER 0
   143:     |   ENDIF
   144:
   145:     |   IF USED("acr")
   146:     |       SELECT acr
   147:     |       SET ORDER TO TAG "cod_per"
   148:     |   ELSE
```

```
149:        SELECT 0
150:        USE (LOCFILE("dbfs\ac_imprt.dbf","DBF","Buscar
ac_imprt?"))
151:            AGAIN ALIAS acr :
152:            ORDER TAG "cod_per"
153:        ENDIF
154:
155:        SELECT rol
156:
157:        SET RELATION OFF INTO dgp
158:        SET RELATION TO cod_per INTO dgp ADDITIVE
159:
160:        SELECT rol
161:
162:        SET RELATION OFF INTO acr
163:        SET RELATION TO cod_per INTO acr ADDITIVE
164:
165:        SELECT rol
166:
167:        *
************************************************************
168:        *        *
169:        *        *              Windows Definiciones de ventana
170:        *        *
171:        *
************************************************************
172:        *
173:
174:        IF NOT WEXIST("ctrl_par") ;
175:            OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.PJX" ;
176:            OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.SCX" ;
177:            OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.MNX" ;
178:            OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.PRG" ;
179:            OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.FRX" ;
180:            OR UPPER(WTITLE("CTRL_PAR")) == "CTRL_PAR.QPR"
181:            DEFINE WINDOW ctrl_par ;
182:            AT  0.000, 0.000  ;
183:            SIZE 17.692,72.800 ;
184:            TITLE "ESCOGER" ;
185:            FONT "MS Sans Serif", 8 ;
186:            FLOAT ;
187:            NOCLOSE ;
188:            MINIMIZE ;
189:            SYSTEM
190:            MOVE WINDOW ctrl_par CENTER
191:        ENDIF
192:
193:
194:        *
************************************************************
195:        *        *
196:        *        *  CTRL_CIE.Windows Código de configuración -
SECCION 2
197:        *        *
198:        *
************************************************************
199:        *
200:
```

```
  201:      #REGION 1
  202:      **********
  203:      SELE mes
  204:      LOCATE FOR mes = vmes
  205:      STOR nom_mes TO mmes
  206:
  207:      #REGION 1
  208:      DEFINE POPUP _r8m19le8b :
  209:         PROMPT FIELD sst.ciudad ;
  210:         SCROLL
  211:
  212:      *
************************************************************
  213:      *        *
  214:      *        *         CTRL_CIE/Windows Distribución de
pantalla
  215:      *        *
  216:      *
************************************************************
  217:      *
  218:
  219:      #REGION 1
  220:     IF WVISIBLE("ctrl_par")
  221:        ACTIVATE WINDOW ctrl_par SAME
  222:     ELSE
  223:        ACTIVATE WINDOW ctrl_par NOSHOW
  224:     ENDIF
  225:      @ 10.769,30.400 SAY "MES....:" ;
  226:         FONT "Courier New", 10 ;
  227:         STYLE "BIT"
  228:      @ 8.846,30.400 SAY "ADO....:" ;
  229:         FONT "Courier New", 10 ;
  230:         STYLE "BIT"
  231:      @ 14.846,10.600 TO 16.923,59.600 ;
  232:         PEN 1, 8 ;
  233:         STYLE "16"
  234:      @ 1.385,14.800 SAY "PARAMETROS PARA   REALIZAR" +
CHR(13) + ;
  235:         "PROCESO DE CIERRE MENSUAL" + CHR(13) + ;
  236:         "DE ROL DE PAGOS" ;
  237:         SIZE 3.000,33.857, 0.000 ;
  238:         PICTURE "@I" ;
  239:         FONT "MS Serif", 10 ;
  240:         STYLE "BIT"
  241:      @ 0.923,8.000 TO 5.231,68.400 ;
  242:         PEN 1, 8 ;
  243:         STYLE "16"
  244:      @ 7.000,4.600 TO 13.000,69.000 ;
  245:         PEN 1, 8
  246:      @ 7.923,5.600 GET ope_rol ;
  247:         PICTURE "@!N" ;
  248:         POPUP _r8m19le8b ;
  249:         SIZE 4.615,21.400 ;
  250:         DEFAULT " " ;
  251:         FONT "MS Sans Serif", 8 ;
  252:         STYLE "" ;
  253:         VALID _r8m19lejl()
  254:      @ 8.923,44.800 GET manio ;
```

```
255:          SIZE 1.000,6.600 :
256:          DEFAULT " " ;
257:          FONT "MS Sans Serif", 8 ;
258:          PICTURE "9999" ;
259:          VALID manio > 1990 AND manio < 2999 ;
260:          ERROR 'ADO INVALIDO '
261:       @ 10.846,44.800 GET vmes ;
262:          SIZE 1.000,2.600 ;
263:          DEFAULT " " ;
264:          FONT "MS Sans Serif", 8 ;
265:          PICTURE "99" ;
266:          VALID _r8m19leob() ;
267:          ERROR 'MES INVALIDO (1..12)'
268:       @ 10.846,49.000 GET mes.nom_mes ;
269:          SIZE 1.000,17.800 ;
270:          DEFAULT " " ;
271:          FONT "MS Sans Serif", 8 ;
272:          PICTURE "@K" ;
273:          WHEN .F.
274:       @ 15.308,12.600 GET opc_scr ;
275:          PICTURE "@*HN \<PROCESAR;\<EDITAR;\<SALIR" ;
276:          SIZE 1.308,12.333,0.667 ;
277:          DEFAULT 1 ;
278:          FONT "MS Sans Serif", 8 ;
279:          STYLE "B" ;
280:          VALID _r8m19leso()
281:
282:       IF NOT WVISIBLE("ctrl_par")
283:          ACTIVATE WINDOW ctrl_par
284:       ENDIF
285:
286:       READ CYCLE ;
287:          VALID .F.
288:
289:       RELEASE WINDOW ctrl_par
290:
291:       *
************************************************************
292:       *        *
293:       *        *              Windows Cerrando bases de datos
294:       *        *
295:       *
************************************************************
296:       *
297:
298:       IF USED("col")
299:          SELECT col
300:          USE
301:       ENDIF
302:
303:       IF USED("mes")
304:          SELECT mes
305:          USE
306:       ENDIF
307:
308:       IF USED("sst")
309:          SELECT sst
310:          USE
```

```
311:     ENDIF
312:
313:     IF USED("dgp")
314:        SELECT dgp
315:        USE
316:     ENDIF
317:
318:     IF USED("men")
319:        SELECT men
320:        USE
321:     ENDIF
322:
323:     IF USED("ctr")
324:        SELECT ctr
325:        USE
326:     ENDIF
327:
328:     IF USED("acr")
329:        SELECT acr
330:        USE
331:     ENDIF
332:
333:        SELECT (m.currarea)
334:
335:        RELEASE POPUPS _r8m191e8b
336:
337:        #REGION 0
338:
339:        SET readborder &rborder
340:
341:     IF m.talkstat = "ON"
342:        SET TALK ON
343:     ENDIF
344:     IF m.compstat = "ON"
345:        SET COMPATIBLE ON
346:     ENDIF
347:
348:
349:        *
************************************************************
350:        *          *
351:        *          *            CTRL_CIE/Windows Código de
limpieza
352:        *          *
353:        *
************************************************************
354:        *
355:
356:        #REGION 1
357:        ********************
358:
359:
360:        *
************************************************************
361:        *          *
362:        *          * CTRL_CIE/Windows Procedimientos y funciones
de soporte
363:        *          *
```

```
364:      *
**********************************************************
365:      *
366:
367:      #REGION 1
368:
*!*************************************************************************
**************
369:  *!
370:  *!       Procedure: PROCESAR
371:  *!
372:  *!           Calls: MEDIDOR.PRG
373:  *!                : CALCULO.PRG
374:  *!                : VALIDAR()              (function in
CTRL_GTO.SPR)
375:  *!                : ABRIR_ARCH             (procedure in
PROC_DMO.SPR)
376:  *!                : GN_MPL                 (procedure in
PROC_DMO.SPR)
377:  *!                : GN_PDMO                (procedure in
PROC_DMO.SPR)
378:  *!                : GN_CTA_RUB             (procedure in
PROC_DMO.SPR)
379:  *!                : GN_ASI_CON             (procedure in
PROC_DMO.SPR)
380:  *!
381:  *!            Uses: SOBRETIE.DBF
382:  *!                : &MARCHI                     Alias: CP
383:  *!                : DSCTOSVA.DBF                Alias: DVAR
384:  *!                : PERSONAL.DBF                Alias: DGP
385:  *!                : LISTBANC.DBF                Alias: BNC
386:  *!                : TOT_ROL.DBF                 Alias: TROL
387:  *!                : PTMO_EMP.DBF
388:  *!                : CTAS_CPE.DBF
389:  *!                : TMP_ACPE.DBF                Alias: TMPPE
390:  *!                : TMP_PE2.DBF
391:  *!                : ROL_IBS.DBF
392:  *!                : ROL_IBST.DBF
393:  *!                : DBF.DBF
394:  *!
395:  *!         Indexes: CODIGO                    (tag in
PUESTOS.CDX)
396:  *!                : CODIGO.CDX
397:  *!                : &TMP_MPL..CDX
398:  *!
399:  *!       CDX files: DSCTOSVA.CDX
400:  *!                : PERSONAL.CDX
401:  *!                : LISTBANC.CDX
402:  *!                : PTMO_EMP.CDX
403:  *!                : ROL_IBS.CDX
404:  *!
405:
*!*************************************************************************
**************
406:      PROCEDURE procesar
407:          ******************
408:          STOR 2 TO ing_egr
409:          vcodrol =  ALLTRIM(STR(vciu)) +;
```

```
410:              ALLTRIM(IIF(vmes >= 10, STR(vmes), '0'+
ALLTRIM(STR(vmes)))) +;
411:              SUBSTR(ALLTRIM(STR(manio)),3,2) +;
412:              ALLTRIM(STR(vtip))
413:           vcodrol1 =  ALLTRIM(STR(vciu)) +;
414:              ALLTRIM(IIF(vmes >= 10, STR(vmes), '0'+
ALLTRIM(STR(vmes)))) +;
415:              SUBSTR(ALLTRIM(STR(manio)),3,2) +;
416:              ALLTRIM(STR(1))
417:           SELECT rol
418:           SET FILTER TO cod_rol = vcodrol
419:           GO TOP
420:           IF EOF()
421:              IF NOT WEXIST("Mproc")
422:                 DEFINE WINDOW mproc :
423:                    FROM  0.000, 0.000  :
424:                    TO 7.40:
425:                    PANEL:
426:                    FONT "MS Sans Serif", 8 :
427:                    NOCLOSE
428:                 MOVE WINDOW mproc CENTER
429:              ENDIF
430:              ACTI WIND mproc
431:              ?CHR(7)
432:              @0,5 SAY 'No existe Informaci¥n para proceso,
verifique par&metros'
433:              @1,15 SAY "Proceso Terminado ... "
434:              STOR '' TO macepta
435:              @ 2.5,17 GET macepta :
436:                 PICTURE "@* \<ACEPTAR" :
437:                 DEFAULT 1
438:              READ MODAL
439:              DEAC WIND mproc
440:              RETURN
441:           ENDIF
442:           SET FILTER TO cod_rol = vcodrol AND !cierre
443:           GO TOP
444:           IF EOF()
445:              IF NOT WEXIST("Mproc")
446:                 DEFINE WINDOW mproc :
447:                    FROM  0.000, 0.000  :
448:                    TO 7.40:
449:                    PANEL:
450:                    FONT "MS Sans Serif", 8 :
451:                    NOCLOSE
452:                 MOVE WINDOW mproc CENTER
453:              ENDIF
454:              ACTI WIND mproc
455:              ?CHR(7)
456:              @0,5 SAY 'Proceso de cierre mensual ya se
realiz¥'
457:              @1,15 SAY "No se puede continuar ... "
458:              STOR '' TO macepta
459:              @ 2.5,17 GET macepta :
460:                 PICTURE "@* \<ACEPTAR" :
461:                 DEFAULT 1
462:              READ MODAL
463:              DEAC WIND mproc
```

```
464:  <==========RETURN
465:        ENDIF
466:        IF NOT WEXIST("Mproc")
467:           DEFINE WINDOW mproc :
468:              FROM  0.000, 0.000  ;
469:              TO 7,40;
470:              PANEL;
471:              FONT "MS Sans Serif", 8 ;
472:              NOCLOSE
473:           MOVE WINDOW mproc CENTER
474:        ENDIF
475:        ACTI WIND mproc
476:        @0,5 SAY 'Realizando proceso de cierre '
477:        @1,15 SAY "Espere un momento por favor... "
478:
479:        STOR 'anio = manio and mes = vmes ' TO mfiltro
480:        STOR 'VAL(anio) = manio and VAL(mes) = vmes ' TO
rfiltro
481:        IF vciu # 9
482:           STOR mfiltro + ' and ciudad = vciu' TO mfiltro
483:           STOR rfiltro + ' and ciudad = vciu' TO rfiltro
484:        ENDIF
485:        SELE rol
486:        SET FILTER TO &rfiltro
487:        TOTAL ON cod_per TO tot_rol FOR &rfiltro
488:        USE tot_rol ALIAS trol IN 0
489:        SELE trol
490:        SET RELATION OFF INTO dgp
491:        SET RELATION TO cod_per INTO dgp ADDITIVE
492:        SET RELATION OFF INTO acr
493:        SET RELATION TO cod_per INTO acr ADDITIVE
494:        SCAN
495:           IF !EOF('dgp')
496:              REPLACE dgp.tiempo_ser WITH :
497:                 dgp.tiempo_ser + IIF(!EOF('trol'),1,0)
498:           ENDIF
499:           IF EOF('acr')
500:              SELE acr
501:              APPEN BLANK
502:              REPLACE acr.cod_per WITH trol.cod_per:
503:                 acr.num_ciu WITH
VAL(LEFT(trol.cod_per,1))
504:           ENDIF
505:           STOR 'acr.gi' + ALLTRIM(IIF(vmes <= 10,
STR(vmes), '0' + ALLTRIM(STR(vmes)))) TO mcampo
506:           ** valor ganado del mes
507:           REPLACE &mcampo WITH   trol.sueldo +
trol.hextras + :
508:                 trol.vac_ant + trol.vac_act + trol.otros_imp
+ trol.sub_antig+:
509:                 trol.sub_refr
510:           ** acumula otros valores para impuesto a la
renta
511:           REPLACE acr.ac_sueldo WITH  acr.ac_sueldo +
trol.sueldo:
512:              acr.ac_hextras WITH acr.ac_hextras +
trol.hextras:
513:              acr.vac_act WITH acr.vac_act + trol.vac_act:
```

```
  514:          ||        acr.vac_ant WITH acr.vac_ant + trol.vac_ant;
  515:          ||        acr.ac_subant WITH acr.ac_subant +
trol.sub_antig;
  516:          ||        acr.ac_subfam WITH acr.ac_subfam +
trol.sub_famil;
  517:          ||        acr.ac_apiess WITH acr.ac_apiess +
trol.apor_iess;
  518:          ||        acr.ac_subtran WITH acr.ac_subtran +
trol.sub_trans;
  519:          ||        acr.ac_100iess WITH acr.ac_100iess +
trol.iess_100;
  520:          ||        acr.ac_boncom  WITH acr.ac_boncom  +
trol.bonif_comp;
  521:          ||        acr.ac_subener WITH acr.ac_subener +
trol.sub_energ;
  522:          ||        acr.ac_bcomis  WITH acr.ac_bcomis  +
trol.bonf_comis;
  523:          ||        acr.ac_subref  WITH acr.ac_subref  +
trol.sub_refr;
  524:          ||        acr.ac_compct  WITH acr.ac_compct  +
trol.comp_sal;
  525:          ||        acr.ac_compcc  WITH acr.ac_compcc  +
trol.comp_salcc;
  526:          ||        acr.ac_imprent WITH acr.ac_imprent +
trol.imp_renta
  527:          ||     SELE trol
  528:          ||  ENDSCAN
  529:             USE
  530:             ERASE tot_rol.dbf
  531:             SELE men
  532:             REPLACE ALL procesado WITH .T. FOR &mfiltro
  533:             SELE rol
  534:             REPLACE ALL rol.cierre WITH .T. FOR &rfiltro
  535:             SELE ctr
  536:             REPLACE ALL control WITH 'C' FOR (VAL(ctr.anio) =
manio  AND VAL(ctr.mes) = vmes)
  537:             SELE rol
  538:             CLEAR
  539:             ?CHR(7)
  540:             @1,15 SAY "Proceso Terminado ... "
  541:             STOR '' TO macepta
  542:             @ 2.5,17 GET macepta :
  543:                PICTURE "@* !<ACEPTAR" :
  544:                DEFAULT 1
  545:             READ MODAL
  546:             RELEASE WIND mproc
  547:  ========RETURN
  548:
  549:             ****************
  550:
*!*************************************************************************
**************
  551:  *!
  552:  *!       Procedure: EDITAR
  553:  *!
  554:  *!       Called by: CREAR                (procedure in
AACT_SST.SPR)
  555:  *!
```

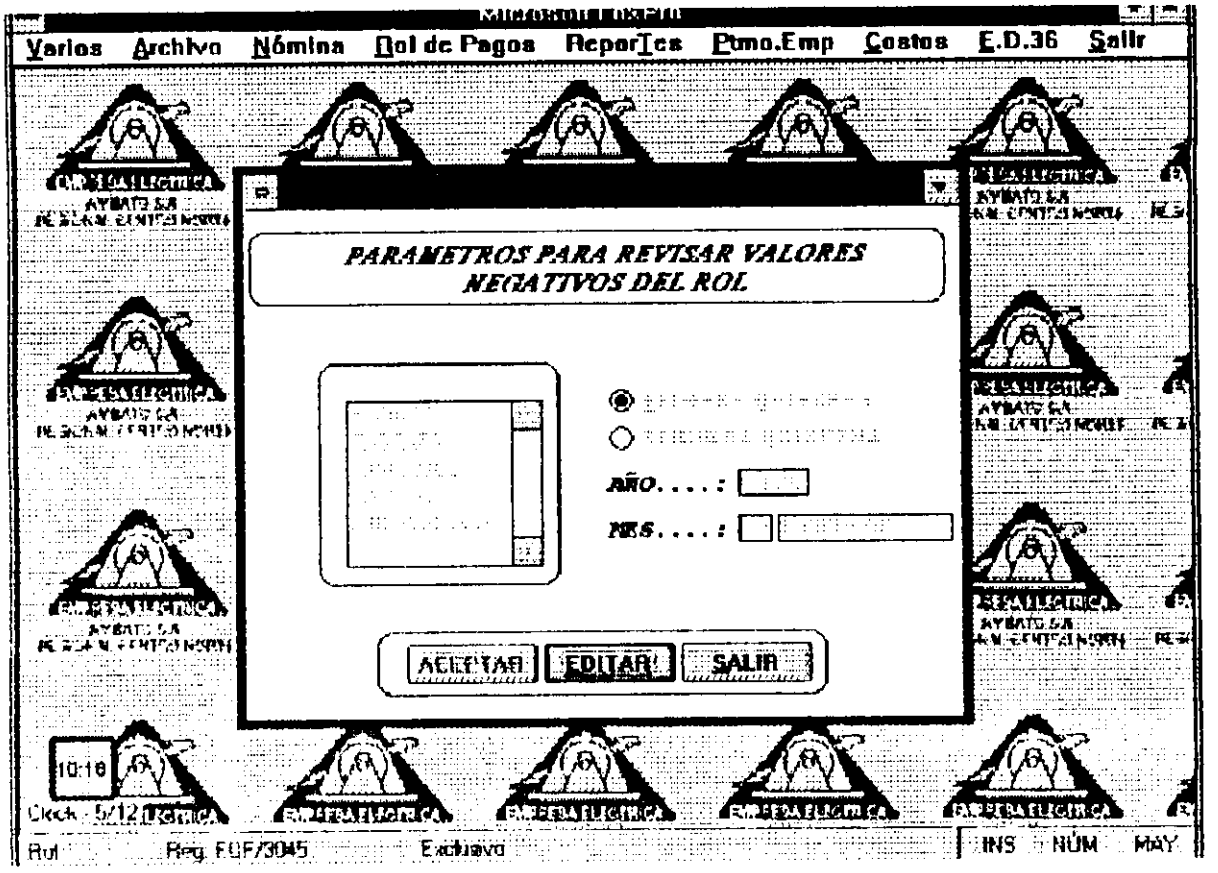
```
  556:  *!          Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
  557:  *!                 : CABEZA           (procedure in
PERSONAL.PRG)
  558:  *!                 : INGRESO          (procedure in
ACT_PTMO.SPR)
  559:  *!                 : CREA_TMP         (procedure in
MO_PDLAB.SPR)
  560:  *!                 : MBROW            (procedure in
MO_PDLAB.SPR)
  561:  *!                 : REF_SCR          (procedure in
MO_PDLAB.SPR)
  562:  *!                 : MENSAJES         (procedure in
MO_PDLAB.SPR)
  563:  *!
  564:
*!*************************************************************************
**************
  565:      PROCEDURE editar
  566:          ****************
  567:          _CUROBJ= OBJNUM(opc_rol)
  568:  <=======RETURN
  569:
  570:          ***************
  571:
*!*************************************************************************
**************
  572:  *!
  573:  *!       Procedure: SALIR
  574:  *!
  575:  *!       Called by: _RAA1732TG()       (function in
AT_ESCLS.SPR)
  576:  *!                : DA_ASISG.PRG
  577:  *!                : DE_EGRQ1.PRG
  578:  *!                : DE_EGRQ2.PRG
  579:  *!                : DE_EGRMN.PRG
  580:  *!                : DE_IEJUB.PRG
  581:  *!
  582:  *!          Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
  583:  *!
  584:
*!*************************************************************************
**************
  585:      PROCEDURE salir
  586:          ***************
  587:          CLEAR READ
  588:  <=======RETURN
  589:
  590:          *
*************************************************************
  591:          *       *
  592:          *       * _R8M19LEJL          opc_rol VALID
  593:          *       *
  594:          *       * Function Origin:
  595:          *       *
  596:          *       * From Platform:      Windows
  597:          *       * From Screen:        CTRL_CIE.
```

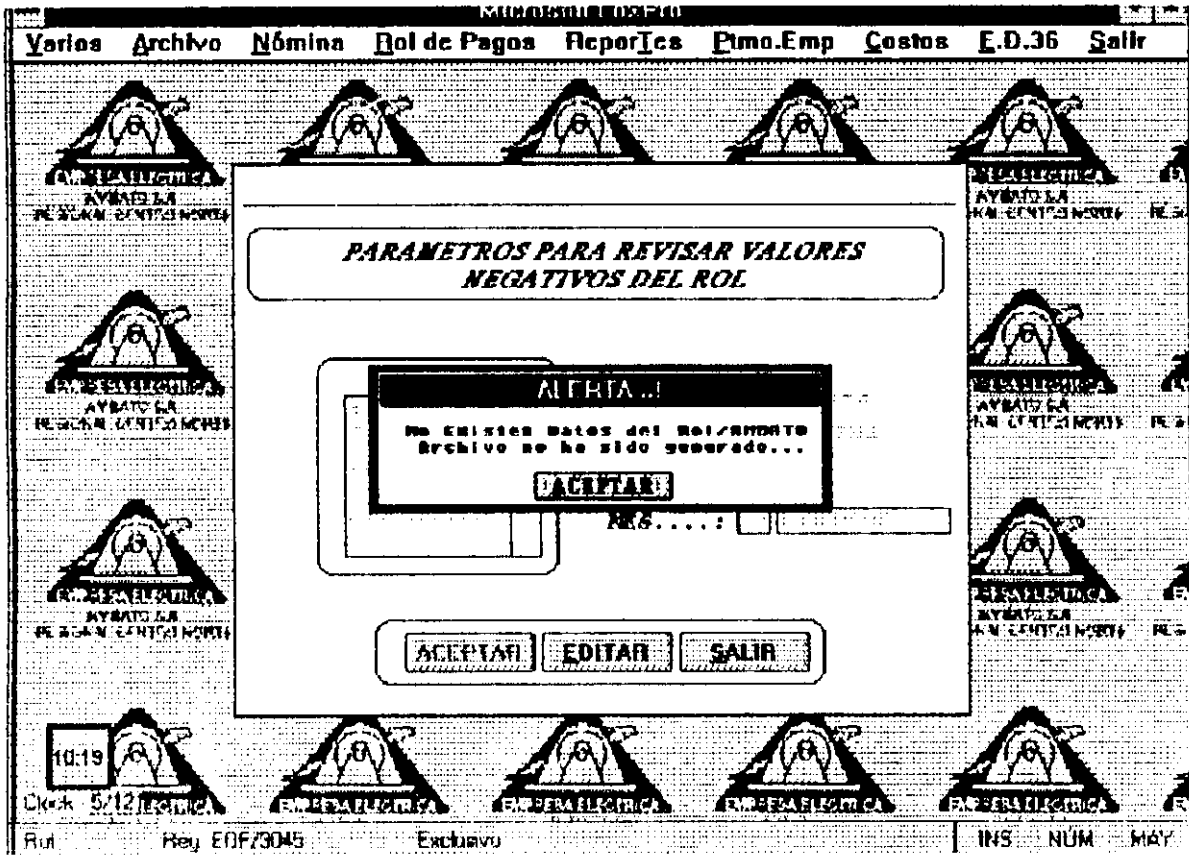
```
Record Number:   17
  598:          *       * Variable:             opc_rol
  599:          *       * Called By:            VALID Clause
  600:          *       * Object Type:          List
  601:          *       * Snippet Number:       1
  602:          *       *
  603:          *
*************************************************************
  604:          *
  605:
*!*************************************************************************
**************
  606:  *!
  607:  *!       Function: _R8M19LEJL
  608:  *!
  609:  *!      Called by: CTRL_CIE.SPR
  610:  *!
  611:
*!*************************************************************************
**************
  612:     FUNCTION _r8m19lejl     &&  opc_rol VALID
  613:         #REGION 1
  614:         STOR sst.cod_ciu TO vciu
  615:         STOR sst.tipo TO vtip
  616:       1=DO CASE
  617:       à====CASE sst.tipo = 0
  618:       |        STOR 2 TO vtip
  619:       à====CASE sst.tipo = 1
  620:       |        STOR 3 TO vtip
  621:       à====CASE sst.tipo = 2
  622:       |        STOR 4 TO vtip
  623:       E=ENDCASE
  624:
  625:
  626:          *
*************************************************************
  627:          *       *
  628:          *       * _R8M19LEOB            VMES VALID
  629:          *       *
  630:          *       * Function Origin:
  631:          *       *
  632:          *       * From Platform:        Windows
  633:          *       * From Screen:          CTRL_CIE.
Record Number:   18
  634:          *       * Variable:             VMES
  635:          *       * Called By:            VALID Clause
  636:          *       * Object Type:          Field
  637:          *       * Snippet Number:       2
  638:          *       *
  639:          *
*************************************************************
  640:          *
  641:
*!*************************************************************************
**************
  642:  *!
  643:  *!       Function: _R8M19LEOB
  644:  *!
```

```
645:  *!       Called by: CTRL_CIE.SPR
646:  *!
647:
*!*************************************************************
**************
648:      FUNCTION _r8m19leob      &&  VMES VALID
649:          #REGION 1
650:          IF vmes > 0 AND vmes < 13
651:              STOR .T. TO vret
652:              SELE mes
653:              LOCATE FOR vmes = mes
654:              STOR nom_mes TO mmes
655:              SHOW GET mes.nom_mes
656:          ELSE
657:              STOR .F. TO vret
658:          ENDIF
659:          RETURN vret
660:
661:          *
************************************************************
662:          *        *
663:          *        * _R8M19LESO            OPC_SCR VALID
664:          *        *
665:          *        * Function Origin:
666:          *        *
667:          *        * From Platform:        Windows
668:          *        * From Screen:          CTRL_CIE.
Record Number:   21
669:          *        * Variable:             OPC_SCR
670:          *        * Called By:            VALID Clause
671:          *        * Object Type:          Push Button
672:          *        * Snippet Number:       3
673:          *        *
674:          *
************************************************************
675:          *
676:
*!*************************************************************
**************
677:  *!
678:  *!       Function: _R8M19LESO
679:  *!
680:  *!      Called by: CTRL_CIE.SPR
681:  *!
682:  *!          Calls: &OPC_SCR.FRG
683:  *!
684:
*!*************************************************************
**************
685:      FUNCTION _r8m19lesc      &&  OPC_SCR VALID
686:          #REGION 1
687:          DO &opc_scr
688:  *:  EOF: CTRL_CIE.scl
```

```
  201:      *
  202:
  203:      #REGION 1
  204:      **********
  205:      SELE mes
  206:      LOCATE FOR mes = vmes
  207:      STOR nom_mes TO mmes
  208:      SELE sst
  209:      STOR sst.ciudad TO opc_rol
  210:
  211:
  212:      #REGION 1
  213:      DEFINE POPUP _r8f1b2nb8 :
  214:         PROMPT FIELD sst.ciudad :
  215:         SCROLL
  216:
  217:      *
************************************************************
  218:      *        *
  219:      *        *        CTRL_CRL/Windows Distribuci?n de
pantalla
  220:      *        *
  221:      *
************************************************************
  222:      *
  223:
  224:      #REGION 1
  225:     ┌IF WVISIBLE("pampdr")
  226:     │   ACTIVATE WINDOW pampdr SAME
  227:     └ELSE
  228:      │   ACTIVATE WINDOW pampdr NOSHOW
  229:      └ENDIF
  230:      @ 13.231,39.200 SAY "MES....:" :
  231:         FONT "Courier New", 10 :
  232:         STYLE "BIT"
  233:      @ 11.308,39.200 SAY "ADO....:" :
  234:         FONT "Courier New", 10 :
  235:         STYLE "BIT"
  236:      @ 6.692,8.000 TO 16.154,34.000 :
  237:         PEN 1, 8 :
  238:         STYLE "16"
  239:      @ 18.154,14.200 TO 21.000,62.600 :
  240:         PEN 1, 8 :
  241:         STYLE "16"
  242:      @ 7.692,39.400 GET opc_qui :
  243:         PICTURE "@*RVN \<PRIMERA QUINCENA;\<SEGUNDA
QUINCENA" :
  244:         SIZE 1.308,25.667,0.308 :
  245:         DEFAULT 1 :
  246:         FONT "MS Serif", 8 :
  247:         STYLE "BI" :
  248:         WHEN mquincena
  249:      @ 11.385,53.600 GET manio :
  250:         SIZE 1.000,6.600 :
  251:         DEFAULT " " :
  252:         FONT "MS Sans Serif", 8 :
  253:         PICTURE "9999" :
  254:         VALID manio > 1990 AND manio < 2999 :
```

```
255:          ERROR 'ADO INVALIDO ' ;
256:          DISABLE
257:       @ 13.308,53.600 GET vmes ;
258:          SIZE 1.000,2.600 ;
259:          DEFAULT " " ;
260:          FONT "MS Sans Serif", 8 ;
261:          PICTURE "99" ;
262:          VALID _r8f1b2nmg() ;
263:          ERROR 'MES INVALIDO (1..12)' ;
264:          DISABLE
265:       @ 18.692,17.400 GET opc_scr ;
266:          PICTURE "@*HN \<ACEPTAR;\<EDITAR;\<SALIR" ;
267:          SIZE 1.769,11.667,0.667 ;
268:          DEFAULT 1 ;
269:          FONT "MS Sans Serif", 8 ;
270:          STYLE "B" ;
271:          VALID _r8f1b2ngl() ;
272:          DISABLE
273:       @ 1.385,11.200 SAY "REGISTRAR DATOS DE DESCUENTOS" +
CHR(13) + ;
274:          "PARA PROCESAR ROLES DE :" ;
275:          SIZE 2.000,38.857, 0.000 ;
276:          PICTURE "@I" ;
277:          FONT "MS Serif", 10 ;
278:          STYLE "BIT"
279:       @ 0.923,0.600 TO 4.154,75.600 ;
280:          PEN 1, 8 ;
281:          STYLE "16"
282:       @ 13.308,57.800 GET mes.nom_mes ;
283:          SIZE 1.000,17.800 ;
284:          DEFAULT " " ;
285:          FONT "MS Sans Serif", 8 ;
286:          PICTURE "@K" ;
287:          WHEN .F.
288:       @ 8.308,11.000 GET opc_rol ;
289:          PICTURE "@&N" ;
290:          POPUP _r8f1b2nb8 ;
291:          SIZE 6.923,20.800 ;
292:          DEFAULT " " ;
293:          FONT "MS Sans Serif", 8 ;
294:          STYLE "" ;
295:          VALID _r8f1b2nvk()
296:
297:       IF NOT WVISIBLE("pampdr")
298:          ACTIVATE WINDOW pampdr
299:       ENDIF
300:
301:       READ CYCLE ;
302:          VALID salir
303:
304:       RELEASE WINDOW pampdr
305:
306:       *
*************************************************************
307:       *        *
308:       *        *              Windows Cerrando bases de datos
309:       *        *
310:       *
```


Varios Archivo Nómina Rol de Pagos Reportes Pmo.Emp Costos E.D.36 Salir
PARAMETROS PARA REVISAR VALORES NEGATIVOS DEL ROL
AÑO. . . . :
MES. . . . :
ACEPTAR
EDITAR
SALIR
10:18
Rol
Reg. FGF/3045
Exclusivo
INS
NÚM
MAY


Varios Archivo Nómina Rol de Pagos ReporTes Ptmo.Emp Costos E.D.36 Salir
PARAMETROS PARA REVISAR VALORES NEGATIVOS DEL ROL
ALERTA..!
No Existen Datos del Rol/ANUATO Archivo no ha sido generado...
ACEPTAR
ACEPTAR
EDITAR
SALIR
10:19
Rol
Rey EDF/3045
Exclusivo
INS
NUM
MAY

```
    1:
*:*************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\CTRL_VNG.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 16/10/1995 at 21:14:22
    8:  *:
    9:  *:  Procs & Fncts: _R4119IU28()
   10:  *:                : _R4119IU69()
   11:  *:                : _R4119IU9Z()
   12:  *:                : _R4119IUEM()
   13:  *:
   14:  *:          Set by: EMPRESA.MPR
   15:  *:
   16:  *:           Calls: _R4119IU28()        (function in
CTRL_VNG.SPR)
   17:  *:                : _R4119IU69()        (function in
CTRL_VNG.SPR)
   18:  *:                : _R4119IU9Z()        (function in
CTRL_VNG.SPR)
   19:  *:                : _R4119IUEM()        (function in
CTRL_VNG.SPR)
   20:  *:
   21:  *:       Documented 19:14:58
FoxDoc version 3.00a
   22:
*:*************************************************************
**************
   23:      *
************************************************************
   24:      *        *
   25:      *        * 10/16/1995            CTRL_VNG.SPR
 21:14:20
   26:      *        *
   27:      *
************************************************************
   28:      *        *
   29:      *        * Author's Name
   30:      *        *
   31:      *        * Copyright (c) 1995 Company Name
   32:      *        * Address
   33:      *        * City,     Zip
   34:      *        *
   35:      *        * Description:
   36:      *        * This program was automatically generated by
GENSCRN.
   37:      *        *
   38:      *
************************************************************
   39:
   40:
   41:      *
************************************************************
   42:      *        *
```

```
*       *          CTRL_VNG/Windows Setup Code - SECTION 1
*       *
*
************************************************************
*
#REGION 1
**********
REGIONAL opc_scr, opc_rol, opc_qui, mmes, vmes, manio, mquincena, mrol,;
   mopc_qui,mfquin,vtip, arrayitem
STOR ' ' TO opc_scr, mmes,opc_rol, fquin, arrayitem
STOR 0 TO opc_qui,vtip
STOR 1 TO mrol
STOR .F. TO mquincena
STOR YEAR(DATE()) TO manio
STOR MONTH(DATE()) TO vmes

**********

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
   SET TALK OFF
   m.talkstat = "ON"
ELSE
   m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.rborder = SET("READBORDER")
SET readborder ON

m.currarea = SELECT()


*
************************************************************
*       *
*       *      CTRL_VNG/Windows Databases, Indexes, Relations
*       *
*
************************************************************
*

IF USED("sst")
   SELECT sst
   SET ORDER TO TAG "tipo_act"
ELSE
   SELECT 0
   USE (LOCFILE("dbfs\sistema.dbf","DBF","Where is sistema?"));
      AGAIN ALIAS sst ;
      ORDER TAG "tipo_act"
```

```
 94:    ENDIF
 95:
 96:    IF USED("rol")
 97:       SELECT rol
 98:       SET ORDER TO TAG "nombre"
 99:    ELSE
100:       SELECT 0
101:       USE (LOCFILE("dbfs\rol.dbf","DBF","Where is
rol?"));
102:          AGAIN ALIAS rol :
103:          ORDER TAG "nombre"
104:    ENDIF
105:
106:    IF USED("mes")
107:       SELECT mes
108:       SET ORDER TO 0
109:    ELSE
110:       SELECT 0
111:       USE (LOCFILE("dbfs\meses.org","DBF","Where is
meses?"));
112:          AGAIN ALIAS mes :
113:          ORDER 0
114:    ENDIF
115:
116:
117:
118:    *
**********************************************************
119:    *        *
120:    *        *                Windows Window definitions
121:    *        *
122:    *
**********************************************************
123:    *
124:
125:    IF NOT WEXIST("pampdr") :
126:          OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.PJX" :
127:          OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.SCX" :
128:          OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.MNX" :
129:          OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.PRG" :
130:          OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.FRX" :
131:          OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.QPR"
132:       DEFINE WINDOW pampdr :
133:          AT  0.000, 0.000  :
134:          SIZE 21.923,77.000 :
135:          FONT "MS Sans Serif", 8 :
136:          FLOAT :
137:          NOCLOSE :
138:          MINIMIZE :
139:          SYSTEM
140:       MOVE WINDOW pampdr CENTER
141:    ENDIF
142:
143:
144:    *
**********************************************************
145:    *        *
146:    *        *          CTRL_VNG/Windows Setup Code -
```

```
SECTION 2
  147:      *        *
  148:      *
************************************************************
  149:      *
  150:
  151:      #REGION 1
  152:      **********
  153:      SELE mes
  154:      LOCATE FOR mes = vmes
  155:      STOR nom_mes TO mmes
  156:      SELE sst
  157:      STOR sst.ciudad TO opc_rol
  158:      SET ORDER TO sist_cod
  159:
  160:      #REGION 1
  161:      DEFINE POPUP _r4i19itwd ;
  162:         PROMPT FIELD sst.ciudad :
  163:         SCROLL
  164:
  165:      *
************************************************************
  166:      *        *
  167:      *        *              CTRL_VNG/Windows Screen Layout
  168:      *        *
  169:      *
************************************************************
  170:      *
  171:
  172:      #REGION 1
  173:     IF WVISIBLE("pampdr")
  174:        ACTIVATE WINDOW pampdr SAME
  175:     ELSE
  176:        ACTIVATE WINDOW pampdr NOSHOW
  177:     ENDIF
  178:      @ 13.231,39.200 SAY "MES....:"  ;
  179:         FONT "Courier New", 10 ;
  180:         STYLE "BIT"
  181:      @ 11.308,39.200 SAY "ADO....:"  ;
  182:         FONT "Courier New", 10 ;
  183:         STYLE "BIT"
  184:      @ 6.692,8.000 TO 16.154,34.000 ;
  185:         PEN 1, 8 ;
  186:         STYLE "16"
  187:      @ 18.154,14.200 TO 21.000,62.600 ;
  188:         PEN 1, 8 ;
  189:         STYLE "16"
  190:      @ 7.692,39.400 GET opc_qui ;
  191:         PICTURE "@*RVN \<PRIMERA QUINCENA:\<SEGUNDA
QUINCENA" ;
  192:         SIZE 1.308,25.667,0.308 ;
  193:         DEFAULT 1 ;
  194:         FONT "MS Serif", 8 ;
  195:         STYLE "BT" ;
  196:         WHEN mquincena ;
  197:         VALID _r4i19iu28()
  198:      @ 11.385,53.600 GET manio ;
  199:         SIZE 1.000,6.600 ;
```

```
200:          DEFAULT " " ;
201:          FONT "MS Sans Serif", 8 ;
202:          PICTURE "9999" ;
203:          VALID manio > 1990 AND manio < 2999 ;
204:          ERROR 'ADO INVALIDO ' ;
205:          DISABLE
206:       @ 13.308,53.600 GET vmes ;
207:          SIZE 1.000,2.600 ;
208:          DEFAULT " " ;
209:          FONT "MS Sans Serif", 8 ;
210:          PICTURE "99" ;
211:          VALID _r4i19iu69() ;
212:          ERROR 'MES INVALIDO (1..12)' ;
213:          DISABLE
214:       @ 18.692,17.400 GET opc_scr ;
215:          PICTURE "@*HN \<ACEPTAR;\<EDITAR;\<SALIR" ;
216:          SIZE 1.769,11.667,0.667 ;
217:          DEFAULT 1 ;
218:          FONT "MS Sans Serif", 8 ;
219:          STYLE "B" ;
220:          VALID _r4i19iu9z() ;
221:          DISABLE
222:       @ 1.385,11.200 SAY "REGISTRAR DATOS DE DESCUENTOS" +
CHR(13) + ;
223:          "PARA PROCESAR ROLES DE :"  ;
224:          SIZE 2.000,38.857, 0.000 ;
225:          PICTURE "@I" ;
226:          FONT "MS Serif", 10 ;
227:          STYLE "BIT"
228:       @ 0.923,0.600 TO 4.154,75.600 ;
229:          PEN 1, 8 ;
230:          STYLE "16"
231:       @ 13.308,57.800 GET mes.nom_mes ;
232:          SIZE 1.000,17.800 ;
233:          DEFAULT " " ;
234:          FONT "MS Sans Serif", 8 ;
235:          PICTURE "@K" ;
236:          WHEN .F.
237:       @ 8.308,11.000 GET opc_rol ;
238:          PICTURE "@^N" ;
239:          POPUP _r4i19itwd ;
240:          SIZE 6.923,20.800 ;
241:          DEFAULT " " ;
242:          FONT "MS Sans Serif", 8 ;
243:          STYLE "" ;
244:          VALID _r4i19iuem()
245:
246:       IF NOT WVISIBLE("pampdr")
247:          ACTIVATE WINDOW pampdr
248:       ENDIF
249:
250:       READ CYCLE ;
251:          DEACTIVATE .F.
252:
253:       RELEASE WINDOW pampdr
254:
255:       *
*************************************************************
```

```
256:      *        *
257:      *        *                    Windows Closing Databases
258:      *        *
259:      *
**************************************************************
260:      *
261:
262:      IF USED("sst")
263:         SELECT sst
264:         USE
265:      ENDIF
266:
267:      IF USED("rol")
268:         SELECT rol
269:         USE
270:      ENDIF
271:
272:      IF USED("mes")
273:         SELECT mes
274:         USE
275:      ENDIF
276:
277:        SELECT (m.currarea)
278:
279:        RELEASE POPUPS _r4i19itwd
280:
281:        #REGION 0
282:
283:        SET readborder &rborder
284:
285:      IF m.talkstat = "ON"
286:         SET TALK ON
287:      ENDIF
288:      IF m.compstat = "ON"
289:         SET COMPATIBLE ON
290:      ENDIF
291:
292:
293:      *
**************************************************************
294:      *        *
295:      *        *                CTRL_VNG/Windows Cleanup Code
296:      *        *
297:      *
**************************************************************
298:      *
299:
300:      #REGION 1
301:      ****************
302:
303:
304:      *
**************************************************************
305:      *        *
306:      *        *  CTRL_VNG/Windows Supporting Procedures and
Functions
307:      *        *
308:      *
```

```
*************************************************************
  309:      *
  310:
  311:      #REGION 1
  312:
*!*************************************************************************
**************
  313:   *!
  314:   *!       Procedure: ACEPTAR
  315:   *!
  316:   *!           Calls: VAL_PRO()          (function in
VREG_PTE.SPR)
  317:   *!                : PREPARAR           (procedure in
VREG_PTE.SPR)
  318:   *!                : DA_ASISG.PRG
  319:   *!                : DE_EGRQ1.PRG
  320:   *!                : DE_EGRQ2.PRG
  321:   *!                : DE_EGRMN.PRG
  322:   *!                : DE_IEJUB.PRG
  323:   *!                : EGR_VARI.PRG
  324:   *!                : ROL_Q1.SPR
  325:   *!                : ROL_Q2.SPR
  326:   *!                : ROL_MEN.SPR
  327:   *!                : ROL_JUB.SPR
  328:   *!                : ACT_PTMO.SPR
  329:   *!                : MO_PDLAB.SPR
  330:   *!
  331:   *!    Report Forms: VAL_NEGA.FRX
  332:   *!
  333:
*!*************************************************************************
**************
  334:      PROCEDURE aceptar
  335:          ******************
  336:          SHOW GETS DISABLE
  337:          SHOW GET opc_scr,2 ENABLE
  338:          SHOW GET opc_scr,3 ENABLE
  339:          SELECT sst
  340:          SEEK mrol
  341:          STOR ' ' TO tip
  342:          STOR .F. TO PRO"
  343:          SELE rol
  344:          SET FILTER TO
  345:          STOR 'Val(Rol.anio) = manio and Val(Rol.mes) =
Vmes :
  346:               AND rol.ciudad = sst.cod_ciu AND rol.tipo =
vtip' TO filtro
  347:          IF val_pro(filtro,1)
  348:             STOR 'Val(Rol.anio) = manio and Val(Rol.mes) =
Vmes :
  349:                  AND rol.ciudad = sst.cod_ciu AND rol.tipo =
vtip:
  350:                  AND total_reci < 0 ' to Filtro
  351:             IF !val_pro(filtro,2)
  352:                RETURN
  353:             ENDIF
  354:             IF NOT WEXIST('VARIOS')
  355:                DEFINE WINDOW varios FROM 0,0 TO 7,35 TITLE
```

```
'ALERTA ..!' SYSTEM
  356:          |  └─ENDIF
  357:          |    MOVE WIND varios CENTER
  358:          |    ACTI WIND varios
  359:          |    STOR 1 TO mcont
  360:          |    @ 2,8 GET mcont;
  361:          |       PICTURE "@*h |<Pantalla;|<Impresora" :
  362:          |       SIZE 1,10 FONT 'R',10 STYLE "BI"
  363:          |    READ MODAL
  364:          |    DEAC WIND varios
  365:          |  ┌─IF mcont = 1
  366:          |  |    BROW FIELD rol.cod_per, rol.nombres,
rol.cargo, ingresos=:
  367:          |  |         (rol.sueldo + rol.hextras +
rol.sub_antig +;
  368:          |  |         rol.otros_ini + rol.iess_100 +
rol.otros_imp +;
  369:          |  |         rol.vac_act + rol.vac_ant + rol.sub_refr
+;
  370:          |  |         rol.bonif_comp + rol.comp_sal +
rol.comp_salcc +;
  371:          |  |         rol.sub_trans + rol.sub_famil +;
  372:          |  |         rol.xvi_sueldo +  rol.sub_energ +
rol.bonf_comis),;
  373:          |  |         rol.p_empresa, rol.r_juzgado,
rol.t_menores,;
  374:          |  |         rol.p_energia, rol.coop_ahor,
rol.alm_labora,;
  375:          |  |         rol.otros_dsct, rol.total_reci;
  376:          |  |         FOR &filtro NOEDIT NOMENU
  377:          |  └─ELSE
  378:          |  |    REPO FORM val_nega FOR &filtro TO PRINT
NOCON NOEJECT
  379:          |  |    *         modi repo val_nega
  380:          |  └─ENDIF
  381:          └ ENDIF
  382: '═══════RETURN
  383:
  384:          ****************
  385:
*!*************************************************************************
**************
  386:  *!
  387:  *!        Function: VAL_PRO
  388:  *!
  389:  *!      Called by: ACEPTAR              (procedure in
ORDE_PTE.PRG)
  390:  *!
  391:
*!*************************************************************************
**************
  392:      FUNCTION val_pro
  393:          ****************
  394:          PARAMETER filtro,opc_f
  395:          malias = ALIAS()
  396:          SELE rol
  397:          vret = .T.
  398:          *     ?manio,vmes,sst.cod_ciu,vtip
```

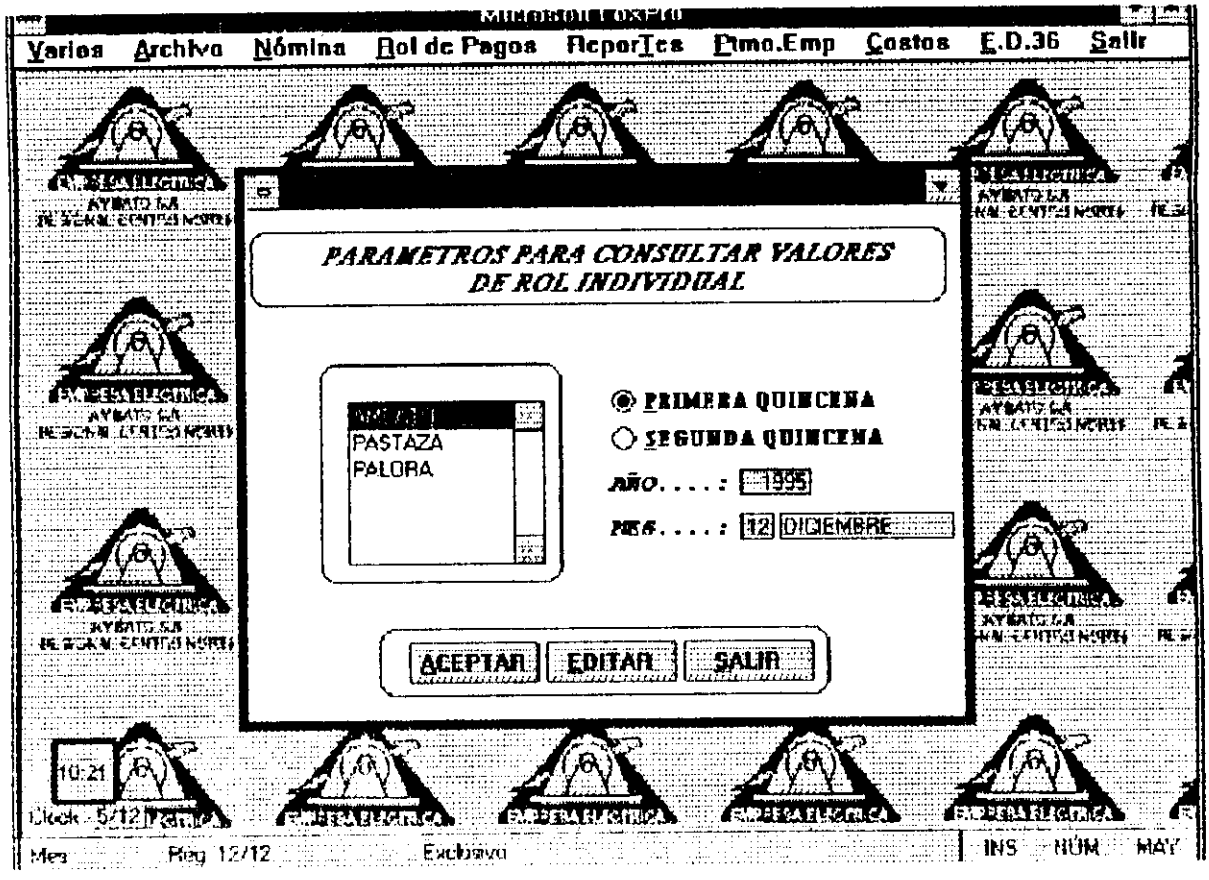
```
  399:          *     wait
  400:          *     Set filter to Val(Rol.anio) = manio and
Val(Rol.mes) = Vmes ;
  401:          AND rol.ciudad = sst.cod_ciu AND rol.tipo = vtip
  402:          SET FILTER TO &filtro
  403:          GO TOP
  404:          IF EOF()
  405:             IF NOT WEXIST('VARIOS')
  406:                DEFINE WINDOW varios FROM 0,0 TO 7,35 TITLE
'ALERTA ..!' SYSTEM
  407:             ENDIF
  408:             ACTIVATE WINDOW varios
  409:             MOVE WINDOW varios CENTER
  410:             IF opc_f = 1
  411:                @ 1,2 SAY 'No Existen Datos del
Rol/'+sst.ciudad
  412:                @ 2,3 SAY 'Archivo no ha sido generado...'
  413:             ELSE
  414:                @ 1,2 SAY 'No Existen Valores Negativos '
  415:                @ 2,3 SAY 'del Rol/'+sst.ciudad
  416:             ENDIF
  417:             STOR 1 TO opc_ac
  418:             @ 4,12 GET opc_ac PICT '@*H \<ACEPTAR';
  419:             SIZE 1.308,12.500,0.308 ;
  420:             DEFAULT 1 ;
  421:             FONT "MS Serif", 8 ;
  422:             STYLE "BT"
  423:             READ MODAL
  424:             RELE WIND varios
  425:             STOR .F. TO vret
  426:          ENDIF
  427:          SELECT &malias
  428: <========RETURN    vret
  429:          ****************
  430:
*!*************************************************************************
**************
  431:  *!
  432:  *!       Procedure: EDITAR
  433:  *!
  434:  *!       Called by: CREAR              (procedure in
AACT_SST.SPR)
  435:  *!
  436:  *!           Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
  437:  *!                : CABECA             (procedure in
PERSONAL.PRG)
  438:  *!                : INGRESO            (procedure in
ACT_PTMO.SPR)
  439:  *!                : CREA_TMP           (procedure in
MO_PDLAB.SPR)
  440:  *!                : MBROW              (procedure in
MO_PDLAB.SPR)
  441:  *!                : REF_SCR            (procedure in
MO_PDLAB.SPR)
  442:  *!                : MENSAJES           (procedure in
MO_PDLAB.SPR)
  443:  *!
```

```
444:
*!*************************************************************************
**************
445:      PROCEDURE editar
446:           ****************
447:           SHOW GETS DISABLE
448:           SHOW GET opc_rol ENABLE
449:  <========RETURN
450:           ****************
451:
*!*************************************************************************
**************
452:  *!
453:  *!        Procedure: SALIR
454:  *!
455:  *!        Called by: _RAA1732TG()        (function in
AT_ESCLS.SPR)
456:  *!                 : DA_ASISG.PRG
457:  *!                 : DE_EGRQ1.PRG
458:  *!                 : DE_EGRQ2.PRG
459:  *!                 : DE_EGRMN.PRG
460:  *!                 : DE_IEJUB.PRG
461:  *!
462:  *!           Calls: DEAC_KEYS            (procedure in
AACT_SST.SPR)
463:  *!
464:
*!*************************************************************************
**************
465:      PROCEDURE salir
466:           ****************
467:           SET DELE ON
468:           CLEAR READ
469:  ========RETURN
470:
471:           *
 ************************************************************
472:           *        *
473:           *        * _R41191U28            OPC_QUI VALID
474:           *        *
475:           *        * Function Origin:
476:           *        *
477:           *        * From Platform:        Windows
478:           *        * From Screen:          CTRL_VNG,
Record Number:    9
479:           *        * Variable:             OPC_QUI
480:           *        * Called By:            VALID Clause
481:           *        * Object Type:          Radio Button
482:           *        * Snippet Number:       1
483:           *        *
484:           *
 ************************************************************
485:           *
486:
*!*************************************************************************
**************
487:  *!
488:  *!        Function: _R41191U28
```

```
489:  *!
490:  *!        Called by: CTRL_VNG.SPR
491:  *!
492:
*!*****************************************************************
**************
493:      FUNCTION _r4i19iu28      &&  OPC_QUI VALID
494:          #REGION 1
495:          STOR opc_qui TO vtip
496:
497:          *
***************************************************************
498:          *         *
499:          *         * _R4I19IU69              VMES VALID
500:          *         *
501:          *         * Function Origin:
502:          *         *
503:          *         * From Platform:          Windows
504:          *         * From Screen:            CTRL_VNG,
Record Number:   11
505:          *         * Variable:               VMES
506:          *         * Called By:              VALID Clause
507:          *         * Object Type:            Field
508:          *         * Snippet Number:         2
509:          *         *
510:          *
***************************************************************
511:          *
512:
*!*****************************************************************
**************
513:  *!
514:  *!        Function: _R4I19IU69
515:  *!
516:  *!       Called by: CTRL_VNG.SPR
517:  *!
518:
*!*****************************************************************
**************
519:      FUNCTION _r4i19iu69      &&  VMES VALID
520:          #REGION 1
521:          IF vmes > 0 AND vmes < 13
522:              STOR .T. TO vret
523:              SELE mes
524:              LOCATE FOR vmes = mes
525:              STOR nom_mes TO mmes
526:              SHOW GET mes.nom_mes
527:          ELSE
528:              STOR .F. TO vret
529:          ENDIF
530:          RETURN vret
531:
532:          *
***************************************************************
533:          *         *
534:          *         * _R4I19IU9Z              OPC_SCR VALID
535:          *         *
536:          *         * Function Origin:
```

```
  537:          *        *
  538:          *        * From Platform:        Windows
  539:          *        * From Screen:          CTRL_VNG.
Record Number:   12
  540:          *        * Variable:             OPC_SCR
  541:          *        * Called By:            VALID Clause
  542:          *        * Object Type:          Push Button
  543:          *        * Snippet Number:       3
  544:          *        *
  545:          *
************************************************************
  546:          *
  547:
*!*************************************************************************
**************
  548:  *!
  549:  *!       Function: _R4I19IU9Z
  550:  *!
  551:  *!      Called by: CTRL_VNG.SPR
  552:  *!
  553:  *!           Calls: &OPC_SCR.PRG
  554:  *!
  555:
*!*************************************************************************
**************
  556:     FUNCTION _r4i19iu9z     &&  OPC_SCR VALID
  557:          #REGION 1
  558:          DO &opc_scr
  559:
  560:          *
************************************************************
  561:          *        *
  562:          *        * _R4I19IUEM            OPC_ROL VALID
  563:          *        *
  564:          *        * Function Origin:
  565:          *        *
  566:          *        * From Platform:        Windows
  567:          *        * From Screen:          CTRL_VNG,
Record Number:   16
  568:          *        * Variable:             OPC_ROL
  569:          *        * Called By:            VALID Clause
  570:          *        * Object Type:          List
  571:          *        * Snippet Number:       4
  572:          *        *
  573:          *
************************************************************
  574:          *
  575:
*!*************************************************************************
**************
  576:  *!
  577:  *!       Function: _R4I19IUEM
  578:  *!
  579:  *!      Called by: CTRL_VNG.SPR
  580:  *!
  581:
*!*************************************************************************
**************
```

```
582:      FUNCTION _r4i19iuem      &&  OPC_ROL VALID
583:          #REGION 1
584:          STOR sst.cod_ciu TO mrol
585:          STOR 0 TO opc_qui,vtip
586:          STOR .F. TO mquincena
587:        1=DO CASE
588:        à====CASE sst.tipo = 0
589:        |         STOR 1 TO opc_qui
590:        |         SHOW GET opc_qui ENABL
591:        |         STOR .T. TO mquincena
592:        à====CASE sst.tipo = 1
593:        |         SHOW GET opc_qui DISABLE
594:        |         STOR .F. TO mquincena
595:        |         STOR 3 TO vtip
596:        à====CASE sst.tipo = 2
597:        |         SHOW GET opc_qui DISABLE
598:        |         STOR .F. TO mquincena
599:        |         STOR 4 TO vtip
600:        E=ENDCASE
601:          STOR IIF(opc_qui = 1, 'PRIMERA
QUINCENA',IIF(opc_qui = 2,;
602:              'SEGUNDA QUINCENA', 'MENSUAL')) TO arrayitem
603:          SHOW GET opc_qui
604:          SHOW GET manio ENABLE
605:          SHOW GET vmes ENABLE
606:          SHOW GET opc_scr ENABLE
607:  *:  EOF: CTRL_VNG.aci
```


Varios Archivo Nómina Rol de Pagos ReporTes Ptmo.Emp Costos E.D.36 Salir
PARAMETROS PARA CONSULTAR VALORES DE ROL INDIVIDUAL
PASTAZA
PALORA
PRIMERA QUINCENA
SEGUNDA QUINCENA
AÑO. . . . : 1995
MES. . . . : 12 DICIEMBRE
ACEPTAR
EDITAR
SALIR
10:21
Mes
Reg 12/12
Exclusivo
INS
NUM
MAY

```
    1:
*:*************************************************************
**************
    2:   *:
    3:   *: Procedure file: C:\DOC\CTRL_CRL.SPR
    4:   *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:   *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:   *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:   *:  Last modified: 03/03/1995 at 21:57:46
    8:   *:
    9:   *:  Procs & Fncts: _R8F1B2NMG()
   10:   *:                : _R8F1B2NQL()
   11:   *:                : _R8F1B2NVK()
   12:   *:
   13:   *:          Set by: EMPRESA.MPR
   14:   *:
   15:   *:           Calls: _R8F1B2NMG()        (function in
CTRL_CRL.SPR)
   16:   *:                : _R8F1B2NQL()        (function in
CTRL_CRL.SPR)
   17:   *:                : _R8F1B2NVK()        (function in
CTRL_CRL.SPR)
   18:   *:
   19:   *:       Documented 19:14:59
FoxDoc version 3.00a
   20:
*:*************************************************************
**************
   21:      *
************************************************************
   22:      *        *
   23:      *        * 03/03/1995           CTRL_CRL.SPR
 21:57:46
   24:      *        *
   25:      *
************************************************************
   26:      *        *
   27:      *        * Nombre del autor
   28:      *        *
   29:      *        * Copyright (c) 1995 Nombre de la empresa
   30:      *        * Direcci$n
   31:      *        * Ciudad.      C.P.
   32:      *        * Pa$s
   33:      *        *
   34:      *        * Description:
   35:      *        * Este programa lo ha generado autom$ticamente
GENSCRN.
   36:      *        *
   37:      *
************************************************************
   38:
   39:
   40:      *
************************************************************
   41:      *        *
   42:      *        *  CTRL_CRL/Windows C$digo de configuraci$n -
SECCION 1
```

```
43:      *        *
44:      *
*************************************************************
45:      *
46:
47:      #REGION 1
48:      **********
49:      REGIONAL opc_scr, opc_rol, opc_qui, mmes, vmes, manio,
mquincena, mrol.:
50:         mopc_qui,mfquin, salir
51:      STOR ' ' TO opc_scr, mmes,opc_rol, fquin
52:      STOR 0 TO opc_qui,mopc_qui
53:      STOR 1 TO mrol
54:      STOR .F. TO mquincena, salir
55:      STOR YEAR(DATE()) TO manio
56:      STOR MONTH(DATE()) TO vmes
57:
58:      **********
59:
60:      #REGION 0
61:      REGIONAL m.currarea, m.talkstat, m.compstat
62:
63:      IF SET("TALK") = "ON"
64:         SET TALK OFF
65:         m.talkstat = "ON"
66:      ELSE
67:         m.talkstat = "OFF"
68:      ENDIF
69:      m.compstat = SET("COMPATIBLE")
70:      SET COMPATIBLE FOXPLUS
71:
72:      m.rborder = SET("READBORDER")
73:      SET readborder ON
74:
75:      m.currarea = SELECT()
76:
77:
78:      *
*************************************************************
79:      *        *
80:      *        *  CTRL_CRL/Windows Bases de datos, Yndices,
relaciones
81:      *        *
82:      *
*************************************************************
83:      *
84:
85:      IF USED("rol")
86:         SELECT rol
87:         SET ORDER TO TAG "cod_per"
88:      ELSE
89:         SELECT 0
90:         USE (LOCFILE("dbfs\rol.dbf","DBF","Buscar  rol?")):
91:            AGAIN ALIAS rol :
92:            ORDER TAG "cod_per"
93:      ENDIF
94:
95:      IF USED("mes")
```

```
 96:        SELECT mes
 97:        SET ORDER TO 0
 98:     ELSE
 99:        SELECT 0
100:        USE (LOCFILE("dbfs\meses.org","DBF","Buscar
meses?"));
101:           AGAIN ALIAS mes :
102:           ORDER 0
103:     ENDIF
104:
105:     IF USED("sst")
106:        SELECT sst
107:        SET ORDER TO TAG "tipo_act"
108:     ELSE
109:        SELECT 0
110:        USE (LOCFILE("dbfs\sistema.dbf","DBF","Buscar
sistema?"));
111:           AGAIN ALIAS sst :
112:           ORDER TAG "tipo_act"
113:     ENDIF
114:
115:     IF USED("dgp")
116:        SELECT dgp
117:        SET ORDER TO TAG "cod_per"
118:     ELSE
119:        SELECT 0
120:        USE (LOCFILE("dbfs\personal.dbf","DBF","Buscar
personal?"));
121:           AGAIN ALIAS dgp :
122:           ORDER TAG "cod_per"
123:     ENDIF
124:
125:     IF USED("men")
126:        SELECT men
127:        SET ORDER TO TAG "mens_cod"
128:     ELSE
129:        SELECT 0
130:        USE (LOCFILE("dbfs\mensual.dbf","DBF","Buscar
mensual?"));
131:           AGAIN ALIAS men :
132:           ORDER TAG "mens_cod"
133:     ENDIF
134:
135:     IF USED("ctr")
136:        SELECT ctr
137:        SET ORDER TO 0
138:     ELSE
139:        SELECT 0
140:        USE (LOCFILE("dbfs\ctrl_rol.dbf","DBF","Buscar
ctrl_rol?"));
141:           AGAIN ALIAS ctr :
142:           ORDER 0
143:     ENDIF
144:
145:     IF USED("acr")
146:        SELECT acr
147:        SET ORDER TO TAG "cod_per"
148:     ELSE
```

```
149:        SELECT 0
150:        USE (LOCFILE("dbfs\ac_imprt.dbf","DBF","Buscar
ac_imprt?"));
151:            AGAIN ALIAS acr ;
152:            ORDER TAG "cod_per"
153:        ENDIF
154:
155:        SELECT sst
156:
157:        SELECT rol
158:        SELECT rol
159:
160:        SET RELATION OFF INTO acr
161:        SET RELATION TO cod_per INTO acr ADDITIVE
162:
163:        SELECT rol
164:
165:        SET RELATION OFF INTO dgp
166:        SET RELATION TO cod_per INTO dgp ADDITIVE
167:
168:        SELECT sst
169:
170:        *
************************************************************
171:        *        *
172:        *        *              Windows Definiciones de ventana
173:        *        *
174:        *
************************************************************
175:        *
176:
177:        IF NOT WEXIST("pampdr") ;
178:            OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.PJX" ;
179:            OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.SCX" ;
180:            OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.MNX" ;
181:            OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.PRG" ;
182:            OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.FRX" ;
183:            OR UPPER(WTITLE("PAMPDR")) == "PAMPDR.QPR"
184:            DEFINE WINDOW pampdr ;
185:            AT  0.000, 0.000  ;
186:            SIZE 21.923,77.000 ;
187:            FONT "MS Sans Serif", 8 ;
188:            FLOAT ;
189:            NOCLOSE ;
190:            MINIMIZE ;
191:            SYSTEM
192:            MOVE WINDOW pampdr CENTER
193:        ENDIF
194:
195:
196:        *
************************************************************
197:        *        *
198:        *        *  CTRL_CRL Windows Código de configuración -
SECCION 2
199:        *        *
200:        *
************************************************************
```

```
  201:      *
  202:
  203:      #REGION 1
  204:      **********
  205:      SELE mes
  206:      LOCATE FOR mes = vmes
  207:      STOR nom_mes TO mmes
  208:      SELE sst
  209:      STOR sst.ciudad TO opc_rol
  210:
  211:
  212:      #REGION 1
  213:      DEFINE POPUP _r8f1b2nb8 ;
  214:         PROMPT FIELD sst.ciudad ;
  215:         SCROLL
  216:
  217:      *
************************************************************
  218:      *        *
  219:      *        *        CTRL_CRL/Windows Distribución de
pantalla
  220:      *        *
  221:      *
************************************************************
  222:      *
  223:
  224:      #REGION 1
  225:     IF WVISIBLE("pampdr")
  226:        ACTIVATE WINDOW pampdr SAME
  227:     ELSE
  228:        ACTIVATE WINDOW pampdr NOSHOW
  229:     ENDIF
  230:      @ 13.231,39.200 SAY "MES....:" ;
  231:         FONT "Courier New", 10 ;
  232:         STYLE "BIT"
  233:      @ 11.308,39.200 SAY "AÑO....:" ;
  234:         FONT "Courier New", 10 ;
  235:         STYLE "BIT"
  236:      @ 6.692,8.000 TO 16.154,34.000 :
  237:         PEN 1, 8 ;
  238:         STYLE "16"
  239:      @ 18.154,14.200 TO 21.000,62.600 :
  240:         PEN 1, 8 :
  241:         STYLE "16"
  242:      @ 7.692,39.400 GET opc_qui :
  243:         PICTURE "@*RVN \<PRIMERA QUINCENA;\<SEGUNDA
QUINCENA" :
  244:         SIZE 1.308,25.667,0.308 :
  245:         DEFAULT 1 ;
  246:         FONT "MS Serif", 8 ;
  247:         STYLE "BT" ;
  248:         WHEN mquincena
  249:      @ 11.385,53.600 GET manio ;
  250:         SIZE 1.000,6.600 ;
  251:         DEFAULT " " ;
  252:         FONT "MS Sans Serif", 8 ;
  253:         PICTURE "9999" ;
  254:         VALID manio > 1990 AND manio < 2999 ;
```

```
    1:
*:*************************************************************
**************
    2:  *:
    3:  *: Procedure file: C:\DOC\ROL_Q1.SPR
    4:  *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:  *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:  *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:  *:  Last modified: 27/02/1995 at  4:30:10
    8:  *:
    9:  *:  Procs & Fncts: _R8A09NFOG()
   10:  *:                : _R8A09NGFW()
   11:  *:
   12:  *:          Set by: ACEPTAR              (procedure in
VREG_PTE.SPR)
   13:  *:
   14:  *:           Calls: ACT_KEYS             (procedure in
AACT_SST.SPR)
   15:  *:                : _R8A09NFOG()         (function in
ROL_Q1.SPR)
   16:  *:                : _R8A09NGFW()         (function in
ROL_Q1.SPR)
   17:  *:
   18:  *:       Documented 19:15:29
FoxDoc version 3.00a
   19:
*:*************************************************************
**************
   20:      *
************************************************************
   21:      *        *
   22:      *        * 27/02/1995            ROL_Q1.SPR
 04:30:07
   23:      *        *
   24:      *
************************************************************
   25:      *        *
   26:      *        * Nombre del autor
   27:      *        *
   28:      *        * Copyright (c) 1995 Nombre de la empresa
   29:      *        * Direcci¥n
   30:      *        * Ciudad,      C.P.
   31:      *        * Pa¥s
   32:      *        *
   33:      *        * Description:
   34:      *        * Este programa lo ha generado autom®ticamente
GENSCRN.
   35:      *        *
   36:      *
************************************************************
   37:
   38:      PARAMETERS vciu, mciu, vtip, mtip, mcierre, manic,
vmes
   39:
   40:      *
************************************************************
   41:      *        *
```

```
   42:      *        *   ROL_Q1/Windows Código de configuración -
SECCION 1
   43:      *        *
   44:      *
************************************************************
   45:      *
   46:
   47:      #REGION 1
   48:      **********
   49:      REGIONAL opc_scr, opc_busc, indices, tag_bus,
atrat_dat, trat_dat,opc_ind,;
   50:         editando
   51:      STORE '' TO opc_scr, opc_busc, trat_dat, opc_ind
   52:      STOR .F. TO editando
   53:      **********
   54:
   55:      #REGION 0
   56:      REGIONAL m.currarea, m.talkstat, m.compstat
   57:
   58:     -IF SET("TALK") = "ON"
   59:      |  SET TALK OFF
   60:      |  m.talkstat = "ON"
   61:     -ELSE
   62:      |  m.talkstat = "OFF"
   63:     -ENDIF
   64:      m.compstat = SET("COMPATIBLE")
   65:      SET COMPATIBLE FOXPLUS
   66:
   67:      m.rborder = SET("READBORDER")
   68:      SET readborder ON
   69:
   70:      m.currarea = SELECT()
   71:
   72:
   73:      *
************************************************************
   74:      *        *
   75:      *        *   ROL_Q1/Windows Bases de datos, Yndices,
relaciones
   76:      *        *
   77:      *
************************************************************
   78:      *
   79:
   80:     -IF USED("dgp")
   81:      |  SELECT dgp
   82:      |  SET ORDER TO TAG "cod_per"
   83:     -ELSE
   84:      |  SELECT 0
   85:      |  USE (LOCFILE("dbfs\personal.dbf","DBF","Buscar
personal?")):
   86:      |     AGAIN ALIAS dgp ;
   87:      |     ORDER TAG "cod_per"
   88:     -ENDIF
   89:
   90:     -IF USED("sec")
   91:      |  SELECT sec
   92:      |  SET ORDER TO TAG "secc_cod"
```

```
 93:     ELSE
 94:         SELECT 0
 95:         USE (LOCFILE("dbfs\seccion.dbf","DBF","Buscar
seccion?"));
 96:             AGAIN ALIAS sec ;
 97:             ORDER TAG "secc_cod"
 98:     ENDIF
 99:
100:     IF USED("pue")
101:         SELECT pue
102:         SET ORDER TO TAG "pues_cod"
103:     ELSE
104:         SELECT 0
105:         USE (LOCFILE("dbfs\puestos.dbf","DBF","Buscar
puestos?"));
106:             AGAIN ALIAS pue ;
107:             ORDER TAG "pues_cod"
108:     ENDIF
109:
110:     IF USED("rol")
111:         SELECT rol
112:         SET ORDER TO TAG "cod_per"
113:     ELSE
114:         SELECT 0
115:         USE (LOCFILE("dbfs\rol.dbf","DBF","Buscar  rol?"));
116:             AGAIN ALIAS rol ;
117:             ORDER TAG "cod_per"
118:     ENDIF
119:
120:     IF USED("men")
121:         SELECT men
122:         SET ORDER TO TAG "mens_cod"
123:     ELSE
124:         SELECT 0
125:         USE (LOCFILE("dbfs\mensual.dbf","DBF","Buscar
mensual?"));
126:             AGAIN ALIAS men ;
127:             ORDER TAG "mens_cod"
128:     ENDIF
129:
130:     SELECT rol
131:
132:     SELECT dgp
133:     SELECT dgp
134:
135:     SET RELATION OFF INTO sec
136:     SET RELATION TO dgp.num_sec INTO sec ADDITIVE
137:
138:     SELECT dgp
139:
140:     SET RELATION OFF INTO pue
141:     SET RELATION TO dgp.cod_pues INTO pue ADDITIVE
142:
143:     SELECT rol
144:     SELECT rol
145:
146:     SET RELATION OFF INTO dgp
147:     SET RELATION TO cod_per INTO dgp ADDITIVE
```

```
148:
149:      SELECT rol
150:
151:      SET RELATION OFF INTO men
152:      SET RELATION TO cod_per INTO men ADDITIVE
153:
154:      SELECT rol
155:
156:      *
**********************************************************
157:      *        *
158:      *        *              Windows Definiciones de ventana
159:      *        *
160:      *
**********************************************************
161:      *
162:
163:     IF NOT WEXIST("cons_rol") ;
164:           OR UPPER(WTITLE("CONS_ROL")) == "CONS_ROL.PJX" ;
165:           OR UPPER(WTITLE("CONS_ROL")) == "CONS_ROL.SCX" ;
166:           OR UPPER(WTITLE("CONS_ROL")) == "CONS_ROL.MNX" ;
167:           OR UPPER(WTITLE("CONS_ROL")) == "CONS_ROL.PRG" ;
168:           OR UPPER(WTITLE("CONS_ROL")) == "CONS_ROL.FRX" ;
169:           OR UPPER(WTITLE("CONS_ROL")) == "CONS_ROL.QPR"
170:        DEFINE WINDOW cons_rol ;
171:           AT 0.000, 0.000 ;
172:           SIZE 29.385,125.600 ;
173:           TITLE "CONSULTA DE VALORES DE ROL DE PAGO" ;
174:           FONT "MS Sans Serif", 8 ;
175:           FLOAT ;
176:           CLOSE ;
177:           MINIMIZE ;
178:           SYSTEM
179:     ENDIF
180:
181:
182:      *
**********************************************************
183:      *        *
184:      *        *   ROL_Q1/Windows Código de configuración -
SECCION 2
185:      *        *
186:      *
**********************************************************
187:      *
188:
189:      #REGION 1
190:      **********
191:      DO set_keys
192:      SELE dgp
193:      GOTO TOP
194:      DIMENSION indices(20)
195:     FOR i = 2 TO 20
196:        IF EMPTY(TAG(i-1))
197:           i = i - 1
198:           DIMENSION indices(i-1)
199:           EXIT
200:        ELSE
```

```
  201:     ||  |     indices(i-1) = TAG(i-1)
  202:     ||  |ENDIF
  203:     |-ENDFOR
  204:      tag_busc=indices(1)
  205:      opc_ind = indices(1)
  206:      SET ORDER TO &tag_busc
  207:      SHOW GETS DISABLE
  208:      SELE men
  209:      SET FILTER TO men.ciudad = vciu AND men.anio = manio
AND men.mes = vmes
  210:      SELE rol
  211:      GO TOP
  212:      SET FILTER TO rol.ciudad = vciu AND VAL(rol.anio) =
manio AND VAL(rol.mes) = vmes AND rol.tipo  = vtip
  213:      SHOW GETS
  214:
  215:      *
*************************************************************
  216:      *        *
  217:      *        *          ROL_Q1/Windows Distribuci?n de
pantalla
  218:      *        *
  219:      *
*************************************************************
  220:      *
  221:
  222:      #REGION 1
  223:     IF WVISIBLE("cons_rol")
  224:        ACTIVATE WINDOW cons_rol SAME
  225:     ELSE
  226:        ACTIVATE WINDOW cons_rol NOSHOW
  227:     ENDIF
  228:      @ 2.462,1.000 TO 20.000,16.800 :
  229:         PEN 1, 8
  230:      @ 1.077,23.000 TO 9.692,123.200 :
  231:         PEN 1, 8
  232:      @ 9.769,10.600 TO 9.846,10.600 :
  233:         PEN 1, 8
  234:      @ 14.077,17.200 TO 23.538,123.800 :
  235:         PEN 1, 8
  236:      @ 2.077,55.400 SAY " C?digo del Empleado
.................................." + CHR(13) + :
  237:         "Nombre del Empleado ..." + CHR(13) + :
  238:         " Cargo ....................." + CHR(13) + :
  239:         "Departamento ..." :
  240:         SIZE 7.000,43.667, 1.000 :
  241:         FONT "MS Sans Serif", 8 :
  242:         STYLE "BT"
  243:      @ 15.154,56.000 SAY "Ret. Juzgado......." :
  244:         FONT "MS Serif", 7 :
  245:         STYLE "BT"
  246:      @ 16.538,55.600 SAY " Trib. Menores....." :
  247:         FONT "MS Serif", 7 :
  248:         STYLE "BT"
  249:      @ 18.000,55.600 SAY " Ptmo.Empresa...." :
  250:         FONT "MS Serif", 7 :
  251:         STYLE "BT"
  252:      @ 19.462,55.600 SAY " Descto.Energ'a.." :
```

```
253:        FONT "MS Serif", 7 ;
254:        STYLE "BT"
255:     @ 1.538,24.200 SAY dgp.foto ;
256:        SIZE 6.923,26.000 ;
257:        CENTER ;
258:        STYLE ""
259:     @ 20.846,55.600 SAY " Alm. Laboral....." ;
260:        FONT "MS Serif", 7 ;
261:        STYLE "BT"
262:     @ 22.231,55.800 SAY " Coop.Ahorro....." ;
263:        FONT "MS Serif", 7 ;
264:        STYLE "BT"
265:     @ 23.769,56.200 SAY "Multas..............." ;
266:        FONT "MS Serif", 7 ;
267:        STYLE "BT"
268:     @ 25.308,56.000 SAY "Cuenta 143 ........." ;
269:        FONT "MS Serif", 7 ;
270:        STYLE "BT"
271:     @ 17.000,89.000 SAY " Sindicato ..........." ;
272:        FONT "MS Serif", 7 ;
273:        STYLE "BT"
274:     @ 18.308,89.000 SAY " Aso.Empleados.." ;
275:        FONT "MS Serif", 7 ;
276:        STYLE "BT"
277:     @ 19.923,89.800 SAY "Club Andinismo.." ;
278:        FONT "MS Serif", 7 ;
279:        STYLE "BT"
280:     @ 21.231,89.000 SAY " Desctos. A.T.I.T.." ;
281:        FONT "MS Serif", 7 ;
282:        STYLE "BT"
283:     @ 22.615,89.400 SAY " Otros Desctos........." ;
284:        FONT "MS Serif", 7 ;
285:        STYLE "BT"
286:     @ 24.846,89.800 SAY " TOTAL D. DESC:....." ;
287:        FONT "MS Serif", 7 ;
288:        STYLE "BT"
289:     @ 14.692,20.400 SAY "VALOR A RECIBIR:" ;
290:        FONT "MS Sans Serif", 8 ;
291:        STYLE "BIT"
292:     @ 12.000,17.200 TO 14.154,123.800 ;
293:        PEN 1, 8
294:     @ 12.077,55.000 TO 28.539,55.000 ;
295:        PEN 1, 8
296:     @ 14.462,13.400 TO 17.308,43.600 ;
297:        PEN 1, 8 ;
298:        STYLE "16"
299:     @ 1.154,54.200 TO 9.616,54.200 ;
300:        PEN 1, 8
301:     @ 8.615,30.000 SAY "FOTOGRAFIA" ;
302:        FONT "Lydian Csv BT", 8 ;
303:        STYLE "BT"
304:     @ 15.077,19.200 SAY "Ant.Sueldo...." ;
305:        FONT "Times New Roman", 8 ;
306:        STYLE "BT"
307:     @ 19.692,19.000 SAY "Xvi Sueldo....." ;
308:        FONT "Times New Roman", 8 ;
309:        STYLE "BT"
310:     @ 21.308,19.200 SAY "Sub.Energia.." ;
```

```
311:          FONT "Times New Roman", 8 ;
312:          STYLE "BT"
313:       @ 22.923,19.000 SAY "Sub.transp....." ;
314:          FONT "Times New Roman", 8 ;
315:          STYLE "BT"
316:       @ 24.538,19.200 SAY "Bonf.Comis..." ;
317:          FONT "Times New Roman", 8 ;
318:          STYLE "BT"
319:       @ 26.385,19.200 SAY "Sub.Familiar" ;
320:          FONT "Times New Roman", 8 ;
321:          STYLE "BT"
322:       @ 14.538,93.000 SAY " OTROS DESCUENTOS" ;
323:          FONT "MS Serif", 8 ;
324:          STYLE "BIT"
325:       @ 2.923,2.400 GET opc_scr ;
326:          PICTURE "@*VN
|!EDITAR;!<BUSCAR;!<REVISAR;!!IMPRIMIR;!!ELIMINAR;!<SALIR" ;
327:          SIZE 2.462,11.333,0.385 ;
328:          DEFAULT 1 ;
329:          FONT "MS Sans Serif", 8 ;
330:          STYLE "B" ;
331:          VALID _r8a09nfog()
332:       @ 15.846,24.000 GET rol.total_reci ;
333:          SIZE 1.000,13.167 ;
334:          DEFAULT 0 ;
335:          FONT "MS Sans Serif", 8 ;
336:          STYLE "BI" ;
337:          PICTURE "@KZ 999,999,999" ;
338:          WHEN editando AND dgp.estado = 1
339:       @ 12.462,25.200 SAY 'I N G R E S O S' ;
340:          SIZE 0.938,25.200 ;
341:          FONT "Futura Md BT", 9 ;
342:          STYLE "BI"
343:       @ 18.077,37.000 GET rol.ant_sueldo ;
344:          SIZE 1.000,14.000 ;
345:          DEFAULT 0 ;
346:          FONT "MS Sans Serif", 8 ;
347:          PICTURE "@KZ 999,999,999" ;
348:          WHEN editando AND dgp.estado = 1
349:       @ 19.615,37.000 GET rol.xvi_sueldo ;
350:          SIZE 1.000,14.200 ;
351:          DEFAULT 0 ;
352:          FONT "MS Sans Serif", 8 ;
353:          PICTURE "@KZ 999,999,999" ;
354:          WHEN editando AND dgp.estado = 1
355:       @ 21.308,37.000 GET rol.sub_energ ;
356:          SIZE 1.000,14.000 ;
357:          DEFAULT 0 ;
358:          FONT "MS Sans Serif", 8 ;
359:          PICTURE "@KZ 999,999,999" ;
360:          WHEN editando AND dgp.estado = 1
361:       @ 23.462,37.000 GET rol.bonf_comis ;
362:          SIZE 1.000,14.200 ;
363:          DEFAULT 0 ;
364:          FONT "MS Sans Serif", 8 ;
365:          PICTURE "@KZ 999,999,999" ;
366:          WHEN editando AND dgp.estado = 1
367:       @ 25.154,36.800 GET rol.sub_famil ;
```

```
368:          SIZE 1.000,14.400 ;
369:          DEFAULT 0 ;
370:          FONT "MS Sans Serif", 8 ;
371:          PICTURE "@KZ 999,999,999" ;
372:          WHEN editando AND dgp.estado = 1
373:       @ 22.769,37.200 GET rol.sub_trans ;
374:          SIZE 1.000,14.000 ;
375:          DEFAULT 0 ;
376:          FONT "MS Sans Serif", 8 ;
377:          PICTURE "@KZ 999,999,999" ;
378:          WHEN editando AND dgp.estado = 1
379:       @ 10.462,61.600 SAY 'Datos del Rol' ;
380:          SIZE 0.938,25.200 ;
381:          FONT "Futura Md BT", 9 ;
382:          STYLE "BI"
383:       @ 15.000,72.800 GET rol.r_juzgado ;
384:          SIZE 1.000,13.000 ;
385:          DEFAULT 0 ;
386:          FONT "MS Sans Serif", 8 ;
387:          PICTURE "@KZ 999,999,999" ;
388:          WHEN editando AND dgp.estado = 1
389:       @ 16.385,72.800 GET rol.t_menores ;
390:          SIZE 1.000,13.000 ;
391:          DEFAULT 0 ;
392:          FONT "MS Sans Serif", 8 ;
393:          PICTURE "@KZ 999,999,999" ;
394:          WHEN editando AND dgp.estado = 1
395:       @ 17.769,72.000 GET rol.p_empresa ;
396:          SIZE 1.000,13.000 ;
397:          DEFAULT 0 ;
398:          FONT "MS Sans Serif", 8 ;
399:          PICTURE "@KZ 999,999,999" ;
400:          WHEN editando AND dgp.estado = 1
401:       @ 19.154,72.800 GET rol.p_energia ;
402:          SIZE 1.000,13.000 ;
403:          DEFAULT 0 ;
404:          FONT "MS Sans Serif", 8 ;
405:          PICTURE "@KZ 999,999,999" ;
406:          WHEN editando AND dgp.estado = 1
407:       @ 20.538,72.800 GET rol.alm_labora ;
408:          SIZE 1.000,13.000 ;
409:          DEFAULT 0 ;
410:          FONT "MS Sans Serif", 8 ;
411:          PICTURE "@KZ 999,999,999" ;
412:          WHEN editando AND dgp.estado = 1
413:       @ 22.000,72.800 GET rol.coop_ahor ;
414:          SIZE 1.000,13.000 ;
415:          DEFAULT 0 ;
416:          FONT "MS Sans Serif", 8 ;
417:          PICTURE "@KZ 999,999,999" ;
418:          WHEN editando AND dgp.estado = 1
419:       @ 23.462,72.800 GET rol.multas ;
420:          SIZE 1.000,13.000 ;
421:          DEFAULT 0 ;
422:          FONT "MS Sans Serif", 8 ;
423:          PICTURE "@KZ 999,999,999" ;
424:          WHEN editando AND dgp.estado = 1
425:       @ 24.923,72.800 GET rol.cta_143 ;
```

```
426:          SIZE 1.000,13.000 ;
427:          DEFAULT 0 ;
428:          FONT "MS Sans Serif", 8 ;
429:          PICTURE "@KZ 999,999,999" ;
430:          WHEN editando AND dgp.estado = 1
431:       @ 7.846,73.600 GET sec.nombre ;
432:          SIZE 1.000,47.200 ;
433:          DEFAULT " " ;
434:          FONT "MS Sans Serif", 8 ;
435:          PICTURE "@K" ;
436:          WHEN .F.
437:       @ 12.538,75.000 SAY 'E G R E S O S' ;
438:          SIZE 0.938,25.200 ;
439:          FONT "Futura Md BT", 9 ;
440:          STYLE "BI"
441:       @ 5.923,80.000 GET dgp.cargo ;
442:          SIZE 1.000,40.800 ;
443:          DEFAULT " " ;
444:          FONT "MS Sans Serif", 8 ;
445:          PICTURE "@K" ;
446:          WHEN .F.
447:       @ 3.846,83.200 GET dgp.nombre ;
448:          SIZE 1.000,37.600 ;
449:          DEFAULT " " ;
450:          FONT "MS Sans Serif", 8 ;
451:          PICTURE "@K" ;
452:          WHEN .F.
453:       @ 16.615,108.600 GET men.desc_sind1 ;
454:          SIZE 1.000,12.800 ;
455:          DEFAULT 0 ;
456:          FONT "MS Sans Serif", 8 ;
457:          PICTURE "@KZ 999,999,999" ;
458:          WHEN editando AND dgp.estado = 1
459:       @ 18.000,108.800 GET men.aso_emple1 ;
460:          SIZE 1.000,12.800 ;
461:          DEFAULT 0 ;
462:          FONT "MS Sans Serif", 8 ;
463:          PICTURE "@KZ 999,999,999" ;
464:          WHEN editando AND dgp.estado = 1
465:       @ 19.462,108.800 GET men.club_andi1 ;
466:          SIZE 1.000,12.800 ;
467:          DEFAULT 0 ;
468:          FONT "MS Sans Serif", 8 ;
469:          PICTURE "@KZ 999,999,999" ;
470:          WHEN editando AND dgp.estado = 1
471:       @ 20.846,109.000 GET men.ati1 ;
472:          SIZE 1.000,12.800 ;
473:          DEFAULT 0 ;
474:          FONT "MS Sans Serif", 8 ;
475:          PICTURE "@KZ 999,999,999" ;
476:          WHEN editando AND dgp.estado = 1
477:       @ 24.462,109.200 GET rol.otros_dsct ;
478:          SIZE 1.000,10.333 ;
479:          DEFAULT 0 ;
480:          FONT "MS Serif", 8 ;
481:          STYLE "B" ;
482:          PICTURE "@KZ 999,999,999" ;
483:          WHEN editando AND dgp.estado = 1
```

```
484:      @ 22.231,109.400 GET men.votrosdes1 ;
485:          SIZE 1.000,12.600 ;
486:          DEFAULT 0 ;
487:          FONT "MS Sans Serif", 8 ;
488:          PICTURE "@KZ 999,999,999" ;
489:          WHEN editando AND dgp.estado = 1
490:      @ 2.077,109.600 GET dgp.cod_per ;
491:          SIZE 1.000,11.200 ;
492:          DEFAULT " " ;
493:          FONT "MS Sans Serif", 8 ;
494:          PICTURE "@K" ;
495:          WHEN .F.
496:
497:      IF NOT WVISIBLE("cons_rol")
498:          ACTIVATE WINDOW cons_rol
499:      ENDIF
500:
501:      READ CYCLE ;
502:          VALID .F. ;
503:          SHOW _r8a09ngfw()
504:
505:      RELEASE WINDOW cons_rol
506:
507:      *
************************************************************
508:      *         *
509:      *         *              Windows Cerrando bases de datos
510:      *         *
511:      *
************************************************************
512:      *
513:
514:      IF USED("dgp")
515:          SELECT dgp
516:          USE
517:      ENDIF
518:
519:      IF USED("sec")
520:          SELECT sec
521:          USE
522:      ENDIF
523:
524:      IF USED("pue")
525:          SELECT pue
526:          USE
527:      ENDIF
528:
529:      IF USED("rol")
530:          SELECT rol
531:          USE
532:      ENDIF
533:
534:      IF USED("men")
535:          SELECT men
536:          USE
537:      ENDIF
538:
539:      SELECT (m.currarea)
```

```
540:
541:
542:      #REGION 0
543:
544:      SET readborder &rborder
545:
546:     IF m.talkstat = "ON"
547:        SET TALK ON
548:     ENDIF
549:     IF m.compstat = "ON"
550:        SET COMPATIBLE ON
551:     ENDIF
552:
553:
554:      *
************************************************************
555:      *        *
556:      *        *            ROL_Q1/Windows Código de limpieza
557:      *        *
558:      *
************************************************************
559:      *
560:
561:      #REGION 1
562:      ******************
563:
564:
565:      *
************************************************************
566:      *        *
567:      *        *  ROL_Q1/Windows Procedimientos y funciones
de soporte
568:      *        *
569:      *
************************************************************
570:      *
571:
572:      #REGION 1
573:
*!*************************************************************************
**************
574:  *!
575:  *!       Procedure: REVISAR
576:  *!
577:  *!           Calls: MSCATTER           (procedure in
AACT_SST.SPR)
578:  *!                : DEAC_KEYS          (procedure in
AACT_SST.SPR)
579:  *!                : ACT_KEYS           (procedure in
AACT_SST.SPR)
580:  *!                : VAL_CAMPO()        (function in
AP_PRMCT.SPR
581:  *!                : REFRESH()          (function in
AP_PRMCT.SPR)
582:  *!                : MBROW              (procedure in
MO_PDLAP.SPR)
593:  *!
594:
```

```
*!*************************************************************
**************
  585:      PROCEDURE revisar
  586:          *****************
  587:          SET ORDER TO nombre
  588:          GOTO TOP
  589:          BROW FIELDS dgp.cod_per, dgp.nombre,dgp.cargo,:
  590:             dgp.ac_sueldo:p='999,999,999',:
  591:             dgp.ac_hextras:p='999,999,999',:
  592:             dgp.ac_subant:p='999,999,999',:
  593:             dgp.ac_subfam:p='999,999,999',:
  594:             dgp.ac_apiess:p='999,999,999',:
  595:             dgp.ac_100iess:p='999,999,999',:
  596:             dgp.ac_subener:p='999,999,999',:
  597:             dgp.ac_gastos:p='999,999,999',:
  598:             dgp.ac_bcomis:p='999,999,999',:
  599:             dgp.ac_subref:p='999,999,999',:
  600:             dgp.ac_compct:p='999,999,999',:
  601:             dgp.ac_compcc:p='999,999,999',:
  602:             dgp.ac_imprent:p='999,999,999';
  603:             NOEDIT NOMENU :
  604:             FOR dgp.estado = 1
  605:          SHOW GETS
  606: <========RETURN
  607:          ****************
  608:
*!*************************************************************
**************
  609:   *!
  610:   *!       Procedure: BUSCAR
  611:   *!
  612:   *!       Called by: SECC()              (function in
PERSONAL.PRG)
  613:   *!                : _R390JW4G3()        (function in
DSC_GRTR.SPR)
  614:   *!                : _RA81R6ER0()        (function in
ACT_PTMO.SPR)
  615:   *!                : _RA81B3G0D()        (function in
ACT_PTMO.SPR)
  616:   *!                : _R6108FBLN()        (function in
MD_PDLAB.SPR)
  617:   *!                : _R5UOM9Z09()        (function in
DSC_CTA.SPR)
  618:   *!
  619:   *!          Calls: VAL_BUSCAR           (procedure in
AACT_SST.SPR)
  620:   *!                 BUSC_CLV()           (function in
DSC_CAR1.SPR)
  621:   *!                 DESPLEGA             (procedure in
PERSONAL.PRG)
  622:   *!                 BUSCARP()            (function in
ACT_PTMO.SPR)
  623:   *!                 REF_SCR              (procedure in
MD_PDLAB.SPR)
  624:   *!                 MENSAJES             (procedure in
MD_PDLAB.SPR)
  625:   *!                 ASIGNAM              (procedure in
MD_PDLAB.SPR)
```

```
626:  *!
627:
*!*****************************************************************
**************
628:      PROCEDURE buscar
629:          ****************
630:          IF NOT WEXIST('BUSCAR') && o definir en el
principal
631:              DEFINE WINDOW buscar FROM 0,0 TO 5,30 TITLE
'BUSCAR' SYSTEM
632:          ENDIF
633:          ACTIVATE WINDOW buscar
634:          MOVE WINDOW buscar CENTER
635:          reg_ant = RECNO()
636:          SET ORDER TO cod_per
637:          mcod_per = SPAC(6)
638:          STOR ' ' TO mcod_per1
639:          STOR 0   TO mcod_per2
640:          DO WHILE .T.
641:              @ 1,1 SAY 'Ingrese el C¾digo: ' GET mcod_per1
642:              @ 1,24 GET mcod_per2 PICT '999'
643:              READ MODAL
644:              STOR mcod_per1 +'00'+IIF(mcod_per2 <
0,'00',IIF(mcod_per2<100,'0',''))+;
645:                   ALLT(STR(mcod_per2)) TO mcod_per
646:              IF LAST() = 27
647:                  GOTO reg_ant
648:                  vret = .F.
649:                  EXIT
650:              ENDIF
651:              SEEK mcod_per
652:              IF !FOUND()
653:                  WAIT WINDOW NOWAIT 'C¾digo no Existe'
654:                  GOTO reg_ant
655:                  LOOP
656:              ELSE
657:                  vret = .T.
658:                  EXIT
659:              ENDIF
660:          ENDDO
661:          DEACTIVATE WINDOW buscar
662:          SHOW GETS
663:          RETURN
664:          ******************
665:
*!*****************************************************************
**************
666:  *!
667:  *!       Procedure: CANCELAR
668:  *!
669:  *!       Called by: ING                (procedure in
PERSONAL.PRG)
670:  *!                : JUBILAR            (procedure in
PERSONAL.PRG)
671:  *!
672:  *!           Calls: MSCATTER           (procedure in
AACT_SST.SPR)
673:  *!                : ACT_KEYS           (procedure in
```

```
AACT_SST.SPR)
  674:  *!                  : DESPLEGA            (procedure in
PERSONAL.PRG)
  675:  *!                  : BORRA_TMP           (procedure in
MO_PDLAB.SPR)
  676:  *!                  : ASIGNAM             (procedure in
MO_PDLAB.SPR)
  677:  *!                  : REF_SCR             (procedure in
MO_PDLAB.SPR)
  678:  *!
  679:
*!*************************************************************
**************
  680:      PROCEDURE cancelar
  681:          ******************
  682:          STOR .F. TO editando
  683:          SHOW GETS DISABLE
  684:          SHOW GET opc_scr,1 PROMPT '\<EDITAR'
  685:          SHOW GET opc_scr,1  ENABLE
  686:          SHOW GET opc_scr,3  ENABLE
  687:          SHOW GET opc_scr,6  ENABLE
  688:          SHOW GET opc_busc,2 ENABLE
  689:          SHOW GET tag_busc   ENABLE
  690:          SHOW GET trat_dat   ENABLE
  691:          SHOW GET opc_ind    ENABLE
  692:  <=======RETURN
  693:          ******************
  694:
*!*************************************************************
**************
  695:  *!
  696:  *!       Procedure: ACT_KEYS
  697:  *!
  698:  *!       Called by: AACT_SST.SPR
  699:  *!                : AACT_DPT.SPR
  700:  *!                : AACT_SEC.SPR
  701:  *!                : DGP_CARG.SPR
  702:  *!                : AC_IMPRT.SPR
  703:  *!                : DSC_GRTR.SPR
  704:  *!                : ELIMINAR            (procedure in
AACT_SST.SCR)
  705:  *!                : REVISAR             (procedure in
AACT_SST.SPR)
  706:  *!                : CTR_POSP            (procedure in
AACT_SST.SPR)
  707:  *!                : GRABAR              (procedure in
AACT_SST.SPR)
  708:  *!                : CANCELAR            (procedure in
AACT_SST.SPR)
  709:  *!                : ROL_01.SPR
  710:  *!                : ROL_02.SPR
  711:  *!                : ACT_PTMO.SPR
  712:  *!                : MO_PDLAB.SPR
  713:  *!                : DSC_CTA.SPR
  714:  *!                : _R6108FBI7()        (function in
MO_PDLAB.SPR)
  715:  *!
  716:  *!           Calls: CTR_POSP            (procedure in
```

```
AACT_SST.SPR)
  754:  *!
  755:  *!          Calls: MSCATTER          (procedure in
AACT_SST.SPR)
  756:  *!               : DEAC_KEYS         (procedure in
AACT_SST.SPR)
  757:  *!               : ASIGNAM           (procedure in
MO_PDLAB.SPR)
  758:  *!               : REF_SCR           (procedure in
MO_PDLAB.SPR)
  759:  *!               : ACT_KEYS          (procedure in
AACT_SST.SPR)
  760:  *!
  761:
*!*********************************************************************
**************
  762:      PROCEDURE ctr_posp
  763:          ********************
  764:          PARAMETER opc_posp, comando
  765:          SELE rol
  766:        ┌─IF BOF()
  767:        │    GOTO TOP
  768:        └─ENDIF
  769:        ┌─IF EOF()
  770:        │    GOTO BOTTOM
  771:        └─ENDIF
  772:          &comando
  773:        ┌─DO CASE
  774:        ├─────CASE opc_posp = 'PRIMERO'
  775:        │         WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
  776:        ├─────CASE opc_posp = 'SIGUIENTE'
  777:        │       ┌─IF EOF()
  778:        │       │    GOTO BOTTOM
  779:        │       │    WAIT WIND NOWAIT 'FIN DE ARCHIVO'
  780:        │       └─ENDIF
  781:        ├─────CASE opc_posp = 'ANTERIOR'
  782:        │       ┌─IF BOF()
  783:        │       │    GOTO TOP
  784:        │       │    WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
  785:        │       └─ENDIF
  786:        ├─────CASE opc_posp = 'ULTIMO'
  787:        │         WAIT WIND NOWAIT 'FIN DE ARCHIVO'
  788:        └─ENDCASE
  789:          SHOW GETS
  790:  <─────RETURN
  791:          ***************
  792:
*!*********************************************************************
**************
  793:  *!
  794:  *!       Procedure: SALIR
  795:  *!
  796:  *!       Called by: _RAA1732TG()        (function in
AT_ESCLS.SPR)
  797:  *!                : DA_ASISG.PRG
  798:  *!                : DE_EGRQ1.PRG
  799:  *!                : DE_EGRQ2.PRG
  800:  *!                : DE_EGRMN.PRG
```

```
  801:  *!                : DE_IEJUB.PRG
  802:  *!
  803:  *!          Calls: DEAC_KEYS          (procedure in
AACT_SST.SPR)
  804:  *!
  805:
*!*************************************************************************
**************
  806:      PROCEDURE salir
  807:          ***************
  808:          DO deac_keys
  809:          CLEAR READ
  810: <=======RETURN
  811:
  812:          *
************************************************************
  813:          *        *
  814:          *        * _R8A09NFOG            opc_scr VALID
  815:          *        *
  816:          *        * Function Origin:
  817:          *        *
  818:          *        * From Platform:        Windows
  819:          *        * From Screen:          ROL_Q1.      Record
Number:   44
  820:          *        * Variable:             opc_scr
  821:          *        * Called By:            VALID Clause
  822:          *        * Object Type:          Push Button
  823:          *        * Snippet Number:       1
  824:          *        *
  825:          *
************************************************************
  826:          *
  827:
*!*************************************************************************
**************
  828:  *!
  829:  *!        Function: _R8A09NFOG
  830:  *!
  831:  *!       Called by: ROL_Q1.SPR
  832:  *!
  833:  *!           Calls: &OPC_SCR.PRG
  834:  *!
  835:
*!*************************************************************************
**************
  836:      FUNCTION _r8a09nfog      &&  opc_scr VALID
  837:          #REGION 1
  838:          DO &opc_scr
  839:
  840:          *
************************************************************
  841:          *        *
  842:          *        * _R8A09NGFW            Nivel de lectura
SHOW
  843:          *        *
  844:          *        * Function Origin:
  845:          *        *
  846:          *        *
```

```
847:          *        * From Platform:        Windows
848:          *        * From Screen:          ROL_Q1
849:          *        * Called By:            READ Statement
850:          *        * Snippet Number:       2
851:          *        *
852:          *
*************************************************************
853:          *
854:
*!*************************************************************************
**************
855:  *!
856:  *!        Function: _R8A09NGFW
857:  *!
858:  *!       Called by: ROL_Q1.SPR
859:  *!
860:
*!*************************************************************************
**************
861:      FUNCTION _r8a09ngfw     && Read Level Show
862:          PRIVATE currwind
863:          STORE WOUTPUT() TO currwind
864:          *
865:          * Código SHOW de la pantalla: ROL_Q1
866:          *
867:          #REGION 1
868:          IF SYS(2016) = "CONS_ROL" OR SYS(2016) = "**"
869:              ACTIVATE WINDOW cons_rol SAME
870:              @ 1.538,24.200 SAY dgp.foto ;
871:                  SIZE 6.923,26.000 ;
872:                  CENTER ;
873:                  STYLE ""
874:          ENDIF
875:          IF NOT EMPTY(currwind)
876:              ACTIVATE WINDOW (currwind) SAME
877:          ENDIF
878:  *:  EOF: ROL_Q1.scl
```

```
    1:
*:*************************************************************
**************
    2:   *:
    3:   *: Procedure file: C:\DOC\ROL_Q2.SPR
    4:   *:          System: SISTEMA DE ROL DE PAGOS DE LA EEASA
RCN
    5:   *:          Author: EL GRUPO FREDDY - HUGO - MERCEDES
    6:   *:       Copyright (c) 10-12-1995, Sisrol Ambato - Ecuador
    7:   *:  Last modified: 04/03/1995 at  0:33:58
    8:   *:
    9:   *:  Procs & Fncts: _R8G017NY3()
   10:   *:                : _R8G017P0Y()
   11:   *:
   12:   *:          Set by: ACEPTAR             (procedure in
VREG_PTE.SPR)
   13:   *:
   14:   *:           Calls: ACT_KEYS            (procedure in
AACT_SST.SPR)
   15:   *:                : _R8G017NY3()        (function in
ROL_Q2.SPR)
   16:   *:                : _R8G017P0Y()        (function in
ROL_Q2.SPR)
   17:   *:
   18:   *:       Documented 19:15:31
FoxDoc version 3.00a
   19:
*:*************************************************************
**************
   20:      *
************************************************************
   21:      *        *
   22:      *        * 04/03/1995             ROL_Q2.SPR
 00:33:55
   23:      *        *
   24:      *
************************************************************
   25:      *        *
   26:      *        * Nombre del autor
   27:      *        *
   28:      *        * Copyright (c) 1995 Nombre de la empresa
   29:      *        * Direcci?n
   30:      *        * Ciudad,      C.P.
   31:      *        * Pa?s
   32:      *        *
   33:      *        * Description:
   34:      *        * Este programa lo ha generado autom?ticamente
GENSCRN.
   35:      *        *
   36:      *
************************************************************
   37:
   38:      PARAMETERS vciu, mciu, vtip, mtip, mcierre, manio,
vmes
   39:
   40:      *
************************************************************
   41:      *        *
```

```
   42:      *        *   ROL_Q2/Windows Código de configuración -
SECCION 1
   43:      *        *
   44:      *
************************************************************
   45:      *
   46:
   47:      #REGION 1
   48:      **********
   49:      REGIONAL opc_scr, opc_busc, indices, tag_bus,
atrat_dat, trat_dat,opc_ind,:
   50:         editando
   51:      STORE '' TO opc_scr, opc_busc, trat_dat, opc_ind
   52:      STOR .F. TO editando
   53:      **********
   54:
   55:      #REGION 0
   56:      REGIONAL m.currarea, m.talkstat, m.compstat
   57:
   58:     ┌IF SET("TALK") = "ON"
   59:     │   SET TALK OFF
   60:     │   m.talkstat = "ON"
   61:     ├ELSE
   62:     │   m.talkstat = "OFF"
   63:     └ENDIF
   64:      m.compstat = SET("COMPATIBLE")
   65:      SET COMPATIBLE FOXPLUS
   66:
   67:      m.rborder = SET("READBORDER")
   68:      SET readborder ON
   69:
   70:      m.currarea = SELECT()
   71:
   72:
   73:      *
************************************************************
   74:      *        *
   75:      *        *   ROL_Q2/Windows Bases de datos, Indices,
relaciones
   76:      *        *
   77:      *
************************************************************
   78:      *
   79:
   80:     ┌IF USED("dgp")
   81:     │   SELECT dgp
   82:     │   SET ORDER TO TAG "cod_per"
   83:     ├ELSE
   84:     │   SELECT 0
   85:     │   USE (LOCFILE("dbfs\personal.dbf","DBF","Buscar
personal")) ;
   86:     │       AGAIN ALIAS dgp ;
   87:     │       ORDER TAG "cod_per"
   88:     └ENDIF
   89:
   90:     ┌IF USED("sec")
   91:     │   SELECT sec
   92:     │   SET ORDER TO TAG "secc_cod"
```

```
   93:     ELSE
   94:         SELECT 0
   95:         USE (LOCFILE("dbfs\seccion.dbf","DBF","Buscar
seccion?"));
   96:             AGAIN ALIAS sec :
   97:             ORDER TAG "secc_cod"
   98:     ENDIF
   99:
  100:     IF USED("pue")
  101:         SELECT pue
  102:         SET ORDER TO TAG "pues_cod"
  103:     ELSE
  104:         SELECT 0
  105:         USE (LOCFILE("dbfs\puestos.dbf","DBF","Buscar
puestos?"));
  106:             AGAIN ALIAS pue ;
  107:             ORDER TAG "pues_cod"
  108:     ENDIF
  109:
  110:     IF USED("rol")
  111:         SELECT rol
  112:         SET ORDER TO TAG "cod_per"
  113:     ELSE
  114:         SELECT 0
  115:         USE (LOCFILE("dbfs\rol.dbf","DBF","Buscar  rol?"));
  116:             AGAIN ALIAS rol :
  117:             ORDER TAG "cod_per"
  118:     ENDIF
  119:
  120:     IF USED("men")
  121:         SELECT men
  122:         SET ORDER TO TAG "mens_cod"
  123:     ELSE
  124:         SELECT 0
  125:         USE (LOCFILE("dbfs\mensual.dbf","DBF","Buscar
mensual?"));
  126:             AGAIN ALIAS men ;
  127:             ORDER TAG "mens_cod"
  128:     ENDIF
  129:
  130:     SELECT rol
  131:
  132:     SELECT dgp
  133:     SELECT dgp
  134:
  135:     SET RELATION OFF INTO sec
  136:     SET RELATION TO dgp.num_sec INTO sec ADDITIVE
  137:
  138:     SELECT dgp
  139:
  140:     SET RELATION OFF INTO pue
  141:     SET RELATION TO dgp.cod_pues INTO pue ADDITIVE
  142:
  143:     SELECT rol
  144:     SELECT rol
  145:
  146:     SET RELATION OFF INTO dgp
  147:     SET RELATION TO cod_per INTO dgp ADDITIVE
```

```
148:
149:      SELECT rol
150:
151:      SET RELATION OFF INTO men
152:      SET RELATION TO cod_per INTO men ADDITIVE
153:
154:      SELECT rol
155:
156:      *
**********************************************************
157:      *        *
158:      *        *              Windows Definiciones de ventana
159:      *        *
160:      *
**********************************************************
161:      *
162:
163:     IF NOT WEXIST("cons_rol") ;
164:           OR UPPER(WTITLE("CONS_ROL")) == "CONS_ROL.PJX" ;
165:           OR UPPER(WTITLE("CONS_ROL")) == "CONS_ROL.SCX" ;
166:           OR UPPER(WTITLE("CONS_ROL")) == "CONS_ROL.MNX" ;
167:           OR UPPER(WTITLE("CONS_ROL")) == "CONS_ROL.PRG" ;
168:           OR UPPER(WTITLE("CONS_ROL")) == "CONS_ROL.FRX" ;
169:           OR UPPER(WTITLE("CONS_ROL")) == "CONS_ROL.QFR"
170:        DEFINE WINDOW cons_rol ;
171:           AT 0.000, 0.000 ;
172:           SIZE 32.385,126.400 ;
173:           TITLE "CONSULTA DE VALORES DE ROL DE PAGO" ;
174:           FONT "MS Sans Serif", 8 ;
175:           FLOAT ;
176:           CLOSE ;
177:           MINIMIZE ;
178:           SYSTEM
179:     ENDIF
180:
181:
182:      *
**********************************************************
193:      *        *
194:      *        *   ROL_Q2/Windows Código de configuración -
SECCION 2
195:      *        *
196:      *
**********************************************************
197:      *
198:
199:      #SECCION 1
200:      **********
191:      DO set_keys
192:      WAIT dsp
193:      CLEAR TYP
194:      DIMENSION indices(20)
195:     FOR i = 2 TO 20
196:        IF EMPTY(TAG(i-1))
197:           i = i - 1
198:           DIMENSION indices(i-1)
199:           EXIT
200:        ELSE
```

```
  201:     ||  |     indices(i-1) = TAG(i-1)
  202:     ||  |  ENDIF
  203:     ||  ENDFOR
  204:        tag_busc=indices(1)
  205:        opc_ind = indices(1)
  206:        SET ORDER TO &tag_busc
  207:        SHOW GETS DISABLE
  208:        SELE men
  209:        SET FILTER TO men.ciudad = vciu AND men.anio = manio
AND men.mes = vmes
  210:        SELE rol
  211:        GO TOP
  212:        SET FILTER TO rol.ciudad = vciu AND VAL(rol.anio) =
manio AND VAL(rol.mes) = vmes AND rol.tipo  = vtip
  213:        GOTO TOP
  214:        SHOW GETS
  215:
  216:        *
************************************************************
  217:        *        *
  218:        *        *          ROL_Q2/Windows Distribución de
pantalla
  219:        *        *
  220:        *
************************************************************
  221:        *
  222:
  223:        #REGION 1
  224:     IF WVISIBLE("cons_rol")
  225:     |     ACTIVATE WINDOW cons_rol SAME
  226:     ELSE
  227:     |     ACTIVATE WINDOW cons_rol NOSHOW
  228:     ENDIF
  229:        @ 0.077,0.400 TO 11.692,17.400 ;
  230:           PEN 1, 8
  231:        @ 1.077,23.200 TO 9.692,123.400 :
  232:           PEN 1, 8
  233:        @ 9.769,10.600 TO 9.846,10.600 ;
  234:           PEN 1, 8
  235:        @ 14.077,0.200 TO 31.692,125.000 :
  236:           PEN 1, 8
  237:        @ 2.077,55.600 SAY " Código del Empleado
.................................." + CHR(13) + ;
  238:           "Nombre del Empleado ..." + CHR(13) + ;
  239:           "Cargo ..................." + CHR(13) + ;
  240:           "Departamento ..." ;
  241:           SIZE 7.000,43.667, 1.000 ;
  242:           FONT "MS Sans Serif", 8 ;
  243:           STYLE "BIT"
  244:        @ 19.231,1.000 SAY "Sueldo............" + CHR(13) + ;
  245:           "" + CHR(13) + ;
  246:           "H.Extras 40%.." + CHR(13) + ;
  247:           "" + CHR(13) + ;
  248:           "H.Extras 50%..." + CHR(13) + ;
  249:           "" + CHR(13) + ;
  250:           "H.Extras 100%.." + CHR(13) + ;
  251:           "" + CHR(13) + ;
  252:           "Sub.Antigued.." + CHR(13) + ;
```

```
253:          "" + CHR(13) + ;
254:          "Subs.Refrigerio.←" ;
255:          SIZE 11.000,14.167, 0.000 ;
256:          FONT "MS Serif", 7 ;
257:          STYLE "BT"
258:       @ 15.154,62.800 SAY "Ret. Juzgado......." ;
259:          FONT "MS Serif", 7 ;
260:          STYLE "BT"
261:       @ 16.538,62.400 SAY " Trib. Menores....." ;
262:          FONT "MS Serif", 7 ;
263:          STYLE "BT"
264:       @ 18.000,62.400 SAY " Ptmo.Empresa...." ;
265:          FONT "MS Serif", 7 ;
266:          STYLE "BT"
267:       @ 19.462,62.400 SAY " Ptmo I.E.S.S......." ;
268:          FONT "MS Serif", 7 ;
269:          STYLE "BT"
270:       @ 26.769,62.600 SAY "Impuesto Renta." ;
271:          FONT "MS Serif", 7 ;
272:          STYLE "BT"
273:       @ 1.538,24.400 SAY dgp.foto ;
274:          SIZE 6.923,26.000 ;
275:          CENTER ;
276:          STYLE ""
277:       @ 16.154,29.000 SAY "Comp.Sal.Cd/Trab." + CHR(13) + ;
278:          "" + CHR(13) + ;
279:          "Comp.Sal.Cnt/Col.." + CHR(13) + ;
280:          "" + CHR(13) + ;
281:          "Bonif.Complement." + CHR(13) + ;
282:          "" ;
283:          SIZE 5.000,16.500, 0.000 ;
284:          FONT "MS Serif", 7 ;
285:          STYLE "BT"
286:       @ 21.077,29.200 SAY "Vac.Actuales........." + CHR(13)
+ ;
287:          "" + CHR(13) + ;
288:          "Vac.Anteriores....." + CHR(13) + ;
289:          "" + CHR(13) + ;
290:          "Otros Ing.Imponib.." + CHR(13) + ;
291:          "" + CHR(13) + ;
292:          "Otros Ing.No/Imp.." + CHR(13) + ;
293:          "" + CHR(13) + ;
294:          "100% I.E.S.S........." + CHR(13) + ;
295:          "" ;
296:          SIZE 9.000,16.833, 0.000 ;
297:          FONT "MS Serif", 7 ;
298:          STYLE "BT"
299:       @ 20.846,62.400 SAY " Alm. Laboral....." ;
300:          FONT "MS Serif", 7 ;
301:          STYLE "BT"
302:       @ 22.231,62.600 SAY " Coop.Ahorro....." ;
303:          FONT "MS Serif", 7 ;
304:          STYLE "BT"
305:       @ 23.769,63.000 SAY "Multas..............." ;
306:          FONT "MS Serif", 7 ;
307:          STYLE "BT"
308:       @ 25.308,62.800 SAY "Cuenta 143 ........" ;
309:          FONT "MS Serif", 7 ;
```

```
310:          STYLE "BT"
311:       @ 28.154,62.200 SAY " Sindicato ..........." :
312:          FONT "MS Serif", 7 :
313:          STYLE "BT"
314:       @ 29.462,62.200 SAY " Aso.Empleados.." :
315:          FONT "MS Serif", 7 :
316:          STYLE "BT"
317:       @ 19.000,93.000 SAY "Club Andinismo.." :
318:          FONT "MS Serif", 7 :
319:          STYLE "BT"
320:       @ 20.308,92.200 SAY " Descfos. A.T.I.T.." :
321:          FONT "MS Serif", 7 :
322:          STYLE "BT"
323:       @ 21.692,92.200 SAY " Otr.Dsc.Aso.Emp." :
324:          FONT "MS Serif", 7 :
325:          STYLE "BT"
326:       @ 24.615,93.400 SAY "Pre-Coop.Pastaza" :
327:          FONT "MS Serif", 7 :
328:          STYLE "BT"
329:       @ 26.154,93.200 SAY "Financonti............." :
330:          FONT "MS Serif", 7 :
331:          STYLE "BT"
332:       @ 27.538,92.600 SAY " Otros Descfos........." :
333:          FONT "MS Serif", 7 :
334:          STYLE "BT"
335:       @ 29.154,92.600 SAY " TOTAL O. DESC:....." :
336:          FONT "MS Serif", 7 :
337:          STYLE "BT"
338:       @ 23.231,92.600 SAY " Otr.Dsc.Sindicato" :
339:          FONT "MS Serif", 7 :
340:          STYLE "BT"
341:       @ 14.615,2.600 SAY "VALOR A RECIBIR:" :
342:          FONT "MS Sans Serif", 8 :
343:          STYLE "BIT"
344:       @ 12.000,0.200 TO 14.154,125.000 :
345:          PEN 1, 8
346:       @ 12.077,61.400 TO 31.692,61.400 :
347:          PEN 1, 8
348:       @ 14.385,0.600 TO 17.231,26.200 :
349:          PEN 1, 8 :
350:          STYLE "16"
351:       @ 1.154,54.400 TO 9.616,54.400 :
352:          PEN 1, 8
353:       @ 3.615,30.200 SAY "FOTOGRAFIA" :
354:          FONT "Lydian Csv BT", 8 :
355:          STYLE "BT"
356:       @ 0,00,2.400 GET opc_cor :
357:          PICTURE "@*VN
"EDITAR:\<BUSCAR:\<REVISAR:\<IMPRIMIR:\<ELIMINAR:\<SALIR" :
358:          SIZE 1.385,11.333,0.385 :
359:          DEFAULT 1 :
360:          FONT "MS Sans Serif", 8 ;
361:          STYLE "B" :
362:          VALID _rBg017ny3()
363:       @ 12.462,15.000 SAY 'I N G R E S O S' :
364:          SIZE 1.000,21.000 :
365:          FONT "", 9 :
366:          STYLE "BI"
```

```
367:      @ 15.769,6.400 GET rol.total_reci ;
368:         SIZE 1.000,13.167 ;
369:         DEFAULT 0 ;
370:         FONT "MS Sans Serif", 8 ;
371:         STYLE "BI" ;
372:         PICTURE "@KZ 999,999,999" ;
373:         WHEN editando AND dgp.estado = 1
374:      @ 21.231,47.800 GET rol.vac_act ;
375:         SIZE 1.000,12.200 ;
376:         DEFAULT 0 ;
377:         FONT "MS Sans Serif", 8 ;
378:         PICTURE "@KZ 999,999,999" ;
379:         WHEN editando AND dgp.estado = 1
380:      @ 22.923,47.800 GET rol.vac_ant ;
381:         SIZE 1.000,12.200 ;
382:         DEFAULT 0 ;
383:         FONT "MS Sans Serif", 8 ;
384:         PICTURE "@KZ 999,999,999" ;
385:         WHEN editando AND dgp.estado = 1
386:      @ 24.538,47.800 GET rol.otros_imp ;
387:         SIZE 1.000,12.200 ;
388:         DEFAULT 0 ;
389:         FONT "MS Sans Serif", 8 ;
390:         PICTURE "@KZ 999,999,999" ;
391:         WHEN editando AND dgp.estado = 1
392:      @ 26.231,47.600 GET rol.otros_ini ;
393:         SIZE 1.000,12.200 ;
394:         DEFAULT 0 ;
395:         FONT "MS Sans Serif", 8 ;
396:         PICTURE "@KZ 999,999,999" ;
397:         WHEN editando AND dgp.estado = 1
398:      @ 28.000,47.800 GET rol.iess_100 ;
399:         SIZE 1.000,12.200 ;
400:         DEFAULT 0 ;
401:         FONT "MS Sans Serif", 8 ;
402:         PICTURE "@KZ 999,999,999" ;
403:         WHEN editando AND dgp.estado = 1
404:      @ 16.000,47.600 GET rol.comp_sal ;
405:         SIZE 1.000,12.200 ;
406:         DEFAULT 0 ;
407:         FONT "MS Sans Serif", 8 ;
408:         PICTURE "@KZ 999,999,999" ;
409:         WHEN editando AND dgp.estado = 1
410:      @ 17.846,47.600 GET rol.comp_salcc ;
411:         SIZE 1.000,12.200 ;
412:         DEFAULT 0 ;
413:         FONT "MS Sans Serif", 8 ;
414:         PICTURE "@KZ 999,999,999" ;
415:         WHEN editando AND dgp.estado = 1
416:      @ 19.615,47.600 GET rol.bonif_comp ;
417:         SIZE 1.000,12.200 ;
418:         DEFAULT 0 ;
419:         FONT "MS Sans Serif", 8 ;
420:         PICTURE "@KZ 999,999,999" ;
421:         WHEN editando AND dgp.estado = 1
422:      @ 19.077,15.200 GET rol.sueldo ;
423:         SIZE 1.000,12.000 ;
424:         DEFAULT 0 ;
```

```
425:          FONT "MS Sans Serif", 6 ;
426:          PICTURE "@KZ 999,999,999" :
427:          WHEN editando AND dgp.estado = 1
428:       @ 20.538,15.200 GET rol.hext_40 ;
429:          SIZE 1.000,12.000 ;
430:          DEFAULT 0 ;
431:          FONT "MS Sans Serif", 8 ;
432:          PICTURE "@KZ 999,999,999" ;
433:          WHEN editando AND dgp.estado = 1
434:       @ 22.154,15.200 GET rol.hext_50 ;
435:          SIZE 1.000,12.000 ;
436:          DEFAULT 0 ;
437:          FONT "MS Sans Serif", 8 ;
438:          PICTURE "@KZ 999,999,999" ;
439:          WHEN editando AND dgp.estado = 1
440:       @ 23.769,15.200 GET rol.hext_100 ;
441:          SIZE 1.000,12.000 ;
442:          DEFAULT 0 ;
443:          FONT "MS Sans Serif", 8 ;
444:          PICTURE "@KZ 999,999,999" ;
445:          WHEN editando AND dgp.estado = 1
446:       @ 25.615,15.200 GET rol.sub_antig ;
447:          SIZE 1.000,12.000 ;
448:          DEFAULT 0 ;
449:          FONT "MS Sans Serif", 8 ;
450:          PICTURE "@KZ 999,999,999" ;
451:          WHEN editando AND dgp.estado = 1
452:       @ 27.385,15.200 GET rol.sub_refr ;
453:          SIZE 1.000,12.000 ;
454:          DEFAULT 0 ;
455:          FONT "MS Sans Serif", 8 ;
456:          PICTURE "@KZ 999,999,999" ;
457:          WHEN editando AND dgp.estado = 1
458:       @ 10.462,19.800 SAY 'Datos del Rol' ;
459:          SIZE 1.000,21.000 ;
460:          FONT "", 9 ;
461:          STYLE "BI"
462:       @ 7.846,73.800 GET sec.nombre ;
463:          SIZE 1.000,47.200 ;
464:          DEFAULT " " ;
465:          FONT "MS Sans Serif", 8 ;
466:          PICTURE "@K" ;
467:          WHEN .F.
468:       @ 5.923,80.200 GET dgp.cargo ;
469:          SIZE 1.000,40.800 ;
470:          DEFAULT " " ;
471:          FONT "MS Sans Serif", 8 ;
472:          PICTURE "@K" ;
473:          WHEN .F.
474:       @ 3.538,83.400 GET dgp.nombre ;
475:          SIZE 1.000,37.600 ;
476:          DEFAULT " " ;
477:          FONT "MS Sans Serif", 8 ;
478:          PICTURE "@K" ;
479:          WHEN .F.
480:       @ 12.538,83.600 SAY 'E G R E S O S' ;
481:          SIZE 1.000,21.000 ;
482:          FONT "", 9 ;
```

```
483:          STYLE "BI"
484:       @ 15.000,79.000 GET rol.r_juzgado ;
485:          SIZE 1.000,12.600 ;
486:          DEFAULT 0 ;
487:          FONT "MS Sans Serif", 8 ;
488:          PICTURE "@KZ 999,999,999" ;
489:          WHEN editando AND dgp.estado = 1
490:       @ 16.385,79.000 GET rol.t_menores ;
491:          SIZE 1.000,12.600 ;
492:          DEFAULT 0 ;
493:          FONT "MS Sans Serif", 8 ;
494:          PICTURE "@KZ 999,999,999" ;
495:          WHEN editando AND dgp.estado = 1
496:       @ 17.769,79.000 GET rol.p_empresa ;
497:          SIZE 1.000,12.600 ;
498:          DEFAULT 0 ;
499:          FONT "MS Sans Serif", 8 ;
500:          PICTURE "@KZ 999,999,999" ;
501:          WHEN editando AND dgp.estado = 1
502:       @ 19.154,79.000 GET rol.ptmo_iess ;
503:          SIZE 1.000,12.600 ;
504:          DEFAULT 0 ;
505:          FONT "MS Sans Serif", 8 ;
506:          PICTURE "@KZ 999,999,999" ;
507:          WHEN editando AND dgp.estado = 1
508:       @ 26.462,79.000 GET rol.imp_renta ;
509:          SIZE 1.000,12.600 ;
510:          DEFAULT 0 ;
511:          FONT "MS Sans Serif", 8 ;
512:          PICTURE "@KZ 999,999,999" ;
513:          WHEN editando AND dgp.estado = 1
514:       @ 20.462,79.000 GET rol.alm_labora ;
515:          SIZE 1.000,12.600 ;
516:          DEFAULT 0 ;
517:          FONT "MS Sans Serif", 8 ;
518:          PICTURE "@KZ 999,999,999" ;
519:          WHEN editando AND dgp.estado = 1
520:       @ 22.000,79.000 GET rol.coop_ahor ;
521:          SIZE 1.000,12.600 ;
522:          DEFAULT 0 ;
523:          FONT "MS Sans Serif", 8 ;
524:          PICTURE "@KZ 999,999,999" ;
525:          WHEN editando AND dgp.estado = 1
526:       @ 23.462,79.000 GET rol.multas ;
527:          SIZE 1.000,12.600 ;
528:          DEFAULT 0 ;
529:          FONT "MS Sans Serif", 8 ;
530:          PICTURE "@KZ 999,999,999" ;
531:          WHEN editando AND dgp.estado = 1
532:       @ 24.923,79.000 GET rol.cta_143 ;
533:          SIZE 1.000,12.600 ;
534:          DEFAULT 0 ;
535:          FONT "MS Sans Serif", 8 ;
536:          PICTURE "@KZ 999,999,999" ;
537:          WHEN editando AND dgp.estado = 1
538:       @ 27.769,79.000 GET rol.sindicato ;
539:          SIZE 1.000,12.600 ;
540:          DEFAULT 0 ;
```

```
541:          FONT "MS Sans Serif", 8 ;
542:          PICTURE "@KZ 999,999,999" ;
543:          WHEN editando AND dgp.estado = 1
544:       @ 29.154,79.200 GET rol.aso_emplea :
545:          SIZE 1.000,12.400 ;
546:          DEFAULT 0 ;
547:          FONT "MS Sans Serif", 8 ;
548:          PICTURE "@KZ 999,999,999" ;
549:          WHEN editando AND dgp.estado = 1
550:       @ 18.538,111.200 GET men.club_andi2 ;
551:          SIZE 1.000,11.800 ;
552:          DEFAULT 0 ;
553:          FONT "MS Sans Serif", 8 ;
554:          PICTURE "@KZ 999,999,999" ;
555:          WHEN editando AND dgp.estado = 1
556:       @ 19.923,111.400 GET men.atit ;
557:          SIZE 1.000,11.600 ;
558:          DEFAULT 0 ;
559:          FONT "MS Sans Serif", 8 ;
560:          PICTURE "@KZ 999,999,999" ;
561:          WHEN editando AND dgp.estado = 1
562:       @ 21.308,111.600 GET men.aso_emple2 ;
563:          SIZE 1.000,11.600 ;
564:          DEFAULT 0 ;
565:          FONT "MS Sans Serif", 8 ;
566:          PICTURE "@KZ 999,999,999" ;
567:          WHEN editando AND dgp.estado = 1
568:       @ 22.692,111.600 GET men.desc_sind2 ;
569:          SIZE 1.000,11.600 ;
570:          DEFAULT 0 ;
571:          FONT "MS Sans Serif", 8 ;
572:          PICTURE "@KZ 999,999,999" ;
573:          WHEN editando AND dgp.estado = 1
574:       @ 24.154,111.600 GET men.pre_coop ;
575:          SIZE 1.000,11.600 ;
576:          DEFAULT 0 ;
577:          FONT "MS Sans Serif", 8 ;
578:          PICTURE "@KZ 999,999,999" ;
579:          WHEN editando AND dgp.estado = 1
580:       @ 25.692,111.600 GET men.financonti ;
581:          SIZE 1.000,11.600 ;
582:          DEFAULT 0 ;
583:          FONT "MS Sans Serif", 8 ;
584:          PICTURE "@KZ 999,999,999" ;
585:          WHEN editando AND dgp.estado = 1
586:       @ 27.154,111.800 GET men.otrosdes2 ;
587:          SIZE 1.000,11.600 ;
588:          DEFAULT 0 ;
589:          FONT "MS Sans Serif", 8 ;
590:          PICTURE "@KZ 999,999,999" ;
591:          WHEN editando AND dgp.estado = 1
592:       @ 28.769,112.000 GET rol.otros_dsct ;
593:          SIZE 1.000,9.667 ;
594:          DEFAULT 0 ;
595:          FONT "MS Serif", 6 ;
596:          STYLE "B" ;
597:          PICTURE "@KZ 999,999,999" ;
598:          WHEN editando AND dgp.estado = 1
```

```
599:      @ 2.077,109.000 GET dgp.cod_per :
600:         SIZE 1.000,11.200 ;
601:         DEFAULT " " ;
602:         FONT "MS Sans Serif", 8 ;
603:         PICTURE "@K" ;
604:         WHEN .F.
605:      @ 15.154,92.800 SAY "Ant.Sueldo...." ;
606:         FONT "MS Serif", 7 ;
607:         STYLE "BT"
608:      @ 14.692,111.000 GET rol.ant_sueldo :
609:         SIZE 1.000,11.800 ;
610:         DEFAULT 0 ;
611:         FONT "MS Sans Serif", 8 ;
612:         PICTURE "@KZ 999,999,999" ;
613:         WHEN editando AND dgp.estado = 1
614:      @ 16.538,92.800 SAY "Aport_iess...." ;
615:         FONT "MS Serif", 7 ;
616:         STYLE "BT"
617:      @ 16.077,111.000 GET rol.apor_iess ;
618:         SIZE 1.000,11.800 ;
619:         DEFAULT 0 ;
620:         FONT "MS Sans Serif", 8 ;
621:         PICTURE "@KZ 999,999,999" ;
622:         WHEN editando AND dgp.estado = 1
623:
624:      IF NOT WVISIBLE("cons_rol")
625:         ACTIVATE WINDOW cons_rol
626:      ENDIF
627:
628:      READ CYCLE ;
629:         VALID .F. ;
630:         SHOW _r3gC17p0y()
631:
632:      RELEASE WINDOW cons_rol
633:
634:      *
*************************************************************
635:      *        *
636:      *        *              Windows Cerrando bases de datos
637:      *        *
638:      *
*************************************************************
639:      *
640:
641:      IF USED("dgp")
642:         SELECT dgp
643:         USE
644:      ENDIF
645:
646:      IF USED("sec")
647:         SELECT sec
648:         USE
649:      ENDIF
650:
651:      IF USED("pue")
652:         SELECT pue
653:         USE
654:      ENDIF
```

```
655:
656:     IF USED("rol")
657:        SELECT rol
658:        USE
659:     ENDIF
660:
661:     IF USED("men")
662:        SELECT men
663:        USE
664:     ENDIF
665:
666:     SELECT (m.currarea)
667:
668:
669:     #REGION 0
670:
671:     SET readborder &rborder
672:
673:     IF m.talkstat = "ON"
674:        SET TALK ON
675:     ENDIF
676:     IF m.compstat = "ON"
677:        SET COMPATIBLE ON
678:     ENDIF
679:
680:
681:     *
************************************************************
682:     *        *
683:     *        *            ROL_Q2/Windows Código de limpieza
684:     *        *
685:     *
************************************************************
686:     *
687:
688:     #REGION 1
689:     *****************
690:
691:
692:     *
************************************************************
693:     *        *
694:     *        *  ROL_Q2 Windows Procedimientos y funciones
de soporte
695:     *        *
696:     *
************************************************************
697:     *
698:
699:     #REGION 1
700:
*!*************************************************************
**************
701:  *!
702:  *!      Procedure: REVISAR
703:  *!
704:  *!          Calls: MSCATTER           (procedure in
AACT_SST.SPR)
```

```
  705:  *!                  : DEAC_KEYS          (procedure in
AACT_SST.SPR)
  706:  *!                  : ACT_KEYS           (procedure in
AACT_SST.SPR)
  707:  *!                  : VAL_CAMPO()        (function in
AP_PRMCT.SPR)
  708:  *!                  : REFRESH()          (function in
AP_PRMCT.SPR)
  709:  *!                  : MBROW              (procedure in
MO_PDLAB.SPR)
  710:  *!
  711:
*!*****************************************************************
**************
  712:      PROCEDURE revisar
  713:          ****************
  714:          SELE rol
  715:          SET ORDER TO nombre
  716:          GOTO TOP
  717:          BROW FIELDS rol.cod_per, dgp.nombre,dgp.cargo,;
  718:             dgp.ac_sueldo:p='999,999,999',;
  719:             dgp.ac_hextras:p='999,999,999',;
  720:             dgp.ac_subant:p='999,999,999',;
  721:             dgp.ac_subfam:p='999,999,999',;
  722:             dgp.ac_apiess:p='999,999,999',;
  723:             dgp.ac_100iess:p='999,999,999',;
  724:             dgp.ac_subener:p='999,999,999',;
  725:             dgp.ac_gastos:p='999,999,999',;
  726:             dgp.ac_bcomis:p='999,999,999',;
  727:             dgp.ac_subref:p='999,999,999',;
  728:             dgp.ac_compct:p='999,999,999',;
  729:             dgp.ac_compcc:p='999,999,999',;
  730:             dgp.ac_imprent:p='999,999,999';
  731:             !NOEDIT NOMENU
  732:          SHOW GETS
  733:          ==========RETURN
  734:          ****************
  735:
*!*****************************************************************
**************
  736:  *!
  737:  *!       Procedure: BUSCAR
  738:  *!
  739:  *!       Called by: SECC()            (function in
PERSONAL.PRG)
  740:  *!                : _R200JW4G3()      (function in
DSC_CRTR.SPR)
  741:  *!                : _RA81M3ER0()      (function in
ACT_PTMO.SPR)
  742:  *!                : _RA81B8GQD()      (function in
ACT_PTMO.SPR)
  743:  *!                : _R5100FBLN()      (function in
MO_PDLAB.SPR)
  744:  *!                : _R5KOM9Z09()      (function in
DSC_CTA.SPR)
  745:  *!
  746:  *!            Calls: VAL_BUSCAR       (procedure in
AACT_SST.SPR)
```

```
  747:  *!                  : BUSC_CLV()          (function in
DGP_CARG.SPR)
  748:  *!                  : DESPLEGA            (procedure in
PERSONAL.FRG)
  749:  *!                  : BUSCARP()           (function in
ACT_PTMO.SPR)
  750:  *!                  : REF_SCR             (procedure in
MO_PDLAB.SPR)
  751:  *!                  : MENSAJES            (procedure in
MO_PDLAB.SPR)
  752:  *!                  : ASIGNAM             (procedure in
MO_PDLAB.SPR)
  753:  *!
  754:
*!*************************************************************
**************
  755:      PROCEDURE buscar
  756:          ****************
  757:        ┌─IF NOT WEXIST('BUSCAR') && o definir en el
principal
  758:        │     DEFINE WINDOW buscar FROM 0,0 TO 5,30 TITLE
'BUSCAR' SYSTEM
  759:        └─ENDIF
  760:          ACTIVATE WINDOW buscar
  761:          MOVE WINDOW buscar CENTER
  762:          reg_ant = RECNO()
  763:          SET ORDER TO cod_per
  764:          mcod_per = SPAC(6)
  765:          STOR ' ' TO mcod_per1
  766:          STOR 0   TO mcod_per2
  767:        ┌─DO WHILE .T.
  768:        │     @ 1,1 SAY 'Ingrese el Código: ' GET mcod_per1
  769:        │     @ 1,24 GET mcod_per2 PICT '999'
  770:        │     READ MODAL
  771:        │     STOR mcod_per1 +'00'+IIF(mcod_per2 <
0.'00',IIF(mcod_per2<100,'0',''))+;
  772:        │          ALLT(STR(mcod_per2)) TO mcod_per
  773:        │   ┌─IF LAST() = 27
  774:        │   │     GOTO reg_ant
  775:        │   │     vret = .F.
  776:        │   ╞═══════EXIT
  777:        │   └─ENDIF
  778:        │     SEEK mcod_per
  779:        │   ┌─IF !FOUND()
  780:        │   │     WAIT WINDOW NOWAIT 'Código no Existe'
  781:        │   │     GOTO reg_ant
  782:        │   ╞═══════LOOP
  783:        │   ├─ELSE
  784:        │   │     vret = .T.
  785:        │   ╞═══════EXIT
  786:        │   └─ENDIF
  787:        └─ENDDO
  788:          DEACTIVATE WINDOW buscar
  789:          SHOW GETS
  790:  ╞═══════RETURN
  791:          ******************
  792:
*!*************************************************************
```

```
**************
  793:  *!
  794:  *!       Procedure: CANCELAR
  795:  *!
  796:  *!       Called by: ING                (procedure in
PERSONAL.PRG)
  797:  *!                : JUBILAR            (procedure in
PERSONAL.PRG)
  798:  *!
  799:  *!           Calls: MSCATTER           (procedure in
AACT_SST.SPR)
  800:  *!                : ACT_KEYS           (procedure in
AACT_SST.SPR)
  801:  *!                : DESPLEGA           (procedure in
PERSONAL.PRG)
  802:  *!                : BORRA_TMP          (procedure in
MO_PDLAB.SPR)
  803:  *!                : ASIGNAM            (procedure in
MO_PDLAB.SPR)
  804:  *!                : REF_SCR            (procedure in
MO_PDLAB.SPR)
  805:  *!
  806:
*!*************************************************************************
**************
  807:      PROCEDURE cancelar
  808:          ******************
  809:          STOR .F. TO editando
  810:          SHOW GETS DISABLE
  811:          SHOW GET opc_scr,1 PROMPT '\<EDITAR'
  812:          SHOW GET opc_scr,1  ENABLE
  813:          SHOW GET opc_scr,3  ENABLE
  814:          SHOW GET opc_scr,6  ENABLE
  815:          SHOW GET opc_busc,2 ENABLE
  816:          SHOW GET tag_busc   ENABLE
  817:          SHOW GET trat_dat   ENABLE
  818:          SHOW GET opc_ind    ENABLE
  819:          RETURN
  820:          ******************
  821:
*!*************************************************************************
**************
  822:  *!
  823:  *!       Procedure: ACT_KEYS
  824:  *!
  825:  *!       Called by: AACT_SST.SPR
  826:  *!                : AACT_DET.SPR
  827:  *!                : AACT_SEC.SPR
  828:  *!                : DGP_CARG.SPR
  829:  *!                : AC_IMPRT.SPR
  830:  *!                : DSC_GRTR.SPR
  831:  *!                : ELIMINAR           (procedure in
AACT_SST.SPR)
  832:  *!                : REVISAR            (procedure in
AACT_SST.SPR)
  833:  *!                : CTR_POSP           (procedure in
AACT_SST.SPR)
  834:  *!                : GRABAR             (procedure in
```

```
AACT_SST.SPR)
  835:  *!                  : CANCELAR          (procedure in
AACT_SST.SPR)
  836:  *!                  : ROL_Q1.SPR
  837:  *!                  : ROL_Q2.SPR
  838:  *!                  : ACT_PTMO.SPR
  839:  *!                  : MO_PDLAB.SPR
  840:  *!                  : DSC_CTA.SPR
  841:  *!                  : _R6106FB17()      (function in
MO_PDLAB.SPR)
  842:  *!
  843:  *!          Calls: CTR_POSP           (procedure in
AACT_SST.SPR)
  844:  *!
  845:
*!*************************************************************************
**************
  846:      PROCEDURE act_keys
  847:          ********************
  848:          ON KEY LABEL pgup DO ctr_posp WITH
'ANTERIOR','SKIP -1'
  849:          ON KEY LABEL pgdn DO ctr_posp WITH
'SIGUIENTE','SKIP'
  850:          ON KEY LABEL home DO ctr_posp WITH 'PRIMERO','GOTO
TOP'
  851:          ON KEY LABEL END DO ctr_posp WITH 'ULTIMO','GOTO
BOTTOM'
  852:          ========RETURN
  853:          **********************
  854:
*!*************************************************************************
**************
  855:  *!
  856:  *!       Procedure: DEAC_KEYS
  857:  *!
  858:  *!       Called by: EDITAR           (procedure in
CTRL_REG.SPR)
  859:  *!                : SALIR            (procedure in
CTRL_REG.SPR)
  860:  *!                : CREAR            (procedure in
AACT_SST.SPR)
  861:  *!                : ELIMINAR         (procedure in
AACT_SST.SPR)
  862:  *!                : REVISAR          (procedure in
AACT_SST.SPR)
  863:  *!                : CTR_POSP         (procedure in
AACT_SST.SPR)
  864:  *!                : _R6108FB17()     (function in
MO_PDLAB.SPR)
  865:  *!
  866:
*!*************************************************************************
**************
  867:      PROCEDURE deac_keys
  868:          ********************
  869:          ON KEY LABEL pgup
  870:          ON KEY LABEL pgdn
  871:          ON KEY LABEL home
```

```
872:          ON KEY LABEL END
873:  <=======RETURN
874:          ********************
875:
*!*************************************************************************
**************
876:  *!
877:  *!       Procedure: CTR_POSP
878:  *!
879:  *!       Called by: ACT_KEYS            (procedure in
AACT_SST.SPR)
880:  *!                : ELIMINAR            (procedure in
AACT_SST.SPR)
881:  *!
882:  *!           Calls: MSCATTER            (procedure in
AACT_SST.SPR)
883:  *!                : DEAC_KEYS           (procedure in
AACT_SST.SPR)
884:  *!                : ASIGNAM             (procedure in
MO_PDLAB.SPR)
885:  *!                : REF_SCR             (procedure in
MO_PDLAB.SPR)
886:  *!                : ACT_KEYS            (procedure in
AACT_SST.SPR)
887:  *!
888:
*!*************************************************************************
**************
889:      PROCEDURE ctr_posp
890:          ********************
891:          PARAMETER opc_posp, comando
892:          SELE rol
893:        ┌─IF BOF()
894:        │     GOTO TOP
895:        └─ENDIF
896:        ┌─IF EOF()
897:        │     GOTO BOTTOM
898:        └─ENDIF
899:          &comando
900:        ┌=DO CASE
901:        ├====CASE opc_posp = 'PRIMERO'
902:        │         WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
903:        ├====CASE opc_posp = 'SIGUIENTE'
904:        │       ┌─IF EOF()
905:        │       │     GOTO BOTTOM
906:        │       │     WAIT WIND NOWAIT 'FIN DE ARCHIVO'
907:        │       └─ENDIF
908:        ├====CASE opc_posp = 'ANTERIOR'
909:        │       ┌─IF BOF()
910:        │       │     GOTO TOP
911:        │       │     WAIT WIND NOWAIT 'INICIO DE ARCHIVO'
912:        │       └─ENDIF
913:        ├====CASE opc_posp = 'ULTIMO'
914:        │         WAIT WIND NOWAIT 'FIN DE ARCHIVO'
915:        └=ENDCASE
916:          SHOW GETS
917:  <=======RETURN
918:          ***************
```

```
  919:
*!*************************************************************************
**************
  920:  *!
  921:  *!       Procedure: SALIR
  922:  *!
  923:  *!       Called by: _RAA1732TG()        (function in
AT_ESCLS.SPR)
  924:  *!                : DA_ASISG.PRG
  925:  *!                : DE_EGRQ1.PRG
  926:  *!                : DE_EGRQ2.PRG
  927:  *!                : DE_EGRMN.PRG
  928:  *!                : DE_IEJUB.PRG
  929:  *!
  930:  *!            Calls: DEAC_KEYS           (procedure in
AACT_SST.SPR)
  931:  *!
  932:
*!*************************************************************************
**************
  933:      PROCEDURE salir
  934:          ***************
  935:          DO deac_keys
  936:          CLEAR READ
  937:  ========RETURN
  938:
  939:          *
*************************************************************
  940:          *        *
  941:          *        * _R8G017NY3            opc_scr VALID
  942:          *        *
  943:          *        * Function Origin:
  944:          *        *
  945:          *        * From Platform:        Windows
  946:          *        * From Screen:          ROL_Q2,      Record
Number:   45
  947:          *        * Variable:             opc_scr
  948:          *        * Called By:            VALID Clause
  949:          *        * Object Type:          Push Button
  950:          *        * Snippet Number:       1
  951:          *        *
  952:          *
*************************************************************
  953:          *
  954:
*!*************************************************************************
**************
  955:  *!
  956:  *!        Function: _R8G017NY3
  957:  *!
  958:  *!       Called by: ROL_Q2.SPR
  959:  *!
  960:  *!            Calls: &OPC_SCR.PRG
  961:  *!
  962:
*!*************************************************************************
**************
  963:      FUNCTION _r8g017ny3      &&  opc_scr VALID
```

```
  964:          #REGION 1
  965:          DO &opc_scr
  966:
  967:          *
*************************************************************
  968:          *        *
  969:          *        * _R8G017P0Y            Nivel de lectura
SHOW
  970:          *        *
  971:          *        * Function Origin:
  972:          *        *
  973:          *        *
  974:          *        * From Platform:        Windows
  975:          *        * From Screen:          ROL_Q2
  976:          *        * Called By:            READ Statement
  977:          *        * Snippet Number:       2
  978:          *        *
  979:          *
*************************************************************
  980:          *
  981:
*!*************************************************************************
***************
  982:   *!
  983:   *!        Function: _R8G017P0Y
  984:   *!
  985:   *!      Called by: ROL_Q2.SPR
  986:   *!
  987:
*!*************************************************************************
***************
  988:      FUNCTION _r8g017p0y      && Read Level Show
  989:          PRIVATE currwind
  990:          STORE WOUTPUT() TO currwind
  991:          *
  992:          * Código SHOW de la pantalla: ROL_Q2
  993:          *
  994:          #REGION 1
  995:        ┌─IF SYS(2016) = "CONS_ROL" OR SYS(2016) = "*"
  996:        │     ACTIVATE WINDOW cons_rol SAME
  997:        │     @ 1.538,24.400 SAY dgp.foto :
  998:        │         SIZE 6.923,26.000 :
  999:        │         CENTER :
 1000:        │         STYLE ""
 1001:        └─ENDIF
 1002:        ┌─IF NOT EMPTY(currwind)
 1003:        │     ACTIVATE WINDOW (currwind) SAME
 1004:        └─ENDIF
 1005:   *:  EOF: ROL_Q2.sc1
```