

*Pontificia Universidad
Católica del Ecuador*

Programas Academicos Ambato

Tecnología en Informática

**TRABAJO PRACTICO
DE GRADO**

TEMA:

Ambiente Integrado para SAP 80

AUTOR:

**Jenny Ibarra de Endara
Elizabeth Arreaga Banda**

DIRECTOR ING: Patricio Beltrán

ASESOR ING: Victor Hugo Paredes

AMBATO - ECUADOR

1993

Aprobado por los Profesores de la
Facultad de Ingeniería de la
Pontificia Universidad Católica de
Chile, Programa Académico Ambiental, luego
de haber dado cumplimiento a los
requisitos exigidos, previa la
obtención del título de Tecnología
en Informática.

DIRECTOR DE TESIS

ASESOR DE TESIS

DIRECTOR DE TESIS

RECTOR DE TESIS

DIRECTOR DE INFORMÁTICA

ACORDADO DIFERITO

El Encargado de la materia más atencioso agradece al Sr. Felicitio Bellaretti, digno Director de nuestro trabajo pedológico y laborioso profesor, haberse ocupado de mi problema en la misma forma educacional, resolviendo todo lo que está a su alcance para conseguir el mejor conocimiento científico, moral y psicológico de la persona para una formación integral de sus alumnos.

Quisiera haber sido posible también nuestra intensa colaboración mutua y personal para que han cooperado directamente y indirectamente en la realización de este trabajo.

JEREMY

7

BELTZARETTI

DEDICATORIA

A DAVID pequeña
proyección de mi vida.

JENNY

Este trabajo dedicó a mi MADRE,
quién con su AMOR y COMPRENSION supo
dirigirme en éste largo caminar, para
la culminación de una etapa más de mi
vida. A mis HERMANOS, quienes fueron mi
inspiración para seguir adelante. A mi
FAMILIA, que siempre estuvo a mi lado
en todo momento.

ELIZABETH

INDICE

CAPITULO I

MANUAL DE OPERACION

| | |
|--|---|
| 1.1. Introducción..... | 1 |
| 1.2. Descripción del Sistema..... | 2 |
| 1.3. Implementación del Sistema en el computador..... | 2 |
| 1.4. Requerimientos de Hardware y Software..... | 3 |

CAPITULO II

MANUAL DEL USUARIO

| | |
|-------------------------------------|---|
| 2.1. Introducción..... | 4 |
| 2.2. Objetivo del Sistema..... | 5 |
| 2.3. Consideraciones Generales..... | 5 |

CAPITULO III

MANUAL DE REFERENCIA TECNICA

| | |
|---|---------|
| 3.1. Arboles de programas..... | 23 - 38 |
| 3.2. Listado de Variables Listado de Programas Fuente Diagramas de Flujo..... | 40 |
| Recomendaciones..... | 336 |
| Bibliografía..... | 337 |

CAPITULO I

MANUAL DE OPERACION

INTRODUCCION.

El presente sistema me lo ha desarrollado, para que se cree en él un archivo de texto con el cual se facilitará el cálculo de estructuras dentro del SAPSO, éste es un programa que realiza el análisis estructural de edificaciones; el trabajo presentado en este sistema, disminuye en sí el tiempo de generación de datos, ya que facilita tanto a los profesionales dentro de la rama de mecánica y construcción, como a los estudiantes que son novatos, en la generación del archivo texto que se necesita para que los cálculos que ingresen dentro de SAPSO vayan de una forma ordenada, y que sea entendible, con cierta lógica de seguimiento en la manera de como se van accedando y llenando los diferentes bloques de datos que se van generando dentro del archivo de datos.

1.2. DESCRIPCION DEL SISTEMA.

El Sistema es un archivo de datos tipo texto, el cual es manipulado mediante procedimientos simples, para su ingreso, consulta, modificación y reportes.

Los mismos que tienen como objetivo general ayudar a los estudiantes de Ingeniería Civil o Ingenieros Civiles; para que mediante esta herramienta creen su propio archivo de datos para SAFBO, con las facilidades que le presta el manejo de los mensajes y de los datos.

El programa ha sido desarrollado en TURBO PASCAL Ver. 6.0 lo que garantiza la utilización de este SOFTWARE en un pequeño computador personal. Esta versión del programa esta diseñada para ser usada en el sistema operativo MS-DOS, implementado en un computador de 512 Kb de memoria y 10 Mb de disco duro.

Todos los datos de entrada ingresan con formato libre, toda las condiciones geométricas y las de carga de los elementos de la *estructura* que son ingresados por el usuario, se graban en el archivo de datos controlado por el sistema.

1.3 IMPLEMENTACION DEL SISTEMA EN EL COMPUTADOR.

Para implementar el sistema en el computador tipo IBM PC X

ó AT) o compatibles no necesita un sistema operativo MS-DOS Ver. 3.13 o cualquier versión superior, el sistema esta conformado por un solo diskette de 5 1/4 Doble Densidad (DD), el cual contiene el archivo único llamado PRINCIPA.EXE, que es el que ejecuta el programa y se le puede copiar en cualquier subdirectorio de nombre opcional, o también se puede trabajar directamente en la unidad de diskette. Además como respaldo de la información el sistema consta de dos diskettes de 5 1/4 Doble Densidad (DD), en los cuales están grabados, los archivos fuentes con extensión .FAC y .TFF respectivamente.

En caso de que sufriese alguna modificación o se deteriorara el archivo ejecutable el usuario deberá hacer una compilación del archivo PRINCIPA.FAC con destinación al disco, para obtener así un nuevo archivo ejecutable.

1.4 REQUERIMIENTOS DE HARDWARE Y SOFTWARE.

HARDWARE.

- Computador IBM/PC/XT ó AT
- Memoria RAM mínima 512 KB.
- Memoria de un disco 1 MB.
- Disco Duro con 1 Mb libre
- 1 Disk Drive de 5 1/4.
- Monitor Monocromático o color.
- Unidad Central de Proceso.
- Impresora de 80 columnas.

SOFTWARE.

- MS-DOS Ver. 3.1 o superior.

CAPITULO 11

MANUAL DEL USUARIO

INTRODUCCION.

El manejo del sistema se lo hace mediante menús de los cuales podemos elegir una opción deseada con el manejo de las teclas del cursor más la tecla [ENTER]. para mejor entendimiento del presente sistema, siempre en las últimas líneas de pantalla aparecen mensajes que orientan al usuario permitiéndole de esta manera el ingreso de sus datos sea correcto.

2.2. OBJETIVO DEL SISTEMA.

El presente sistema está realizado en el lenguaje de programación PASCAL 6.0; el cual persigue como sus objetivos principales el ingreso de datos, consultas, modificaciones y reportes de una manera eficaz y rápida.

Los datos ingresados se envían a un archivo tipo texto, cuyo nombre es generado por el usuario y está relacionado con sus requerimientos; existirán tantos archivos, como tantos grupos de datos sean ingresados por el (los) usuario(s).

2.3 CONSIDERACIONES GENERALES.

Posterior a copiar el sistema en el disco duro ó se encuentre trabajando en la unidad de diskette se va a acceder al mismo tan solo con teclear PRINCIPA más la tecla [ENTER].
Ejemplo:

```
A:PRINCIPA >
```

Aparecerá un menú del cual el usuario elegirá la opción deseada con el movimiento de las teclas del cursor más la tecla [ENTER].

Si el usuario escogió la primera opción (INGRESO DE DATOS), éste realizará el ingreso de los mismos previo a indicar la

unidad de disco a grabar más el nombre del archivo que contendrá los datos a ingresar.

Ejemplo:

```
Ingrese la Unidad de Disco: ==> A:
Nombre del Archivo       : ==> PRUEBAS
```

El Nombre del Archivo debe tener máximo 8 Caracteres

Si el archivo existe y su ingreso es completo el sistema automáticamente le pedirá que ingrese otro nombre de archivo ya que éste está completo. En caso de que el usuario cree un nuevo archivo, empezará el ingreso de sus datos indicando primero el título que llevará (TITLE) y será considerado como una línea informativa. Posterior a su ingreso debe indicar el control de salida de información (SYSTEM) representado con las variables:

P = Dispositivo de salida, ingresará el valor de uno (1) si el usuario desea que SAP80 envíe los resultados del análisis a la consola o monitor; ó ingresará el valor de cero (0) si los resultados van hacer enviados al disco.

L = En esta variable deberá ingresar el número de estados de carga.

N = Aquí ingresará el número total de nudos de la estructura.

En cada uno de estos ingresos el sistema le preguntará si desea INGRESO HA CILLO VALLETO, en caso de que NO el usuario deberá presionar la tecla (B, ó, n) e ingresar el nuevo valor.

Luego de este ingreso pasa el sistema a pedir la información de las coordenadas de los nudos (JOINTS); tantas como el número de nudos ingresados, se debe indicar el número de nudo, las coordenadas en las direcciones X,Y,Z globales; además, tiene la opción al usuario de hacer generaciones, pudiendo ser estas lineales o cuadráticas, teniendo como orden lógico el siguiente:

GENERACION LINEAL: (G=)

Nudo Inicial,
Nudo Final,
Incremento de Nudo.

GENERACION CUADRATICA: (Q=)

Nudo Horizontal Inicial,
Nudo Horizontal Final,
Nudo Vertical Inicial,
Nudo Vertical Final,
Incremento Horizontal,
Incremento Vertical.

Tiene la posibilidad el usuario de presionar la tecla [B ESPACIADORA], en caso de que desee terminar el ingreso y regresar al Menú Principal grabando lo ingresado.

Para el mejor entendimiento de estos ingresos, el sistema siempre le estará presentando por pantalla mensajes que le servirán como guía para el ingreso correcto de sus datos.

Ejemplo:

| NUMERO TOTAL DE NUDOS A INGRESAR : 10 | | | | | |
|---------------------------------------|-----------------|------|--------|---------------------|-----|
| NUMERO DE NUDO | COORDENADAS EN: | | ESCALA | GENERACION | |
| | X= | Y= | | Z= | S/N |
| 1 | 0.00 | 0.00 | 1.00 | Lineal ó Cuadrática | |
| 4 | 15.00 | 0.00 | 0.00 | G=1.00, 4.00, 1.00 | |
| 5 | 0.00 | 7.00 | 1.00 | | |

Presione MAYÚS ESPACIADORA en número de nudo y regrese al MENÚ PRINCIPAL ...

Posteriormente pasará a pedir las Restricciones (RESTRIC) de los nudos, entendiéndose por esto a los seis componentes desplazamientos, tres traslaciones globales, y tres giros globales. Estas Restricciones, pueden tomar los valores de 0 ó 1, 0 desplazamientos permitidos e impedidos, respectivamente. El sistema realiza un conteo interno para que usuario ingrese solo

La cantidad de nudos que declaró al inicio de su ingreso. El orden de ingreso en este bloque es el siguiente:

- Numero del primer nudo
- Numero del último nudo
- Incremento de número de nudo
- Restricción para traslación X
- Restricción para traslación Y
- Restricción para traslación Z.
- Restricción para rotación X
- Restricción para rotación Y
- Restricción para rotación Z.

Ejemplos:

| RESTRICCIONES DE LOS NUDOS | | | |
|----------------------------|------------|------------|-----------------|
| ID Nudo | 73 Nudo | Incrmento | Restricciones |
| 1111111111 | 1111111111 | 1111111111 | Rest,11111111 |
| 4444444444 | 1111111111 | 1111111111 | Rest,4,1,1111,9 |

Sus datos son correctos S/N: 1

Seguidamente le pedirá que indique el número de apoyo especiales (SPECIALS), si los desea declarar, caso contrario deberá presionar [ENTER] o ingresar el valor de cero (0) si no desea declarar apoyos especiales. Para el ingreso de estos en orden lógico sea:

- Número del primer nudo
- Número del último nudo
- Incremento del número de nudo
- Constante de Resorte para traslación X
- Constante de Resorte para traslación Y
- Constante de Resorte para traslación Z.
- Constante de Resorte para rotación X
- Constante de Resorte para rotación Y
- Constante de Resorte para rotación Z.

Para este ingreso el permite al usuario utilizar la [BAK] [ESPACIADORA] en el número del primer nudo, en caso de que dese concluir el ingreso y salir al Menú Principal.

Ejemplo:

```

      APOYOS ESPECIALES
Ingrese la cantidad de apoyos a declarar: 000

Nudo      Nudo      Incremento      K
Inicial   Fijado
          X           Y           Z
          Kx      Ky      Kz      Krx      Kry      Krz
000       000       000      1.0E+06  1.0E+06  1.0E+06  0.0E+00  0.0E+00  0.0E+00

K = Constante de Resorte:
Kx = Para Traslación-x ; Ky = Para Traslación-y ; Kz = Para Traslación-z
Krx = Para Rotación-x ; Kry = Para Rotación-y ; Krz = Para Rotación-z
  
```

Al terminar este ingreso pasará a pedir que se indique el número de concatenaciones de los nudos a declarar (CONSTRAINTS) debiendo el usuario ingresar cero (0) o [ENTER] si no desea declarar concatenación de nudos; caso contrario el orden d

ingreso es el siguiente:

- Número del primer nudo
- Número del último nudo
- Incremento de nudos
- Junta dependiente del desplazamiento X
- Junta dependiente del desplazamiento Y
- Junta dependiente del desplazamiento Z.
- Junta dependiente de la rotación X
- Junta dependiente de la rotación Y
- Junta dependiente de la rotación Z.

Incremento en el nudo para las series:

- Nudo C1
- Nudo C2
- Nudo C3
- Nudo C4
- Nudo C5
- Nudo C6.

Ejemplo:

| ENCADENACION DE NUDOS | | | | | | | | | | | | | | | |
|---|------------|------------|----|----|----|----|----|----|----|----|----|----|----|----|--|
| Ingrese la cantidad de concatenaciones a declarar : 3 | | | | | | | | | | | | | | | |
| Nudo Inicial | Nudo Final | Incremento | 1 | | | | | | 2 | | | | | | |
| | | | C1 | C2 | C3 | C4 | C5 | C6 | C1 | C2 | C3 | C4 | C5 | C6 | |
| 1 | 4 | 1 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 4 | 1 | 2 | |
| 2 | 5 | 1 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 4 | 1 | 2 | |
| 3 | 6 | 1 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 4 | 1 | 2 | |

Presione MAYÚSCULA ESPACIADORA en número de nudo y regrese al MENU PRINCIPAL ...

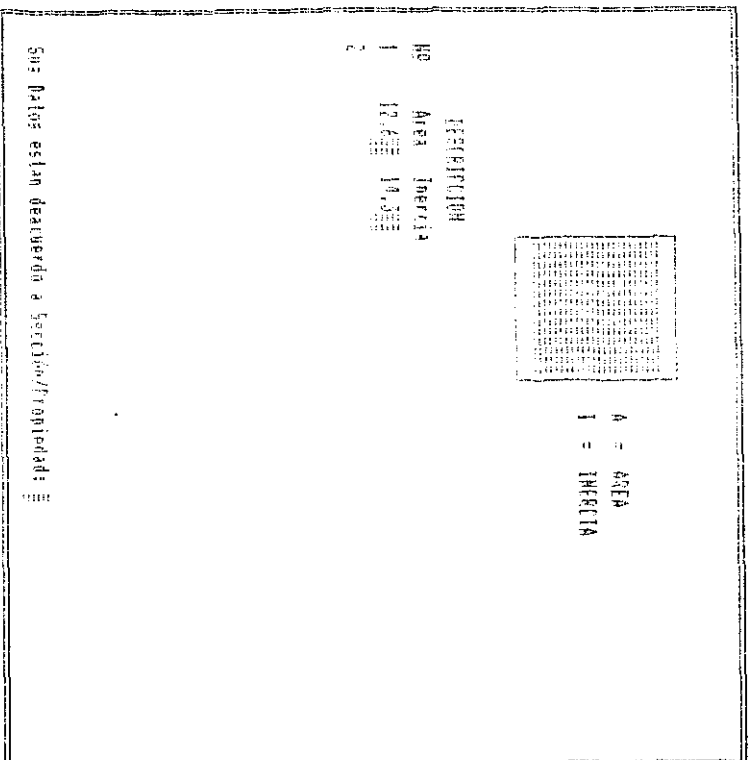
Posterior a este ingreso deberá indicar el número de propiedades de sección de los miembros, el módulo de elasticidad

y el número de cargas de vano (distribuidas), bloque de datos conocido como (FEAHE), las cuales ingresarán de la siguiente manera:

PROPIEADES DE SECCION. (MFE)

Tiene dos posibilidades de ingreso que son: por sección donde el sistema le presenta los gráficos de los tipos estructura que puede ingresar y debiendo el usuario dar los valores, para las bases y alturas correspondientes; y por propiedad debiendo ingresar el Area y la Inercia respectiva.

Ejemplo:

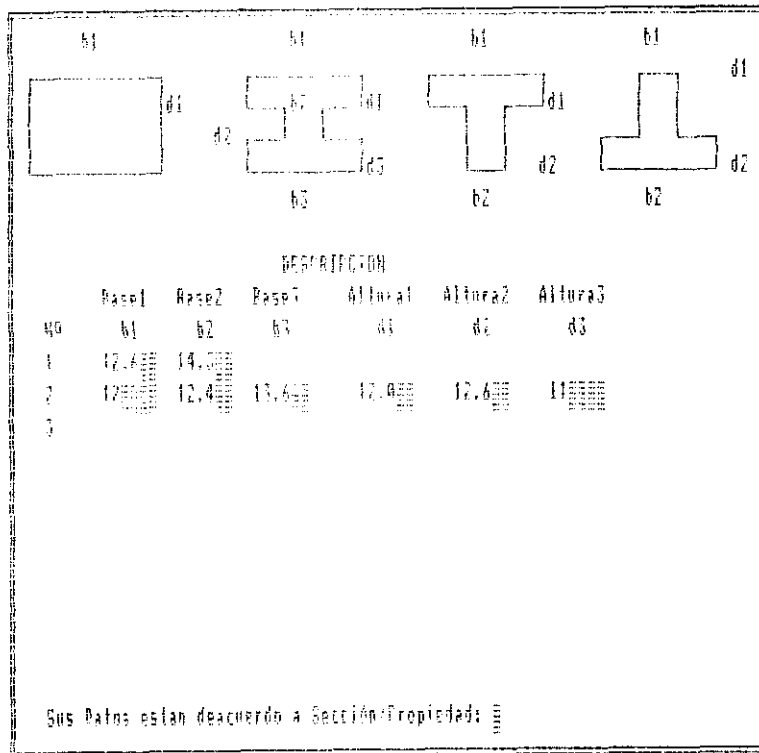


NUMERO DE CARGAS DISTRIBUIDAS. (MFE)

El usuario declarará tantas como ingreso al inicio de ese bloque y el orden de ingreso es el siguiente:

Carga distribuida en dirección X,Y,Z.

Ejemplo:



Luego el usuario ingresará el número de Tipo Miembros que desea declarar. Los cuales serán declarados según los requerimientos del usuario, siendo el orden de ingreso:

- Número de miembro
- Primer nudo
- Segundo nudo
- Tipo de miembro
- Generación:
 - Número de miembros a generar
 - Miembros a incrementar
 - Incremento de nudo en el extremo i
 - Incremento de nudo en el extremo j
- Coordenadas X,Y,Z
- Estados de carga:
 - Carga Viva
 - Carga Muerta
 - Carga Sismica

Ejemplo:

```

NUMERO TOTAL DE MIEMBROS A INGRESAR : 000

Material  (E)  (M)  Tipo material  Generación  Coordenadas  Estados Caras
      Edo  Mdo
      S/N
      S/N
      S/N

000000  200  500  M-100  4-1-1-1-1-1-1-1-1-1
000000  400  800  M-100
  
```

A continuación deberá indicar la cantidad de cáscaras, placas, y membranas a declarar o ingresar (QUAD). Pudiendo ingresar cero (0) si no desea declarar; caso contrario el orden de ingreso es:

- Factor gravitacional para X
- Factor gravitacional para Y
- Factor gravitacional para Z
- Módulo de Elasticidad
- Módulo de Poisson
- Peso sobre volumen

Ejemplo:

```

P. CASCARAS, PLACAS Y MEMBRANAS

Ingrese la cantidad de tipos de materiales a declarar: 2000

Factor Gravitacional      Módulo      Módulo      Peso/      Espesor
X      Y      Z      Elasticidad  Poisson     Volumen

100000  100000  200000  2000      3000      1000      210000

Presione PAUSA (SPACIARORA en Factor Gravitacional) X
y RETORNA al MENU PRINCIPAL ....
  
```

Seguido a este ingreso deberá indicar el número de carga concentradas a dar lugar (LOADS), pudiendo ingresar cero (0) si no desea declarar datos; caso contrario el orden de ingreso es:

- Número de primera junta
- Número de segunda junta
- Incremento de número de junta
- Número de estado de carga
- Rotación aplicada en dirección X
- Fuerza aplicada en dirección Y
- Fuerza aplicada en dirección Z.
- Momento aplicado alrededor del eje X
- Momento aplicado alrededor del eje Y
- Momento aplicado alrededor del eje Z.
- Desplazamiento aplicado en dirección X
- Desplazamiento aplicado en dirección Y
- Desplazamiento aplicado en dirección Z.
- Rotación aplicada alrededor del eje X
- Rotación aplicada alrededor del eje Y
- Rotación aplicada alrededor del eje Z.

Ejemplo:

| CARGAS CONCENTRADAS EN LOS NUDOS | | | | | | | | | |
|----------------------------------|-------|-----------------------------------|--------|---------|-------|----------|------------|-------|--------|
| Ingreso | | La cantidad de cargas a declarar: | | ZONES | | | | | |
| Nudos | Punto | Est. | Fuerza | direcc. | Mom. | alreded. | Desplazam. | Rot. | al eje |
| 19 | 29 | | Carga | X | Y | Z | X | Y | Z |
| 19 | 29 | | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 19 | 29 | | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |

d = Regiones donde Aplicadas en las Direcciones X, Y, Z
 r = Rotaciones Aplicadas Alrededor de los Ejes X, Y, Z

Ejemplos:

| GRAFICO DE LA ESTADISTICA | | | |
|-----------------------------|-----|---------------|---------------|
| Vectores de vista para modo | | Modo 1 vector | Modo 2 vector |
| 1 | 2 | horizontal | horizontal |
| DES | DES | DES | DES |

Presione PARA ESPACIAR en Vector de vista para Modo 1

Para el ingreso de cada uno de estos bloques de datos, el sistema le da la posibilidad al usuario de corregir los datos ingresados, en caso de que hubiese algún error; además de presentarle continuamente mensajes a los cuales el usuario deberá poner mucha atención para que le facilite el ingreso.

El sistema le da la opción de presionar la [BARRA ESPACIADORA], en momentos indicados si el usuario desea abandonar el ingreso y retornar al menú principal.

En caso de que el usuario por un error abandono el ingreso desea continuar el mismo, deberá elegir en el menú principal la opción INGRESO DE DATOS e ingresar con el nombre de archivo de

cual alabarlo en ingreso y automáticamente el sistema, mostrará en una pantalla el último de los bloques de datos que ingreso con todos sus datos.

Si el usuario eligió la opción cambios y consultas, aparecerá un menú que contiene cada uno de los bloques de datos ingresados, a lo que accederá utilizando las teclas de movimiento de cursor más la tecla [ENTER]. Al ingresar a cada una de] sub-opciones, el sistema le pedirá indique la unidad de disco y nombre del archivo del cual va a consultar o modificar. Si nombre del archivo es erróneo le mostrará un mensaje que indica que el archivo no existe y le pide, ingrese otro nombre archivo. Posteriormente le aparecerá un mensaje que le indica si desea realizar consultas ó modificaciones a ese bloque de datos, debiendo presionar el usuario la tecla correspondiente la letra que el sistema le muestra en color más intenso.

Ejemplo:

| | |
|--|--|
| FONTOPIA PRESENCIA CAJUELA DEL ECUADOR | |
| PROGRAMAS DE APRENDIZAJE | |
| ANALISIS DE ESTADISTICAS | |
| MENU DE CAMBIOS Y CONSULTAS | |
| OPCIONES | |
| Nombre de la estructura | |
| Control de salida de Interacción | |
| Coordenadas de los Nodos | |
| Especificaciones de los Nodos | |
| Apuvos Especiales | |
| Concestrucción de Nodos | |
| Estrategias de los Miembros | |
| P.Lasercas, Placas y Membranas | |
| Cargas Coordinadas en los Nodos | |
| Combinación de Cargas (fuerzas) | |
| Análisis de la estructura | |
| Valor <input type="checkbox"/> Meno <input type="checkbox"/> Principal | |

Consulta y/o Modifica el nombre de la estructura

En caso de escoger consultas el sistema le presenta todo el bloque de datos ingresado pidiéndole que presione cualquier tecla para retornar al menú anterior.

En caso de escoger modificaciones el sistema le muestra el bloque completo de sus datos y las últimas líneas de la pantalla están destinadas para que usuario indique que línea va a modificar e ingrese los datos modificados.

Ejemplo de Consultas y Modificaciones.

| COORDENADAS DE LOS NUDOS | | | | | | |
|-------------------------------------|--------------|-----------------|--------|--------|--------|------------|
| NUMERO TOTAL DE NUDOS INGRESADOS 15 | | | | | | |
| N | # | COORDENADAS EN: | | | ESCALA | GENERACION |
| LINIA | NUDO | X= | Y= | Z= | | |
| 1 | 1 | X=0.00 | Y=0.00 | Z=1.00 | 5=1 | |
| 2 | 2 | X=1.20 | Y=0.35 | Z=0.12 | | |
| 3 | 3 | X=0.50 | Y=0.50 | Z=0.90 | | |
| 4 | 4 | X=0.25 | Y=0.56 | Z=0.37 | | |
| 5 | 5 | X=0.00 | Y=0.00 | Z=0.00 | | |
| 6 | FIN DE DATOS | | | | | |

QUE NO DE LINEA DESEA MODIFICAR : 000

Al igual que en el ingreso de los datos, el sistema le brinda la posibilidad de corregir sus datos, si cometi6 algùn error, caso contrario el sistema grabará su nuevo ingreso y retornará al menú anterior.

El análisis que se muestra por el menú principal, indica que luego que el usuario ha creado su archivo de texto, deberá salir al sistema operativo y tipiar SAPBO más nombre del archivo generado y ENTERAR.

Si el usuario elige los Reportes le aparecerá un mensaje la pantalla que lo pedirá al usuario que indique la unidad disco y el nombre del archivo del cual va a sacar el reporte, caso de que el archivo no exista, el sistema le indicará y pedirá ingreso otro nombre de archivo; si este es correcto aparecerá un mensaje que lo indicará que prepare la impresora presione una tecla cualquiera para iniciar el reporte, recultado de este reporte es un listado de todos los datos ingresados por el usuario.

Ejemplo:

```

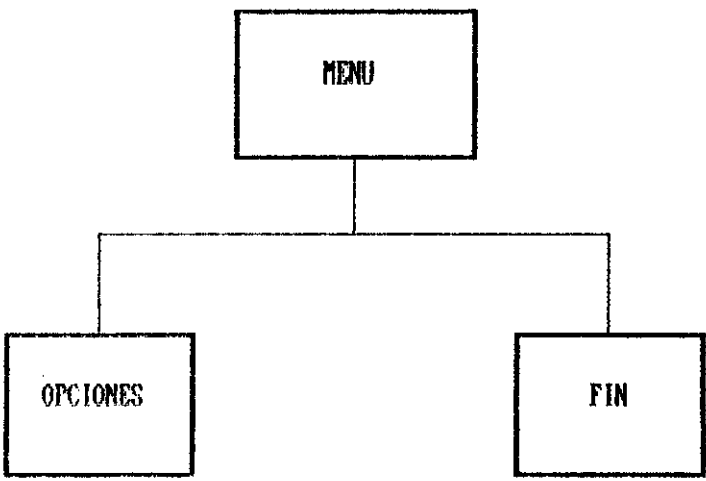
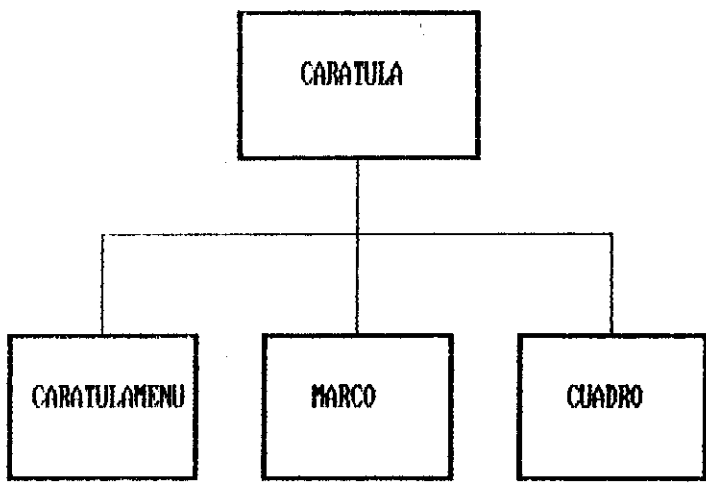
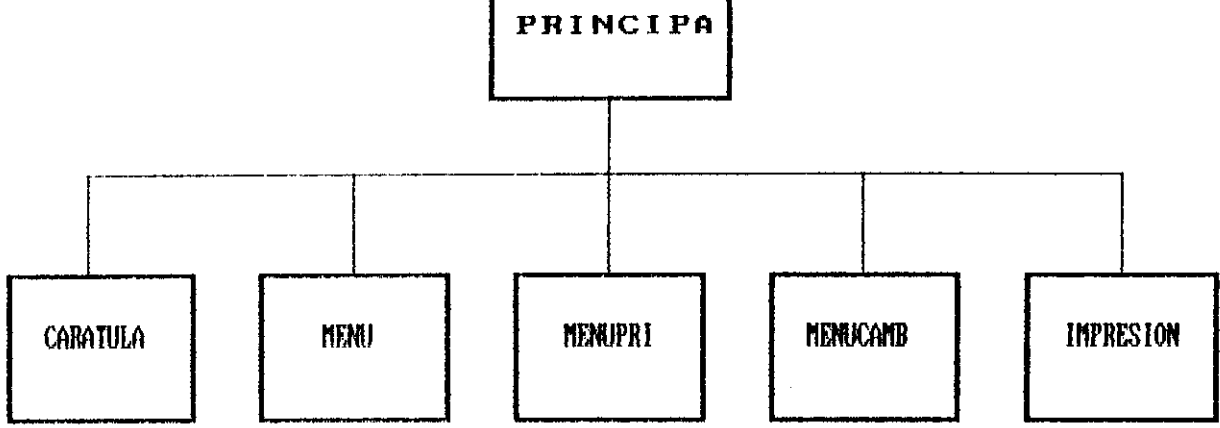
MENU : Leona Pastores
SYSTEM : UNIDAD DE SALIDA DE LA OPERACION
F=1  W=3  I=2
C FIN DE SISTEMA
REPORTES : CATEGORIAS DE LOS REPORTES
1  000.00  1=0.00  1=1.00  S=1
2  000.00  1=0.00  1=0.00
3  000.00  1=0.00  1=0.00
4  000.00  1=0.00  1=0.00
5  000.00  1=0.00  1=0.00
C FIN DE REPORTES
DETALLES : FECHAS DE LOS REPORTES
1  2  1  R=1,1,1,1,1
2  3  1  R=0,0,0,0,1
C FIN DE DETALLES
SPECIALS : AFINOS ESPECIALES ?
1  2  3  A=0,000,3,000,3,000,3,000,3,000,3,000
2  3  1  E=0,000,1,000,2,000,1,000,3,000,1,000
C FIN DE SPECIALS
CONSULTAS : CONSULTACION DE REPORTES 1

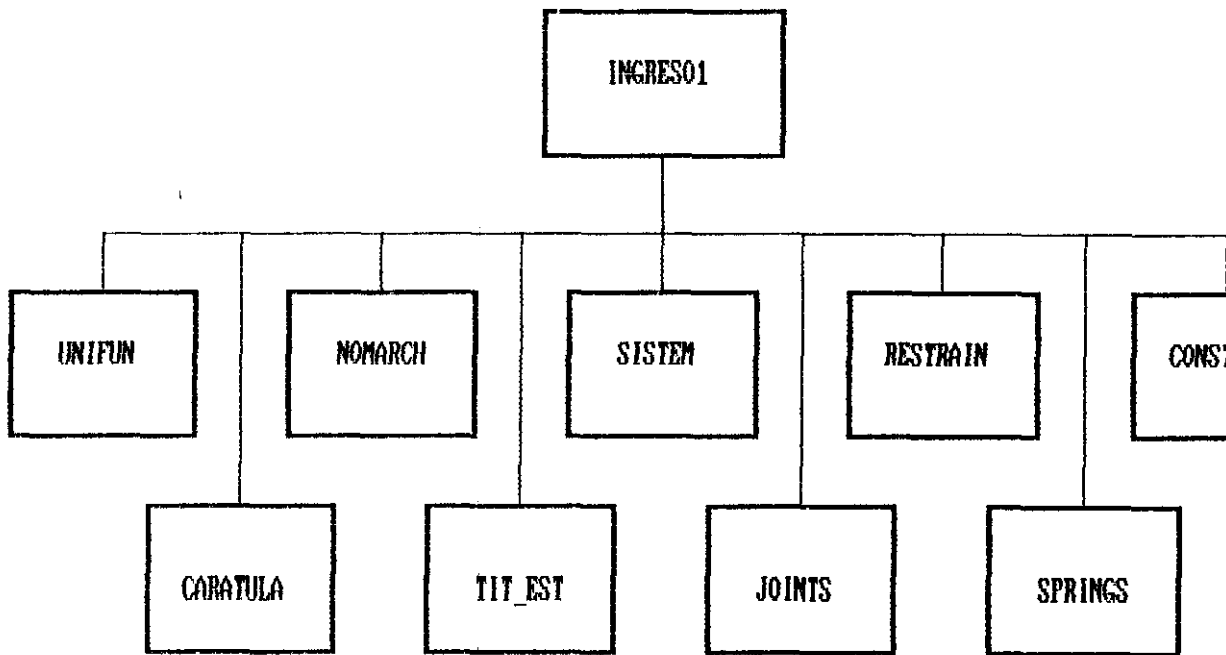
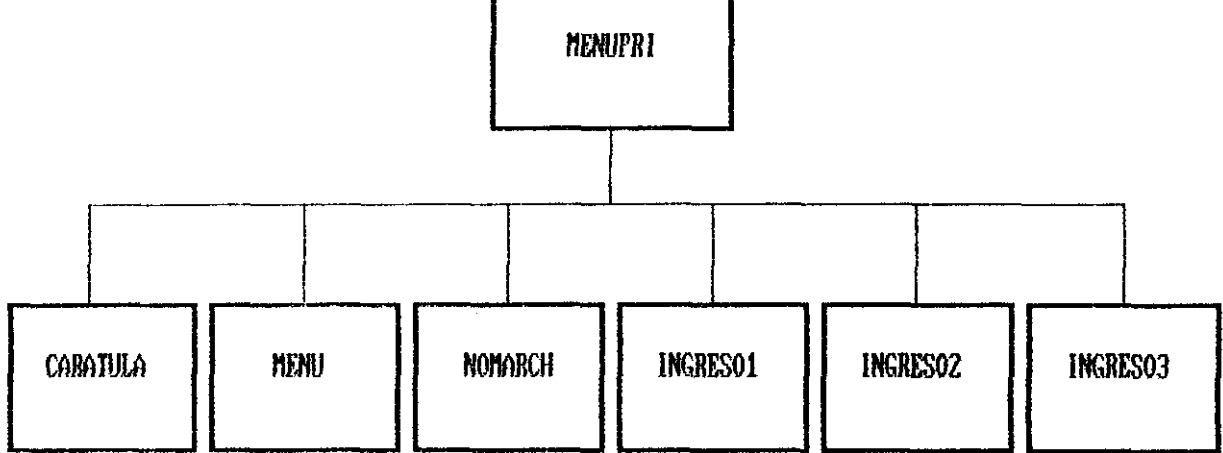
```

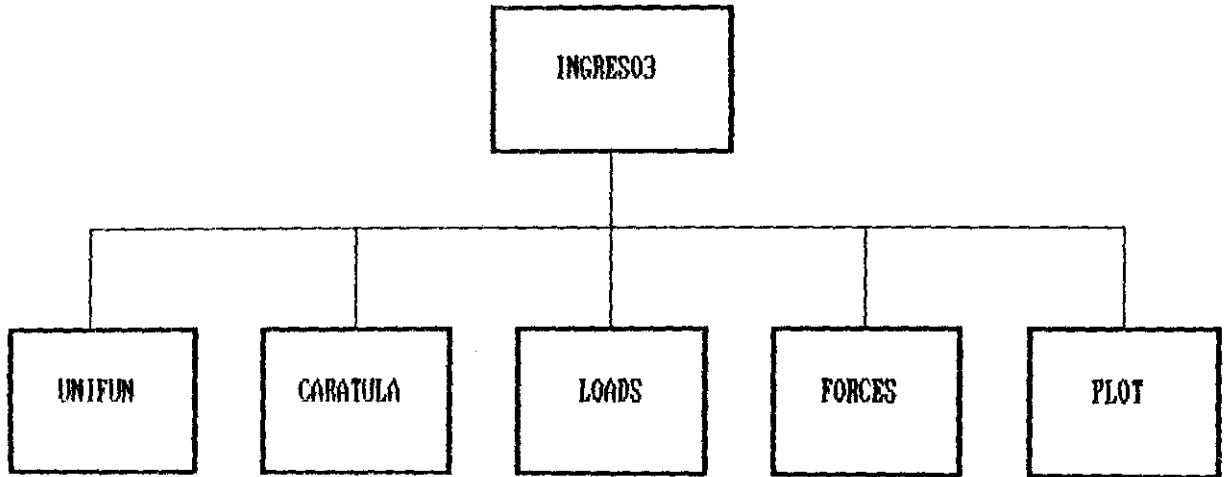
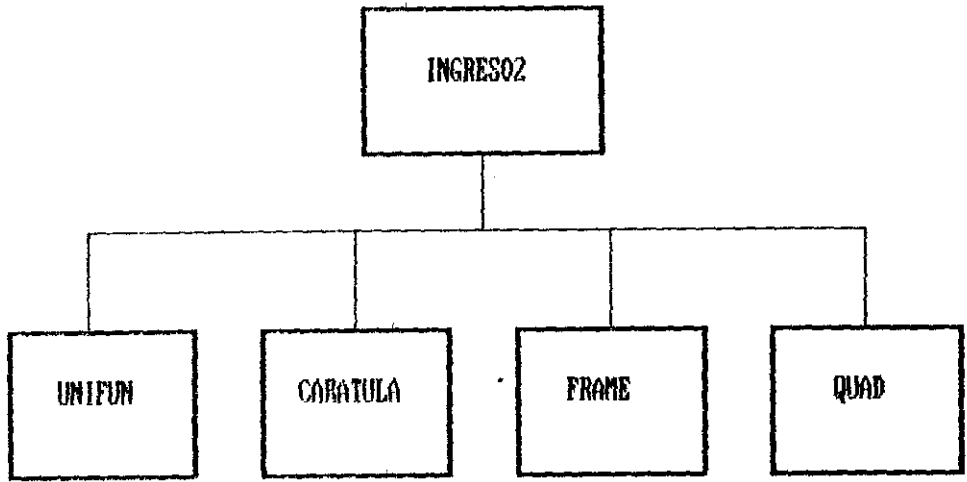
Si el usuario elige salir del sistema este automáticamente terminará y volverá al sistema operativo.

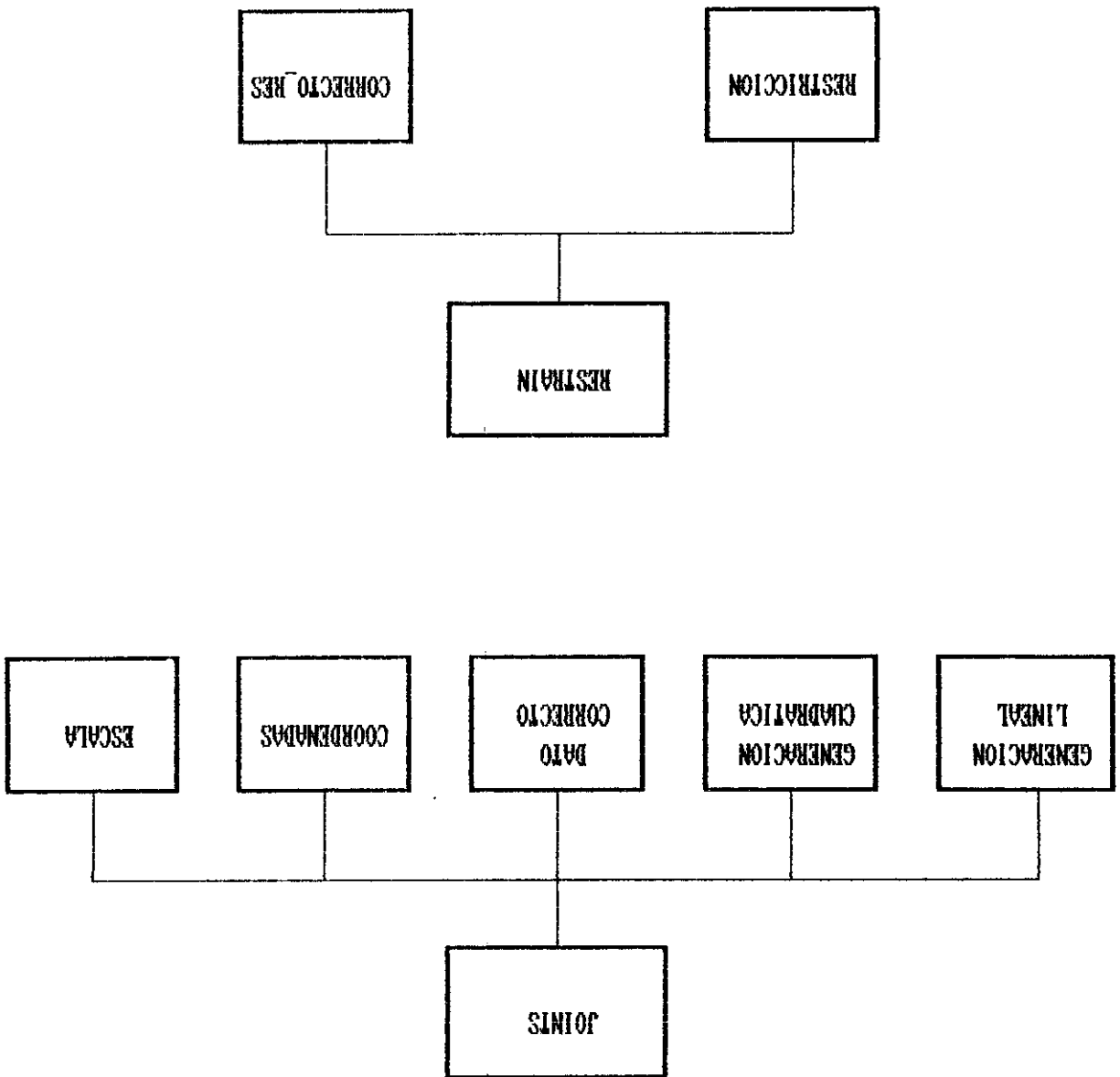
CAPITULO III

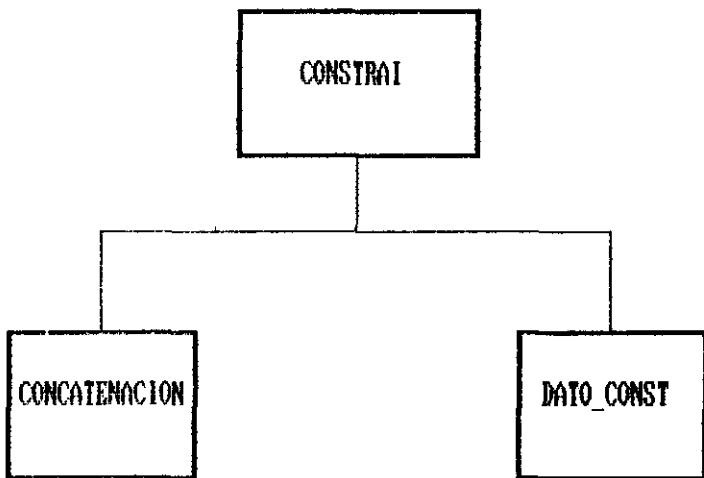
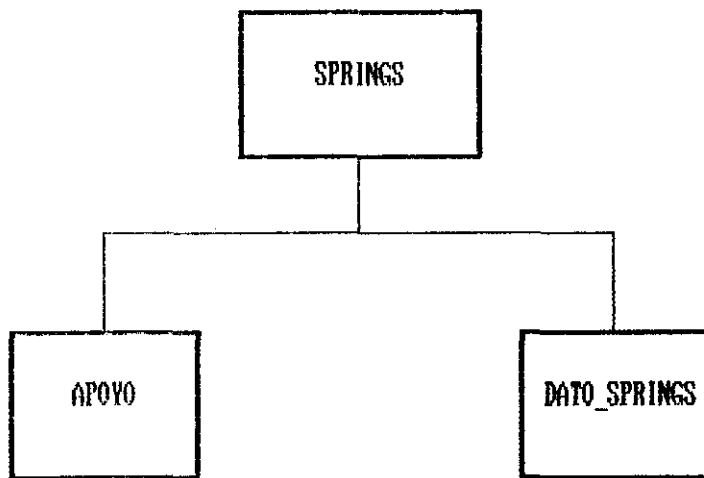
MANUAL DE REFERENCIA TECNICA

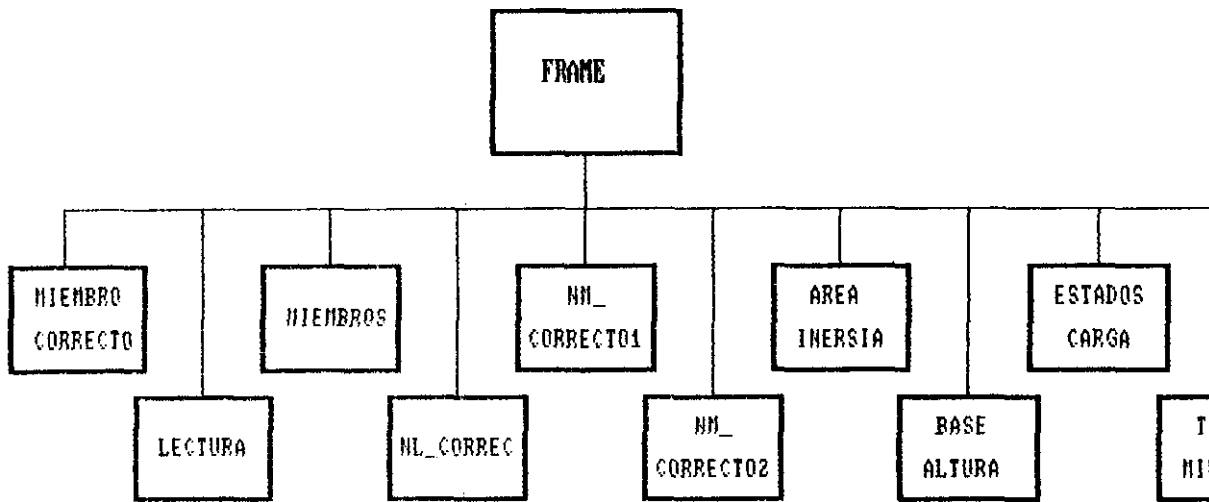


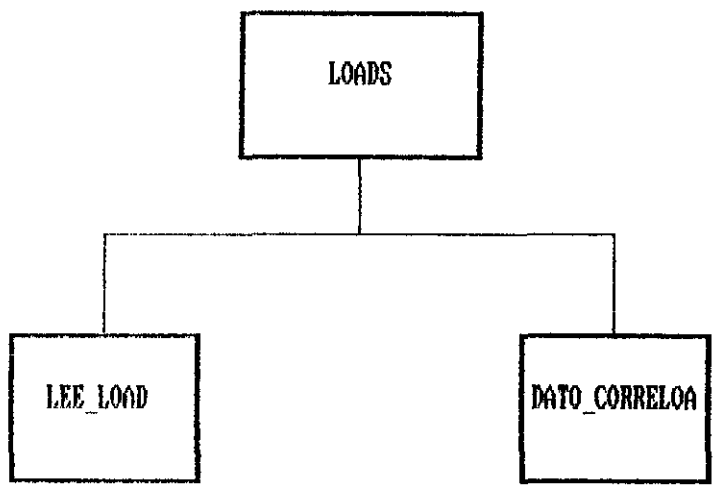
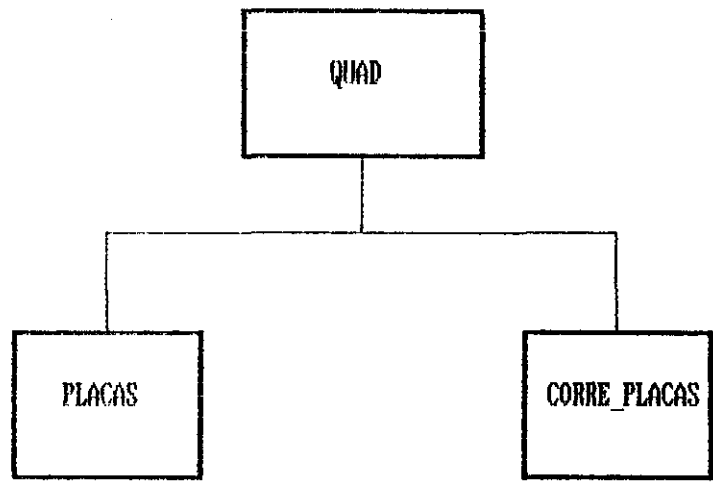


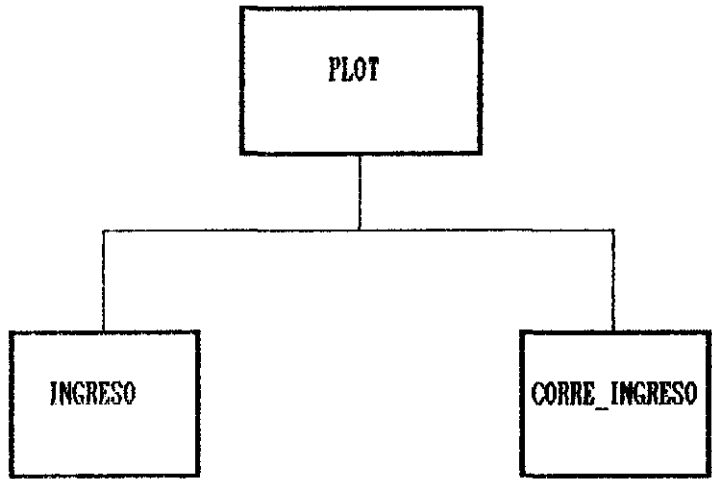
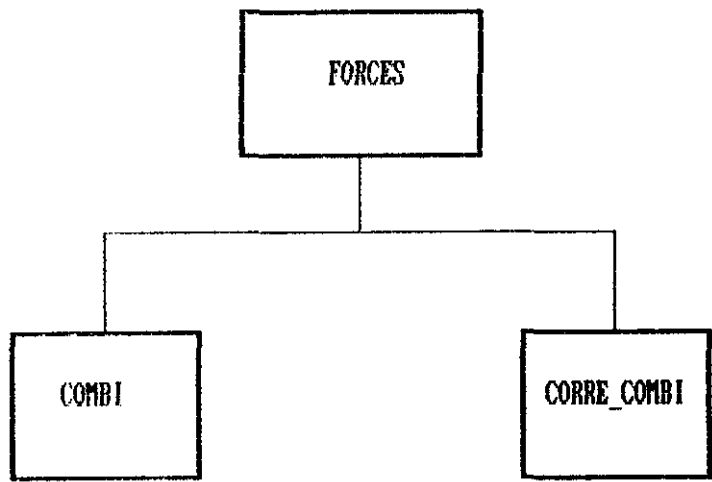


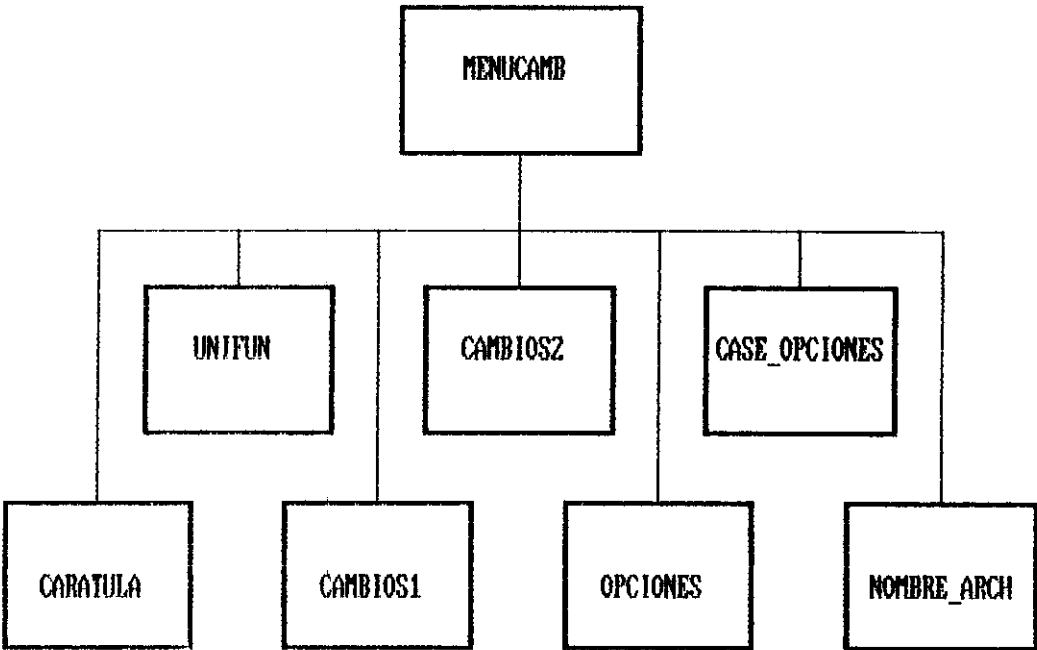


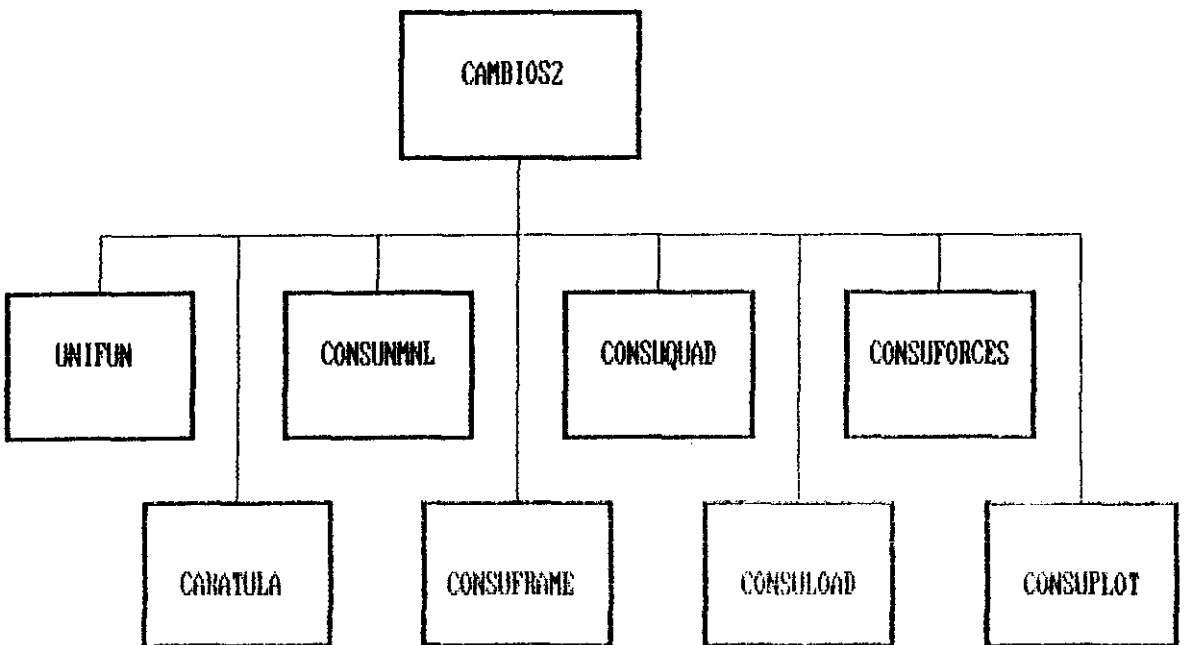
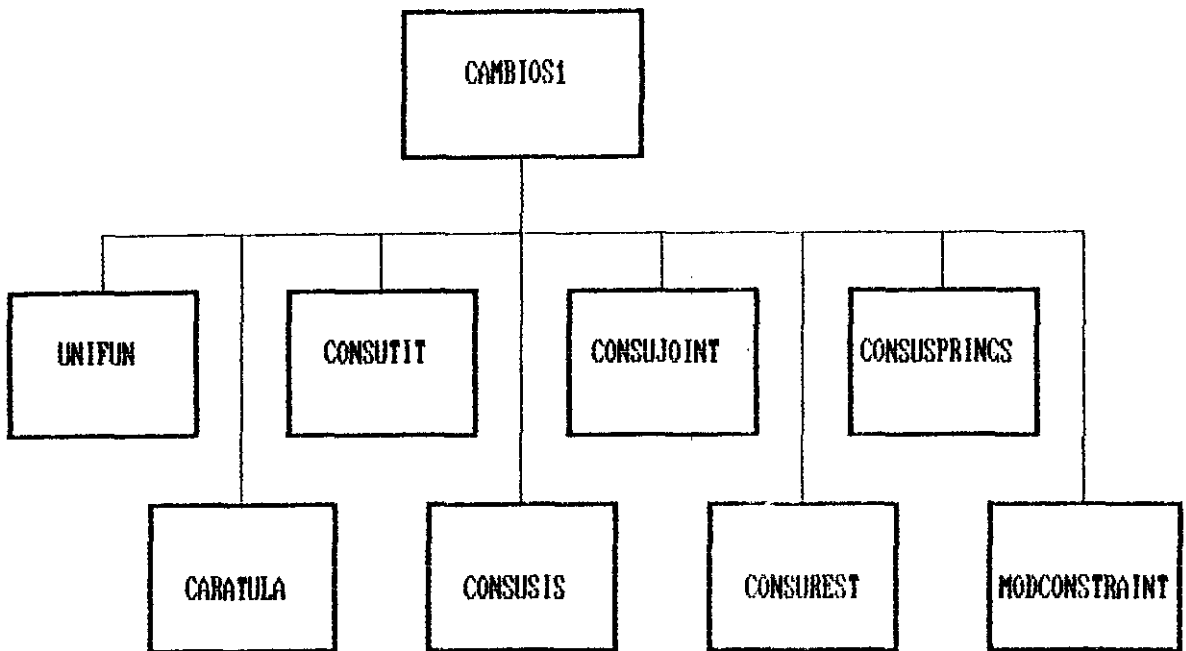


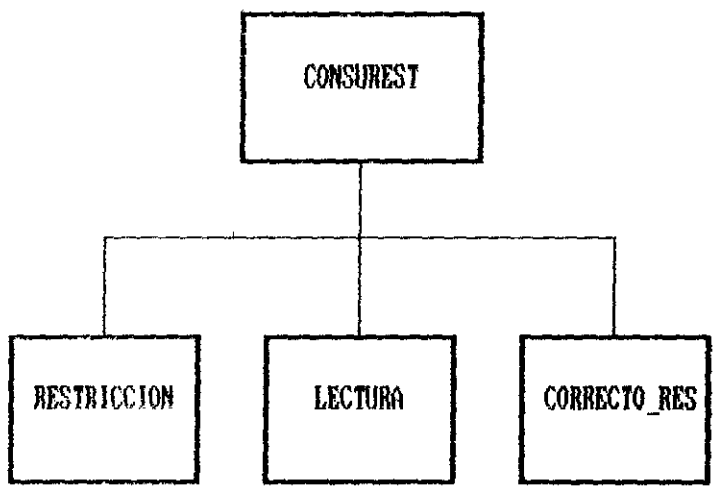
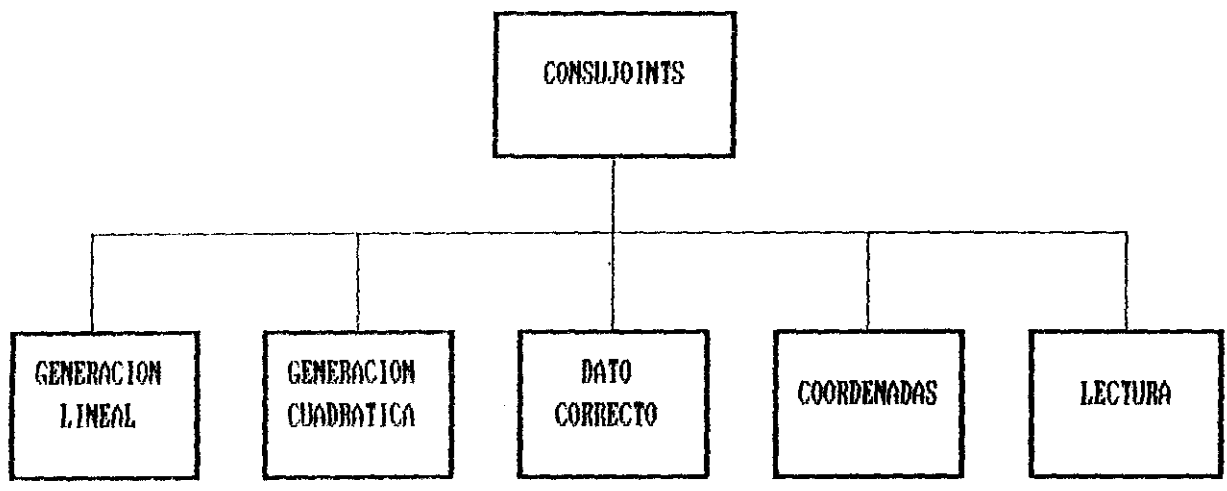


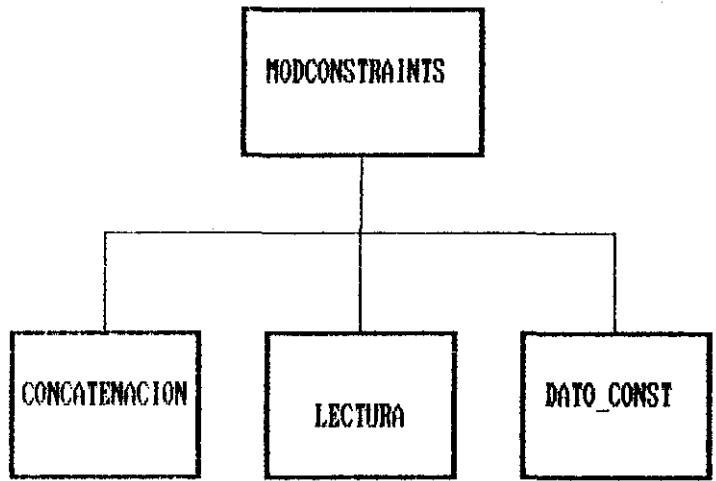
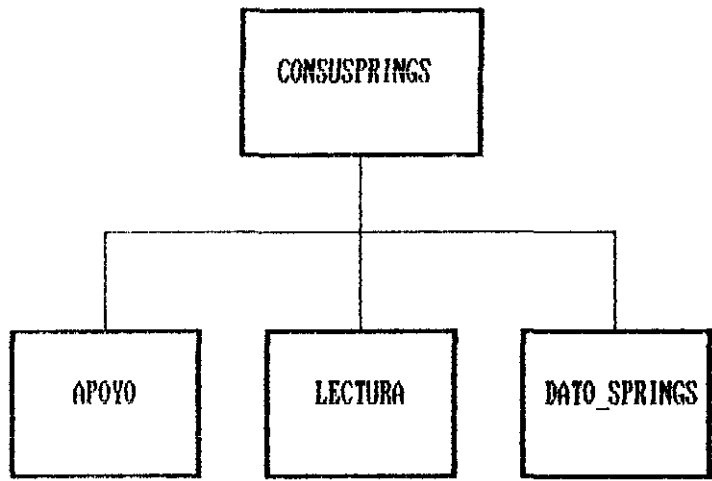


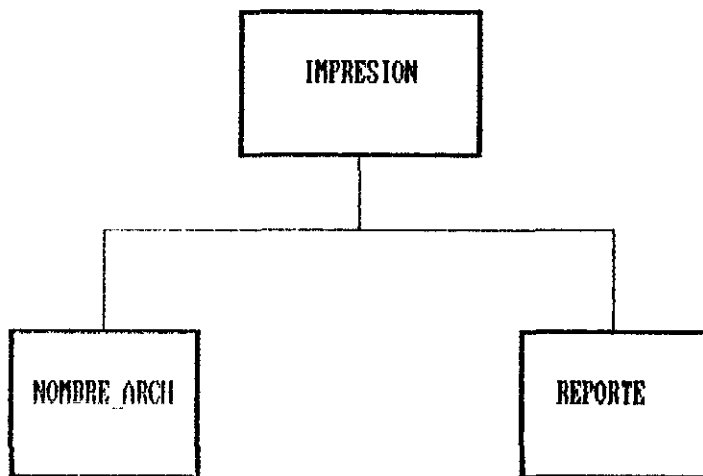
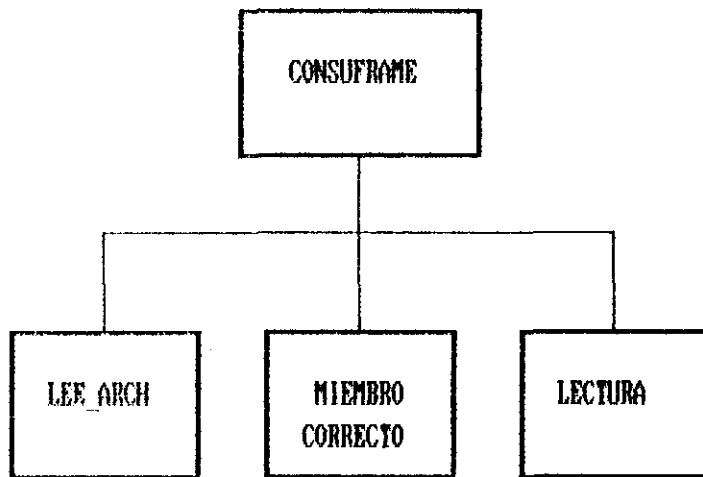


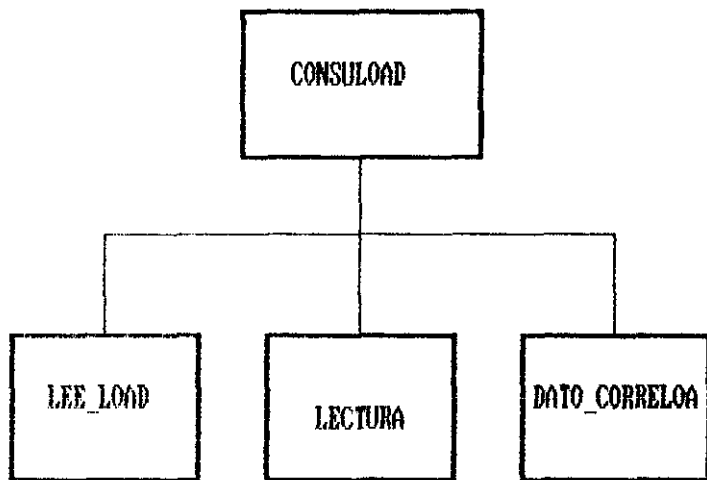
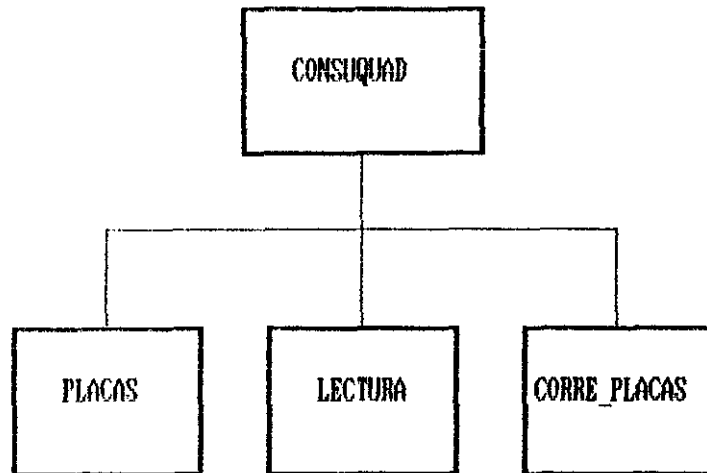


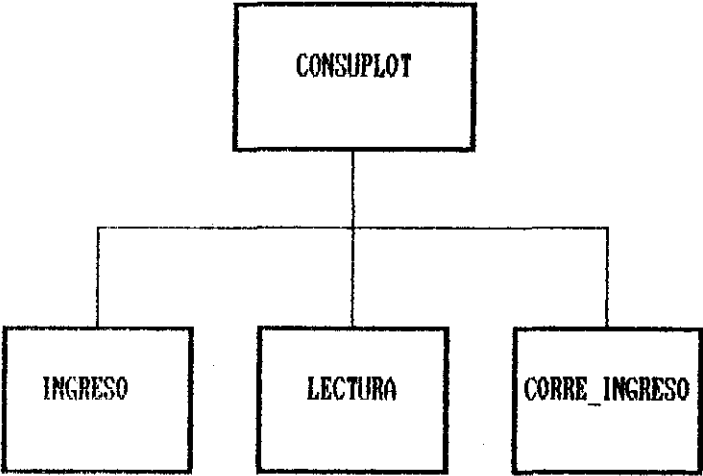
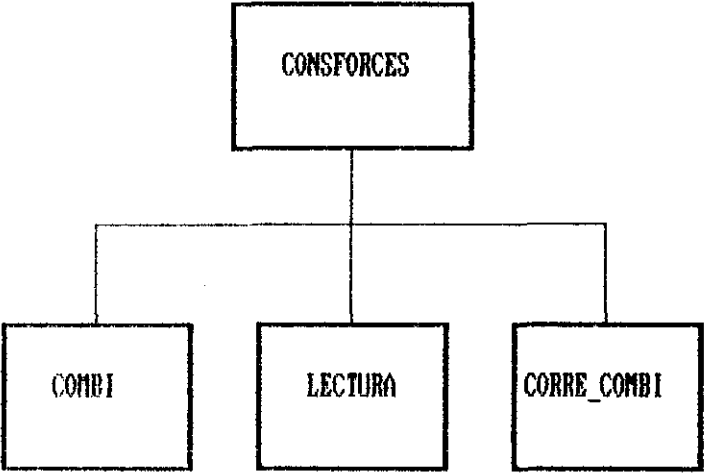


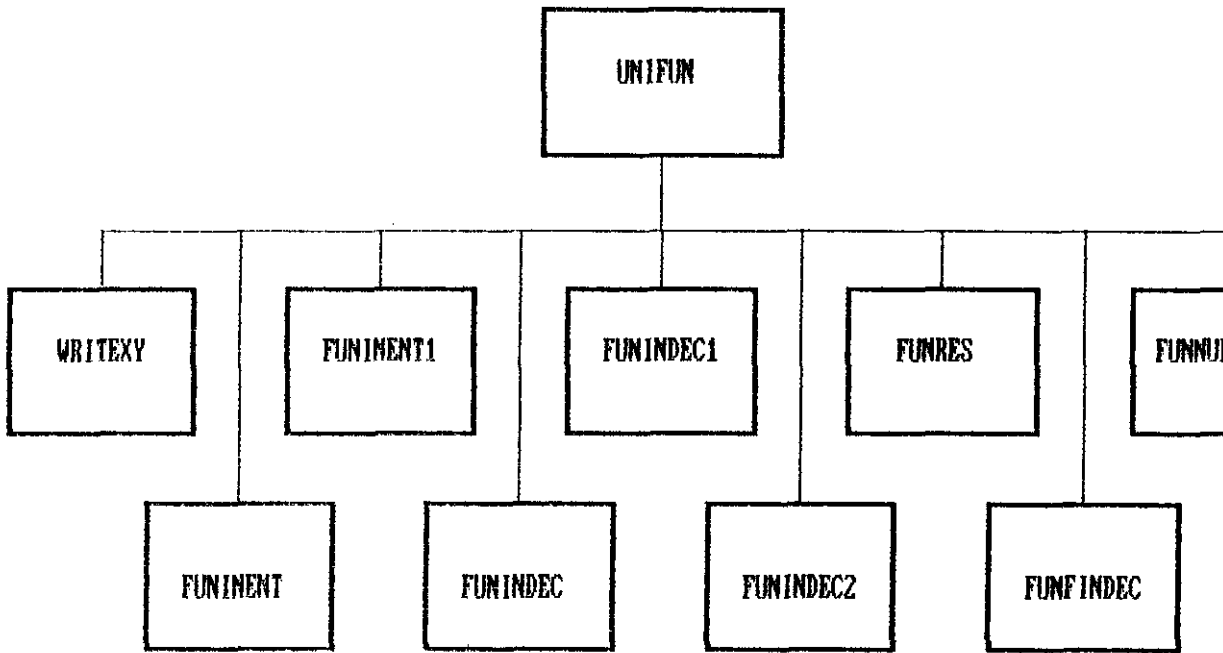












LISTA DE VARIABLES

VARIABLES GLOBALES

| NOMBRE | TIPO | DESCRIPCION |
|--------------|---------|---|
| LC | Byte | Valor del ordinal de lund |
| LN | Integer | Número de líneas |
| LCO (línea) | Integer | Para varias líneas |
| LC (columna) | Integer | Para varias columnas |
| EXTEN | String | Para aceptar nombre del archivo |
| PROHIBIT | String | Para leer nombre del archivo |
| ARCHIVO(S) | Text | Variable de archivo |
| BATERIA | Char | Para aceptar dato correcto Y/N |
| CONTRIBUIR | Char | El dato para corregir |
| SW | Char | Ayuda |
| LEER | Byte | Para la posición de lectura en el archivo |
| CONT | Integer | Variable para leer |

LISTA DE VARIABLES

NOMBRE DEL PROGRAMA: ELBIBLIOTECA

| NOMBRE | TIPO | DESCRIPCION |
|--------------|------|-----------------------|
| LEGALE | Byte | Variable de control |
| DEL | Byte | Variable de control |
| AYUDA | Byte | |
| SW | Char | Switch de ayuda |
| OPCION | Char | Variable de selección |
| ELBIBLIOTECA | Char | Variable de opciones |

```
PROCEDURE PRINCIPA;
USES CRT,CARATULA,MENU,MENUPRIA,MENUCAMB,IMPRESION;
VAR
```

```
LUGAR,FER,SITUA : BYTE;
SW,MUEVE,ELECCION:CHAR;
```

```
PROCEDURE CASE OPCIONES;
```

```
BEGIN
```

```
  CASE SITUA OF
```

```
  15: BEGIN
```

```
    TEXTCOLOR(14);
    GOTOXY(22,15);WRITE('          INGRESO DE DATOS');
    GOTOXY(15,23); WRITE('Selecciona el archivo e ingreso de datos');
    TEXTCOLOR(7);
```

```
  END;
```

```
  16: BEGIN
```

```
    TEXTCOLOR(14);
    GOTOXY(22,16);WRITE('          CAMBIOS Y CONSULTAS');
    GOTOXY(15,23); WRITE('Consulta y Pregunta si desea realizar algún cambio');
    TEXTCOLOR(7);
```

```
  END;
```

```
  17: BEGIN
```

```
    TEXTCOLOR(14);
    GOTOXY(22,17);WRITE('          ANALISIS');
    GOTOXY(15,23); WRITE('Enlaza al SAP80 y realiza los cálculos');
    TEXTCOLOR(7);
```

```
  END;
```

```
  18: BEGIN
```

```
    TEXTCOLOR(14);
    GOTOXY(22,18);WRITE('          REPORTES');
    GOTOXY(15,23); WRITE('Muestra el listado de los datos ingresados');
    TEXTCOLOR(7);
```

```
  END;
```

```
  19: BEGIN
```

```
    TEXTCOLOR(14);
    GOTOXY(22,19);WRITE('  VOLVER AL SISTEMA OPERATIVO');
    GOTOXY(15,23); WRITE('Termina su sesión de trabajo');
    TEXTCOLOR(7);
```

```
  END;
```

```
END; #DEL CASE#
```

```
END; #DEL PROCEDURE CASE OPCIONES#
```

```
(***PROGRAMA PRINCIPAL DE PRINCIPA***)
```

```
BEGIN #del procedure menu#
```

```
  while true do
```

```
  begin
```

```
    SW:='S';
```

```
    WHILE (SW='S') OR (SW='s') DO
```

```
    BEGIN
```

```
      CLRSCR;
```

```
      CARATULAMENU;
```

```
      OPCIONES;
```

```
      SITUA:=15;
```

```
      TEXTCOLOR(14);
```

```
      GOTOXY(22,15);WRITE('          INGRESO DE DATOS');
```

```
      GOTOXY(15,23); WRITE('Selecciona el archivo e ingreso de datos');
```

```
      TEXTCOLOR(7);
```

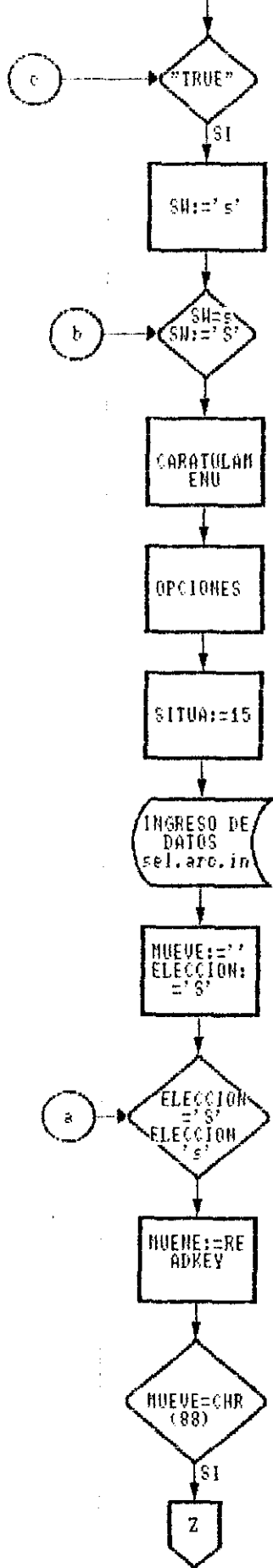
```
      MUEVE:=' '; ELECCION:='S';
```

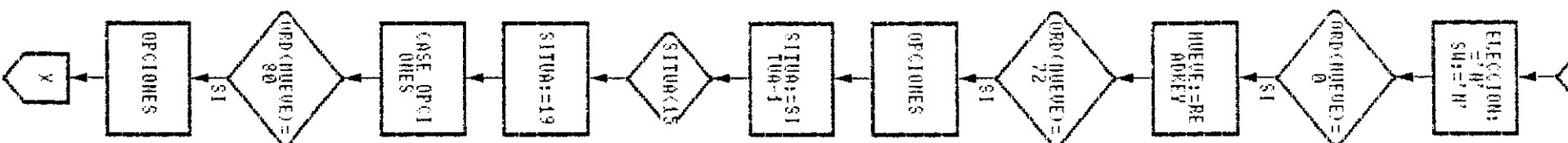
```
      WHILE (ELECCION='S') OR (ELECCION='s') DO
```

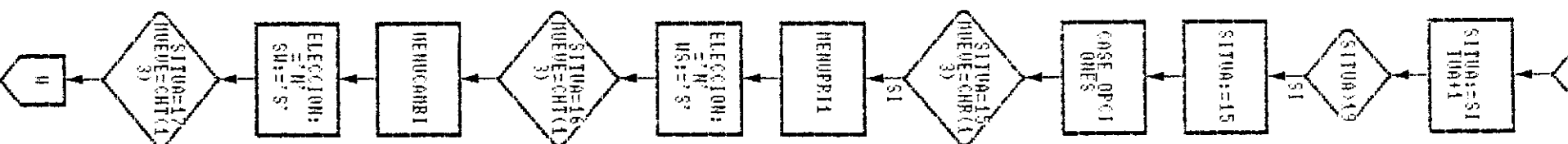
```

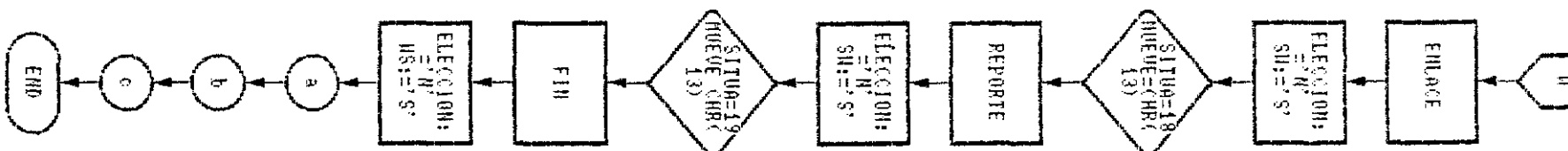
BEGIN
  MUEVE:=READKEY;
  IF UPCASE(MUEVE)=CHR(88) THEN ELECCION:='N'; SW:='N';
  IF ORD(MUEVE)=0 THEN
  BEGIN
    MUEVE:=READKEY;
    IF ORD(MUEVE)=72 THEN
    BEGIN
      OPCIONES;
      GOTOXY(15,23);WRITE('
      SITUA:=SITUA-1;
      IF SITUA<15 THEN SITUA:=19;
      CASE OPCIONES;
    END; *IF ORD(MUEVE)=72*
    IF ORD(MUEVE)=80 THEN
    BEGIN
      OPCIONES;GOTOXY(15,23);WRITE('
      SITUA:=SITUA+1;
      IF SITUA>19 THEN SITUA:=15;
      CASE OPCIONES;
    END; *IF ORD(MUEVE)=80*
  END; *IF ORD(MUEVE)=0*
  IF (SITUA=15) AND (MUEVE=CHR(13)) THEN
  BEGIN
    *MENU1;*CLRSCL;
    ELECCION:='N';SW:='S';
  END; *SITUA=15*
  IF (SITUA=16) AND (MUEVE=CHR(13)) THEN
  BEGIN
    MENU2;CLRSCL;
    ELECCION:='N';SW:='S';
  END;
  *IF (FER2.SITUA=17) AND (MUEVE=CHR(13)) THEN
  BEGIN
    ANALISIS;
    ELECCION:='N';SW:='S';
  END;*
  IF (SITUA=18) AND (MUEVE=CHR(13)) THEN
  BEGIN
    REPORTE;
    ELECCION:='N';SW:='S';
  END;
  IF (SITUA=19) AND (MUEVE=CHR(13)) THEN
  BEGIN
    FIN;
    ELECCION:='N';SW:='S';
  END;
  END; *WHILE ELECCION=SA
  END; *WHILE SW=SA
  END;*WHILE TRUE*
END. *DEL PROCEDURE PRINCIPAL

```









```

UNIT CARATULA;
INTERFACE
USES
    CRT, GRAPH;
VAR
    CO, FI: INTEGER;
    NUEVE, E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, E16, E17, E18, E19, E20, E21, E22, E23, E24, E25, E26, E27, E28, E29, E30, E31, E32, E33, E34, E35, E36, E37, E38, E39, E40, E41, E42, E43, E44, E45, E46, E47, E48, E49, E50, E51, E52, E53, E54, E55, E56, E57, E58, E59, E60, E61, E62, E63, E64, E65, E66, E67, E68, E69, E70, E71, E72, E73, E74, E75, E76, E77, E78, E79, E80, E81, E82, E83, E84, E85, E86, E87, E88, E89, E90, E91, E92, E93, E94, E95, E96, E97, E98, E99, E100;

```

```

PROCEDURE PRIMAN;
PROCEDURE SEPRAN;
PROCEDURE CARATULAMENU;
PROCEDURE MARCHO;
PROCEDURE CUADRO;

```

```
IMPLEMENTATION
```

```

PROCEDURE PRIMAN;
BEGIN
    CLRSCR;
    TEXTCOLOR(14);
    TEXTBACKGROUND(128);
    SETTEXTSTYLE(1,HORIZDIR,3);
    OUTTEXTXY(10,60,'BIENVENIDOS AL AMBIENTE');
    SETTEXTSTYLE(1,HORIZDIR,3);
    OUTTEXTXY(10,120,'INTEGRADO PARA SAPRO');
    SETTEXTSTYLE(2,HORIZDIR,6);
    OUTTEXTXY(300,190,'Version 1.0');
    SETTEXTSTYLE(2,HORIZDIR,6);
    OUTTEXTXY(210,270,'Todos los Derechos son Reservados');
    TEXTCOLOR(0);
    SETTEXTSTYLE(2,HORIZDIR,3);
    OUTTEXTXY(300,330,'Presione (Enter) para continuar... ');
END;
    * Fin de la Primera Pantalla 8

```

```

PROCEDURE SEPRAN;
BEGIN
    CLRSCR;
    TEXTBACKGROUND(128);
    SETTEXTSTYLE(1,HORIZDIR,3);
    OUTTEXTXY(85,1,'FACULTAD UNIVERSIDAD CATOLICA DEL ECUADOR');
    OUTTEXTXY(135,20,'PROGRAMAS ACADMICOS AMBATO');
    OUTTEXTXY(250,60,'TESTS DE GRADO');
    SETTEXTSTYLE(1,HORIZDIR,2);
    OUTTEXTXY(100,100,'REALIZADO POR:');
    OUTTEXTXY(100,130,'DIRECTOR DE TESTS:');
    OUTTEXTXY(100,230,'ASESOR DE TESTS:');
    SETTEXTSTYLE(2,HORIZDIR,6);
    OUTTEXTXY(320,130,'APREHA B. ELIZABETH');
    OUTTEXTXY(320,150,'TERRA V. JENNY');
    OUTTEXTXY(320,210,'Ing. CAROLINA PELERAN');
    OUTTEXTXY(320,240,'Ing. VICTOR RUGO FAREDES');
    SETTEXTSTYLE(2,HORIZDIR,3);
    OUTTEXTXY(250,305,'AMBATO - 1973');
    OUTTEXTXY(330,335,'Presione Enter para Continuar.....');
END;
    * Fin de la Segunda Pantalla 8

```

```

PROCEDURE CARATULAMENU;
BEGIN
    CLRSCR;
    FOR CO:=1 TO 80 DO
    BEGIN

```



```

GOTOY(60,13);WRITE(1,1);
GOTOY(60,21);WRITE(1,1);
GOTOY(60,21);WRITE(1,1);
END;

```

```

PROCEDURE MARCO;
VAR
  I,J,K:INTEGER;
BEGIN
  READLN(N);
  FOR CO:=1 TO 90 DO
  BEGIN
    GOTOY(CO,1);WRITE(1,1);
    GOTOY(CO,24);WRITE(1,1);
  END;
  FOR I:=1 TO 100 DO
  BEGIN
    GOTOY(I,1);WRITE(1,1);
    GOTOY(CO,1);WRITE(1,1);
  END;
  GOTOY(1,1);WRITE(1,1);
  GOTOY(10,1);WRITE(1,1);
  GOTOY(1,24);WRITE(1,1);
  GOTOY(10,24);WRITE(1,1);
END; * Fin de la Procedure Marco;

```

```

PROCEDURE LINDA;
VAR
  COLE:INTEGER;
BEGIN
  I:=0;
  FOR CO:=1 TO 90 DO
  BEGIN
    GOTOY(CO,6);WRITE(1,1);
    GOTOY(CO,12);WRITE(1,1);
  END;
  FOR I:=1 TO 12 DO
  BEGIN
    GOTOY(I,1);WRITE(1,1);
    GOTOY(70,1);WRITE(1,1);
  END;
  GOTOY(10,6);WRITE(1,1);
  GOTOY(70,6);WRITE(1,1);
  GOTOY(10,12);WRITE(1,1);
  GOTOY(70,12);WRITE(1,1);
  TEXT(0,0,0);
END; * Fin de la Procedure Linda;

```

END.

UNIT MENU;

INTERFACE

USES CRT,CARATULA;

VAR

SITUA:BYTE;

PROCEDURE OPCIONES;

PROCEDURE FIN;

IMPLEMENTATION

PROCEDURE OPCIONES;

BEGIN

GOTOXY(22,15);WRITE(' INGRESO DE DATOS');
GOTOXY(22,16);WRITE(' CAMBIOS Y CONSULTAS');
GOTOXY(22,17);WRITE(' ANALISIS');
GOTOXY(22,18);WRITE(' REPORTES');
GOTOXY(22,19);WRITE(' VOLVER AL SISTEMA OPERATIVO');

END;

PROCEDURE FIN; (**PANTALLA PARA DESPEDIDA**)

VAR

FJ,CO,FINAL:BYTE;

BEGIN

CLRSR;

FINAL:=1;

WHILE FINAL<=2 DO

BEGIN

FOR FJ:= 1 TO 24 DO

BEGIN

GOTOXY(1,FJ); WRITE(' ');

GOTOXY(79,FJ); WRITE(' ');

END;

FOR CO:= 1 TO 79 DO

BEGIN

GOTOXY(CO,1);WRITE('=');

GOTOXY(CO,24);WRITE('=');

END;

GOTOXY(1,1); WRITE(' ');

GOTOXY(79,1); WRITE(' ');

GOTOXY(1,24); WRITE(' ');

GOTOXY(79,24); WRITE(' ');

FOR CO:=2 TO 78 DO

BEGIN

FOR FJ:=2 TO 23 DO

BEGIN

GOTOXY(CO,FJ); WRITE(' ');

END;

GOTOXY(CO,FJ); WRITE(' ');

END;

TEXTCOLOR(14);

GOTOXY(34,18); WRITE('THATS ALL FOLCKS');

GOTOXY(27,12); WRITE('SEE YOU, UNTIL THE NEXT PROGRAM');

TEXTCOLOR(7);

FINAL:=FINAL+1;

END;

HALT(1);

END; #DEL PROCEDURE FIN#

| NOMBRE DEL PROGRAMA | DOCUMENTO | | |
|---------------------|-----------|-------------------------------|----|
| NOMBRE | TIPO | DESCRIPCION | |
| LEFAR | Byte | Variable de control | de |
| PER | Byte | Variable de control | de |
| NTUA | Byte | | |
| OW | Char | Switch de ayuda | |
| MEVE | Char | Variable de selección | de |
| ELMCTOP | Char | Variable de operadores | de |
| MEHBER | String | Variable para los comentarios | |
| H | Integer | Número de nodos | |
| FRMOCHE | String | Para leer archivos | |

UNIT MENUPRI;

INTERFACE

USES CRT,CARATULA,MENU,INGRESO1,INGRESO2,INGRESO3;
 PROCEDURE MENUPRI1;

IMPLEMENTATION

PROCEDURE MENUPRI1;

VAR

LUGAR,FER,SITUA : BYTE;
 SW,MUEVE,ELECCION:CHAR;
 ULTIMOC:STRING#25A;
 N:INTEGER;
 NOMEST1:STRING#23A;

LABEL 1,2,3,4,5,6,7,8,9,10,11,25,30,50;

BEGIN ~~adel~~ procedure menupri1

```

30: NOMARCH;
    ASSIGN(ARCHIVOSAP,NOMBRE);
    IF I=1
    RESET(ARCHIVOSAP);
    IF I=2
    IF IORESULT<>0 THEN
        BEGIN
            REWRITE(ARCHIVOSAP);
            CLOSE(ARCHIVOSAP);
            GOTO 1;
        END IF IORESULT<>0
    ELSE
        BEGIN
            LUGAR:=0;
            RESET(ARCHIVOSAP);
            FOR CODE:=1 TO 3 DO
                READLN(ARCHIVOSAP,NOMEST);
            NOMEST:=COPY(NOMEST,1,80);
            LUGAR:=POS('N=',NOMEST);
            IF LUGAR=0 THEN GOTO 2
            ELSE
                BEGIN
                    VAL(COPY(NOMEST,LUGAR+2,3),N, CODE);
                    LUGAR:=POS('L=',NOMEST);
                    IF LUGAR <> 0 THEN VAL(COPY(NOMEST,LUGAR+2,3),L,CODE);
                END;
            WHILE NOT EOF(ARCHIVOSAP) DO
                BEGIN
                    READLN(ARCHIVOSAP,NOMEST);
                    LUGAR:=POS('C ',NOMEST);
                    IF LUGAR<>0 THEN ULTIMOC:=COPY(NOMEST,LUGAR+2,22);
                END; WHILE
            CLOSE (ARCHIVOSAP);
            IF ULTIMOC='FIN DE SYSTEM' THEN
                BEGIN
                    FER:=1; GOTO 3;
                END;
            IF ULTIMOC='FIN DE JOINTS' THEN
                BEGIN
                    FER:=1; GOTO 4;
                END;
        END;
    
```

```

END;
IF ULTIMOC='FIN DE RESTRAINTS' THEN
BEGIN
    FER:=1; GOTO 5;
END;
IF ULTIMOC='FIN DE SPRINGS' THEN
BEGIN
    FER:=1; GOTO 6;
END;
IF ULTIMOC='FIN DE CONSTRAINTS' THEN
BEGIN
    FER:=1; GOTO 7;
END;
IF ULTIMOC='FIN DE FRAME' THEN
BEGIN
    FER:=1; GOTO 8;
END;
IF ULTIMOC='FIN DE QUAD' THEN
BEGIN
    FER:=1; GOTO 9;
END;
IF ULTIMOC='FIN DE LOADS' THEN
BEGIN
    FER:=1; GOTO 10;
END;
IF ULTIMOC='FIN DE FORCES' THEN
BEGIN
    FER:=1; GOTO 11;
END;
IF ULTIMOC='FIN DE PLOT' THEN
BEGIN
    GOTOXY(5,23);
    TEXTCOLOR(14);
    WRITE('ARCHIVO COMPLETO..PRESIONE <ENTER> E INGRESE OTRO NOMBRE DE ARCHIVO');
    TEXTCOLOR(7);
    READLN;
    GOTO 30;
END;
END; #IORESULTS
1: TIT_EST(NOMBRE);
2: SISTEM(NOMBRE,N);
FER:=0;
3: JOINTS(NOMBRE,N,L,FER);
IF FER=2 THEN
BEGIN
    FER:=0; GOTO 4;
END
ELSE
    GOTO 50;
4: RESTRAIN(NOMBRE,N,L,FER);
IF FER=2 THEN
BEGIN
    FER:=0; GOTO 5;
END
ELSE
    GOTO 50;
5: SPRINGS(NOMBRE,N,L,FER);
IF FER=2 THEN
BEGIN
    ENDFER:=0; GOTO 6;

```

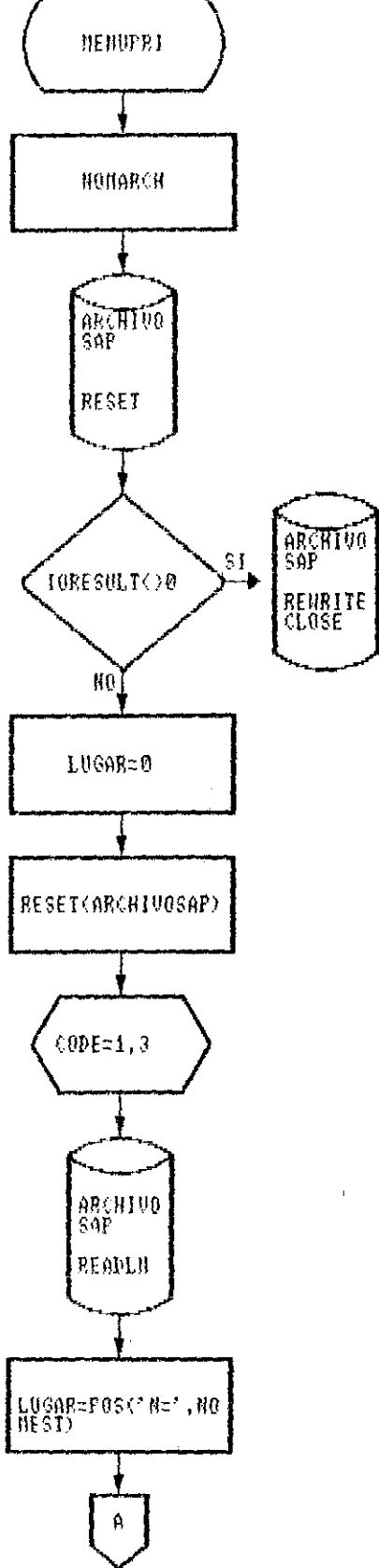
```

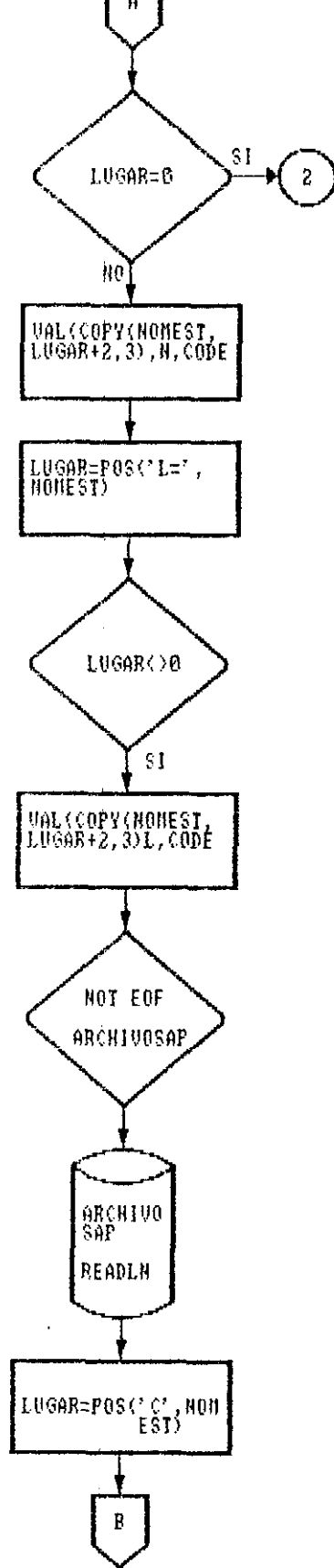
ELSE
  GOTO 50;
6: CONSTRAI(NOMBRE,N,L,FER);
  IF FER=2 THEN
  BEGIN
    FER:=0; GOTO 7;
  END
ELSE
  GOTO 50;
7: FRAME(NOMBRE,N,L,FER);
  IF FER=2 THEN
  BEGIN
    FER:=0; GOTO 8;
  END
ELSE
  GOTO 50;
8: QUAD(NOMBRE,N,L,FER);
  IF FER=2 THEN
  BEGIN
    FER:=0; GOTO 9;
  END
ELSE
  GOTO 50;
9: LOADS(NOMBRE,N,L,FER);
  IF FER=2 THEN
  BEGIN
    FER:=0; GOTO 10;
  END
ELSE
  GOTO 50;
10: FORCES(NOMBRE,N,L,FER);
  IF FER=2 THEN
  BEGIN
    FER:=0; GOTO 11;
  END
ELSE
  GOTO 50;
11: PLOT(NOMBRE,N,L,FER);

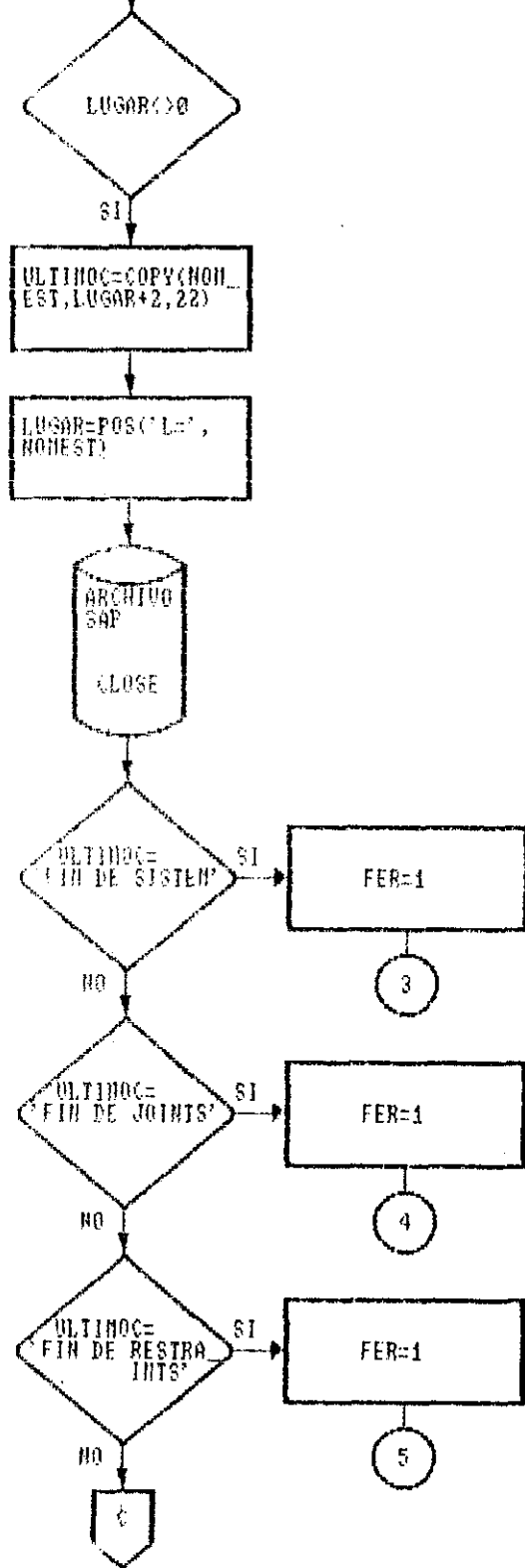
50: END; #DEL PROCEDURE MENU3

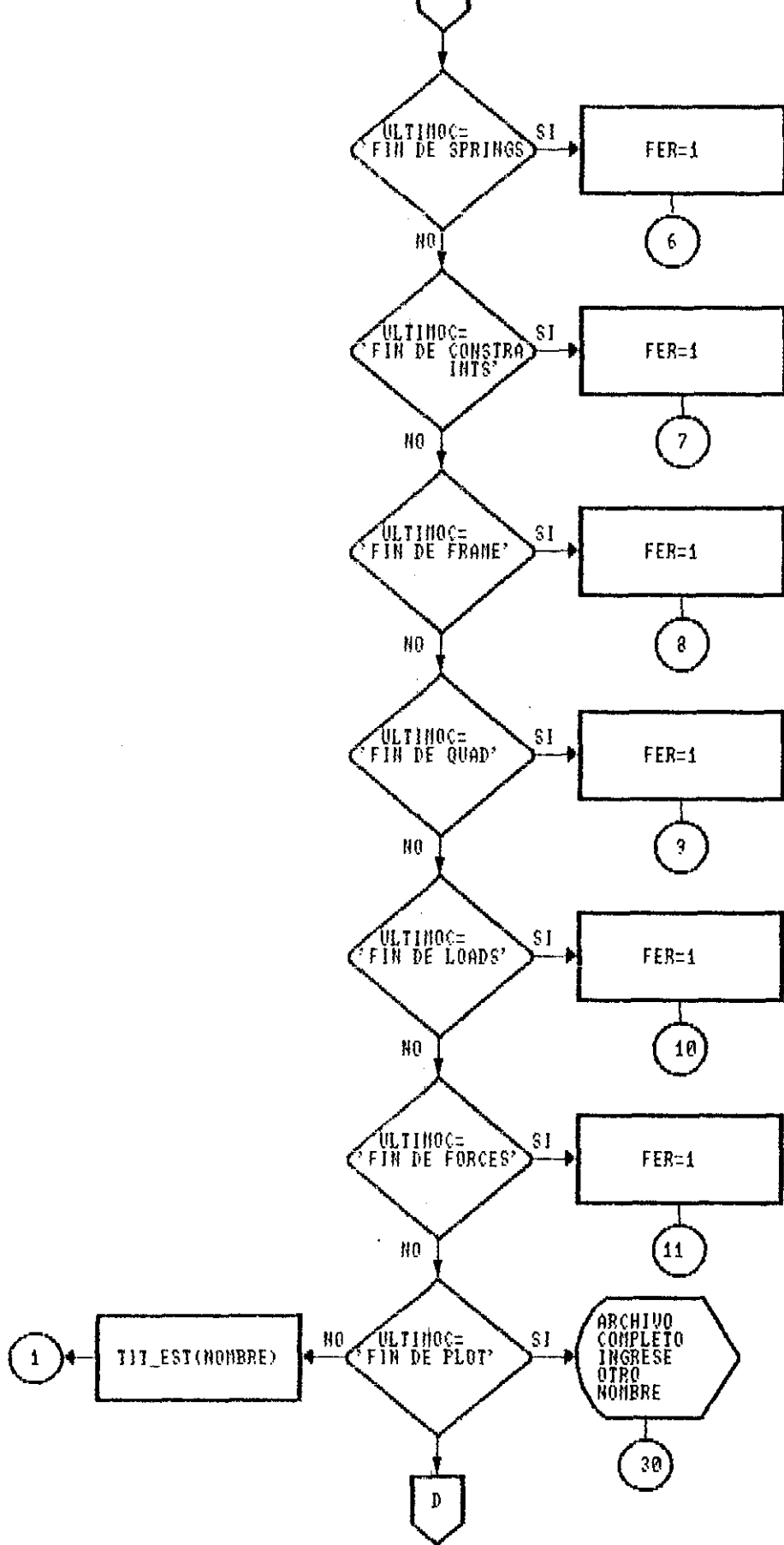
END. #DE LA UNIDAD MENUFRIA

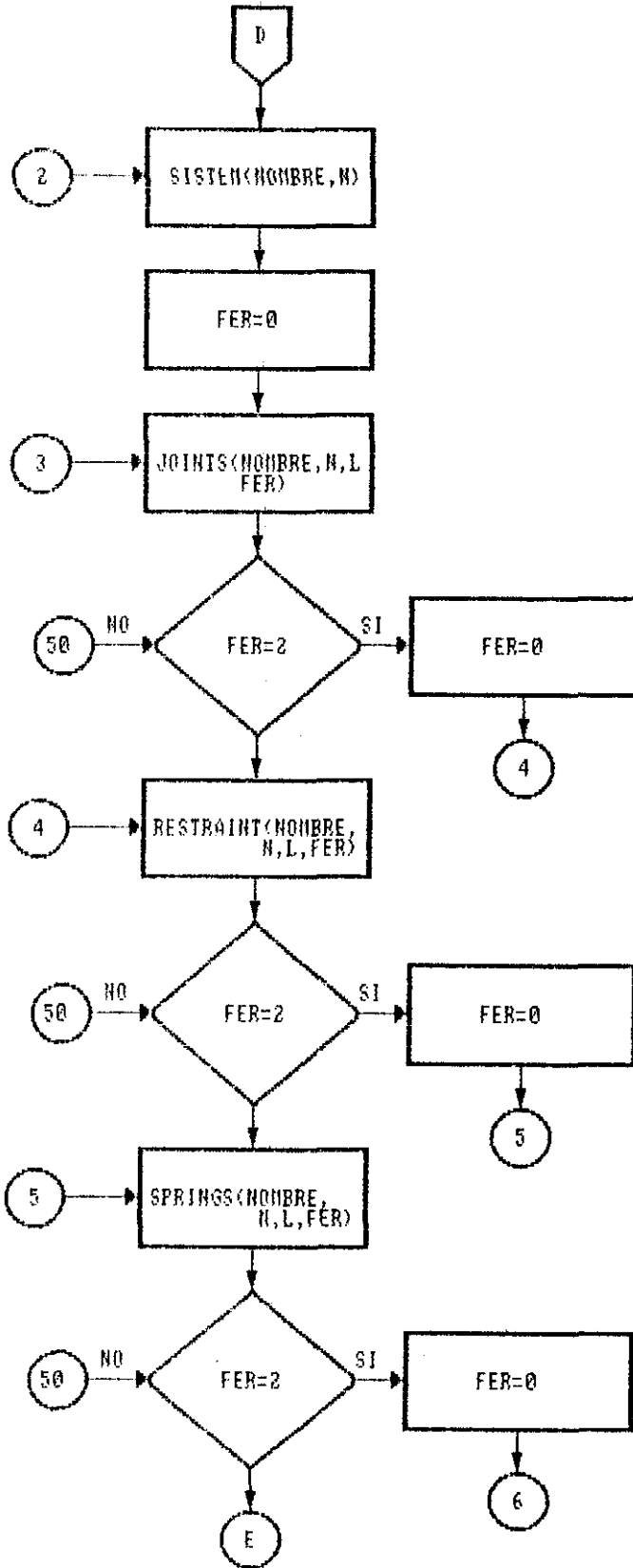
```

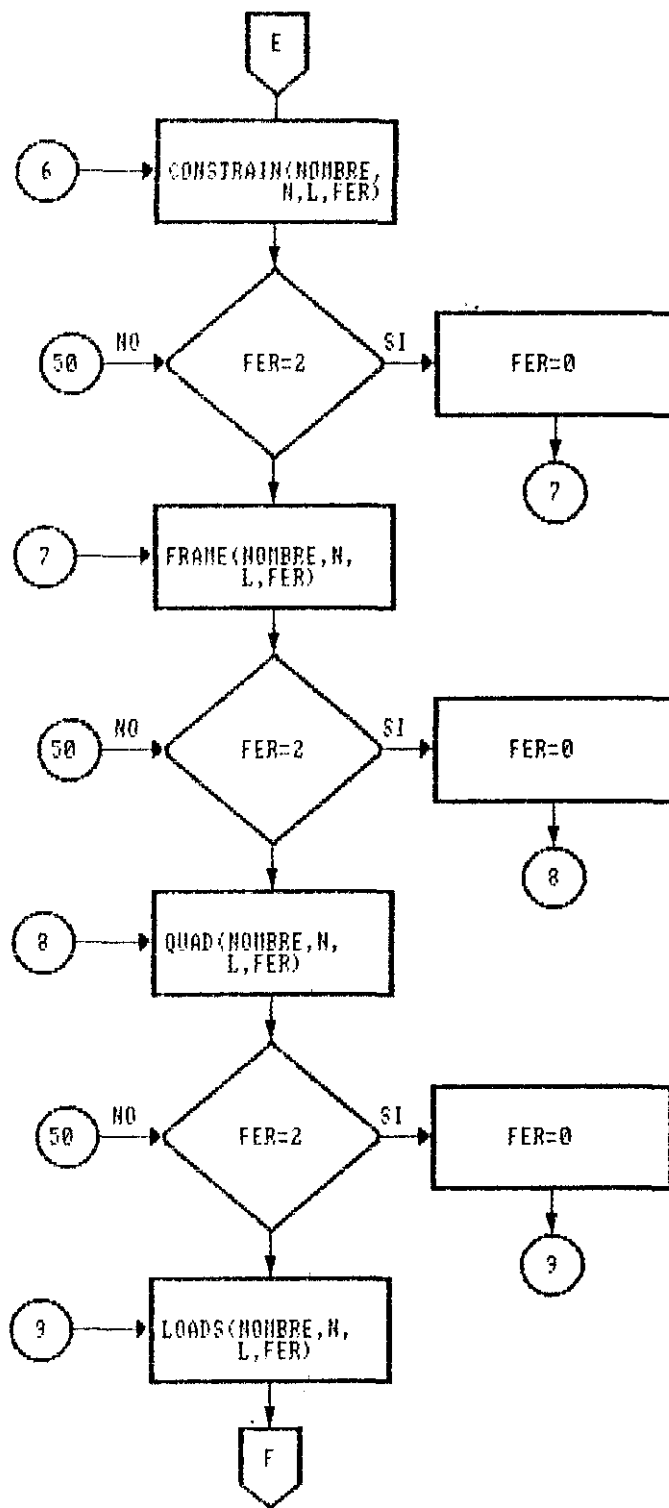


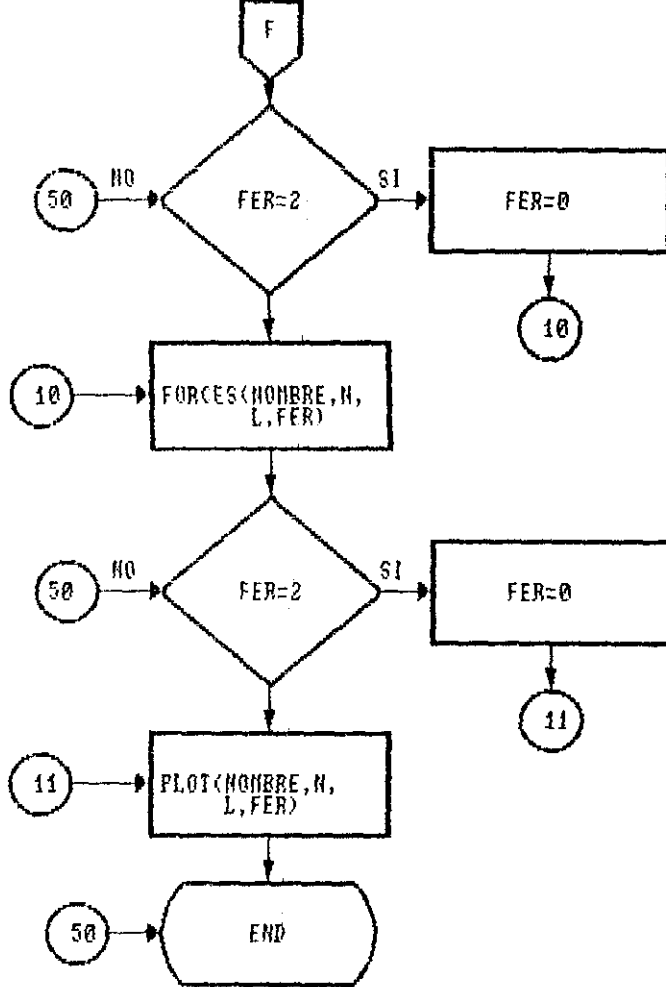












```
UNIT INGRESO1;  
INTERFACE  
Uses Crt, Unifun, Caratula;  
TYPE
```

```
    STR10=STRING#14A;  
VAR  
    NOMBRE:STR10;  
    CODE,POSICION,COLUMNA:INTEGER;  
    L,LUGAR:BYTE;  
    NOMEST:STRING#80A;  
    ARCHIVOSAP:TEXT;  
    DATOCD,CORREGIR,SW:CHAR;  
    CONTA,ADRI,MITAD:BYTE;
```

```
PROCEDURE NOMARCH;  
PROCEDURE TIT_EST(NOMBRE:STR10);  
PROCEDURE SISTEM(NOMBRE:STR10; VAR N:INTEGER);  
PROCEDURE JOINTS(NOMBRE:STR10; VAR N:INTEGER;VAR L,FER:BYTE);  
PROCEDURE RESTRAIN(NOMBRE:STR10;VAR N:INTEGER;VAR L,FER:BYTE);  
PROCEDURE SPRINGS(NOMBRE:STR10; VAR N:INTEGER;VAR L,FER:BYTE);  
PROCEDURE CONSTRAI(NOMBRE:STR10;VAR N:INTEGER;VAR L,FER:BYTE);
```

IMPLEMENTATION

```
PROCEDURE NOMARCH;  
VAR
```

```
    SW1:char;  
    NOMBRE1:STRING#80A;  
    UNIDAD:CHAR;
```

```
BEGIN
```

```
    CLRSCR;
```

```
    CUAORO;
```

```
    GOTOXY(15,8); WRITE('Ingrese la Unidad de Disco: ==> ');
```

```
    GOTOXY(15,10);WRITE('Nombre del Archivo      : ==> ');
```

```
    TEXTCOLOR(14);
```

```
    GOTOXY(15,8); WRITE(' ');GOTOXY(26,8);WRITE('U');GOTOXY(36,8); WRITE('D');
```

```
    GOTOXY(15,10);WRITE('N');GOTOXY(26,10);WRITE('A');
```

```
    TEXTCOLOR(7);
```

```
    SW1:='N';
```

```
    WHILE (SW1='N') OR (SW1='n') DO
```

```
    BEGIN
```

```
        REPEAT
```

```
            TEXTCOLOR(14);
```

```
            GOTOXY(49,8); WRITE('###');
```

```
            TEXTCOLOR(7);
```

```
            GOTOXY(5,22);WRITE('La unidad de disco puede ser: A: B: C: ó D:');
```

```
            TEXTCOLOR(14);
```

```
            GOTOXY(8,22);WRITE('U');GOTOXY(18,22);WRITE('D'); GOTOXY(35,22);WRITE('A: B: C:');
```

```
            GOTOXY(50,22);WRITE('D:');
```

```
            GOTOXY(49,8);UNIDAD:=READKEY;
```

```
            TEXTCOLOR(7);
```

```
            CASE UNIDAD OF
```

```
                'A','B','C','D','a','b','c','d':
```

```
                BEGIN
```

```
                    TEXTCOLOR(14);
```

```
                    GOTOXY(49,8);WRITE(UNIDAD);
```

```
                    TEXTCOLOR(7);
```

```
                END;
```

```
            END;  aDel Casea
```

```
        UNTIL (UNIDAD='A') OR (UNIDAD='a') OR (UNIDAD='B') OR (UNIDAD='b') OR (UNIDAD='C') OR (UNIDAD='c')  
        GOTOXY(8,22);UNIDAD:=READKEY;
```

```

GOTOXY(5,23);CLREOL;
GOTOXY(5,23);WRITE('Su Ingreso es Correcto S/N: █');
TEXTCOLOR(14);
GOTOXY(28,23);WRITE('S');GOTOXY(30,23);WRITE('N');
GOTOXY(33,23);SW1:=READKEY;
TEXTCOLOR(7);
END; aWHILE (SW1='N')a
SW1:='N';
WHILE (SW1='N') OR (SW1='n') DO
BEGIN
GOTOXY(5,23);WRITE('El nombre del archivo debe tener máximo 8 caracteres');
TEXTCOLOR(14);
GOTOXY(49,10);WRITE('██████████');
GOTOXY(8,23);WRITE('N');GOTOXY(19,23);WRITE('A');GOTOXY(45,23);WRITE('B C');
GOTOXY(49,10);READLN(NOMBRE1);
TEXTCOLOR(7);
GOTOXY(5,22);CLREOL;
GOTOXY(5,23);CLREOL;
GOTOXY(5,23);WRITE('Su Ingreso es Correcto S/N: █');
TEXTCOLOR(14);
GOTOXY(28,23);WRITE('S');GOTOXY(30,23);WRITE('N');
GOTOXY(33,23);SW1:=READKEY;
TEXTCOLOR(7);
END; aWHILE (SW1='N')a
NOMBRE:=UNIDAD+' '+NOMBRE1;
end; adel procedure Nomarcha

```

```
PROCEDURE TIT_EST;
```

```
VAR
```

```
TITULO:STRING#30A;
```

```
SW:CHAR;
```

```
BEGIN a De la Procedure Tit_Esta
```

```
CUADRO;
```

```
SW:='N';
```

```
WHILE (SW='N') OR (SW='n') DO
```

```
BEGIN
```

```
GOTOXY(5,20);WRITE('Para el Nombre de la Estructura debe ingresar máximo 30 caracteres');
```

```
TEXTCOLOR(14);
```

```
GOTOXY(13,20);WRITE('N');GOTOXY(26,20);WRITE('E');GOTOXY(58,20);WRITE('30');
```

```
TEXTCOLOR(7);
```

```
GOTOXY(13,9);WRITE('Nombre de la estructura:');
```

```
GOTOXY(38,9);WRITE('████████████████████████████████████████');
```

```
TEXTCOLOR(14);
```

```
GOTOXY(38,9);READLN(TITULO);
```

```
GOTOXY(5,24);WRITELN('Su ingreso es correcto S/N: █');
```

```
GOTOXY(28,24);WRITELN('S');
```

```
GOTOXY(30,24);WRITELN('N');
```

```
TEXTCOLOR(7);
```

```
GOTOXY(33,24);SW:=READKEY;
```

```
END;
```

```
ASSIGN(ARCHIVOSAP,NOMBRE);
```

```
APPEND(ARCHIVOSAP);
```

```
WRITELN(ARCHIVOSAP,'TITULO : ',TITULO);
```

```
CLOSE (ARCHIVOSAP);
```

```
END; adel procedure Tit_Esta
```

```
PROCEDURE SISTEM;
```

```
VAR
```

```
SW,NUMP,NUML:CHAR;
```

```
REG#BYT#del Programa Principal
```

```

CLRSCR;
CUADRO;
GOTOXY(5,19);WRITE('P = Dirección de los datos: P=1 ==> Disco ; P=0 ==> Impresora');
GOTOXY(5,20);WRITE('L = Tipo de Carga; máximo deben existir 9 tipos de cargas');
GOTOXY(5,21);WRITE('N = Cantidad total de Nudos a ingresar');
TEXTCOLOR(14);
GOTOXY(5,19);WRITE('P = D');GOTOXY(33,19);WRITE('P=1');GOTOXY(41,19);WRITE('D');
GOTOXY(51,19);WRITE('P=0');GOTOXY(59,19);WRITE('I');
GOTOXY(5,20);WRITE('L = T');GOTOXY(17,20);WRITE('C');
GOTOXY(5,21);WRITE('N = C');GOTOXY(27,21);WRITE('N');
TEXTCOLOR(7);
SW:='N';
WHILE (SW='N') OR (SW='n') DO
  BEGIN
    TEXTCOLOR(14);
    GOTOXY(35,8);WRITE('P = ');
    TEXTCOLOR(7);
    GOTOXY(39,8);WRITE('   ');
    REPEAT
      GOTOXY(39,8);NUMP:=READKEY;
      CASE NUMP OF
        '0','1': BEGIN
          GOTOXY(39,8);WRITE(NUMP);
          P:=ORD(NUMP)-48;
        END;
      END;
    UNTIL (NUMP='0') OR (NUMP='1');
    GOTOXY(5,24);WRITE('Su Ingreso es Correcto S/N:   ');
    TEXTCOLOR(14);
    GOTOXY(28,24);WRITE('S');GOTOXY(30,24);WRITE('N');
    GOTOXY(33,24);SW:=READKEY;
    TEXTCOLOR(7);
  END;
SW:='N';
WHILE (SW='N') OR (SW='n') DO
  BEGIN
    TEXTCOLOR(14);
    GOTOXY(35,9);WRITELN('L = ');
    TEXTCOLOR(7);
    GOTOXY(39,9);WRITE('   ');
    REPEAT
      GOTOXY(39,9);NUML:=READKEY;
      CASE NUML OF
        '1'..'9': BEGIN
          GOTOXY(39,9);WRITE(NUML);
          L:=ORD(NUML)-48;
        END;
      END;
    UNTIL (NUML<='9') AND (NUML>='1');
    GOTOXY(34,24);SW:=READKEY;
  END;
TEXTCOLOR(14);
GOTOXY(35,10);WRITE('N = ');
TEXTCOLOR(7);
GOTOXY(5,24);CLREOL;
GOTOXY(5,24);WRITE('Para Corregir el Número de Nudos Utilice');
TEXTCOLOR(14);GOTOXY(46,24);WRITE('BACKSPACE');TEXTCOLOR(7);
SW:='N';
N:=1;
WHILE (SW='N') OR (SW='n') DO

```

```

N:=FunInEnt(39,10);
GOTOXY(5,24);WRITE(' ');
GOTOXY(5,24);WRITE('Su Ingreso es Correcto S/N: ');
TEXTCOLOR(14);
GOTOXY(28,24);WRITE('S');GOTOXY(30,24);WRITE('N');
GOTOXY(33,24);SW:=READKEY;
TEXTCOLOR(7);
END;
DEL WHILEA
ASSIGN(ARCHIVOSAP,NOMBRE);
APPEND(ARCHIVOSAP);
WRITELN(ARCHIVOSAP,'SYSTEM : CONTROL DE SALIDA DE LA INFORMACION');
WRITELN(ARCHIVOSAP,'P=',NUMP:1,' N=',N:3,' L=',NUML:1);
WRITELN(ARCHIVOSAP,'C FIN DE SYSTEM');
CLOSE(ARCHIVOSAP);
END;
De la Procedure System

PROCEDURE JOINTS;
VAR
  NUDO,NUDO1,N1,NF,NH1,NHF,NV1,NVF:INTEGER;
  IH,IV,IL,LON,S:BYTE;
  X,Y,Z:REAL;
  NUDOAX,N1AX,NFAX,NH1AX,NHFAUX,NV1AX,NVFAUX:STRING#3A;
  IHAX,IVAX,ILAX:STRING#2A;
  XAX,YAX,ZAX:STRING#6A;
LABEL 5;

PROCEDURE COORDENADAS(POSICION:INTEGER; VAR NUDO1,N:INTEGER;VAR ADRI,FER:BYTE);FORWARD;
PROCEDURE DATO_CORRECTO(POSICION:INTEGER; VAR NUDO1,N:INTEGER;VAR ADRI,FER:BYTE);FORWARD;

PROCEDURE GENERACION_CUADRATICA(VAR POSICION,NUDO1:INTEGER);
VAR
  SW:CHAR;
BEGIN
  SW='N';
  WHILE UPCASE(SW)='N' DO
  BEGIN
    TEXTCOLOR(14);GOTOXY(57,POSICION);WRITE('Q=');TEXTCOLOR(7);
    GOTOXY(59,POSICION);WRITE(' ');
    GOTOXY(5,21);WRITE(' ');
    GOTOXY(5,22);WRITE(' ');
    GOTOXY(5,23);WRITE(' ');
    TEXTCOLOR(14);
    GOTOXY(3,21);WRITE('Q=Generación Cuadrática. ');
    TEXTCOLOR(7);
    GOTOXY(28,21);WRITE('El orden de ingreso de los nudos es el siguiente: ');
    TEXTCOLOR(14);
    GOTOXY(5,22); WRITE('Nudo horizontal inicial, Nudo horizontal final, Nudo vertical inicial');
    GOTOXY(5,23); WRITE('Nudo vertical final, Incremento horizontal, Incremento vertical');
    TEXTCOLOR(7);
    COLUMNA:=59;NHI:=FunInEnt(columna,posicion);
    COLUMNA:=63;NHF:=FunInEnt(columna,posicion);
    COLUMNA:=67;NVI:=FunInEnt(columna,posicion);
    COLUMNA:=71;NVF:=FunInEnt(columna,posicion);
    COLUMNA:=75;IH:=FunInEnt(columna,posicion);
    COLUMNA:=78;IV:=FunInEnt(columna,posicion);
    GOTOXY(3,21);WRITE(' ');
    GOTOXY(3,22);WRITE(' ');
    GOTOXY(3,23);WRITE(' ');
    GOTOXY(5,23);WRITE('Datos de Generacion Correctos S/N: ');
    TEXTCOLOR(14);WRITE('S');GOTOXY(37,23);WRITE('N');
  END;

```

```

    GOTOXY(40,23);SW:=READKEY;
    TEXTCOLOR(7);
END; aDel While8
NUDD1:= NUDD1+(((NF-NI) DIV (IH))+1) * (((NVF-NVI) DIV (IV))+1)-4);
GOTOXY(3,21);WRITE(' ');
GOTOXY(3,22);WRITE(' ');
GOTOXY(3,23);WRITE(' ');
adri:=3;
END; aDe la Procedure Gen.Cuadratica

PROCEDURE GENERACION_LINEAL(VAR POSICION,NUDD1:INTEGER);
VAR
    SW:CHAR;
BEGIN
    SW='N';
    WHILE (SW='N') OR (SW='n') DO
        BEGIN
            TEXTCOLOR(14);GOTOXY(62,POSICION);WRITE('G=');TEXTCOLOR(7);
            GOTOXY(64,POSICION);WRITE('  ', '  ', '  ');
            TEXTCOLOR(14);
            GOTOXY(5,22);WRITE(' ');
            GOTOXY(5,23);WRITE(' ');
            GOTOXY(5,22);WRITE('G=Generación lineal;');
            TEXTCOLOR(7);
            GOTOXY(27,22);WRITE('Para generación lineal el orden de ingreso de los');
            GOTOXY(5,23);WRITE(' ');
            GOTOXY(5,23);WRITE('nudos es el siguiente:');
            TEXTCOLOR(14);
            GOTOXY(28,23);WRITE('Nudo inicial, Nudo final, incremento entre nudos');
            TEXTCOLOR(7);
            COLUMNA:=64;NI:=FunInEnt(columna, posicion);
            COLUMNA:=68;NF:=FunInEnt(columna, posicion);
            COLUMNA:=72;IL:=FunInEnt(columna, posicion);
            GOTOXY(5,21);WRITE(' ');
            GOTOXY(5,22);WRITE(' ');
            GOTOXY(5,23);WRITE(' ');
            GOTOXY(5,23);WRITE('Datos de Generacion Correctos S/N: ');
            TEXTCOLOR(14);
            GOTOXY(35,23);WRITE('S');GOTOXY(37,23);WRITE('N');
            GOTOXY(40,23);SW:=READKEY;
            TEXTCOLOR(7);
        END;
        NUDD1:=NUDD1+(((NF-NI)div(IL))-1);
        GOTOXY(5,21);WRITE(' ');
        GOTOXY(5,22);WRITE(' ');
        GOTOXY(5,23);WRITE(' ');
        adri:=2;
    END;
END;

PROCEDURE DATO_CORRECTO(POSICION:INTEGER;VAR NUDD1,N:INTEGER;VAR ADRI,FER:BYTE);
label 2;
BEGIN
    GOTOXY(5,22);WRITE(' ');
    GOTOXY(5,23);WRITE(' ');
    GOTOXY(5,23);WRITE('Nudo y Coordenadas Correctos S/N: ');
    TEXTCOLOR(14);
    GOTOXY(34,23);WRITE('S');
    GOTOXY(36,23);WRITE('N');
    TEXTCOLOR(7);
    GOTOXY(40,23);SW:=READKEY;

```

```

BEGIN
  GOTOXY(5,23);WRITE(' ');
  GOTOXY(5,23);WRITE(' ');
  GOTOXY(5,23);WRITE('Corregir: Nudo; Coordenadas: X, Y, ó Z: ');
  TEXTCOLOR(14);
  GOTOXY(15,23);WRITE('N');
  GOTOXY(34,23);WRITE('X');
  GOTOXY(37,23);WRITE('Y');
  GOTOXY(42,23);WRITE('Z');
  TEXTCOLOR(7);
  GOTOXY(45,23);CORREGIR:=READKEY;
  CASE CORREGIR OF
    'N','n': BEGIN
      COLUMNA:=12;NUDO:=FunNudoFin(columna,posicion);
      if nudo = maxint then GOTO 2;
    END;
    'X','x': BEGIN
      COLUMNA:=25;X:=FunInDec(columna,posicion);
    END;
    'Y','y': BEGIN
      COLUMNA:=35;Y:=FunInDec(columna,posicion);
    END;
    'Z','z': BEGIN
      COLUMNA:=45;Z:=FunInDec(columna,posicion);
    END;
  END;  *End_Case*
  DATO_CORRECTO(POSICION,NUDO1,N,ADRI,FER);
END  *End_If_Dato_Correcto*
ELSE
BEGIN
  ASSIGN(ARCHIVOSAP,NOMBRE);
  APPEND(ARCHIVOSAP);
  NUDDAUX:=' ';LON:=0;
  STR(NUDD,NUDDAUX);
  LON:=LENGTH(NUDDAUX);
  WHILE LON<=3 DO
  BEGIN
    NUDDAUX:=NUDDAUX+' ';
    lon:=lon+1;
  END;
  XAUX:=' ';LON:=0;
  STR(X:1:2,XAUX);
  LON:=LENGTH(XAUX);
  WHILE LON<=6 DO
  BEGIN
    XAUX:=XAUX+' ';
    lon:=lon+1;
  END;
  YAUX:=' ';LON:=0;
  STR(Y:1:2,YAUX);
  LON:=LENGTH(YAUX);
  WHILE LON<=6 DO
  BEGIN
    YAUX:=YAUX+' ';
    lon:=lon+1;
  END;
  ZAUX:=' ';LON:=0;
  STR(Z:1:2,ZAUX);
  LON:=LENGTH(ZAUX);
  WHILE LON<=6 DO

```

```

        ZAux:=ZAux+' ';
        lon:=lon+1;
END;
if adri = 0 then
    writeln(ARCHIVOSAP,NUDDAux,' X=',XAux,' Y=',YAux,' Z=',ZAux);
if adri = 1 then
begin
    writeln(ARCHIVOSAP,NUDDAux,' X=',XAux,' Y=',YAux,' Z=',ZAux,' S=',S:1);
    adri:=0;
end;
if adri=2 then
begin
    STR(NI:1,NIAux);
    STR(NF:1,NFAux);
    STR(IL:1,ILAux);
    writeln(ARCHIVOSAP,NUDDAux,' X=',XAux,' Y=',YAux,' Z=',ZAux,' B=',NI:3,' ',NF:3);
    adri:=0;
end;
if adri=3 then
begin
    STR(NHI:1,NHIAux);
    STR(NHF:1,NHFAux);
    STR(NVI:1,NVIAux);
    STR(NVF:1,NVFAux);
    STR(IH:1,IHAux);
    STR(IV:1,IVAux);
    writeln(ARCHIVOSAP,NUDDAux,' X=',XAux,' Y=',YAux,' Z=',ZAux,
        B=',NHI:3,' ',NHF:3,' ',NVI:3,' ',NVF:3,' ',IH:2,' ',IV:2);
    adri:=0;
end;
CLOSE(ARCHIVOSAP);
GOTOXY(5,23);WRITE(' ');
END;  #End DATOCOD
2: END;  #End_Procedure Dato_Correcto

```

```

PROCEDURE COORDENADAS(POSICION:INTEGER;VAR NUDDI,N:INTEGER;VAR ADRI,FER:BYTE);
VAR
    GENERAR,FORMA:CHAR;
    label 20,3,4;
BEGIN
    FOR POSICION:=9 TO 19 DO
        BEGIN
            IF NUDDI>N THEN
                BEGIN
                    POSICION:=21;
                    FER:=2;
                    goto 20;
                END
            ELSE
                BEGIN
                    GOTOXY(5,22);WRITE(' ');
                    GOTOXY(5,23);WRITE(' ');
                    GOTOXY(5,22);WRITE('Presione Barra Espaciadora en número de nudo');
                    GOTOXY(5,23);WRITE('y regresa al Menú Principal ....');
                    TEXTCOLOR(14);
                    GOTOXY(14,22);WRITE('BARRA ESPACIADORA');
                    GOTOXY(18,23);WRITE('MENU PRINCIPAL ');
                    TEXTCOLOR(7);
                    X:=0; Y:=0; Z:=0;FER:=1;
                    COLUMNA:=12;NUDD:=FunNudoFin(columna,posicion);
                    if nudo = maxint then GOTO 3;
                END
            END
        END
    END

```

```

COLUMNNA:=25;X:=FunInDec(columnna,posicion);
COLUMNNA:=35;Y:=FunInDec(columnna,posicion);
COLUMNNA:=45;Z:=FunInDec(columnna,posicion);
IF NUDD1=1 THEN GOTO 4;
GOTOXY(66,POSICION);GENERAR:=READKEY;
IF (GENERAR = 'S') OR (GENERAR = 's') THEN
BEGIN
  GOTOXY(60,8);WRITE('lineal ó uadrática █');
  TEXTCOLOR(14);
  GOTOXY(59,8);WRITE('L');
  GOTOXY(68,8);WRITE('C');
  GOTOXY(79,8);FORMA:=READKEY;
  TEXTCOLOR(7);
  IF (FORMA = 'L') OR (FORMA='l') THEN
    GENERACION_LINEAL(POSICION,NUDD1);
  IF (FORMA = 'C') OR (FORMA='c') THEN
    GENERACION_CUADRATICA(POSICION,NUDD1);
  END;adel if primero GENERAR
4:  DATO_CORRECTO(POSICION,NUDD1,N,ADRI,FER);
  NUDD1:=NUDD1+1;
  END;adel IF PRIMERO
END;adel For
CLSCR;
MARCO;
TEXTCOLOR(14);
GOTOXY(5,3);WRITE('NUMERO TOTAL DE NUDOS A INGRESAR :');
GOTOXY(41,3);WRITE(N);
GOTOXY(5,5);
WRITELN('NUMERO DE NUDD      COORDENADAS EN:      ESCALA      GENERACION');
GOTOXY(5,7);
WRITELN('                X=      Y=      Z=                S/N ');
TEXTCOLOR(7);
POSICION:=8;
GOTOXY(52,7);WRITE(S);
COORDENADAS(POSICION,NUDD1,N,ADRI,FER);
20: if posicion=21 then
  BEGIN
    ASSIGN(ARCHIVOSAP,NOMBRE);
    APPEND(ARCHIVOSAP);
    WRITELN(ARCHIVOSAP,'C FIN DE JOINTS');
    CLOSE(ARCHIVOSAP);
  END;
3: END; adel del Procedure

PROCEDURE ESCALA (VAR POSICION,NUDD1,N:INTEGER;VAR S,ADRI,FER:BYTE);
VAR
  SCONT:BYTE;
BEGIN
  TEXTCOLOR(14);
  GOTOXY(5,23);WRITELN('ESCALA: 100=Centimetros; 10=Decimetros; 1=unidades');
  COLUMNNA:=52;
  S:=FunInEnt(columnna,posicion-1);
  TEXTCOLOR(7);
  IF (S=1) OR (S=10) OR (S=100) THEN
  BEGIN
    adri:=1;
    COORDENADAS(POSICION,NUDD1,N,ADRI,FER)
  END
  ELSE
  BEGINTEXTCOLOR(14); SCONT:=0;

```

```

FOR SCONT:=1 TO 250 DO
  BEGIN
    GOTOXY(5,23);WRITE('El Dato es Erróneo, Ingrese nuevamente ');
  END;
  TEXTCOLOR(7);
  ESCALA(POSICION,NUDD1,N,S,ADRI,FER);
END; adel [f]
END; adel Procedure

```

```

(***PROGRAMA PRINCIPAL***)

```

```

BEGIN
  IF FER=0 THEN
    BEGIN
      CLSCR;
      NUDD1:=1;
      MARCO;
      GOTOXY(5,3);WRITE('NUMERO TOTAL DE NUDDOS A INGRESAR :');
      TEXTCOLOR(14);
      GOTOXY(41,3);WRITE(N);
      GOTOXY(5,5);
      WRITELN(' NUMERO DE NUDDO      COORDENADAS EN:          ESCALA      GENERACION');
      GOTOXY(5,7);
      WRITELN('                X=          Y=          Z=          S/N ');
      TEXTCOLOR(7);
      POSICION:=0;
      ADRI:=0;
      ASSIGN(ARCHIVOSAP,NUMBRE);
      APPEND(ARCHIVOSAP);
      WRITELN(ARCHIVOSAP,'JOINTS : COORDENADAS DE LOS NUDDOS');
      CLOSE(ARCHIVOSAP);
      ESCALA(POSICION,NUDD1,N,S,ADRI,FER);
    END
  ELSE
    BEGIN
      IF FER=1 THEN
        BEGIN
          ADRI:=0;NUDD1:=1;
          ASSIGN(ARCHIVOSAP,NUMBRE);
          RESET(ARCHIVOSAP);
          CODE:=0;CONTA:=0;
          WHILE NOT EOF (ARCHIVOSAP) DO
            BEGIN
              READLN(ARCHIVOSAP,NOMEST);
              CONTA:=CONTA+1;
              LUGAR:=POS('B=',NOMEST);
              IF LUGAR<>0 THEN
                BEGIN
                  VAL(COPY(NOMEST,LUGAR+2,3),NI,CODE);
                  VAL(COPY(NOMEST,LUGAR+6,3),NF,CODE);
                  VAL(COPY(NOMEST,LUGAR+10,2),IL,CODE);
                  NUDD1:=NUDD1+(((NF-NI)DIV(IL))-1);
                END;
              LUGAR:=POS('O=',NOMEST);
              IF LUGAR<>0 THEN
                BEGIN
                  VAL(COPY(NOMEST,LUGAR+2,3),NHI,CODE);
                  VAL(COPY(NOMEST,LUGAR+6,3),NHF,CODE);
                  VAL(COPY(NOMEST,LUGAR+10,3),NVI,CODE);
                  VAL(COPY(NOMEST,LUGAR+10,3),NHFCODE);
                END;
            END;
        END;
      END;
    END;
  END;

```

```

        VAL(COPY(NOMEST,LUGAR+21,2),IV,CODE);
        NUDD1:= NUDD1+((((NMF-NHI) DIV (IH))+1) * (((NVF-NVI) DIV (IV))+1))-4);
    END;
END; %WHILE NOT EOF;
CLOSE(ARCHIVOSAP);
MITAD:=8;
MITAD:=CONTA-5;
NUDD1:=NUDD1+MITAD;
CLRSCR;
MARCO;
TEXTCOLOR(14);
GOTOXY(25,2);WRITE('COORDENADAS DE LOS NUDOS');
GOTOXY(5,3);WRITE('NUMERO TOTAL DE NUDOS A INGRESAR :',N);
GOTOXY(5,4);WRITE('NUMERO TOTAL DE NUDOS INGRESADOS :',NUDD1-1);
GOTOXY(3,5);WRITE(' # NUDO      COORDENADAS EN:      ESCALA GENERACION');
GOTOXY(3,6);WRITE('      X=          Y=          Z=          S/N ');
TEXTCOLOR(7);WRITELN;
ASSIGN(ARCHIVOSAP,NOMBRE);
RESET(ARCHIVOSAP);
FOR CODE:=1 TO 5 DO
    READLN(ARCHIVOSAP,NOMEST);
REPEAT
    READLN(ARCHIVOSAP,NOMEST);
    WRITELN('||',NOMEST);
UNTIL EOF(ARCHIVOSAP);
WRITELN;WRITELN('|| PRESIONE <ENTER> PARA CONTINUAR EL INGRESO');
SW:=READKEY;
CLOSE(ARCHIVOSAP);
CLRSCR;
MARCO;
TEXTCOLOR(14);
GOTOXY(5,3);WRITE('NUMERO TOTAL DE NUDOS A INGRESAR :',N);
GOTOXY(5,4);WRITE('NUMERO TOTAL DE NUDOS INGRESADOS :',NUDD1-1);
GOTOXY(5,6);WRITE('NUMERO DE NUDO      COORDENADAS EN:      ESCALA      GENERACION');
GOTOXY(5,7);WRITE('      X=          Y=          Z=          S/N ');
TEXTCOLOR(7);
POSICION:=10;
COORDENADAS(POSICION,NUDD1,N,ADRI,FER);
END;
END;%DEL ELSE FER=1;
END; %De la Procedure Joints;

```

PROCEDURE RESTRAIN;

VAR

```

    CUENTANUDOS,J1,J2:INTEGER;
    INCREMENTO,R1,R2,R3,R4,R5,R6,LON:BYTE;
    J1AUX,J2AUX,J3AUX:STRING#3A;
    R1L,R2L,R3L,R4L,R5L,R6L:STRING#1A;

```

PROCEDURE RESTRICCION(POSICION:INTEGER;VAR CUENTANUDOS,N:INTEGER);FORWARD;

PROCEDURE CORRECTO_RES(POSICION:INTEGER;VAR CUENTANUDOS,N:INTEGER);

label 2;

BEGIN

```

    GOTOXY(2,21);WRITE(' ');
    GOTOXY(2,22);WRITE(' ');
    GOTOXY(2,23);WRITE(' ');
    GOTOXY(5,23);WRITE('Sus Datos son Correctos S/N: ');
    TEXTCOLOR(14);
    GOTOXY(29,23);WRITE('S');
    GOTOXY(23,23);WRITE('N');

```

```

GOTOXY(34,23);DATOCO:=READKEY;
IF (DATOCO='N') OR (DATOCO='n') THEN
BEGIN
  GOTOXY(5,23);WRITE(' ');
  GOTOXY(2,23);WRITE('CORREGIR:Nudo Inicial, Final; Incremento; Restricciones:R1,R2,R3,R4,R5,R6: ');
  TEXTCOLOR(14);
  GOTOXY(16,23);WRITE('I');
  GOTOXY(25,23);WRITE('F');
  GOTOXY(33,23);WRITE('N');
  GOTOXY(59,23);WRITE('1');
  GOTOXY(62,23);WRITE('2');
  GOTOXY(65,23);WRITE('3');
  GOTOXY(68,23);WRITE('4');
  GOTOXY(71,23);WRITE('5');
  GOTOXY(74,23);WRITE('6');
  TEXTCOLOR(7);
  GOTOXY(77,23);CORREGIR:=READKEY;
  CASE CORREGIR OF
    'I','i': BEGIN
      COLUMNA:=18;J1:=FunNudoFin(columna,posicion);
      if j1 = maxint then
      BEGIN
        FER:=1;
        GOTO 2;
      END;
    END;
    'F','f': BEGIN
      COLUMNA:=30;J2:=FunInEnt(columna,posicion);
    END;
    'N','n': BEGIN
      COLUMNA:=41;INCREMENTO:=FunInEnt(columna,posicion);
    END;
    '1': BEGIN
      COLUMNA:=55;R1:=FunRes(columna,posicion);
    END;
    '2': BEGIN
      COLUMNA:=57;R2:=FunRes(columna,posicion);
    END;
    '3': BEGIN
      COLUMNA:=59;R3:=FunRes(columna,posicion);
    END;
    '4': BEGIN
      COLUMNA:=61;R4:=FunRes(columna,posicion);
    END;
    '5': BEGIN
      COLUMNA:=63;R5:=FunRes(columna,posicion);
    END;
    '6': BEGIN
      COLUMNA:=65;R6:=FunRes(columna,posicion);
    END;
  END; *End_Case*
  CORRECTO_RES(POSICION,CUENTANUDOS,N);
END *End_if_Dato_Correcto*
ELSE
BEGIN
  JIAUX:=' ';LON:=0;
  STR(J1,JIAUX);
  LON:=LENGTH(JIAUX);
  WHILE LON<=3 DO
  BEGIN
    JIAUX:=JIAUX+' ';
  
```

```

END;
J2AUX:=' ';LON:=0;
STR(J2,J2AUX);
LON:=LENGTH(J2AUX);
WHILE LON<=3 DO
BEGIN
  J2AUX:=J2AUX+' ';
  lon:=lon+1;
END;
J3AUX:=' ';LON:=0;
STR(INCREMENTO,J3AUX);
LON:=LENGTH(J3AUX);
WHILE LON<=3 DO
BEGIN
  J3AUX:=J3AUX+' ';
  lon:=lon+1;
END;
STR(R1,R1L);STR(R2,R2L);STR(R3,R3L);
STR(R4,R4L);STR(R5,R5L);STR(R6,R6L);
ASSIGN(ARCHIVOSAP,NUMBRE);
RESET(ARCHIVOSAP);
APPEND(ARCHIVOSAP);
WRITELN(ARCHIVOSAP,J1AUX+' '+J2AUX+' '+J3AUX+' '+R1+' '+R2+' '+R3+' '+R4+' '+R5+' '+R6L);
CLOSE(ARCHIVOSAP);
GOTOXY(5,23);WRITE(' ');
END;
2: END;

```

```

PROCEDURE RESTRICCION(POSICION:INTEGER;VAR CUENTANUDOS,N:INTEGER);
label 20,3;
BEGIN
  FOR POSICION:=7 TO 19 DO
  BEGIN
    IF CUENTANUDOS=N THEN
    BEGIN
      POSICION:=21;
      goto 20;
    END
  ELSE
  BEGIN
    GOTOXY(5,22);WRITE(' ');
    GOTOXY(5,23);WRITE(' ');
    GOTOXY(5,22);WRITE('Presione Barra Espaciadora en el 19 Nudo');
    GOTOXY(5,23);WRITE('y regresa al Menú Principal ....');
    TEXTCOLOR(14);
    GOTOXY(14,22);WRITE('BARRA ESPACIADORA');
    GOTOXY(18,23);WRITE('MENU PRINCIPAL ');
    TEXTCOLOR(7);
    COLUMNA:=18; J1:=FunNudoFin(columna,posicion);
    if j1=maxint then
    BEGIN
      FER:=1;
      GOTO 3;
    END;
    COLUMNA:=30;
    J2:=FunInEnt(columna,posicion);
    COLUMNA:=41;
    INCREMENTO:=FunInEnt(columna,posicion);
    TEXTCOLOR(14);
    GOTOXY(18,POSICION);WRITE('R= , , , , ');
  END;

```

```

GOTOXY(5,22);WRITE(' ');
GOTOXY(5,23);WRITE(' ');
TEXTCOLOR(14);
GOTOXY(5,22);WRITE('Valores para las Restricciones');
TEXTCOLOR(7);
GOTOXY(5,23);WRITE(' = Desplazamiento permitido');
GOTOXY(36,23);WRITE(' = Desplazamiento impedido');
TEXTCOLOR(14);
GOTOXY(5,23);WRITE('0');
GOTOXY(36,23);WRITE('1');
TEXTCOLOR(7);
COLUMNA:=55;R1:=FunRes(columna,posicion);
COLUMNA:=57;R2:=FunRes(columna,posicion);
COLUMNA:=59;R3:=FunRes(columna,posicion);
COLUMNA:=61;R4:=FunRes(columna,posicion);
COLUMNA:=63;R5:=FunRes(columna,posicion);
COLUMNA:=65;R6:=FunRes(columna,posicion);
CORRECTO_RES(POSICION,CUENTANUDOS,N);
CUENTANUDOS:=CUENTANUDOS+((J2-J1)DIV(INCREMENTO))+1;
END; *Del If Primero*
END; *Del For*
CLRSCR;
MARCO;
TEXTCOLOR(14);
GOTOXY(16,2);WRITE('R E S T R I C C I O N E S   D E   L O S   N U D O S');
GOTOXY(15,5);WRITE('19 Nudo    29 Nudo  Incremento  Restricciones');
TEXTCOLOR(7);
POSICION:=5;
RESTRICCION(POSICION,CUENTANUDOS,N);
20: if posicion = 21 then
  BEGIN
    ASSIGN(ARCHIVOSAP,NOMBRE);
    APPEND(ARCHIVOSAP);
    WRITELN(ARCHIVOSAP,'C FIN DE RESTRAINTS');
    CLOSE(ARCHIVOSAP);
    FER:=2;
  END;
3: END; *De la Procedure Restricción*

BEGIN *PRINCIPAL DE RESTRINA
  IF FER=0 THEN
    BEGIN
      CLRSCR;
      POSICION:=5; CUENTANUDOS:=0;
      MARCO;
      TEXTCOLOR(14);
      GOTOXY(16,2);WRITE('R E S T R I C C I O N E S   D E   L O S   N U D O S');
      GOTOXY(15,5);WRITE('19 Nudo    29 Nudo  Incremento  Restricciones');
      TEXTCOLOR(7);
      GOTOXY(5,23);WRITE('Número Total de Nudos = ');
      GOTOXY(36,23);WRITE(N);
      ASSIGN(ARCHIVOSAP,NOMBRE);
      RESET(ARCHIVOSAP);
      APPEND(ARCHIVOSAP);
      WRITELN(ARCHIVOSAP,'RESTRAINTS : RESTRICCIONES DE LOS NUDOS');
      CLOSE(ARCHIVOSAP);
      RESTRICCION(POSICION,CUENTANUDOS,N);
    END
  ELSE
    BEGIN FER=1 THEN

```

```

BEGIN
  ASSIGN(ARCHIVOSAP,NOMBRE);
  RESET(ARCHIVOSAP);
  CONTA:=0;CUENTANUDOS:=0;
  REPEAT
    READLN(ARCHIVOSAP,NOMEST);
    LUGAR:=POS('RESTRAINTS : ',NOMEST);
    CONTA:=CONTA+1;
  UNTIL LUGAR<>0;
  REPEAT
    READLN(ARCHIVOSAP,NOMEST);
    VAL(COPY(NOMEST,1,3),J1,CODE);
    VAL(COPY(NOMEST,5,3),J2,CODE);
    VAL(COPY(NOMEST,10,3),INCREMENTO,CODE);
    GOTOXY(50,20);WRITE('J1= ',J1);READLN;
    GOTOXY(50,20);WRITE('J2= ',J2);READLN;
    GOTOXY(50,20);WRITE('INC.= ',INCREMENTO);READLN;
    CUENTANUDOS:=CUENTANUDOS+((J2-J1)DIV(INCREMENTO))+1;
  UNTIL EOF(ARCHIVOSAP);
  CLOSE(ARCHIVOSAP);
  CLRSCR;
  MARCO;
  TEXTCOLOR(14);
  GOTOXY(20,2);WRITE('R E S T R I C C I O N E S   D E   L O S   N U D O S');
  GOTOXY(5,3);WRITE('NUMERO TOTAL DE NUDOS A INGRESAR :',N);
  GOTOXY(5,4);WRITE('NUMERO TOTAL DE NUDOS INGRESADOS :',CUENTANUDOS);
  GOTOXY(5,6);
  WRITELN('# NUDO');
  TEXTCOLOR(7);
  ASSIGN(ARCHIVOSAP,NOMBRE);
  RESET(ARCHIVOSAP);
  WHILE NOT EOF (ARCHIVOSAP) DO
  BEGIN
    FOR MITAD:= 1 TO CONTA DO
      READLN(ARCHIVOSAP,NOMEST);
      READLN(ARCHIVOSAP,NOMEST);
      WRITELN('||      ',NOMEST);
    END;
  CLOSE(ARCHIVOSAP);
  WRITELN;WRITELN('|| PRESIONE <ENTER> PARA CONTINUAR EL INGRESO');
  SW:=READKEY;
  CLRSCR;
  MARCO;
  TEXTCOLOR(14);
  GOTOXY(20,2);WRITE('R E S T R I C C I O N E S   D E   L O S   N U D O S');
  GOTOXY(5,3);WRITE('NUMERO TOTAL DE NUDOS A INGRESAR :',N);
  GOTOXY(5,4);WRITE('NUMERO TOTAL DE NUDOS INGRESADOS :',CUENTANUDOS);
  GOTOXY(15,5);WRITE('10 Nudo      20 Nudo      Incremento      Restricciones');
  TEXTCOLOR(7);
  POSICION:=7;
  RESTRICION(POSICION,CUENTANUDOS,N);
  END; #DEL IF FER=10
  END; #DEL ELSE#
END; #Del Programa Principal de Restrain#

PROCEDURE SPRINGS;
VAR
  NS,JESP,IS,LON:BYTE;
  PN,UN:INTEGER;
  PRAUX,STRNNGERY,KRZ:REAL;

```

```
PROCEDURE APOYO(POSICION,NS,JESP:INTEGER); FORWARD;
```

```
PROCEDURE DATO_SPRINGS(POSICION,NS,JESP:INTEGER);
```

```
LABEL 2;
```

```
BEGIN
```

```
  GOTOXY(5,21);WRITE('');
  GOTOXY(5,22);WRITE('');
  GOTOXY(5,23);WRITE('');
  GOTOXY(5,23);WRITE('Sus Datos están Correctos S/N: ');
  TEXTCOLOR(14);
  GOTOXY(31,23);WRITE('S');
  GOTOXY(33,23);WRITE('N');
  TEXTCOLOR(7);
  GOTOXY(36,23);DATOCO:=READKEY;
  IF (DATOCO='N') OR (DATOCO='n') THEN
    BEGIN
      GOTOXY(5,22);WRITE('Corregir: Nudo Inicial; Nudo Final; Incremento; Constante Kx;');
      GOTOXY(5,23);WRITE('Constante Ky; Constante Kz; Constante Krx; Constante Kry; Constante Krz: ');
      TEXTCOLOR(14);
      GOTOXY(21,22);WRITE('I');GOTOXY(35,22);WRITE('F');GOTOXY(47,22);WRITE('M');GOTOXY(55,22);WRITE('O')
      GOTOXY(7,23);WRITE('N');GOTOXY(22,23);WRITE('S');GOTOXY(37,23);WRITE('T');
      GOTOXY(53,23);WRITE('A');GOTOXY(71,23);WRITE('E');
      TEXTCOLOR(7);
      GOTOXY(78,23);CORREGIR:=READKEY;
      CASE CORREGIR OF
        'I','i' : BEGIN
          COLUMNA:=9;PN:=FunNudoFin(columna,posicion);
          if pn=maxint then
            BEGIN
              FER:=1;
              GOTO 2;
            END;
          END;
        'F','f' : BEGIN
          COLUMNA:=18;UN:=FunInEnt(columna,posicion);
          END;
        'M','m' : BEGIN
          COLUMNA:=30;IS:=FunInEnt(columna,posicion);
          END;
        'O','o' : BEGIN
          COLUMNA:=38;KX:=FunInDec2(columna,posicion);
          END;
        'N','n' : BEGIN
          COLUMNA:=44;KY:=FunInDec2(columna,posicion);
          END;
        'S','s' : BEGIN
          COLUMNA:=49;KZ:=FunInDec2(columna,posicion);
          END;
        'T','t' : BEGIN
          COLUMNA:=54;KRX:=FunInDec2(columna,posicion);
          END;
        'A','a' : BEGIN
          COLUMNA:=60;KRY:=FunInDec2(columna,posicion);
          END;
        'E','e' : BEGIN
          COLUMNA:=66;KRZ:=FunInDec2(columna,posicion);
          END;
      END; #End_Case#
      DATO_SPRINGS(POSICION,NS,JESP);
    END #End_If_Datoco#
```

```

ELSE
BEGIN
  ASSIGN(ARCHIVOSAP,NOMBRE);
  APPEND(ARCHIVOSAP);
  PNAUX:=' ';LON:=0;
  STR(PN,PNAUX);
  LON:=LENGTH(PNAUX);
  WHILE LON<=3 DO
  BEGIN
    PNAUX:=PNAUX+' ';
    LON:=LON+1;
  END;
  WRITELN(ARCHIVOSAP,PNAUX,' ',UN:3,' ',IS:2,' K=',KX:4:3,',',KY:4:3,',',KZ:4:3,',',
  KRX:4:3,',',KRY:4:3,',',KRZ:4:3);
  JESP:=JESP+1;
  CLOSE(ARCHIVOSAP);
END;
2: END;

```

```

PROCEDURE APOYO(POSICION,NS,JESP:INTEGER);

```

```

label 20,3;

```

```

BEGIN

```

```

  FOR POSICION:= 9 TO 19 DO

```

```

  BEGIN

```

```

    IF JESP>NS THEN

```

```

    BEGIN

```

```

      POSICION:=21;

```

```

      goto 20;

```

```

    END

```

```

  ELSE

```

```

  BEGIN

```

```

    GOTOXY(5,22);WRITE(' ');
```

```

    GOTOXY(5,23);WRITE(' ');
```

```

    GOTOXY(5,22);WRITE('Presione Barra Espaciadora en Nudo Inicial');
```

```

    GOTOXY(5,23);WRITE('y regresa al Menú Principal ....');
```

```

    TEXTCOLOR(14);

```

```

    GOTOXY(14,22);WRITE('BARRA ESPACIADORA');
```

```

    GOTOXY(18,23);WRITE('MENU PRINCIPAL ');
```

```

    TEXTCOLOR(7);

```

```

    COLUMNA:=9;PN:=FunNudoFin(columna,posicion);
```

```

    if pn=maxint then

```

```

    BEGIN

```

```

      FER:=1;

```

```

      GOTO 3;

```

```

    END;

```

```

    COLUMNA:=18;UN:=FunInEnt(columna,posicion);
```

```

    COLUMNA:=30;IS:=FunInEnt(columna,posicion);
```

```

    GOTOXY(5,21);WRITE('K = Constante de Resortes');
```

```

    GOTOXY(5,22);WRITE('Kx = Para Traslación-x ; Ky = Para Traslación-y ; Kz = Para Traslación-z');
```

```

    GOTOXY(5,23);WRITE('Krx = Para Rotación-x ; Kry = Para Rotación-y ; Krz = Para Rotación-z');
```

```

    TEXTCOLOR(14);

```

```

    GOTOXY(5,21); WRITE('K = C');GOTOXY(22,21);WRITE('R');
```

```

    GOTOXY(5,22); WRITE('Kx');GOTOXY(16,22);WRITE('T');GOTOXY(27,22);WRITE('x');
```

```

    GOTOXY(31,22);WRITE('Ky');GOTOXY(42,22);WRITE('T');GOTOXY(53,22);WRITE('y');
```

```

    GOTOXY(57,22);WRITE('Kz');GOTOXY(68,22);WRITE('T');GOTOXY(79,22);WRITE('z');
```

```

    GOTOXY(5,23); WRITE('Krx');GOTOXY(16,23);WRITE('R');GOTOXY(25,23);WRITE('x');
```

```

    GOTOXY(31,23);WRITE('Kry');GOTOXY(42,23);WRITE('R');GOTOXY(53,23);WRITE('y');
```

```

    GOTOXY(57,23);WRITE('Krz');GOTOXY(68,23);WRITE('R');GOTOXY(77,23);WRITE('z');
```

```

    TEXTCOLOR(7);

```

```

    COLUMNA:=30;KY:=FunInDec2(columna,posicion);

```

```

        COLUMNA:=49;KZ:=FunInDec2(columna,posicion);
        COLUMNA:=54;KRX:=FunInDec2(columna,posicion);
        COLUMNA:=60;KRY:=FunInDec2(columna,posicion);
        COLUMNA:=66;KRZ:=FunInDec2(columna,posicion);
        DATO_SPRINGS(POSICION,NS,JESP);
        JESP:=JESP+1;
    END;#End_jesP#
END;#End_For#
CLRSCL;
MARCO;
TEXTCOLOR(14);
GOTOXY(20,4);WRITE('A P O Y O S   E S P E C I A L E S');
GOTOXY(6,6);WRITE(' Nudo      Nudo      Incremento      K');
GOTOXY(6,7);WRITE(' Inicial    Final                Kx   Ky   Kz   Krx   Kry   Krz');
TEXTCOLOR(7);
POSICION:=8;
APOYO(POSICION,NS,JESP);
20: if posicion=21 then
    BEGIN
        ASSIGN(ARCHIVOSAP,NOMBRE);
        APPEND(ARCHIVOSAP);
        WRITELN(ARCHIVOSAP,'C FIN DE SPRINGS');
        CLOSE(ARCHIVOSAP);
        FER:=2;
    END;
3: END;#End_Procedure#

#PROGRAMA PRINCIPAL#
BEGIN
    IF FER=0 THEN
        BEGIN
            CLRSCL;
            MARCO;
            TEXTCOLOR(14);
            GOTOXY(20,2);WRITE('A P O Y O S   E S P E C I A L E S');
            GOTOXY(6,4);WRITE(' Ingrese la cantidad de apoyos a declarar:');
            TEXTCOLOR(7);
            GOTOXY(5,22);WRITE(' Ingrese 0 si no quiere ingresar datos');
            TEXTCOLOR(14);GOTOXY(13,22);WRITE('0');TEXTCOLOR(7);
            NS:=0;COLUMNA:=48;POSICION:=4;
            NS:=FunInEnt(columna,posicion);
            TEXTCOLOR(14);
            GOTOXY(6,6);WRITE(' Nudo      Nudo      Incremento      K');
            GOTOXY(6,7);WRITE(' Inicial    Final                Kx   Ky   Kz   Krx   Kry   Krz');
            TEXTCOLOR(7);
            JESP:=1;POSICION:=0;
            ASSIGN(ARCHIVOSAP,NOMBRE);
            APPEND(ARCHIVOSAP);
            WRITELN(ARCHIVOSAP,'SPRINGS : APOYOS ESPECIALES ',NS);
            CLOSE (ARCHIVOSAP);
            APOYO(POSICION,NS,JESP);
        END
    ELSE
        BEGIN
            IF FER=1 THEN
                BEGIN
                    ASSIGN(ARCHIVOSAP,NOMBRE);
                    RESET(ARCHIVOSAP);
                    CODE:=0;CONTA:=0;
                    REPEATADLN(ARCHIVOSAP,NOMEST);

```

```

        CONTA:=CONTA+1;
        LUGAR:=POS(' SPRINGS ',NOMEST);
UNTIL LUGAR<>#;
CLOSE(ARCHIVOSAP);
VAL(COPY(NOMEST,LUGAR+28,3),NS,CODE);
ASSIGN(ARCHIVOSAP,NOMBRE);
RESET(ARCHIVOSAP);
CODE:=#;
WHILE NOT EOF (ARCHIVOSAP) DO
BEGIN
    READLN(ARCHIVOSAP,NOMEST);
    CODE:=CODE+1;
END;
CLOSE(ARCHIVOSAP);
MITAD:=CODE-CONTA;
CLRSCR;
MARCO;
TEXTCOLOR(14);
GOTOXY(28,2);WRITE('A P O Y O S   E S P E C I A L E S');
GOTOXY(6,4);WRITE('Número total de apoyos a declarar: ',NS);
GOTOXY(6,5);WRITE('Número de apoyos ingresados:      ',MITAD);
GOTOXY(3,6);WRITE('Nudo   Nudo Incr.   K = Constante de resortes');
GOTOXY(3,7);WRITE(' 1       2 ');
Writeln;
TEXTCOLOR(7);
ASSIGN(ARCHIVOSAP,NOMBRE);
RESET(ARCHIVOSAP);
FOR CODE:=1 TO CONTA DO
    READLN(ARCHIVOSAP,NOMEST);
REPEAT
    READLN(ARCHIVOSAP,NOMEST);
    Writeln('|| ',NOMEST);
UNTIL EOF(ARCHIVOSAP);
Writeln;Writeln('|| PRESIONE <ENTER> PARA CONTINUAR EL INGRESO');
SW:=READKEY;
CLOSE(ARCHIVOSAP);
CLRSCR;
MARCO;
TEXTCOLOR(14);
GOTOXY(28,2);WRITE('A P O Y O S   E S P E C I A L E S');
GOTOXY(6,4);WRITE('Número total de apoyos a declarar: ',NS);
GOTOXY(6,5);WRITE('Número de apoyos ingresados:      ',MITAD);
GOTOXY(6,6);WRITE(' Nudo   Nudo   Incremento           K');
GOTOXY(6,7);WRITE(' Inicial   Final           Kx   Ky   Kz   Krx   Kry   Krz');
TEXTCOLOR(7);
POSICION:=9;
JESP:=1;
JESP:=JESP+MITAD;
APOYO(POSICION,NS,JESP);
END; #DEL IF FER=1#
END; #DEL ELSE#
END; #del procedure springs#

PROCEDURE CONSTRA1;
VAR
    NC,JCONS,IC,C1,C2,C3,C4,C5,C6,I1,I2,I3,I4,I5,I6,LON:BYTE;
    PNC,UNC:INTEGER;
    PNCAUX:STRING#3A;
PROCEDURE CONCATENACION(POSICION:INTEGER;NC,JCONS:BYTE); FORWARD;
PROCEDURE DATO_CONST(POSICION:INTEGER;NC,JCONS:BYTE);

```

LABEL 2;

BEGIN

```
GOTOXY(5,21);WRITE(' ');
GOTOXY(5,22);WRITE(' ');
GOTOXY(5,23);WRITE(' ');
GOTOXY(5,23);WRITE('Sus Datos están Correctos S/N: ');
TEXTCOLOR(14);
GOTOXY(31,23);WRITE('S');
GOTOXY(33,23);WRITE('N');
TEXTCOLOR(7);
GOTOXY(36,23);DATOCO:=READKEY;
IF (DATOCO='N') OR (DATOCO='n') THEN
BEGIN
  GOTOXY(5,23);WRITE(' ');
  GOTOXY(5,21);WRITE('Corregir: N.Inicial; N.Final; Incremento; Desplaza.c1; Desplaza.c2');
  GOTOXY(5,22);WRITE('Desplaza.c3; Rota.c4, Rota.c5, Rota.c6; Nudo i1; Nudo i2; Nudo i3');
  GOTOXY(5,23);WRITE('Nudo i4; Nudo i5; Nudo i6: ');
  TEXTCOLOR(14);
  GOTOXY(18,21);WRITE('I');GOTOXY(29,21);WRITE('F');GOTOXY(38,21);WRITE('C');GOTOXY(48,21);WRITE('D');
  GOTOXY(62,21);WRITE('E');
  GOTOXY(7,22);WRITE('S');GOTOXY(18,22);WRITE('R');GOTOXY(28,22);WRITE('O');GOTOXY(38,22);WRITE('T');
  GOTOXY(51,22);WRITE('1');GOTOXY(60,22);WRITE('2');GOTOXY(69,22);WRITE('3');
  GOTOXY(11,23);WRITE('4');GOTOXY(20,23);WRITE('5');GOTOXY(29,23);WRITE('6');
  TEXTCOLOR(7);
  GOTOXY(32,23);CORREGIR:=READKEY;
  CASE CORREGIR OF
    'I','i' : BEGIN
      GOTOXY(4,POSICION);WRITE(' ');
      COLUMNA:=4; PNC:=FunNudoFin(columna,posicion);
      if pnc=maxint then
      BEGIN
        FER:=1;GOTO 2;
      END;
    END;
    'F','f' : BEGIN
      GOTOXY(11,POSICION);WRITE(' ');
      COLUMNA:=11; UNC:=FunInEnt(columna,posicion);
    END;
    'C','c' : BEGIN
      GOTOXY(19,POSICION);WRITE(' ');
      COLUMNA:=19; IC:=FunInEnt(columna,posicion);
    END;
    'D','d' : BEGIN
      GOTOXY(26,POSICION);WRITE(' ');
      COLUMNA:=26; CI:=FunInEnt(columna,posicion);
    END;
    'E','e' : BEGIN
      GOTOXY(30,POSICION);WRITE(' ');
      COLUMNA:=30; C2:=FunInEnt(columna,posicion);
    END;
    'S','s' : BEGIN
      GOTOXY(34,POSICION);WRITE(' ');
      COLUMNA:=34; C3:=FunInEnt(columna,posicion);
    END;
    'R','r' : BEGIN
      GOTOXY(38,POSICION);WRITE(' ');
      COLUMNA:=38; C4:=FunInEnt(columna,posicion);
    END;
    'O','o' : BEGIN
      GOTOXY(42,POSICION);WRITE(' ');
      COLUMNA:=42; C5:=FunInEnt(columna,posicion);
```

```

END;
't', 't' : BEGIN
    GOTDXY(46, POSICION); WRITE(' ');
    COLUMNA:=46; C6:=FunInEnt(columna, posicion);
END;
'1' : BEGIN
    GOTDXY(53, POSICION); WRITE(' ');
    COLUMNA:=53; I1:=FunInEnt(columna, posicion);
END;
'2' : BEGIN
    GOTDXY(57, POSICION); WRITE(' ');
    COLUMNA:=57; I2:=FunInEnt(columna, posicion);
END;
'3' : BEGIN
    GOTDXY(61, POSICION); WRITE(' ');
    COLUMNA:=61; I3:=FunInEnt(columna, posicion);
END;
'4' : BEGIN
    GOTDXY(65, POSICION); WRITE(' ');
    COLUMNA:=65; I4:=FunInEnt(columna, posicion);
END;
'5' : BEGIN
    GOTDXY(69, POSICION); WRITE(' ');
    COLUMNA:=69; I5:=FunInEnt(columna, posicion);
END;
'6' : BEGIN
    GOTDXY(73, POSICION); WRITE(' ');
    COLUMNA:=73; I6:=FunInEnt(columna, posicion);
END;
END; #End_Case#
DATO_CONST(POSICION, NC, JCONS);
END #End_ff_Datoco#
ELSE
BEGIN
    ASSIGN(ARCHIVOSAP, NOMBRE);
    APPEND(ARCHIVOSAP);
    PNCAUX:=' '; LON:=0;
    STR(PNC, PNCAUX);
    LON:=LENGTH(PNCAUX);
    WHILE LON<=3 DO
    BEGIN
        PNCAUX:=PNCAUX+' ';
        LON:=LON+1;
    END;
    WRITELN(ARCHIVOSAP, PNCAUX, ' ', UNC:3, ' ', IC:2, ' C=', C1:3, ' ', C2:3, ' ', C3:3, ' ', C4:3, ' ', C5:3, ' ', C6:3, ' ', I1:3, ' ', I2:3, ' ', I3:3, ' ', I4:3, ' ', I5:3, ' ', I6:3);
    CLOSE(ARCHIVOSAP);
    JCONS:=JCONS+1;
    GOTDXY(5, 23); WRITE(' ');
END; #End_else_datoco#
2: END; #End_Procedure#

```

```

PROCEDURE CONCATENACION(POSICION: INTEGER; NC, JCONS: BYTE);

```

```

LABEL 20, 3;

```

```

BEGIN

```

```

    FOR POSICION:= 9 TO 20 DO

```

```

        BEGIN

```

```

            IF JCONS>NC THEN

```

```

                BEGIN

```

```

                    GOTDXY(21);

```

```

END
ELSE
BEGIN
    GOTOXY(5,22);WRITE(' ');
    GOTOXY(5,23);WRITE(' ');
    GOTOXY(5,22);WRITE(' Presione Barra Espaciadora en número de nudo');
    GOTOXY(5,23);WRITE(' y regresa al Menú Principal ....');
    TEXTCOLOR(14);
    GOTOXY(14,22);WRITE(' BARRA ESPACIADORA');
    GOTOXY(18,23);WRITE(' MENU PRINCIPAL ');
    TEXTCOLOR(7);
    COLUMNA:=4; PNC:=FunNudoFin(columna,posicion);
    if pnc=maxint then
    BEGIN
        FER:=1;GOTO 3;
    END;
    COLUMNA:=11; UNC:=FunInEnt(columna,posicion);
    COLUMNA:=19; IC:=FunInEnt(columna,posicion);
    GOTOXY(5,21);WRITE(' C = Número de junta independiente del ');
    GOTOXY(5,22);WRITE(' c1 = Desplazamiento-x ; c2 = Desplazamiento-y ; c3 = Desplazamiento-z');
    GOTOXY(5,23);WRITE(' c4 = Rotación-x ; c5 = Rotación-y ; c6 = Rotación-z');
    TEXTCOLOR(14);
    GOTOXY(5,21); WRITE(' C = N');GOTOXY(20,21);WRITE(' J');GOTOXY(26,21);WRITE(' I');
    GOTOXY(5,22); WRITE(' c1');GOTOXY(10,22);WRITE(' D');GOTOXY(25,22);WRITE(' X');
    GOTOXY(29,22);WRITE(' c2');GOTOXY(34,22);WRITE(' D');GOTOXY(49,22);WRITE(' Y');
    GOTOXY(53,22);WRITE(' c3');GOTOXY(58,22);WRITE(' D');GOTOXY(73,22);WRITE(' Z');
    GOTOXY(5,23); WRITE(' c4');GOTOXY(10,23);WRITE(' R');GOTOXY(19,23);WRITE(' X');
    GOTOXY(29,23);WRITE(' c5');GOTOXY(34,23);WRITE(' R');GOTOXY(43,23);WRITE(' Y');
    GOTOXY(53,23);WRITE(' c6');GOTOXY(58,23);WRITE(' R');GOTOXY(67,23);WRITE(' Z');
    TEXTCOLOR(7);
    COLUMNA:=26; C1:=FunInEnt(columna,posicion);
    COLUMNA:=30; C2:=FunInEnt(columna,posicion);
    COLUMNA:=34; C3:=FunInEnt(columna,posicion);
    COLUMNA:=38; C4:=FunInEnt(columna,posicion);
    COLUMNA:=42; C5:=FunInEnt(columna,posicion);
    COLUMNA:=46; C6:=FunInEnt(columna,posicion);
    GOTOXY(5,21);WRITE(' ');
    GOTOXY(5,22);WRITE(' ');
    GOTOXY(5,23);WRITE(' ');
    GOTOXY(5,21);WRITE(' I = Incremento en el número del nudo para las series ');
    GOTOXY(5,22);WRITE(' i1 = Nudo c1 ; i2 = Nudo c2 ; i3 = Nudo c3 ; ');
    GOTOXY(5,23);WRITE(' i4 = Nudo c4 ; i5 = Nudo c5 ; i6 = Nudo c6 ; ');
    TEXTCOLOR(14);
    GOTOXY(5,21);WRITE(' I = I');GOTOXY(27,21);WRITE(' N');GOTOXY(38,21);WRITE(' N');GOTOXY(52,21);WR
    GOTOXY(5,22);WRITE(' i1');GOTOXY(10,22);WRITE(' N');GOTOXY(15,22);WRITE(' c1');
    GOTOXY(20,22);WRITE(' i2');GOTOXY(25,22);WRITE(' N');GOTOXY(30,22);WRITE(' c2');
    GOTOXY(35,22);WRITE(' i3');GOTOXY(40,22);WRITE(' N');GOTOXY(45,22);WRITE(' c3');
    GOTOXY(5,23);WRITE(' i4');GOTOXY(10,23);WRITE(' N');GOTOXY(15,23);WRITE(' c4');
    GOTOXY(20,23);WRITE(' i5');GOTOXY(25,23);WRITE(' N');GOTOXY(30,23);WRITE(' c5');
    GOTOXY(35,23);WRITE(' i6');GOTOXY(40,23);WRITE(' N');GOTOXY(45,23);WRITE(' c6');
    TEXTCOLOR(7);
    COLUMNA:=53; I1:=FunInEnt(columna,posicion);
    COLUMNA:=57; I2:=FunInEnt(columna,posicion);
    COLUMNA:=61; I3:=FunInEnt(columna,posicion);
    COLUMNA:=65; I4:=FunInEnt(columna,posicion);
    COLUMNA:=69; I5:=FunInEnt(columna,posicion);
    COLUMNA:=73; I6:=FunInEnt(columna,posicion);
    DATO_CONST(POSICION,NC,JCONS);
    JCONS:=JCONS+1;
    END;End_JCONS ELSE
END;End_For

```

```

CLRSCR;
MARCO;
TEXTCOLOR(14);
GOTOXY(10,2);WRITE('C O N C A T E N A C I O N   D E   N U D O S');
GOTOXY(6,4);WRITE('Ingrese la cantidad de concatenaciones a declarar :');
TEXTCOLOR(7);
GOTOXY(60,4);WRITE(NC);
TEXTCOLOR(14);
GOTOXY(2,6);WRITE(' Nudo   Nudo   Incre-           C           I');
GOTOXY(2,7);WRITE('Inicial Final mento   c1 c2 c3 c4 c5 c6   i1 i2 i3 i4 i5 i6');
TEXTCOLOR(7);
POSICION:=8;
CONCATENACION(POSICION,NC,JCONS);
20: if posicion=21 then
  BEGIN
    ASSIGN(ARCHIVOSAP,NOMBRE);
    RESET(ARCHIVOSAP);
    APPEND(ARCHIVOSAP);
    WRITELN(ARCHIVOSAP,'C FIN DE CONSTRAINTS');
    CLOSE(ARCHIVOSAP);
    FER:=2;
  END;
3: ENO;≠End_Procedure≠

≠PROGRAMA PRINCIPAL≠
BEGIN
  IF FER=0 THEN
    BEGIN
      CLRSCR;
      MARCO;
      TEXTCOLOR(14);
      GOTOXY(10,2);WRITE('C O N C A T E M A C I O N   D E   N U D O S');
      GOTOXY(6,4);WRITE('Ingrese la cantidad de concatenaciones a declarar :');
      TEXTCOLOR(7);
      GOTOXY(5,22);WRITE('Ingrese 0 si no desea ingresar datos');
      TEXTCOLOR(14);GOTOXY(13,22);WRITE('0');TEXTCOLOR(7);
      NC:=0; COLUMNA:=50; POSICION:=4;
      NC:=FunInEnt(columna,posicion);
      TEXTCOLOR(14);
      GOTOXY(2,6);WRITE(' Nudo   Nudo   Incre-           C           I');
      GOTOXY(2,7);WRITE('Inicial Final mento   c1 c2 c3 c4 c5 c6   i1 i2 i3 i4 i5 i6');
      TEXTCOLOR(7);
      JCONS:=1; POSICION:=8;
      ASSIGN(ARCHIVOSAP,NOMBRE);
      APPEND(ARCHIVOSAP);
      WRITELN(ARCHIVOSAP,'CONSTRAINTS ; CONCATENACION DE NUDOS ',NC);
      CLOSE(ARCHIVOSAP);
      CONCATENACION(POSICION,NC,JCONS);
    END
  ELSE
    BEGIN
      IF FER=1 THEN
        BEGIN
          ASSIGN(ARCHIVOSAP,NOMBRE);
          RESET(ARCHIVOSAP);
          CODE:=0;CONTA:=0;
          REPEAT
            READLN(ARCHIVOSAP,NOMEST);
            CONTA:=CONTA+1;
          UNTIL LOGAR(0);
          LOGAR:=POSICION,'CONSTRAINTS ',NOMEST);
          UNTIL LOGAR(0);
        END
      END
    END
  END
END

```

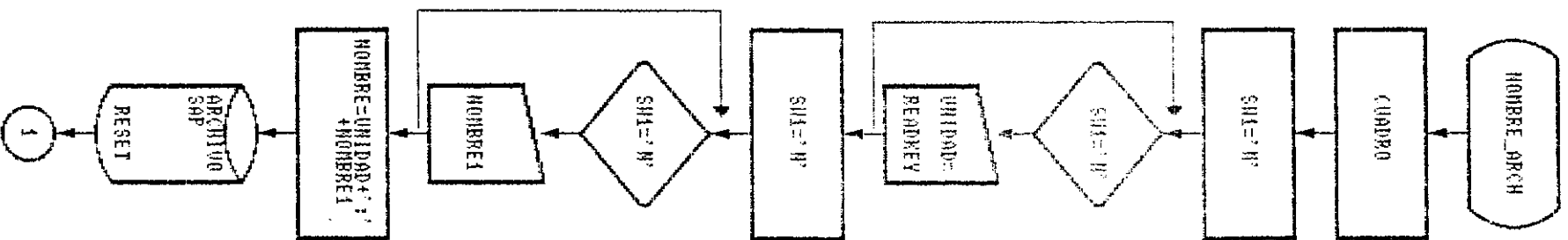
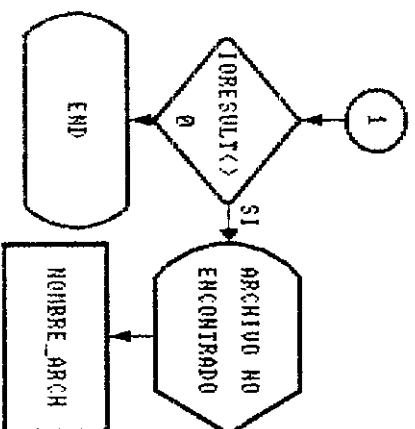
```

CLOSE(ARCHIVOSAP);
VAL(COPY(NOMEST,LUGAR+37,3),NC,CODE);
ASSIGN(ARCHIVOSAP,NOMBRE);
RESET(ARCHIVOSAP);
CODE:=0;
WHILE NOT EOF (ARCHIVOSAP) DO
BEGIN
    READLN(ARCHIVOSAP,NOMEST);
    CODE:=CODE+1;
END;
CLOSE(ARCHIVOSAP);
MITAD:=CODE-CONTA;
CLASCR;
MARCO;
TEXTCOLOR(14);
GOTOXY(10,2);WRITE('C O N C A T E N A C I O N      D E      N U D O S');
GOTOXY(6,4);WRITE('Número total de concatenaciones a declarar: ',NC);
GOTOXY(6,5);WRITE('Número de concatenaciones ingresados:      ',MITAD);
GOTOXY(3,6);WRITE('Nudo  Nudo  Inc.  C = Juntas Dependientes      I = Series de Nudos');
GOTOXY(3,7);WRITE(' 1      2');
WRITELN;
TEXTCOLOR(7);
ASSIGN(ARCHIVOSAP,NOMBRE);
RESET(ARCHIVOSAP);
FOR CODE:= 1 TO CONTA DO
    READLN(ARCHIVOSAP,NOMEST);
POSICION:=7;
REPEAT
    READLN(ARCHIVOSAP,NOMEST);
    WRITELN('|| ',NOMEST);
UNTIL EOF(ARCHIVOSAP);
WRITELN;WRITELN('|| PRESIONE <ENTER> PARA CONTINUAR EL INGRESO');
SW:=READKEY;
CLOSE(ARCHIVOSAP);
CLASCR;
MARCO;
TEXTCOLOR(14);
GOTOXY(10,2);WRITE('C O N C A T E N A C I O N      D E      N U D O S');
GOTOXY(6,4);WRITE('Número total de concatenaciones a declarar: ',NC);
GOTOXY(6,5);WRITE('Número de concatenaciones ingresadas:      ',MITAD);
TEXTCOLOR(14);
GOTOXY(2,6);WRITE(' Nudo  Nudo  Incre-      C      I');
GOTOXY(2,7);WRITE(' Inicial Final  mento  c1 c2 c3 c4 c5 c6      i1 i2 i3 i4 i5 i6');
TEXTCOLOR(7);
POSICION:=9;
JCONS:=1;
JCONS:=JCONS+MITAD;
CONCATENACION(POSICION,NC,JCONS);
END; *DEL IF FER=18
END; *DEL ELSE3
END; *de la procedure constrai3
END. *DE LA UNIDAD INGRESOR

```

LISTA DE VARIABLES

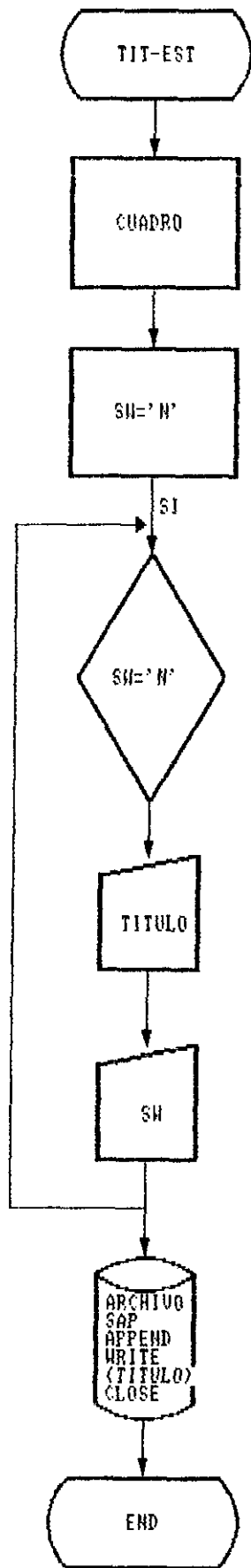
| BOHRRE DEL PROGRAMA: EJEMPLO | TIPO | DESCRIPCION |
|------------------------------|---------|------------------|
| BOHRRE | CHAR | Ayuda |
| BOHRRE | NUMERIC | Para Leer nombre |
| BOHRRE | CHAR | Lee unidad de |
| BOHRRE | CHAR | diaco |
| BOHRRE | CHAR | Para saber en |
| | | d o n d e |
| | | eventual |



LISTA DE VARIABLES

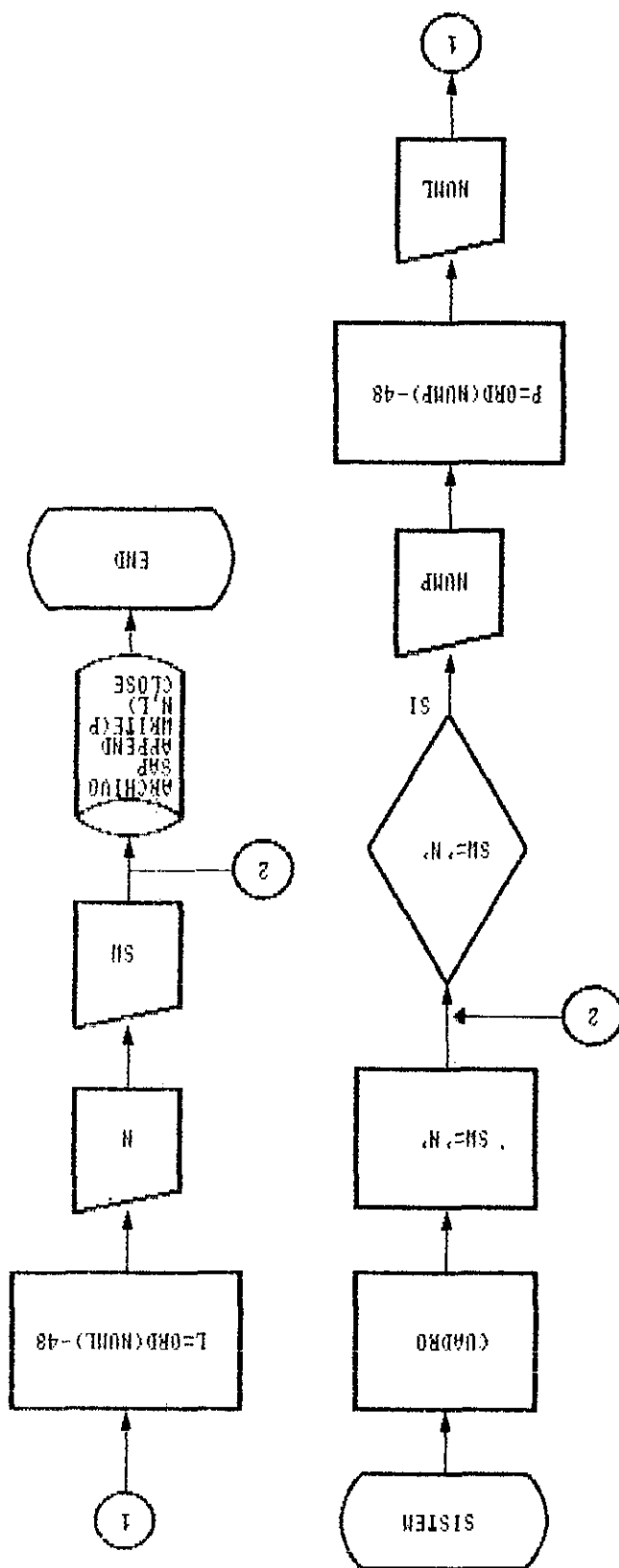
NOMBRE DEL PROGRAMA: TEE.BAT

| NOMBRE | TIPO | DESCRIPCION |
|--------|------|-------------------------|
| TITULO | Char | Nombre de la estructura |
| SW | Char | Switch de ayuda |



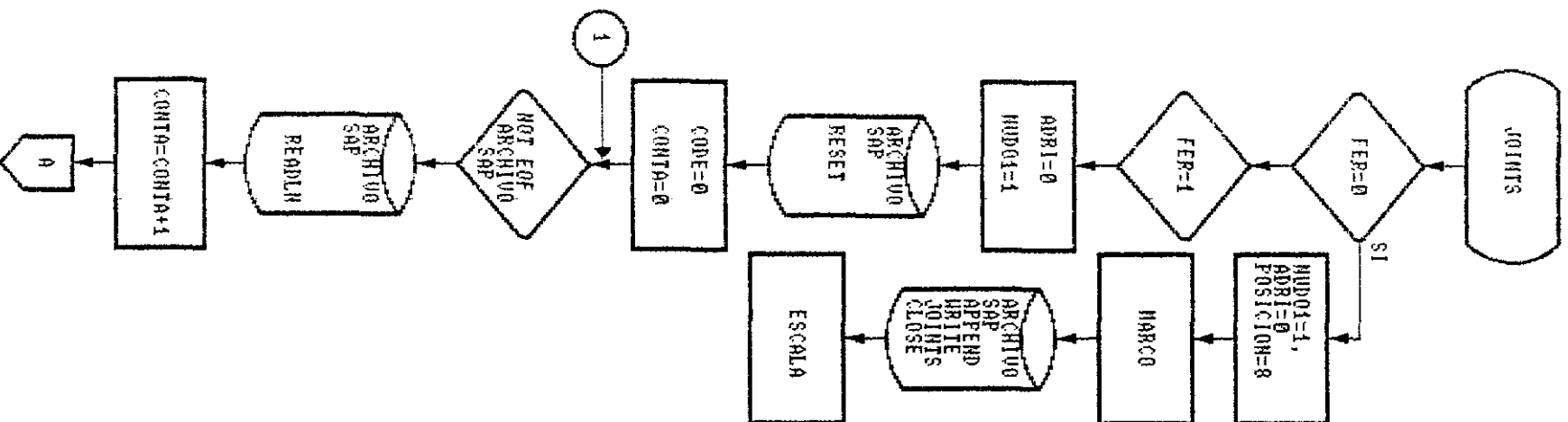
HOMBRE DEL PROGRAMA. ESTADO

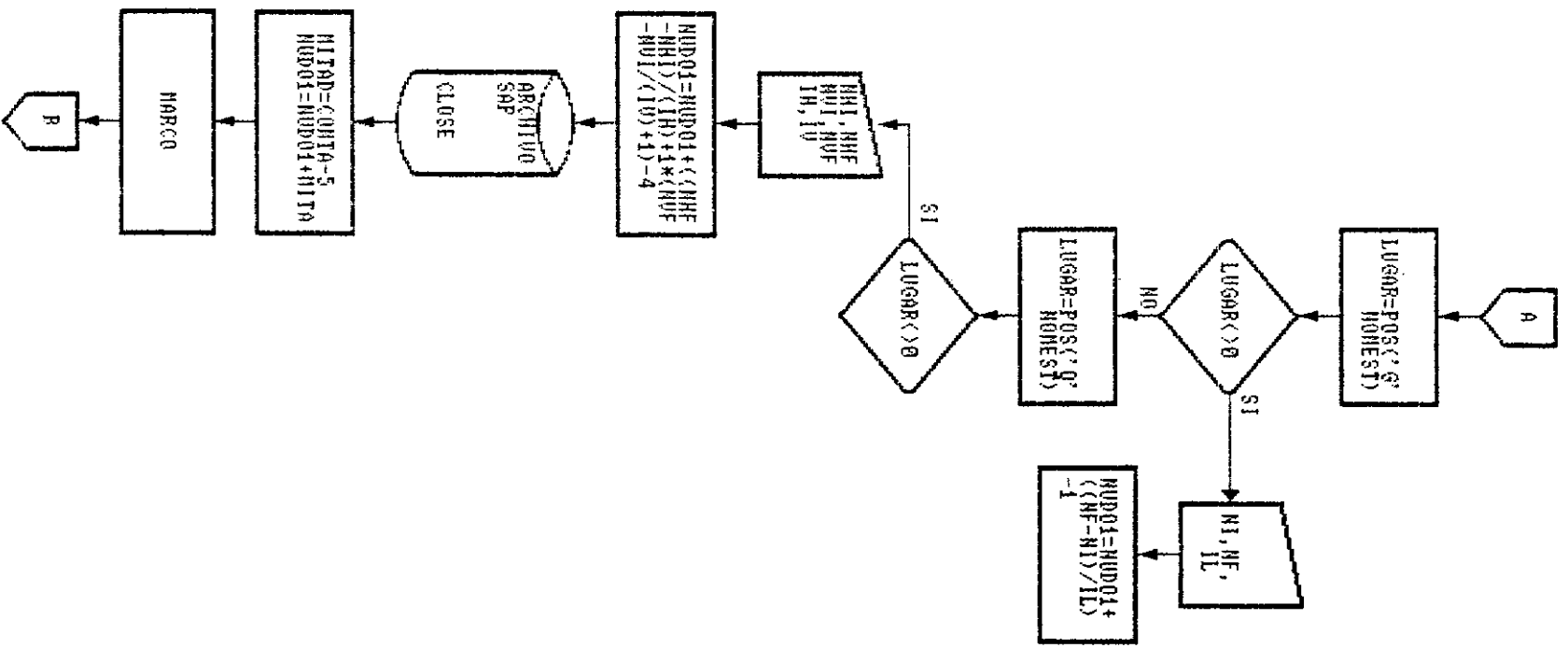
| HOMBRE | TIPO | DESCRIPCION |
|--------|------|------------------|
| .32 | Char | Switch de estado |
| P00P | Char | Para Leer P |
| P00L | Char | Para Leer L |
| P | Byte | Ordinal de Hom |

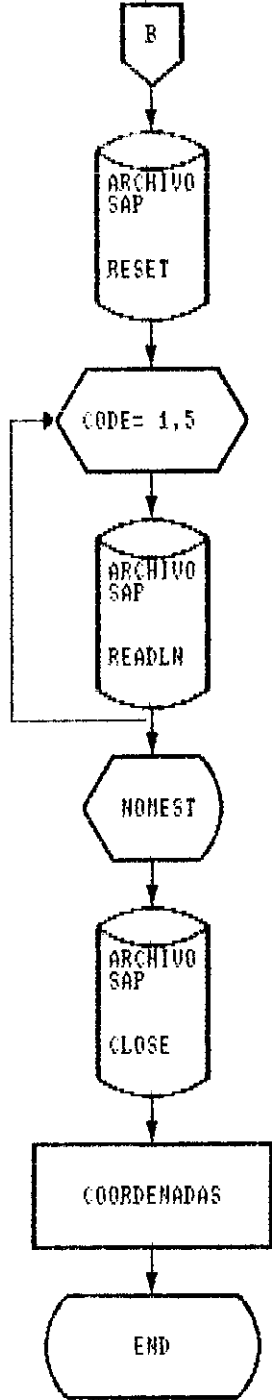


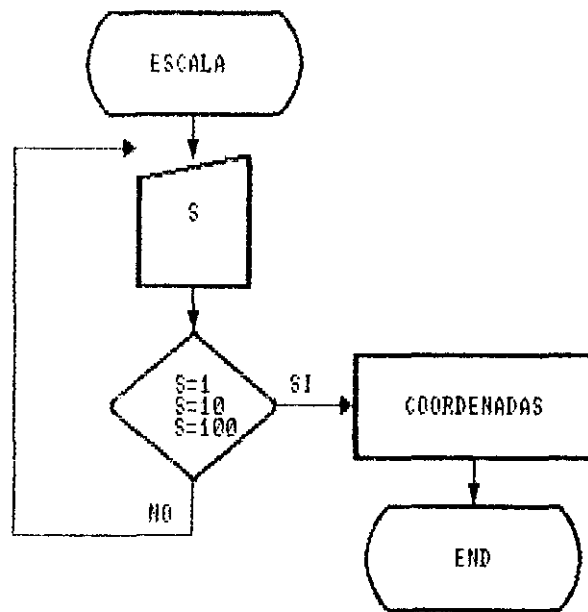
NOMBRE DEL PROGRAMA: DIBUJO

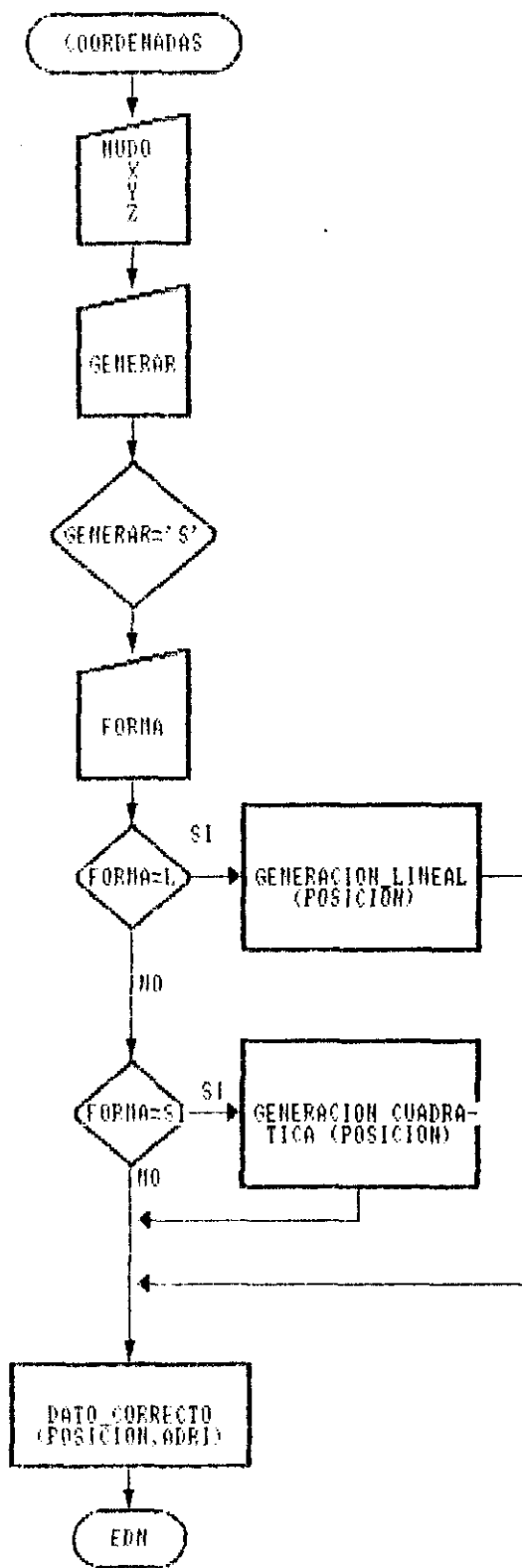
| NOMBRE | TIPO | DESCRIPCION |
|---------------|-------------|----------------------------|
| ARCHI | Byte | Para guardar en el archivo |
| AY | Char | Para ayuda |
| NUMERO | Integer | Número de nudo que ingresa |
| E | Byte | Escala |
| X | Real | Valor de X |
| Y | Real | Valor de Y |
| Z | Real | Valor de Z |
| HI | Integer | Nudo Inicial |
| HF | Integer | Nudo Final |
| HHI | Integer | Nudo Horizontal Inicial |
| HHF | Integer | Nudo Horizontal Final |
| HVI | Integer | Nudo Vertical Inicial |
| HVF | Integer | Nudo Vertical Final |
| II | Byte | Incremento Horizontal |
| IV | Byte | Incremento Vertical |
| LI | Byte | Incremento Lineal |
| NUMCADX | String | Para guardar en el archivo |

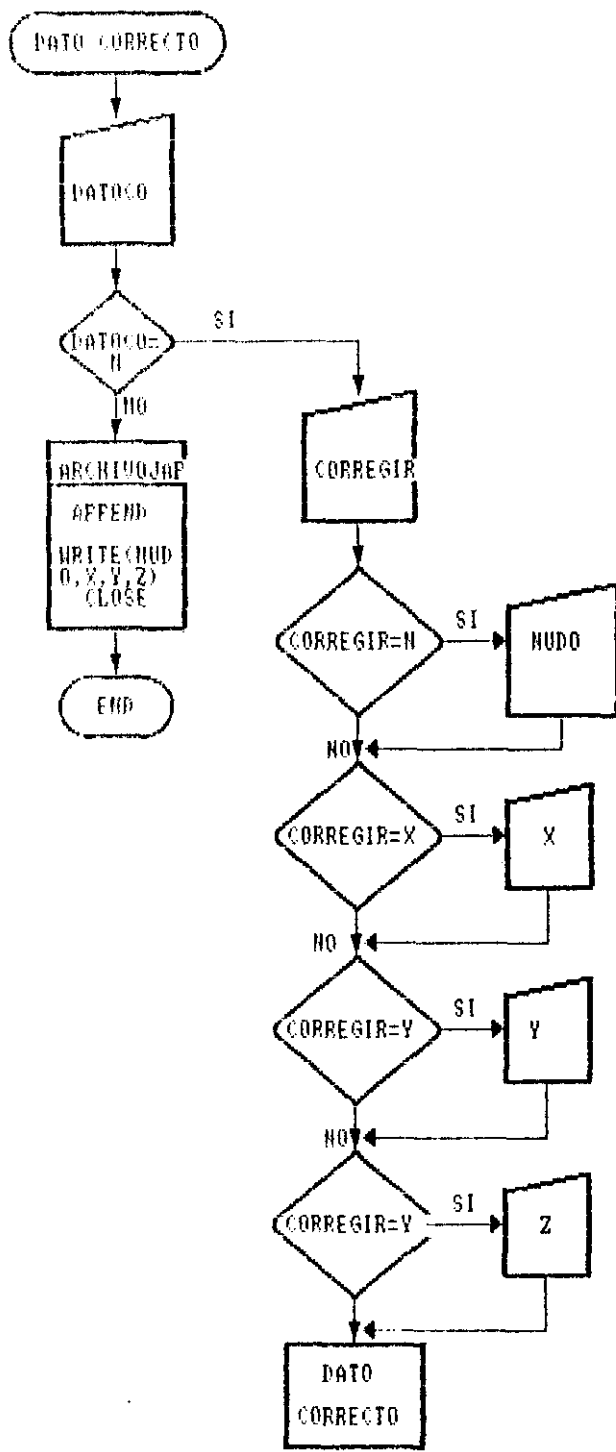


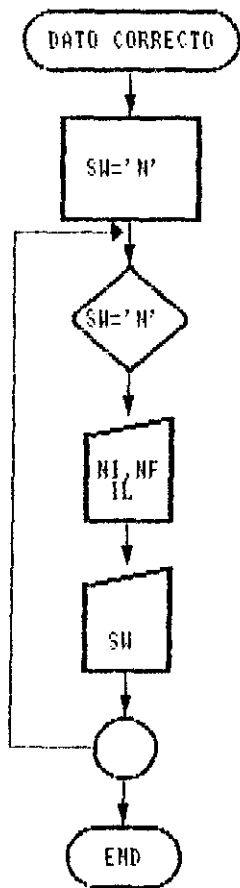


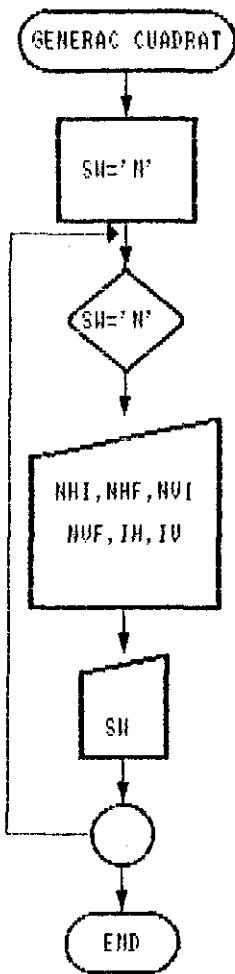








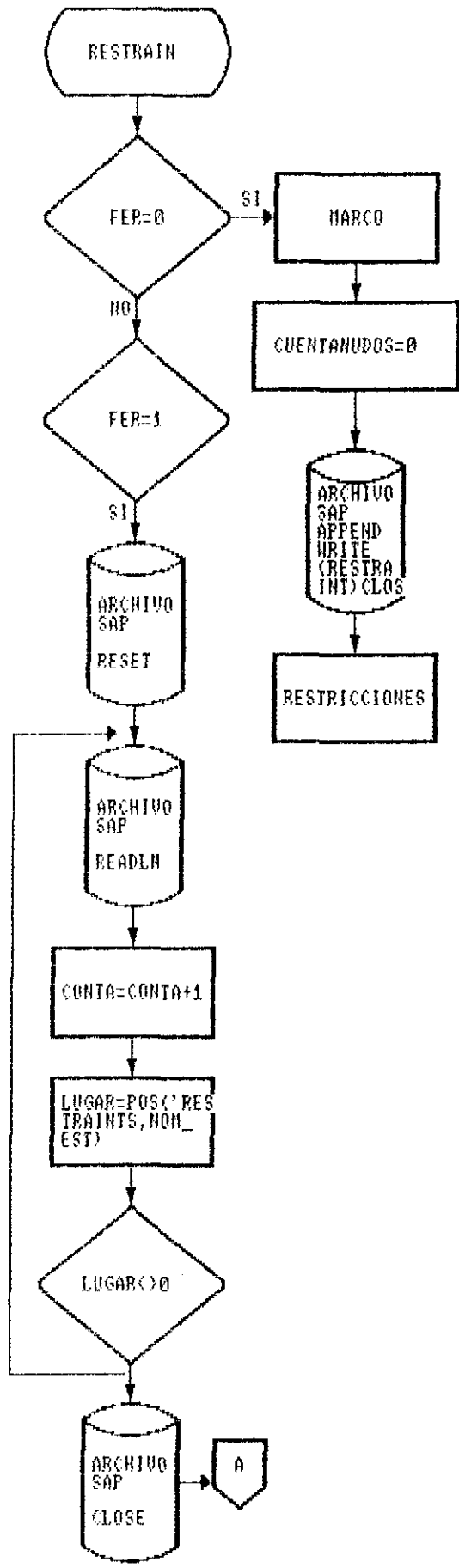


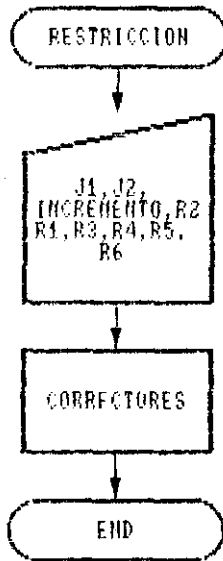


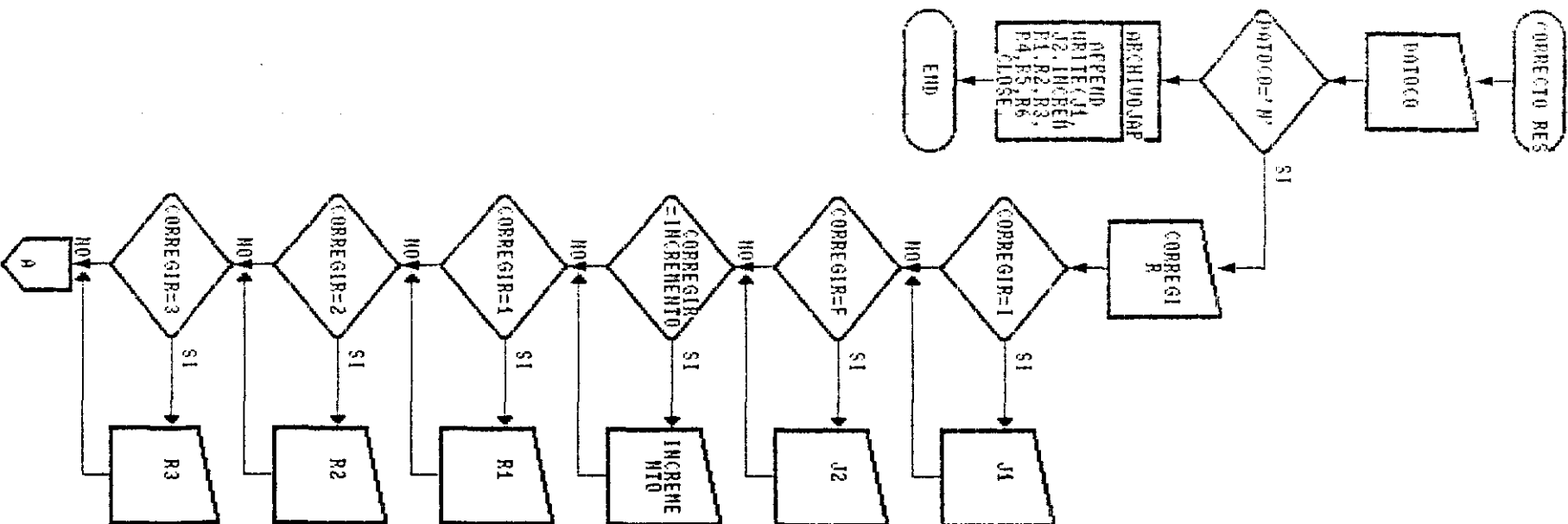
NOBRE DEL PROGRAMA: NETRAIN

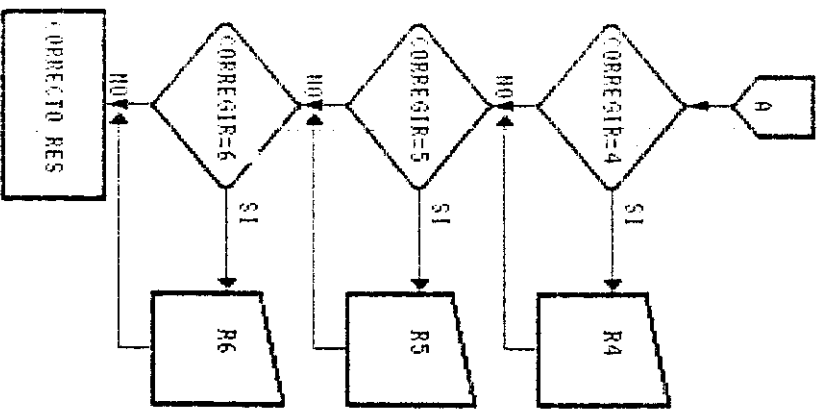
| NOBRE | TIPO | DESCRIPCION |
|------------|---------|--|
| CANTIDADN | Integer | Para contar el número de nudos generados |
| J1 | Integer | Primer Nudo |
| J2 | Integer | Segundo Nudo |
| R1 | Byte | Restricciones de los nudos |
| R2 | Byte | Restricciones de los nudos |
| R3 | Byte | Restricciones de los nudos |
| R4 | Byte | Restricciones de los nudos |
| R5 | Byte | Restricciones de los nudos |
| R6 | Byte | Restricciones de los nudos |
| INCREMENTO | Byte | Incremento de los nudos |
| JIAUX | String | Para grabar |
| JCAUX | String | Para grabar |
| IOI | Byte | Para grabar |
| JCAUX | String | Para grabar incremento |
| RUI | String | Para grabar |

| | | |
|-----|--------|-------------|
| RGB | String | Para global |
| RGB | String | Para global |
| RGB | String | Para global |
| RGB | String | Para global |
| RGB | String | Para global |



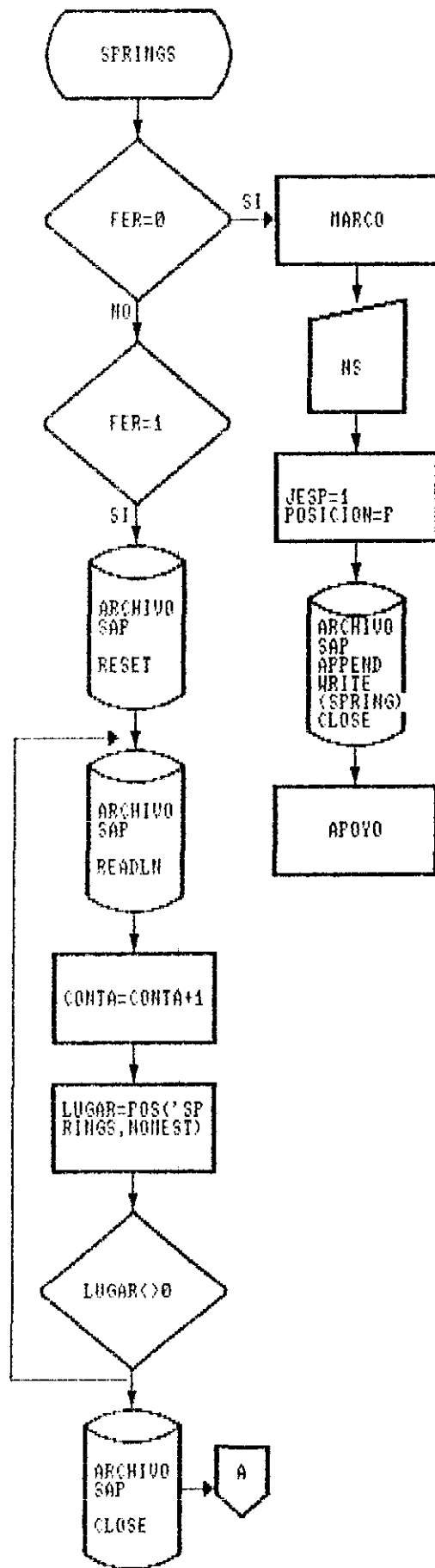


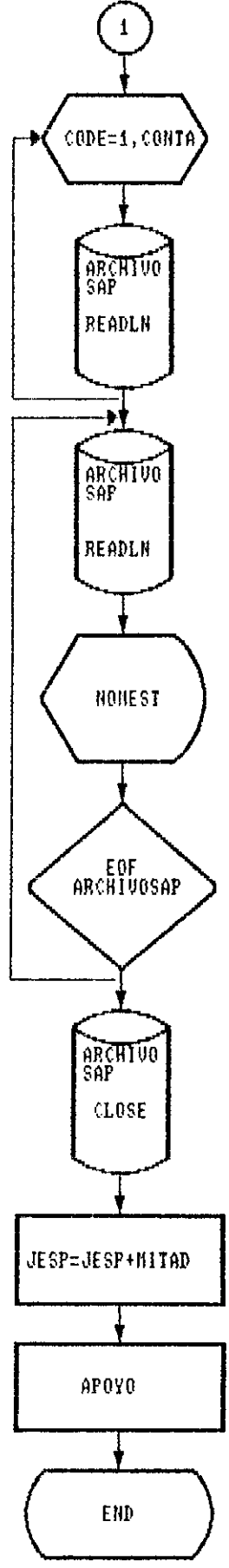
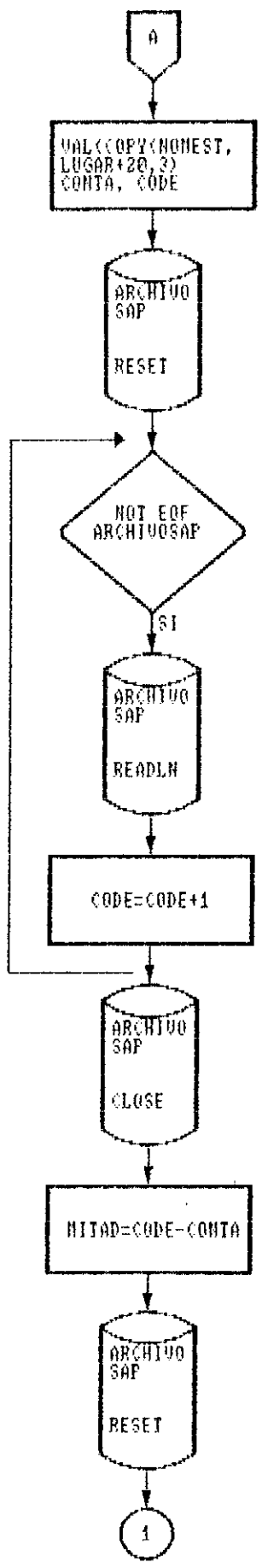


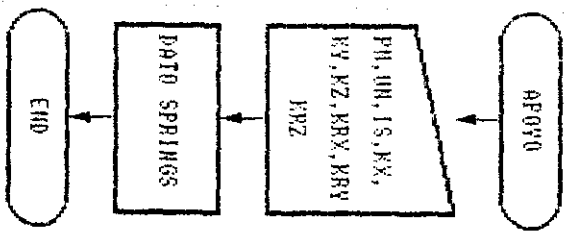


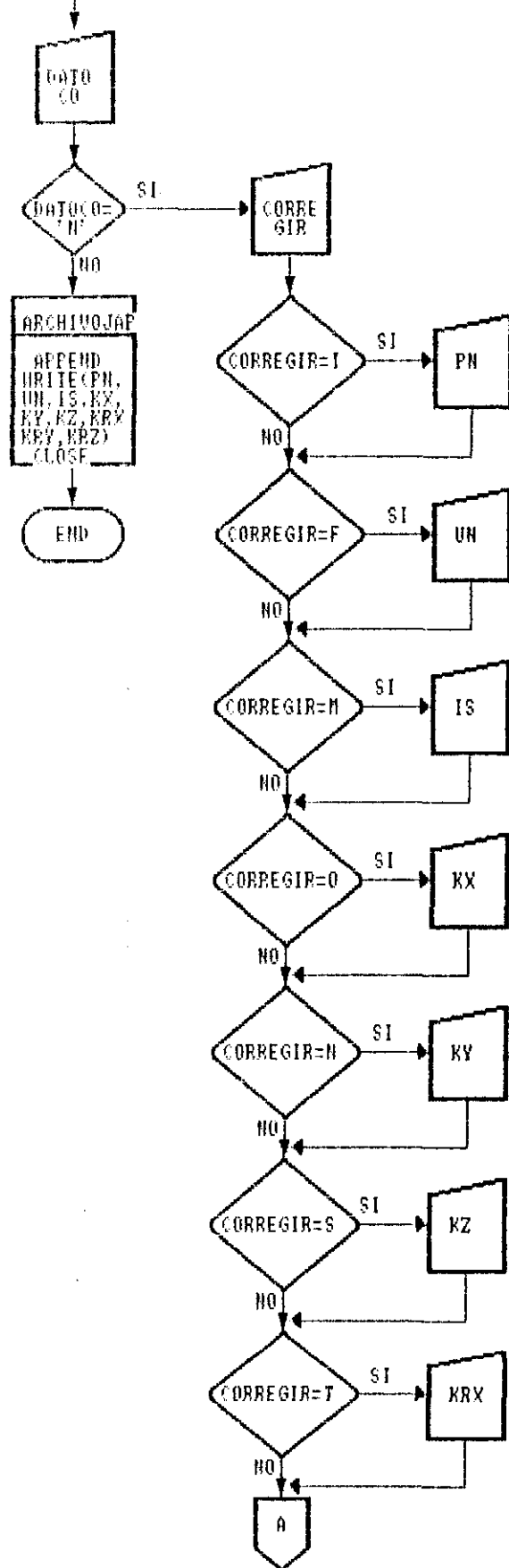
NOMBRE DEL PROGRAMA: CTE11937

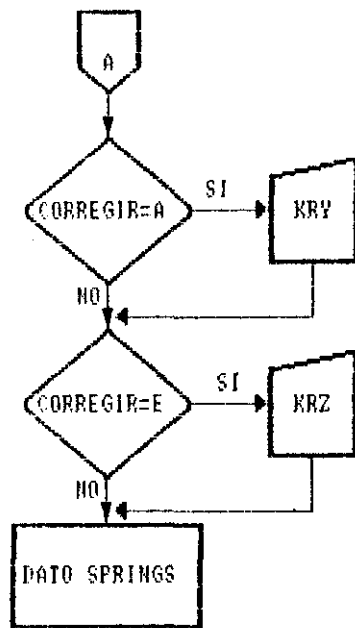
| NOMBRE | TIPO | DESCRIPCION |
|--------|---------|--------------------------------|
| NU | Byte | Número a declarar |
| ICNT | Byte | Contador |
| PI | Integer | Primer índice |
| PII | Integer | Ultimo índice |
| IG | Byte | Incremento |
| EE | Real | Constante para traslación X |
| EY | Real | Constante para traslación Y |
| EZ | Real | Constante para traslación Z |
| EEK | Real | Constante para rotación X |
| EKY | Real | Constante para rotación Y |
| EKZ | Real | Constante para rotación Z |









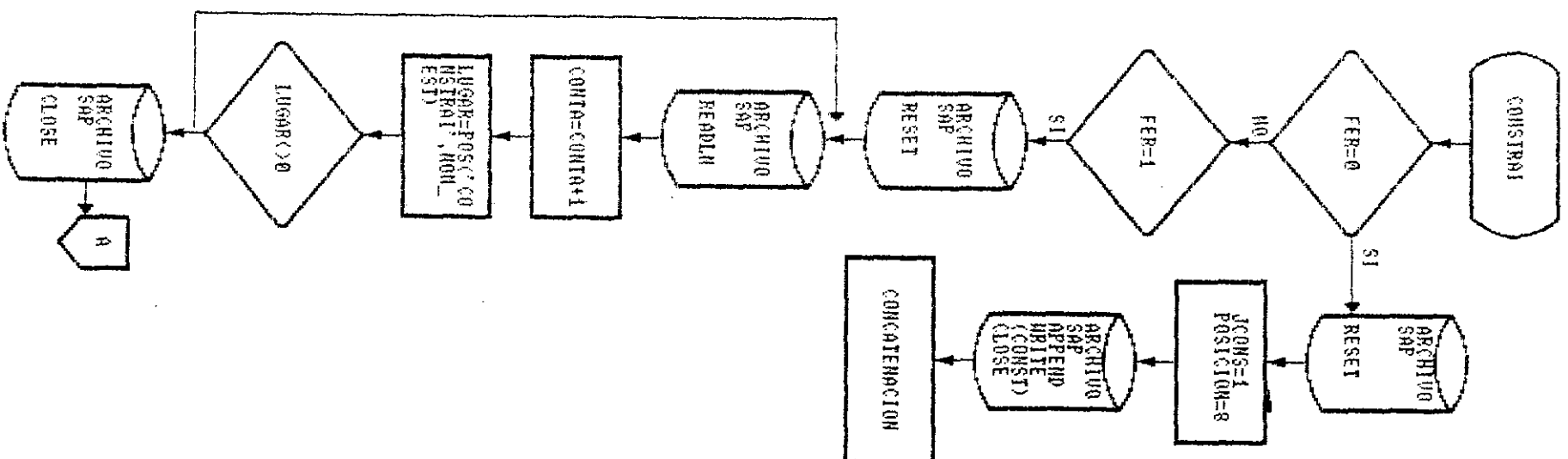


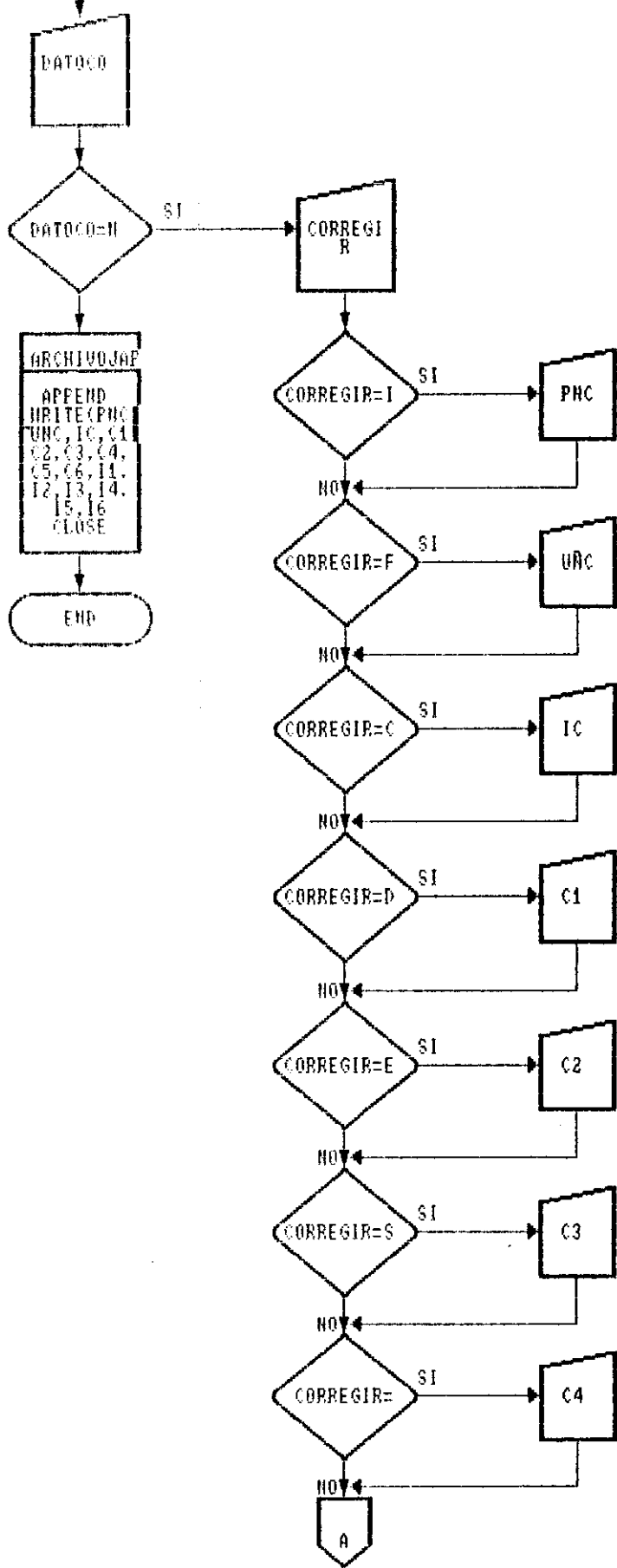
LISTA DE VARIABLES

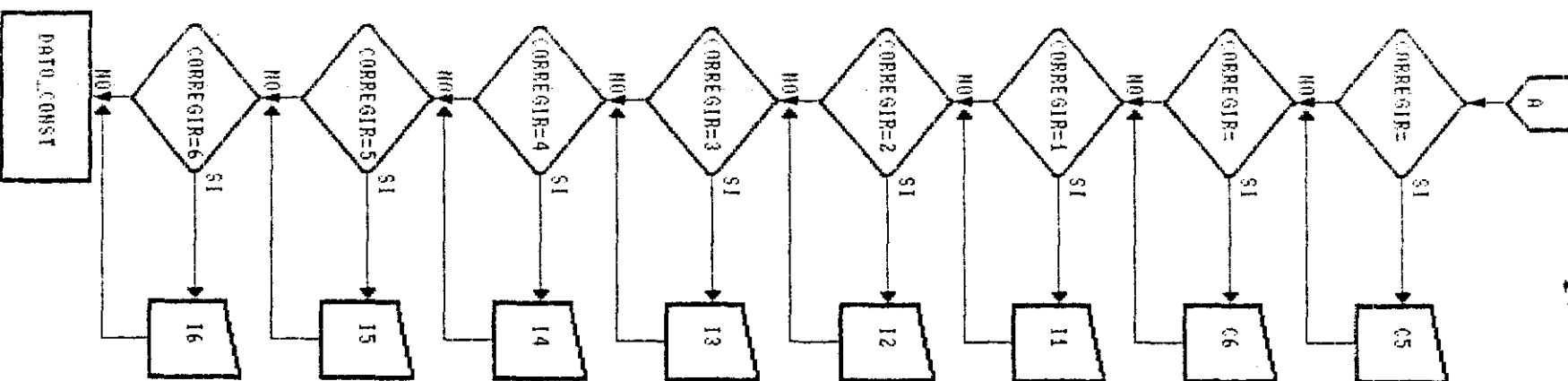
PROGRAMA DE PROBABILIDAD

| ABRABR | TIPO | DESCRIPCION |
|--------|---------|---|
| 00 | Byte | Número a declarar |
| 00000 | Byte | Contador |
| 0001 | Integer | Modo Inicial |
| 0002 | Integer | Modo Final |
| 00 | Byte | Incremento |
| 001 | Byte | Número de Junta |
| | | Independiente de desplazamiento X |
| 002 | Byte | Número de Junta |
| | | Independiente de desplazamiento Y |
| 003 | Byte | Número de Junta |
| | | Independiente de rotación Z |
| 004 | Byte | Número de Junta |
| | | Independiente de rotación X |
| 005 | Byte | Número de Junta |
| | | Independiente de rotación Y |
| 006 | Byte | Número de Junta |
| | | Independiente de rotación Z |
| 007 | Byte | Incremento en el número de modo para el giro 01 |

| | | |
|----|------|--|
| 12 | Byte | Incremento en el número de nudos para el giro 02 |
| 13 | Byte | Incremento en el número de nudos para el giro 03 |
| 14 | Byte | Incremento en el número de nudos para el giro 04 |
| 15 | Byte | Incremento en el número de nudos para el giro 05 |
| 16 | Byte | Incremento en el número de nudos para el giro 06 |







INTERFACÉ

Uses Crt, Unifun, Caratula;

TYPE

STR10=STRING#14A;

VAR

POSICION,COLUMNA,M,COFE:INTEGER;

CONTA,MITAD,L:BYTE;

NOMEST:STRING#80A;

ARCHIVOSAP:TEXT;

DATOCO,CORREGIR,SW:CHAR;

LUGAR:BYTE;

PROCEDURE FRAME(NOMBRE:STR10;VAR N:INTEGER;VAR L,FER:BYTE);

PROCEDURE QUAD(NOMBRE:STR10;VAR N:INTEGER;VAR L,FER:BYTE);

IMPLEMENTATION

PROCEDURE FRAME;

VAR

CMV,E,DELN,ALN:INTEGER;

CM,CM1,M,NMG,INI,INF,IN,LP1,LP2,LP3,CV,CMU,CS:BYTE;

NM,NL,J,ADRI,ADRI1,ADRI2,ADRI3,ADRI4,LON:BYTE;

NMAUX,NLAUX:STRING#3A;

B1,B2,B3,D1,D2,D3,A,I,WX,WY,WZ:REAL;

SWS,SW,SECPRO,CONTROLTIPO,GENER,COORDE,CARGA:CHAR;

J AUX,CMVAUX,DELNAUX,ALNAUX,MAUX,NMGAUX,INIAUX,INFAUX,IMAUX,LP1AUX,LP2AUX,LP3AUX,CVAUX,CMUAUX,CSAUX:STR

WXAUX,WYAUX,WZAUX,AAUX,IAUX,B1AUX,B2AUX,B3AUX,D1AUX,D2AUX,D3AUX:STRING#6A;

PROCEDURE MIEMBROS(POSICION:INTEGER;NM,NL,J:BYTE);FORWARD;

PROCEDURE ESTADOS_CARGA(POSICION:INTEGER;NL,J:BYTE);FORWARD;

PROCEDURE TIPO_MIEMBRO(POSICION:INTEGER;NM,NL,J:BYTE); FORWARD;

PROCEDURE LECTURA(POSICION:INTEGER;NM,NL,CM,CM1,adri:byte);FORWARD;

PROCEDURE MIEMBRO_CORRECTO(POSICION:INTEGER;NM,NL,J,CM:BYTE);

label 2;

BEGIN

GOTOXY(5,23);WRITE(' ');

GOTOXY(5,23);WRITE('Sus Datos están Correctos S/N: ');

TEXTCOLOR(14);

GOTOXY(31,23);WRITE('S');

GOTOXY(33,23);WRITE('N');

TEXTCOLOR(7);

GOTOXY(36,23);DATOCO:=READKEY;

IF (DATOCO='N') OR (DATOCO='n') THEN

BEGIN

WRITEXY(5,23,' ');

WRITEXY(5,23,'Corregir: Miembro, Del Nudo, Al Nudo : ');

TEXTCOLOR(14);

GOTOXY(15,23);WRITE('M');

GOTOXY(24,23);WRITE('D');

GOTOXY(34,23);WRITE('A');

TEXTCOLOR(7);

GOTOXY(43,23);CORREGIR:=READKEY;

CASE CORREGIR OF

'M','m': BEGIN

COLUMNA:=5;CMV:=FunNudoFin(columna,posicion);

if cmv=maxint then

BEGIN

ERR=2;

```

END;
END;
'D', 'd': BEGIN
    COLUMNA:=13; DELN:=FunInEnt(columna, posicion);
END;
'B', 'a': BEGIN
    COLUMNA:=19; ALN:=FunInEnt(columna, posicion);
END;
END;  aEnd_Casea
MIEMBRO_CORRECTO(POSICION, NM, NL, J, CM);
END  aEnd_If Dato Correcto
ELSE
    J:=J+1;
2: END;  aEnd_Procedure MIEMBRO_CORRECTO

```

```

PROCEDURE LECTURA(POSICION: INTEGER; NM, NL, CM, CM1, adri: byte);
label 10, 20, 3;

```

```

PROCEDURE SIEMPRE;
BEGIN

```

```

    CMVAUX:= ' '; LON:=0;
    STR(CMV, CMVAUX);
    LON:=LENGTH(CMVAUX);
    WHILE LON<=3 DO
    BEGIN
        CMVAUX:=CMVAUX+' ';
        lon:=lon+1;
    END;
    DELNAUX:= ' '; LON:=0;
    STR(DELN, DELNAUX);
    LON:=LENGTH(DELNAUX);
    WHILE LON<=3 DO
    BEGIN
        DELNAUX:=DELNAUX+' ';
        lon:=lon+1;
    END;
    ALNAUX:= ' '; LON:=0;
    STR(ALN, ALNAUX);
    LON:=LENGTH(ALNAUX);
    WHILE LON<=3 DO
    BEGIN
        ALNAUX:=ALNAUX+' ';
        lon:=lon+1;
    END;
END;

```

```

END;
PROCEDURE OPCION1;
BEGIN

```

```

    MAUX:= ' '; LON:=0;
    STR(M, MAUX);
    LON:=LENGTH(MAUX);
    WHILE LON<=3 DO
    BEGIN
        MAUX:=MAUX+' ';
        lon:=lon+1;
    END;
END;

```

```

END;
PROCEDURE OPCION2;
BEGIN
    STR(NMG:1, NMGAUX);
    STR(INB:1, INGAUX);

```

```

    STR(IM:1, IMAUX);
END;
PROCEDURE OPCION3;
BEGIN
    STR(LP1:1, LP1AUX);
    STR(LP2:1, LP2AUX);
    STR(LP3:1, LP3AUX);
END;
PROCEDURE OPCION4;
BEGIN
    STR(CV:1, CVAUX);
    STR(CMU:1, CMUAUX);
    STR(CS:1, CSAUX);
END;

BEGIN xdel procedure lectura
    TEXTCOLOR(14);
    GOTOXY(4,5);WRITE('Miembro Del Al Tipo miembro Generación Coordenadas Estados Carga');
    GOTOXY(4,6);WRITE(' Nudo Nudo S/N S/N S/N S/N ');
    TEXTCOLOR(7);
    POSICION:=0;
    FOR POSICION:=POSICION TO 19 DO
        BEGIN
            IF CM1>CM THEN
                BEGIN
                    posicion:=21;
                    goto 10
                END
            ELSE
                BEGIN
                    GOTOXY(5,22);WRITE('Presione BARRA ESPACIADORA en Miembro');
                    GOTOXY(5,23);WRITE(' y regresa al MENU PRINCIPAL');
                    TEXTCOLOR(14);
                    GOTOXY(14,22);WRITE('BARRA ESPACIADORA');
                    GOTOXY(19,23);WRITE('MENU PRINCIPAL');
                    TEXTCOLOR(7);
                    COLUMNA:=5; CMV:=FunNudoFin(columna,posicion);
                    if cmv=maxint then
                        BEGIN
                            FER:=1;
                            GOTO 3;
                        END;
                    GOTOXY(5,22);WRITE(' ');
                    GOTOXY(5,23);WRITE(' ');
                    COLUMNA:=13; DELN:=FunInEnt(columna,posicion);
                    COLUMNA:=19; ALN:=FunInEnt(columna,posicion);
                    MIEMBRO_CORRECTO(POSICION, NM, NL, J, CM);
                    ADRI:=0;
                    GOTOXY(30, POSICION); CONTROLTIPO:=READKEY;
                    adril:=0;
                    IF (CONTROLTIPO='S') OR (CONTROLTIPO='s') THEN
                        BEGIN
                            GOTOXY(28, POSICION);WRITE('M=');
                            COLUMNA:=30; SMS='S';
                            WHILE SMS='S' DO
                                begin
                                    M:=FunInEnt(columna,posicion);
                                    if m>nm then sms='S';
                                    if m<=nm then sms='N';
                                end;
                            while

```

```

END; #if CONTROLTIPOA
GOTOXY(44,POSICION);GENER:=READKEY;
adri2:=0;
IF (GENER = 'S') OR (GENER = 's') THEN
BEGIN
    sw:='N';
    While (sw='N') or (sw='n') do
    Begin
        GOTOXY(5,20);WRITE('Orden de ingreso para Generación');
        GOTOXY(5,21);WRITE('Número de miembros a generar, Incremento al nudo inicial,');
        GOTOXY(5,22);WRITE('Incremento al nudo final, Incremento de miembro');
        TEXTCOLOR(14);
        GOTOXY(27,20);WRITE('G');
        GOTOXY(5,21);WRITE('N');
        GOTOXY(35,21);WRITE('I');
        GOTOXY(5,22);WRITE('I');
        GOTOXY(31,22);WRITE('I');
        GOTOXY(40,POSICION);WRITE('G= , , ');
        TEXTCOLOR(7);
        COLUMNA:=42; NMG:=FunInEnt(columna,posicion);
        COLUMNA:=45; INI:=FunInEnt(columna,posicion);
        COLUMNA:=48; INF:=FunInEnt(columna,posicion);
        COLUMNA:=51; IM:=FunInEnt(columna,posicion);
        GOTOXY(5,20);WRITE(' ');
        GOTOXY(5,21);WRITE(' ');
        GOTOXY(5,22);WRITE(' ');
        GOTOXY(5,23);WRITE('Datos de Generación Correctos S/N: ');
        TEXTCOLOR(14);
        GOTOXY(35,23);WRITE('S');gotoxy(37,23);write('N');
        GOTOXY(40,23);sw:=readkey;
        TEXTCOLOR(7);
    end; #WhileA
    adri2:=2;
END; #if GENERA
GOTOXY(56,POSICION);COORD:=READKEY;
adri3:=0;
IF (COORD = 'S') OR (COORD = 's') THEN
BEGIN
    sw:='N';
    While (sw='N') or (sw='n') do
    Begin
        GOTOXY(5,23);WRITE(' ');
        GOTOXY(5,21);WRITE('Orden de ingreso para Coordenadas de los miembros');
        GOTOXY(5,22);WRITE('Coordenadas en X, Y, Z');
        TEXTCOLOR(14);
        GOTOXY(27,21);WRITE('C');
        GOTOXY(45,21);WRITE('M');
        GOTOXY(20,22);WRITE('X');
        GOTOXY(23,22);WRITE('Y');
        GOTOXY(26,22);WRITE('Z');
        GOTOXY(54,POSICION);WRITE('LP= , , ');
        TEXTCOLOR(7);
        COLUMNA:=57; LP1:=FunInEnt(columna,posicion);
        COLUMNA:=60; LP2:=FunInEnt(columna,posicion);
        COLUMNA:=63; LP3:=FunInEnt(columna,posicion);
        GOTOXY(5,20);WRITE(' ');
        GOTOXY(5,21);WRITE(' ');
        GOTOXY(5,22);WRITE(' ');
        GOTOXY(5,23);WRITE('Datos de Coordenadas Correctos S/N: ');
        GOTOXY(35,23);WRITE('S');gotoxy(38,23);write('N');

```

```

        GOTOXY(41,23);sw:=readkey;
        TEXTCOLOR(7);
    end; #While (sw='N')#
    adri3:=3;
END; #if COORDE#
GOTOXY(67,POSICION);CARGA:=READKEY;
adri4:=0;
IF ((CARGA = 'S') OR (CARGA = 's')) THEN
BEGIN
    sw:='N';
    While (sw='N') or (sw='n') do
    Begin
        GOTOXY(5,23);WRITE(' ');
        GOTOXY(5,20);WRITE('Orden de ingreso para Cargas');
        GOTOXY(5,21);WRITE('Cargas: Viva, Muerta, Sismica');
        TEXTCOLOR(14);
        GOTOXY(27,20);WRITE('C');
        GOTOXY(13,21);WRITE('V');
        GOTOXY(19,21);WRITE('M');
        GOTOXY(27,21);WRITE('S');
        GOTOXY(68,POSICION);WRITE('NSL= , , ');
        TEXTCOLOR(7);
        COLUMNA:=72; sws:='S';
        While sws='S' do
        Begin
            CV:=FunInEnt(columna,posicion);
            if cv > n1 then sws:='S';
            if cv <= n1 then sws:='N';
        end; #While sws#
        COLUMNA:=75; sws:='S';
        While sws='S' do
        Begin
            CMU:=FunInEnt(columna,posicion);
            if cmu > n1 then sws:='S';
            if cmu <= n1 then sws:='N';
        end; #While sws#
        COLUMNA:=78; sws:='S';
        While sws='S' do
        Begin
            CS:=FunInEnt(columna,posicion);
            if cs > n1 then sws:='S';
            if cs <= n1 then sws:='N';
        end;
        GOTOXY(5,20);WRITE(' ');
        GOTOXY(5,21);WRITE(' ');
        GOTOXY(5,23);WRITE(' ');
        GOTOXY(5,23);WRITE('Datos de Cargas Correctos S/N: ');
        TEXTCOLOR(14);
        GOTOXY(31,23);WRITE('S');gotoxy(33,23);write('N');
        GOTOXY(36,23);sw:=readkey;
        TEXTCOLOR(7);
    end; #While (sw='N')#
    adri4:=4;
END; #if CARGA#
ASSIGN(ARCHIVOSAP,NOMBRE);
APPEND(ARCHIVOSAP);
IF ((adri1=1) and (adri2=2) and (adri3=3) and (adri4=4)) THEN
BEGIN
    SIEMPRE;OPCION1;OPCION2;OPCION3;OPCION4;
    WRITELN(ARCHIVOSAP,CMV#N#,'C#ADR#N#CMU#C#LY#CAUX,'M#,'MAUX#',LP#AUX#MGAUX#P#AUX#N#AUX#P#

```

```

goto 20;
END;
IF ((adri1=1) and (adri2=2) and (adri3=3)) THEN
BEGIN
SIEMPRE;OPCION1;OPCION2;OPCION3;
WRITELN(ARCHIVOSAP,CMVAUX,' ',DELNAUX,' ',ALNAUX,' M=',MAUX,' G=',NMG AUX,' ',INIAUX,' ',
IAUX,' NSL=',CVAUX,' ',CMUAUX,' ',CSAUX);
LP=',LP1AUX,' ',LP2AUX,' ',LP3AUX);
goto 20;
END;
IF (adri1=1) and (adri2=2) and (adri4=4) THEN
BEGIN
SIEMPRE;OPCION1;OPCION2;OPCION4;
WRITELN(ARCHIVOSAP,CMVAUX,' ',DELNAUX,' ',ALNAUX,' M=',MAUX,' G=',NMG AUX,' ',INIAUX,' ',
IAUX,' NSL=',CVAUX,' ',CMUAUX,' ',CSAUX);
goto 20;
END;
IF ((adri1=1) and (adri3=3) and (adri4=4)) THEN
BEGIN
SIEMPRE;OPCION1;OPCION3;OPCION4;
WRITELN(ARCHIVOSAP,CMVAUX,' ',DELNAUX,' ',ALNAUX,' M=',MAUX,' G=',NMG AUX,' ',INIAUX,' ',
IAUX,' NSL=',CVAUX,' ',CMUAUX,' ',CSAUX,' LP=',LP1AUX,' ',LP2AUX,' ',LP3AUX);
goto 20;
END;
IF ((adri1=1) and (adri4=4)) THEN
BEGIN
SIEMPRE;OPCION1;OPCION4;
WRITELN(ARCHIVOSAP,CMVAUX,' ',DELNAUX,' ',ALNAUX,' M=',MAUX,' G=',NMG AUX,' ',INIAUX,' ',
IAUX,' NSL=',CVAUX,' ',CMUAUX,' ',CSAUX);
goto 20;
END;
IF ((adri1=1) and (adri3=3)) THEN
BEGIN
SIEMPRE;OPCION1;OPCION3;
WRITELN(ARCHIVOSAP,CMVAUX,' ',DELNAUX,' ',ALNAUX,' M=',MAUX,' G=',NMG AUX,' ',INIAUX,' ',
IAUX,' NSL=',CVAUX,' ',CMUAUX,' ',CSAUX,' LP=',LP1AUX,' ',LP2AUX,' ',LP3AUX);
goto 20;
END;
IF ((adri1=1) and (adri2=2)) THEN
BEGIN
SIEMPRE;OPCION1;OPCION2;
WRITELN(ARCHIVOSAP,CMVAUX,' ',DELNAUX,' ',ALNAUX,' M=',MAUX,' G=',NMG AUX,' ',INIAUX,' ',
IAUX,' NSL=',CVAUX,' ',CMUAUX,' ',CSAUX,' LP=',LP1AUX,' ',LP2AUX,' ',LP3AUX);
goto 20;
END;
IF ((adri2=2) and (adri3=3) and (adri4=4)) THEN
BEGIN
SIEMPRE;OPCION2;OPCION3;OPCION4;
WRITELN(ARCHIVOSAP,CMVAUX,' ',DELNAUX,' ',ALNAUX,' M=',MAUX,' G=',NMG AUX,' ',INIAUX,' ',INFAUX,' ',
IAUX,' NSL=',CVAUX,' ',CMUAUX,' ',CSAUX,' LP=',LP1AUX,' ',LP2AUX,' ',LP3AUX);
goto 20;
END;
IF ((adri2=2) and (adri3=3)) THEN
BEGIN
SIEMPRE;OPCION2;OPCION3;
WRITELN(ARCHIVOSAP,CMVAUX,' ',DELNAUX,' ',ALNAUX,' M=',MAUX,' G=',NMG AUX,' ',INIAUX,' ',INFAUX,' ',
IAUX,' NSL=',CVAUX,' ',CMUAUX,' ',CSAUX,' LP=',LP1AUX,' ',LP2AUX,' ',LP3AUX);
goto 20;
END;
IF ((adri2=2) and (adri4=4)) THEN
BEGIN
SIEMPRE;OPCION2;OPCION4;
WRITELN(ARCHIVOSAP,CMVAUX,' ',DELNAUX,' ',ALNAUX,' M=',MAUX,' G=',NMG AUX,' ',INIAUX,' ',INFAUX,' ',
IAUX,' NSL=',CVAUX,' ',CMUAUX,' ',CSAUX,' LP=',LP1AUX,' ',LP2AUX,' ',LP3AUX);
goto 20;
END;

```

```

        IMAUX,'        NSL=' ,CVAUX,' ,',CMUAUX,' ,',CSAUX);
    goto 20;
END;
IF ((adri3=3) and (adri4=4)) THEN
BEGIN
    SIEMPRE;OPCION3;OPCION4;
    WRITELN(ARCHIVOSAP,CMVAUX,' ',DELNAUX,' ',ALNAUX,' ',
            NSL=' ,CVAUX,' ,',CMUAUX,' ,',CSAUX,' ',LP=' ,LP1AUX,' ,',LP2AUX,' ,',LP3AUX);
    goto 20;
END;
IF (adri4=4) THEN
BEGIN
    SIEMPRE;OPCION4;
    WRITELN(ARCHIVOSAP,CMVAUX,' ',DELNAUX,' ',ALNAUX,' ',
            NSL=' ,CVAUX,' ,',CMUAUX,' ,',CSAUX);
    goto 20;
END;
IF (adri3=3) THEN
BEGIN
    SIEMPRE;OPCION3;
    WRITELN(ARCHIVOSAP,CMVAUX,' ',DELNAUX,' ',ALNAUX,' ',
            LP=' ,LP1AUX,' ,',LP2AUX,' ,',LP3AUX);
    goto 20;
END;
IF adri2=2 THEN
BEGIN
    SIEMPRE;OPCION2;
    WRITELN(ARCHIVOSAP,CMVAUX,' ',DELNAUX,' ',ALNAUX,' ',G=' ,NMGVAUX,' ,',INIAUX,' ,',IN);
    goto 20;
END;
IF (ADRI1=1) THEN
BEGIN
    SIEMPRE;OPCION1;
    WRITELN(ARCHIVOSAP,CMVAUX,' ',DELNAUX,' ',ALNAUX,' ',M=' ,MAUX);
    goto 20;
END;
IF (ADRI=0) THEN
BEGIN
    SIEMPRE;
    WRITELN(ARCHIVOSAP,CMVAUX,' ',DELNAUX,' ',ALNAUX);
    goto 20;
END;
20:   CLOSE(ARCHIVOSAP);
      CM1:=CM1+1;
      END; *DEL IF PRIMERO$
END; *del For$
CLRSCL;
MARCO;
GOTOXY(25,2);WRITE('CARACTERISTICAS DE LOS MIEMBROS');
POSICION:=0;
LECTURA(POSICION,NM,NL,CM,CM1,adri);
10: if posicion=21 then
    BEGIN
        ASSIGN(ARCHIVOSAP,NOMBRE);
        APPEND(ARCHIVOSAP);
        WRITELN(ARCHIVOSAP,'C FIN DE FRAME');
        CLOSE (ARCHIVOSAP);
        FER:=2;
    END;
3: END; *Del del Procedure$

```

PROGRAMA PRINCIPAL SEGUNDA PARTE

PROCEDURE MIEMBROS(POSICION:INTEGER;NM,NL,J:BYTE);

BEGIN

CLRSR;

CM:=0; MARCO;

GOTOXY(4,3);WRITE('NUMERO TOTAL DE MIEMBROS A INGRESAR :');

TEXTCOLOR(14);

COLUMNA:=42; POSICION:=3;

CM:=FunInEnt(columna, posicion);

M:=0; DELN:=0; ALN:=0; CMV:=0; CM1:=1;

TEXTCOLOR(7);

POSICION:=7;

GOTOXY(4,5);

TEXTCOLOR(14);

WRITELN('Miembro Del Al Tipo miembro Generación Coordenadas Estados Carga');

GOTOXY(4,6);

WRITELN(' Nudo Nudo S/N S/N S/N S/N ');

TEXTCOLOR(7);

ASSIGN(ARCHIVOSAP, NUMBRE);

APPEND(ARCHIVOSAP);

WRITELN(ARCHIVOSAP, 'c NUMERO DE MIEMBROS ', CM);

CLOSE(ARCHIVOSAP);

LECTURA(POSICION, NM, NL, CM, CM1, adri);

END; *De la Procedure Miembros

PROCEDURE NL_CORRECTO(POSICION:INTEGER;NM,NL,J:BYTE);

BEGIN

GOTOXY(5,23);WRITE(' ');

GOTOXY(5,23);WRITE('Sus Datos están Correctos S/N: ');

TEXTCOLOR(14);

GOTOXY(31,23);WRITE('S');

GOTOXY(33,23);WRITE('N');

TEXTCOLOR(7);

GOTOXY(36,23);DATOCO:=READKEY;

IF (DATOCO='N') OR (DATOCO='n') THEN

BEGIN

GOTOXY(5,23);WRITE(' ');

GOTOXY(5,23);WRITE(' ');

GOTOXY(5,23);WRITE('Corregir: MX, WY, WZ: ');

TEXTCOLOR(14);

GOTOXY(16,23);WRITE('X');

GOTOXY(20,23);WRITE('Y');

GOTOXY(24,23);WRITE('Z');

TEXTCOLOR(7);

GOTOXY(27,23);CORREGIR:=READKEY;

CASE CORREGIR OF

'X', 'x': BEGIN

COLUMNA:=39; MX:=FunInDec(columna, posicion)

END;

'Y', 'y': BEGIN

COLUMNA:=46; WY:=FunInDec(columna, posicion);

END;

'Z', 'z': BEGIN

COLUMNA:=53; WZ:=FunInDec(columna, posicion);

END;

END; *End_Case

NL_CORRECTO(POSICION, NM, NL, J);

END *End_if Dato Correcto

ELSE

BEGIN

```

ASSIGN(ARCHIVOSAP,NOMBRE);
APPEND(ARCHIVOSAP);
JAUX:=' ':LON:=0;
STR(1,JAUX);
LON:=LENGTH(JAUX);
WHILE LON<=3 00
BEGIN
    JAUX:=JAUX+' ';
    LON:=LON+1;
END;
STR(MX;1;2,NXAJUX);
STR(MY;1;2,MVAUX);
STR(MZ;1;2,MZAJUX);
WRITELN(ARCHIVOSAP,JAUX, ' MG=',MVAUX, ', ',MVAUX, ', ',MZAJUX);
CLOSE(ARCHIVOSAP);
JAUX:=' ':LON:=0;
STR(1,JAUX);
LON:=LENGTH(JAUX);
WHILE LON<=3 00
BEGIN
    JAUX:=JAUX+' ';
    LON:=LON+1;
END;
PROCEDURE NH_CORRECTOZ(POSITION:INTEGER;NM,J:BYTE);
BEGIN
    GOTXY(5,23);WRITE('
GOTXY(5,23);WRITE('Sus Datos estan Correctos S/N: ');
TEXTCLR(14);
GOTXY(31,23);WRITE('S');
GOTXY(33,23);WRITE('N');
TEXTCLR(7);
GOTXY(36,23);DATOC:=READY;
IF (DATOC='N') OR (DATOC='n') THEN
BEGIN
    GOTXY(5,23);WRITE('
GOTXY(5,23);WRITE('
GOTXY(5,23);WRITE('Corregir Area o Inicia: ');
TEXTCLR(14);
GOTXY(15,23);WRITE('R');
GOTXY(22,23);WRITE('N');
TEXTCLR(7);
GOTXY(30,23);CORREG:=READY;
CASE CORREG OF
    'R','r': BEGIN
        COLUMN:=1; A:=FunFindDec(columna,position);
        END;
    'N','n': BEGIN
        COLUMN:=19; J:=FunIndec(columna,position);
        END;
    END;
    aEnd_Case;
    NH_CORRECTOZ(POSITION,NM,J);
END aEnd_If Data Correcta
ELSE
BEGIN
    ASSIGN(ARCHIVOSAP,NOMBRE);
    APPEND(ARCHIVOSAP);
    JAUX:=' ':LON:=0;
    STR(1,JAUX);
    LON:=LENGTH(JAUX);
    WHILE LON<=3 00
    BEGIN
        JAUX:=JAUX+' ';
        LON:=LON+1;
    END;
    JAUX:=' ':LON:=0;
    LON:=LON+1;
END;

```

```

STR(A:1:2,AAUX);
LON:=LENGTH(AAUX);
WHILE LON<=6 DO
BEGIN
  AAUX:=AAUX+' ';
  lon:=lon+1;
END;
IAUX:=' ':LON:=0;
STR(1:1:2,IAUX);
LON:=LENGTH(IAUX);
WHILE LON<=6 DO
BEGIN
  IAUX:=IAUX+' ';
  lon:=lon+1;
END;
WRITELN(ARCHIVOSAP,JAUX,' A=',AAUX,' I=',IAUX);
CLOSE(ARCHIVOSAP);
J:=J+1;
END; *End IF DATOCOR
END; *End_Procedure NM_CORRECTO2A

```

```

PROCEDURE NM_CORRECTO1(POSICION:INTEGER;NM,J:BYTE);
BEGIN
  GOTOXY(5,23);WRITE(' ');
  GOTOXY(5,23);WRITE('Sus Datos están Correctos S/N: ');
  TEXTCOLOR(14);
  GOTOXY(31,23);WRITE('S');
  GOTOXY(33,23);WRITE('N');
  TEXTCOLOR(7);
  GOTOXY(36,23);DATOCO:=READKEY;
  IF (DATOCO='N') OR (DATOCO='n') THEN
  BEGIN
    GOTOXY(5,23);WRITE(' ');
    GOTOXY(5,23);WRITE(' ');
    GOTOXY(5,23);WRITE('Corregir: Base1, Base2, Base3, Altura1, Altura2, Altura3: ');
    TEXTCOLOR(14);
    GOTOXY(15,23);WRITE('R');
    GOTOXY(24,23);WRITE('S');
    GOTOXY(32,23);WRITE('E');
    GOTOXY(36,23);WRITE('A');
    GOTOXY(47,23);WRITE('T');
    GOTOXY(57,23);WRITE('U');
    TEXTCOLOR(7);
    GOTOXY(63,23);CORREGIR:=READKEY;
    CASE CORREGIR OF
      'R','r': BEGIN
        COLUMNA:=11; R1:=FunInDec(columna,posicion);
        END;
      'S','s': BEGIN
        COLUMNA:=19; R2:=FunInDec(columna,posicion);
        END;
      'E','e': BEGIN
        COLUMNA:=27; R3:=FunInDec(columna,posicion);
        END;
      'A','a': BEGIN
        COLUMNA:=38; D1:=FunInDec(columna,posicion);
        END;
      'T','t': BEGIN
        COLUMNA:=48; D2:=FunInDec(columna,posicion);
        END;
      'U','u': BEGIN

```

COLUMNA:=59; 03:=FunIndec(columna,position);

END;

END; #del Case4

MM_CORRECTO((POSITION,MM,1));

END #end If Dato Correcto4

ELSE

BEGIN

ASSIGN(ARCHIVO\$AP,NOMBRE);

APPEND(ARCHIVO\$AP);

JAUX:=''; LON:=0;

STR(1,JAUX);

LON:=LENGTH(JAUX);

WHILE LON<=3 DO

BEGIN

JAUX:=JAUX+' ';

LON:=LON+1;

END;

BAUX:=''; LON:=0;

STR(1,1:2,BAUX);

LON:=LENGTH(BAUX);

WHILE LON<=6 DO

BEGIN

BAUX:=BAUX+' ';

LON:=LON+1;

END;

BZ AUX:=''; LON:=0;

STR(1,2:2,BZ AUX);

LON:=LENGTH(BZ AUX);

WHILE LON<=6 DO

BEGIN

BZ AUX:=BZ AUX+' ';

LON:=LON+1;

END;

B3 AUX:=''; LON:=0;

STR(1,3:1:2,B3 AUX);

LON:=LENGTH(B3 AUX);

WHILE LON<=6 DO

BEGIN

B3 AUX:=B3 AUX+' ';

LON:=LON+1;

END;

D1 AUX:=''; LON:=0;

STR(1,1:1:2,D1 AUX);

LON:=LENGTH(D1 AUX);

WHILE LON<=6 DO

BEGIN

D1 AUX:=D1 AUX+' ';

LON:=LON+1;

END;

D2 AUX:=''; LON:=0;

STR(1,2:1:2,D2 AUX);

LON:=LENGTH(D2 AUX);

WHILE LON<=6 DO

BEGIN

D2 AUX:=D2 AUX+' ';

LON:=LON+1;

END;

D3 AUX:=''; LON:=0;

STR(1,3:1:2,D3 AUX);

LON:=LENGTH(D3 AUX);

```

WHILE LON<=6 00
BEGIN
  D3AUX:=D3AUX+' ';
  LON:=LON+1;
END;
WRITE(NV(ARCHIVOSAP, JANY, ' R1=', R1AUX, ' R2=', R2AUX, ' R3=', R3AUX,
      ' R1=', D1AUX, ' D2=', D2AUX, ' D3=', D3AUX));
CLOSE(ARCHIVOSAP);
J:=J+1;
END;
END;
END;

```

```

END;
END;
END;

```

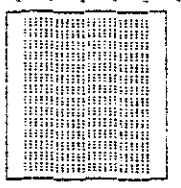
PROCEDURE AREA_INERSIA(POSICION:INTEGER;NM,J:BYTE);CALLA A NM_CORRECTOZA

BEGIN

```

  GOT0XY(3,2);WRITE(' ');
  GOT0XY(3,3);WRITE(' ');
  GOT0XY(3,4);WRITE(' ');
  GOT0XY(3,5);WRITE(' ');
  GOT0XY(3,6);WRITE(' ');
  GOT0XY(3,7);WRITE(' ');
  GOT0XY(3,8);WRITE(' ');
  GOT0XY(20,9);WRITE(' ');
  GOT0XY(3,10);WRITE(' ');
  GOT0XY(5,11);WRITE(' ');
  TEXTCOLOR(14);
  GOT0XY(25,3);WRITE(' ');
  GOT0XY(25,4);WRITE(' ');
  GOT0XY(25,5);WRITE(' ');
  GOT0XY(25,6);WRITE(' ');
  GOT0XY(25,7);WRITE(' ');
  GOT0XY(25,8);WRITE(' ');
  GOT0XY(13,10);WRITE(' DESCRIPCION ');
  TEXTCOLOR(7);
  GOT0XY(5,11);WRITE(' No Area Inercia ');
  COLUMNA:=11; A:=FunIndec(columna, posicion);
  COLUMNA:=19; I:=FunIndec(columna, posicion);
  NM_CORRECTOZ(POSICION, NM, J);
END;

```



END;

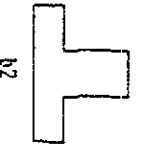
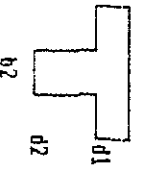
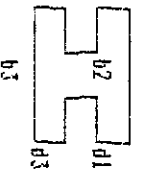
PROCEDURE BASE_ALTURA (POSICION:INTEGER;NM,J:BYTE);

BEGIN

```

  GOT0XY(3,10);WRITE(' ');
  TEXTCOLOR(14);
  GOT0XY(3,2);WRITE(' b1 ');
  GOT0XY(3,3);WRITE(' d1 ');
  GOT0XY(3,4);WRITE(' ');
  GOT0XY(3,5);WRITE(' ');
  GOT0XY(3,6);WRITE(' ');
  GOT0XY(3,7);WRITE(' ');
  GOT0XY(3,8);WRITE(' ');
  GOT0XY(3,9);WRITE(' ');
  GOT0XY(30,9);WRITE(' DESCRIPCION ');
  GOT0XY(5, 10);WRITE(' Base1 Base2 Base3 Altural Altura2 Altura3 ');
  TEXTCOLOR(7);
  GOT0XY(5,11);WRITE(' No b1 b2 b3 d1 d2 ');
  COLUMNA:=11; B1:=FunIndec(columna, posicion);
  COLUMNA:=19; B2:=FunIndec(columna, posicion);
  COLUMNA:=27; B3:=FunIndec(columna, posicion);
  COLUMNA:=38; D1:=FunIndec(columna, posicion);
  COLUMNA:=48; D2:=FunIndec(columna, posicion);
  COLUMNA:=58; D3:=FunIndec(columna, posicion);

```



```

NM_CORRECTO(POSICION,NM,J);
END;

PROCEDURE ESTADOS_CARGA(POSICION:INTEGER;NL,J:BYTE);
LABEL 1;
BEGIN
  FOR POSICION:= POSICION TO 19 DO
    BEGIN
      IF J>NL THEN
        BEGIN
          POSICION:=19;
          CLRSCR;
          J:=1;POSICION:=8;
          MIEMBROS(POSICION,NM,NL,J);
          GOTO 1;
        END
      ELSE
        BEGIN
          TEXTCOLOR(14);
          GOTOXY(30,POSICION);WRITE(J);
          TEXTCOLOR(7);
          GOTOXY(35,POSICION);WRITE('WG=      ,      ,      ');
          COLUMNA:=39; WX:=FunInDec(columna,posicion);
          COLUMNA:=46; WY:=FunInDec(columna,posicion);
          COLUMNA:=53; WZ:=FunInDec(columna,posicion);
          NL_CORRECTO(POSICION,NM,NL,J);
          END; *DEL IF#
          J:=J+1;
        END; *DEL FOR#
      CLRSCR;
      MARCO;
      POSICION:=9;
      TEXTCOLOR(14);
      GOTOXY(20,3);WRITE('CARACTERISTICAS DE LOS MIEMBROS');
      GOTOXY(25,4);WRITE('Número de Cargas Distribuidas a ingresar : ',NL);
      GOTOXY(35,6);WRITE('  CARGAS DISTRIBUIDAS');
      GOTOXY(35,7);WRITE('    X    Y    Z');
      TEXTCOLOR(7);
      GOTOXY(5,23);WRITE('    = Cargas Distribuidas');
      TEXTCOLOR(14);
      GOTOXY(5,23);WRITE('WG');
      TEXTCOLOR(7);
      ESTADOS_CARGA(POSICION,NL,J);
1: END; *DEL PROCEDURE#

```

```

PROCEDURE TIPO_MIEMBRO(POSICION:INTEGER;NM,NL,J:BYTE);
LABEL 2;
BEGIN
  FOR POSICION:= POSICION TO 20 DO
    BEGIN
      IF J>NM THEN
        BEGIN
          POSICION:=20;
          CLRSCR;
          MARCO;
          TEXTCOLOR(14);
          GOTOXY(25,3);WRITE('CARACTERISTICAS DE LOS MIEMBROS');
          GOTOXY(20,4);WRITE('Número de Cargas Distribuidas a ingresar : ',NL);
          GOTOXY(35,6);WRITE('  CARGAS DISTRIBUIDAS');
          GOTOXY(35,7);WRITE('    X    Y    Z');

```

```

GOTOXY(5,23);WRITE(' = Cargas Distribuidas');
TEXTCOLOR(14);
GOTOXY(5,23);WRITE('WG');
TEXTCOLOR(7);
POSICION:=7;
J:=1;
ESTADOS_CARGA(POSICION,NL,J);
GOTO 2;
END
ELSE
BEGIN
TEXTCOLOR(14);
GOTOXY(5,POSICION);WRITE(J);
TEXTCOLOR(7);
GOTOXY(5,23); WRITE('Sus Datos estan deacuerdo a Sección/Propiedad: ');
TEXTCOLOR(14);
GOTOXY(33,23); WRITE('S'); GOTOXY(41,23); WRITE('P');
TEXTCOLOR(7);
GOTOXY(52,23); SECPRO:=READKEY;
IF (SECPRO='S') OR (SECPRO='s') THEN
BASE_ALTURA (POSICION,NM,J);
IF (SECPRO='P') OR (SECPRO='p') THEN
AREA_INERBIA (POSICION,NM,J);
END;#Del IfS
J:=J+1;
END; #Del ForS
CLRSR;
MARCO;
GOTOXY(3,10);WRITE('NO Propiedades'); POSICION:=12;
TIPO_MIEMBRO(POSICION,NM,NL,J);
2: END; #Del Procedura

```

*PROGRAMA PRINCIPAL PRIMERA PARTEA

```

BEGIN
IF FER=0 THEN
BEGIN
CLRSR;
MARCO;
ADRI:=0;
POSICION:=12; NM:=1; J:=1;
TEXTCOLOR(14);
GOTOXY(25,3);WRITE('CARACTERISTICAS DE LOS MIEMBROS');
TEXTCOLOR(7);
GOTOXY(20,5);WRITE('Ingrese la cantidad de tipos de miembros :');
TEXTCOLOR(14);
COLUMNA:=64; POSICION:=5;
NM:=FunInEnt(columna,posicion);
TEXTCOLOR(7);
GOTOXY(20,7);WRITE('Ingrese el Módulo de Elasticidad :');
TEXTCOLOR(14);
COLUMNA:=55; POSICION:=7;
E:=FunInEnt1(columna,posicion);
TEXTCOLOR(7);
GOTOXY(20,9);WRITE('Ingrese el # Cargas Distribuidas : ');
TEXTCOLOR(14);
COLUMNA:=55; POSICION:=9;
NL:=FunInEnt(columna,posicion);
POSICION:=12;
GOTOXY(5,11);WRITE('NO Propiedades');
TEXTCOLOR(7);
ASSIGN(ARCHIVOSAP,NOMBRE);

```

```

APPEND(ARCHIVOSAP);
NMAUX:= ' ';LON:=0;
STR(NM,NMAUX);
LON:=LENGTH(NMAUX);
WHILE LON<=3 DO
BEGIN
    NMAUX:=NMAUX+' ';
    lon:=lon+1;
END;
NLAUX:= ' ';LON:=0;
STR(NL,NLAUX);
LON:=LENGTH(NLAUX);
WHILE LON<=3 DO
BEGIN
    NLAUX:=NLAUX+' ';
    lon:=lon+1;
END;
WRITELN(ARCHIVOSAP,'FRAME : CARACTERISTICAS DE LOS MIEMBROS - CARGAS DISTRIBUIDAS E INCIDENCIAS');
WRITELN(ARCHIVOSAP,'NM=' ,NM,'      NL=' ,NL,'      E=' ,E:1);
CLOSE(ARCHIVOSAP);
TIPO_MIEMBRO(POSICION,NM,NL,J);
END)
ELSE
BEGIN
    IF FER=1 THEN
    BEGIN
        ASSIGN(ARCHIVOSAP,NOMBRE);
        RESET(ARCHIVOSAP);
        CODE:=0;CONTA:=0;
        REPEAT
            READLN(ARCHIVOSAP,NOMEST);
            CONTA:=CONTA+1;
            LUGAR:=POS('c NUMERO ',NOMEST);
        UNTIL LUGAR<>0;
        CLOSE(ARCHIVOSAP);
        VAL(COPY(NOMEST,LUGAR+24,4),CH,CODE);
        ASSIGN(ARCHIVOSAP,NOMBRE);
        RESET(ARCHIVOSAP);
        CODE:=0;
        WHILE NOT EOF(ARCHIVOSAP) DO
        BEGIN
            READLN(ARCHIVOSAP,NOMEST);
            CODE:=CODE+1;
        END;
        CLOSE(ARCHIVOSAP);
        MITAD:=CODE-CONTA;
        ASSIGN(ARCHIVOSAP,NOMBRE);
        RESET(ARCHIVOSAP);
        WHILE NOT EOF(ARCHIVOSAP) DO
        BEGIN
            READLN(ARCHIVOSAP,NOMEST);
            LUGAR:=POS('NM=',NOMEST);
        END;
        CLOSE(ARCHIVOSAP);
        IF LUGAR<>0 THEN VAL(COPY(NOMEST,LUGAR+3,2),NM,CODE);
        ASSIGN(ARCHIVOSAP,NOMBRE);
        RESET(ARCHIVOSAP);
        WHILE NOT EOF(ARCHIVOSAP) DO
        BEGIN
            READLN(ARCHIVOSAP,NOMEST);
            LUGAR:=POS('NL=',NOMEST);

```

```

END;
CLOSE(ARCHIVOSAP);
IF LUGAR<>0 THEN VAL(COPY(NOMEST,LUGAR+3,2),NL,CODE);
CLRSCR;
MARCO;
TEXTCOLOR(14);
GOTOXY(25,2);WRITE('CARACTERISTICAS DE LOS MIEMBROS');
GOTOXY(5,3);WRITE('Número de miembros a ingresar: ',CM);
GOTOXY(5,4);WRITE('Número de miembros ingresados: ',MITAD);
GOTOXY(3,5);WRITE(' NR Del AI Tipo Generación Coordenadas Estados Carga');
GOTOXY(3,6);WRITE(' Nudo Nudo Miembro ');
TEXTCOLOR(7);
WRITELN;
ASSIGN(ARCHIVOSAP,NOMBRE);
RESET(ARCHIVOSAP);
FOR CODE:=1 TO CUNTA DO
  READLN(ARCHIVOSAP,NOMEST);
  REPEAT
    READLN(ARCHIVOSAP,NOMEST);
    WRITELN('|| ',NOMEST);
  UNTIL EOF(ARCHIVOSAP);
  CLOSE (ARCHIVOSAP);
  WRITELN(WRITELN('|| PRESIONE <ENTER> PARA CONTINUAR EL INGRESO'));
  READLN;
  POSICION:=9;
  CMI:=1;
  CMI:=CMI+MITAD;
  CLRSCR;
  MARCO;
  ADRI:=0;
  POSICION:=7;
  TEXTCOLOR(14);
  GOTOXY(25,2);WRITE('CARACTERISTICAS DE LOS MIEMBROS');
  GOTOXY(5,3);WRITE('Número de miembros a ingresar: ',CM);
  GOTOXY(5,4);WRITE('Número de miembros ingresados: ',MITAD);
  TEXTCOLOR(7);
  LECTURA(POSICION,NH,NL,(M,CMI,adri);
  END; #DEL IF FER=14
  END; #DEL ELSEA
END; #Del Programa Principal del Framed

```

```

PROCEDURE QUAD;
VAR
  JQ,NMAT,LON:BYTE;
  QEL,QU,QW:INTEGER;
  QX,QY,QZ,TH:REAL;
  QXAUX,QYAUX,QZAUX,THAUX:STRING#6A;
  QELAUX,QUAUX,QWAUX:STRING#3A;

```

```
PROCEDURE PLACAS(POSICION:INTEGER;NMAT,JQ:BYTE); FORWARD;
```

```
PROCEDURE CORRE_FLACAS(POSICION:INTEGER;NMAT,JQ:BYTE);
```

```
Label 2;
```

```
BEGIN
```

```

  GOTOXY(5,21);WRITE(' ');
  GOTOXY(5,22);WRITE(' ');
  GOTOXY(5,23);WRITE(' ');
  GOTOXY(5,23);WRITE('Sus Datos están Correctos S/N: |||');
  TEXTCOLOR(14);
  GOTOXY(31,23);WRITE('S');
  GOTOXY(33,23);WRITE('N');

```

```

TEXTCOLOR(7);
GOTOXY(36,23);DATOCO:=READKEY;
IF (DATOCO='N') OR (DATOCO='n') THEN
BEGIN
    GOTOXY(5,23);WRITE(' ');
    GOTOXY(5,22);WRITE('Corregir: Factor gravitacional: X, Y, Z; Módulos de: Elasticidad ó Poisson');
    GOTOXY(5,23);WRITE('Peso/Volumen; Espesor: ');
    TEXTCOLOR(14);
    GOTOXY(37,22);WRITE('X');GOTOXY(40,22);WRITE('Y');GOTOXY(43,22);WRITE('Z');GOTOXY(50,22);WRITE('E')
    GOTOXY(72,22);WRITE('P');
    GOTOXY(10,23);WRITE('V');GOTOXY(20,23);WRITE('S');
    TEXTCOLOR(7);
    GOTOXY(20,23);CORREGIR:=READKEY;
    CASE CORREGIR OF
    'X','x' : BEGIN
        COLUMNA:=5;
        QX:=FunFinDec(columna,posicion);
        if Qx=maxint then
        BEGIN
            FER:=1;
            GOTO 2;
        END;
    END;
    'Y','y' : BEGIN
        COLUMNA:=13;
        QY:=FunInDec(columna,posicion);
    END;
    'Z','z' : BEGIN
        COLUMNA:=21;
        QZ:=FunInDec(columna,posicion);
    END;
    'E','e' : BEGIN
        COLUMNA:=32;
        QEI:=FunInEnt(columna,posicion);
    END;
    'P','p' : BEGIN
        COLUMNA:=44;
        QU:=FunInEnt(columna,posicion);
    END;
    'V','v' : BEGIN
        COLUMNA:=57;
        QV:=FunInEnt(columna,posicion);
    END;
    'S','s' : BEGIN
        COLUMNA:=64;
        TH:=FunInDec(columna,posicion);
    END;
    END; #End_Case#
    CORRE_PLACAS(POSICION,MMAT,IR);
END #End_If_Datoco#
ELSE
BEGIN
    ASSIGN(ARCHIVOSAP,NOMBRE);
    APPEND(ARCHIVOSAP);
    QXAUX:=' ';LON:=0;
    STR(QX:1:2,QXAUX);
    LON:=LENGTH(QXAUX);
    WHILE LON<=6 DO
    BEGIN
        QXAUX:=QXAUX+' ';
        lon:=lon+1;
    END;

```

```

END;
QY AUX:= ' ';LON:=0;
STR(QY:1:2,QY AUX);
LON:=LENGTH(QY AUX);
WHILE LON<=6 DO
BEGIN
  QY AUX:=QY AUX+' ';
  lon:=lon+1;
END;
QZ AUX:= ' ';LON:=0;
STR(QZ:1:2,QZ AUX);
LON:=LENGTH(QZ AUX);
WHILE LON<=6 DO
BEGIN
  QZ AUX:=QZ AUX+' ';
  lon:=lon+1;
END;
QEL AUX:= ' ';LON:=0;
STR(QEL,QEL AUX);
LON:=LENGTH(QEL AUX);
WHILE LON<=3 DO
BEGIN
  QEL AUX:=QEL AUX+' ';
  lon:=lon+1;
END;
QU AUX:= ' ';LON:=0;
STR(QU,QU AUX);
LON:=LENGTH(QU AUX);
WHILE LON<=3 DO
BEGIN
  QU AUX:=QU AUX+' ';
  lon:=lon+1;
END;
QW AUX:= ' ';LON:=0;
STR(QW,QW AUX);
LON:=LENGTH(QW AUX);
WHILE LON<=3 DO
BEGIN
  QW AUX:=QW AUX+' ';
  lon:=lon+1;
END;
TH AUX:= ' ';LON:=0;
STR(TH:1:2,TH AUX);
LON:=LENGTH(TH AUX);
WHILE LON<=6 DO
BEGIN
  TH AUX:=TH AUX+' ';
  lon:=lon+1;
END;
WRITELN(ARCHIVOSAP,' X=',QX AUX,' Y=',QY AUX,' Z=',QZ AUX,' E=',QEL AUX,' U=',QU AUX,' W=',QW AUX,' T
CLOSE(ARCHIVOSAP);
JQ:=JQ+1;
END;#End_else_datoca
2: END; #End_Procedurea

```

```

PROCEDURE PLACAS(POSICION:INTEGER;MMAT,JQ:BYTE);
label 20,3;
BEGIN
  FOR POSICION:=POSICION+1 TO 20 DO
  BEGIN
    IF JQ>MMAT THEN

```

```

BEGIN
    POSICION:=21;
    goto 20;
END
ELSE
BEGIN
    GOTOXY(5,22);WRITE(' ');
    GOTOXY(5,23);WRITE(' ');
    GOTOXY(5,22);WRITE('Presione Barra Espaciadora en Factor Gravitacional X');
    GOTOXY(5,23);WRITE('y regresa al Menú Principal ....');
    TEXTCOLOR(14);
    GOTOXY(14,22);WRITE('BARRA ESPACIADORA');
    GOTOXY(18,23);WRITE('MENU PRINCIPAL ');
    TEXTCOLOR(7);
    COLUMNA:=5;QX:=FunInDec(columna,posicion);
    if Qx=maxint then
    BEGIN
        FER:=1;
        GOTO 3;
    END;
    COLUMNA:=13;QY:=FunInDec(columna,posicion);
    COLUMNA:=21;QZ:=FunInDec(columna,posicion);
    COLUMNA:=32;QEL:=FunInEnt(columna,posicion);
    COLUMNA:=46;QU:=FunInEnt(columna,posicion);
    COLUMNA:=57;QW:=FunInEnt(columna,posicion);
    COLUMNA:=66;QH:=FunInDec(columna,posicion);
    CORRE_PLACAS(POSICION,NMAT,JQ);
    JQ:=JQ+1;
END;#End_JQB
END;#End_ForA
CLASCR;
MARC0;
TEXTCOLOR(14);
GOTOXY(18,4);WRITE('B. C A S C A R A S, P L A C A S Y M E M B R A N A S');
GOTOXY(5,6);WRITE('Factor Gravitacional      Módulo      Módulo      Peso/      Espesor');
GOTOXY(5,7);WRITE(' X      Y      Z      Elasticidad      Poisson      Volumen');
TEXTCOLOR(7);
POSICION:=8;
PLACAS(POSICION,NMAT,JQ);
20: if posicion = 71 then
    BEGIN
        ASSIGN(ARCHIVOSAP,NOMBRE);
        APPEND(ARCHIVOSAP);
        WRITELN(ARCHIVOSAP,'C FIN DE QUAD');
        CLOSE (ARCHIVOSAP);
        FER:=2;
    END;
3: END;#End_ProcedureA

(*PROGRAMA PRINCIPAL*)
BEGIN
    IF FER=0 THEN
    BEGIN
        CLASCR;
        MARC0;
        TEXTCOLOR(14);
        GOTOXY(18,2);WRITE('B. C A S C A R A S, P L A C A S Y M E M B R A N A S');
        GOTOXY(6,4);WRITE('Ingrese la cantidad de tipos de materiales a declarar:');
        TEXTCOLOR(7);
        GOTOXY(5,22);WRITE('Ingrese 0 si no desea ingresar datos');
        TEXTCOLOR(14);GOTOXY(13,22);WRITE('0');TEXTCOLOR(7);
    
```

```

NMAT:=0;
COLUMNA:=62;POSICION:=4;
NMAT:=FunInEnt(columna, posicion);
TEXTCOLOR(14);
GOTOXY(4,6);WRITE('Factor Gravitacional      Módulo      Módulo      Peso/      Espesor');
GOTOXY(4,7);WRITE(' X          Y          Z      Elasticidad      Poisson      Volumen');
TEXTCOLOR(7);
JO:=1;
POSICION:=8;
ASSIGN(ARCHIVOSAP,NOMBRE);
APPEND(ARCHIVOSAP);
WRITELN(ARCHIVOSAP,'QUAD : R. CASCARAS, PLACAS Y MEMBRANAS ',NMAT);
CLOSE(ARCHIVOSAP);
PLACAS(POSICION,NMAT,JO);
END
ELSE
BEGIN
  IF FER=1 THEN
  BEGIN
    ASSIGN(ARCHIVOSAP,NOMBRE);
    RESET(ARCHIVOSAP);
    CODE:=0;CONTA:=0;
    REPEAT
      READLN(ARCHIVOSAP,NOMEST);
      CONTA:=CONTA+1;
      LUGAR:=POS('QUAD ',NOMEST);
    UNTIL LUGAR<>0;
    CLOSE(ARCHIVOSAP);
    VAL(COPY(NOMEST,LUGAR+57,3),NMAT,CODE);
    ASSIGN(ARCHIVOSAP,NOMBRE);
    RESET(ARCHIVOSAP);
    CODE:=0;
    WHILE NOT EOF (ARCHIVOSAP) DO
    BEGIN
      READLN(ARCHIVOSAP,NOMEST);
      CODE:=CODE+1;
    END;
    CLOSE(ARCHIVOSAP);
    MITAD:=CODE-CONTA;
    CLRSCR;
    MARCO;
    TEXTCOLOR(14);
    GOTOXY(10,2);WRITE('B. C A S C A R A S, P L A C A S Y M E M B R A N A S');
    GOTOXY(6,4);WRITE('Número de tipos de materiales a ingresar : ',NMAT);
    GOTOXY(6,5);WRITE('Número de tipos de materiales declarados : ',MITAD);
    GOTOXY(3,6);WRITE('      Factor Gravitacional      Módulo      Módulo      Peso/      Espesor');
    GOTOXY(3,7);WRITE(' X          Y          Z      Elasti. Poiss. Volú. ');
    TEXTCOLOR(7);
    WRITELN;
    ASSIGN(ARCHIVOSAP,NOMBRE);
    RESET(ARCHIVOSAP);
    FOR CODE:=1 TO CONTA DO
      READLN(ARCHIVOSAP);
    REPEAT
      READLN(ARCHIVOSAP,NOMEST);
      WRITELN('|| ',NOMEST);
    UNTIL EOF(ARCHIVOSAP);
    CLOSE(ARCHIVOSAP);
    WRITELN;WRITELN('|| PRESIONE <ENTER> PARA CONTINUAR EL INGRESO');
    READLN;
    CLRSCR;

```

```

MARCO;
TEXTCOLOR(14);
GOTOXY(10,2);WRITE('B. C A S C A R A S, P L A C A S Y M E M B R A N A S');
GOTOXY(6,4);WRITE('Número de tipos de materiales a ingresar : ',NMAT);
GOTOXY(6,5);WRITE('Número de tipos de materiales declarados : ',MITAD);
GOTOXY(4,6);WRITE('Factor Gravitacional      Módulo      Módulo      Peso/      Espesor');
GOTOXY(4,7);WRITE(' X      Y      Z      Elasticidad      Poisson      Volúmen');
TEXTCOLOR(7);
POSICION:=8;
JQ:=1;
JQ:=JQ+MITAD;
PLACAS(POSICION,NMAT,JQ);
END; #DEL IF FER=12
END; #DEL ELSE2
END; #del procedure QUAD2

END. #FIN DE LA UNIDAD INGRESO22

```

NUMERO DEL PROGRAMA CLAVE

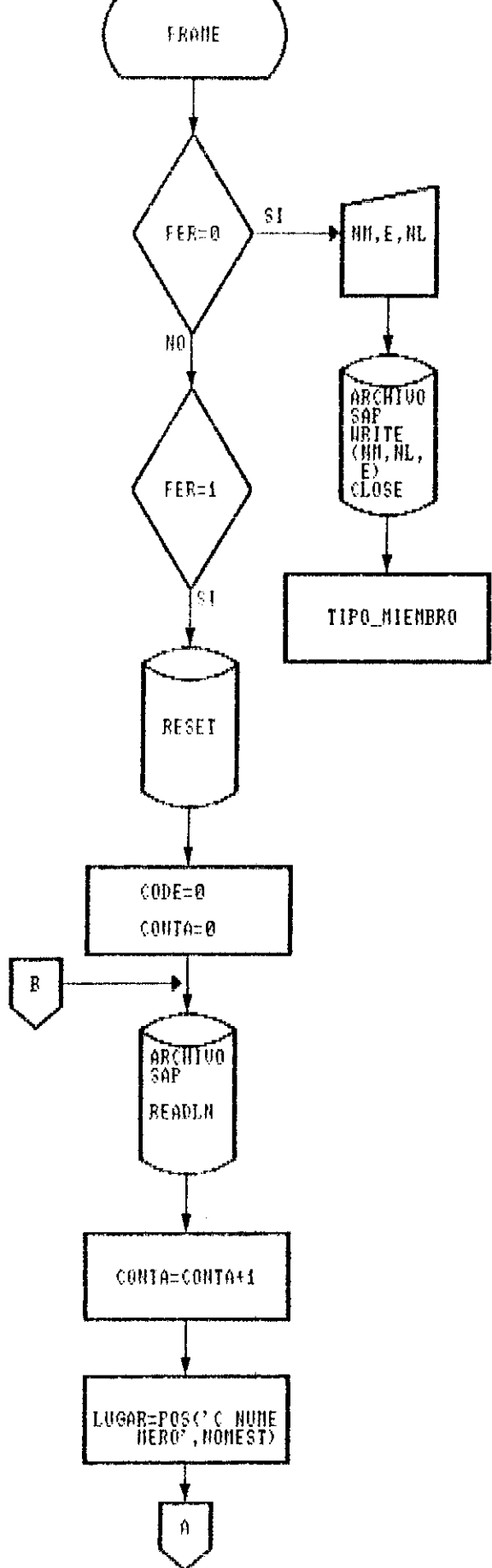
| NUMERO | TIPO | DESCRIPCION |
|---------------|-------------|------------------------------|
| 01 | Integer | Numero de miembros |
| 02 | Byte | Cantidad de miembros |
| 03 | Byte | Condicion |
| 04 | Byte | Numero de miembros |
| 05 | Integer | Dado Inicial |
| 06 | Integer | Dado Final |
| 07 | Byte | Estado de carga |
| 08 | Byte | Numero de miembros generados |
| 09 | Byte | Incremento de estado Inicial |
| 10 | Byte | Incremento de estado final |
| 11 | Byte | Incremento de miembros |
| 12 | Byte | Coordenada en X |
| 13 | Byte | Coordenada en Y |
| 14 | Byte | Coordenada en Z |
| 05 | Integer | Carga viva |
| 06 | Integer | Carga muerta |
| 07 | Integer | Carga Química |
| 08 | Byte | Tipo de miembro |

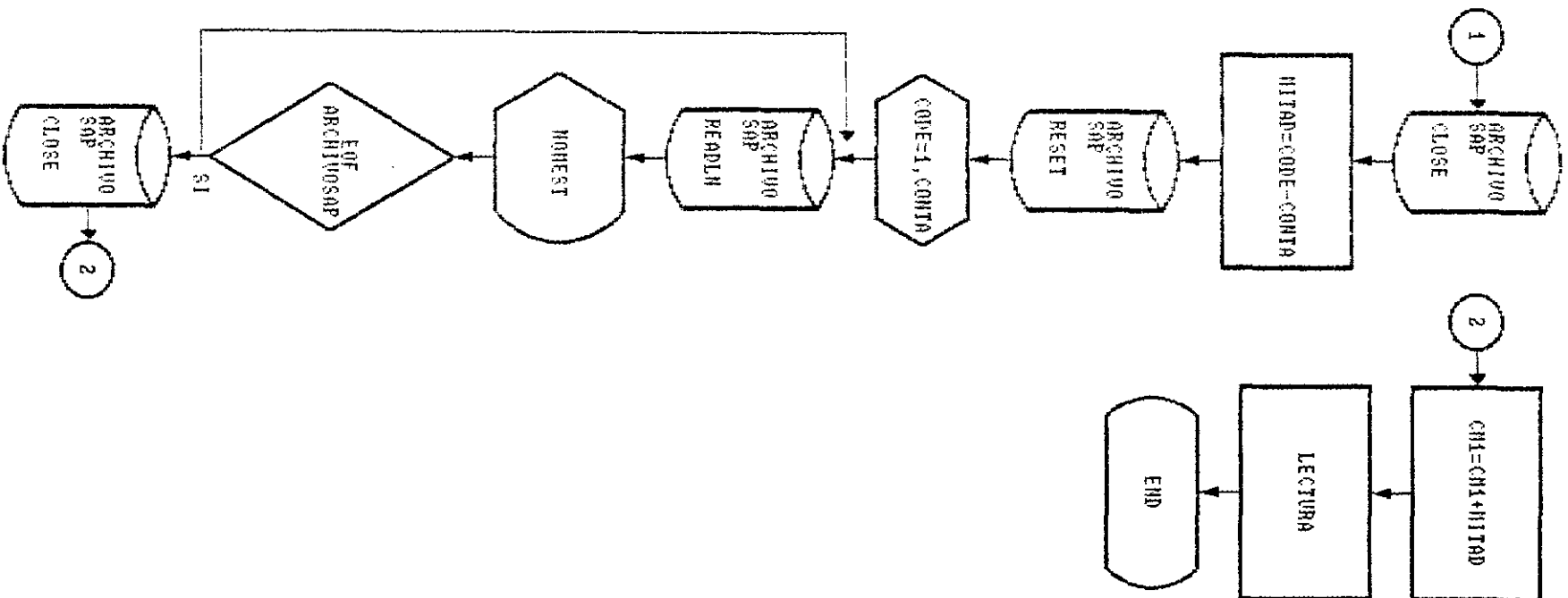
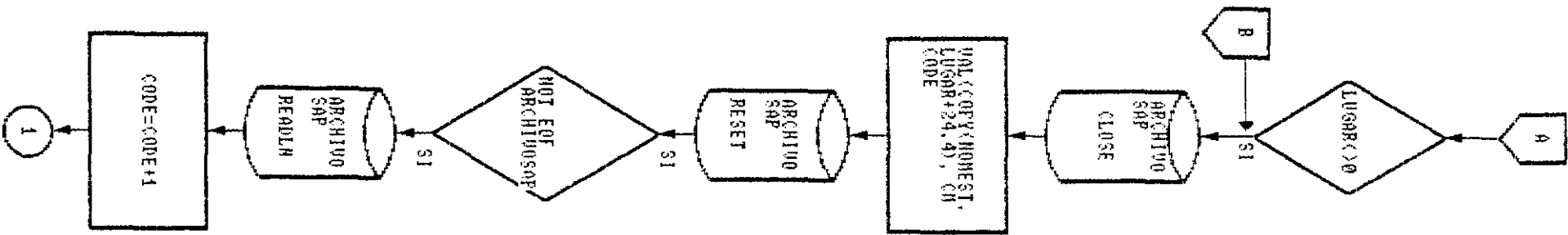
| | | |
|-----------|------|--------------------------------|
| UJ | Byte | Estado de carga |
| P | Byte | Base |
| H | Byte | Altura |
| WX | Byte | Carga distribuida en X |
| WT | Byte | Carga distribuida en Y |
| WZ | Byte | Carga distribuida en Z |
| B1 | Real | Base 1 |
| B2 | Real | Base 2 |
| B3 | Real | Base 3 |
| B4 | Real | Altura 1 |
| B5 | Real | Altura 2 |
| B6 | Real | Altura 3 |
| A | Real | Area |
| I | Real | Inercia |
| J | Byte | Contador |
| EWG | Char | Para aceptar datos de ayuda |
| EW | Char | Para aceptar datos de ayuda |
| EWGEM | Char | Para aceptar datos de ayuda |
| EWGEMTTPQ | Char | Ayuda |
| EWGEMR | Char | Ayuda |
| EWGEMRPF | Char | Ayuda |
| EWGEMRPA | Char | Ayuda |

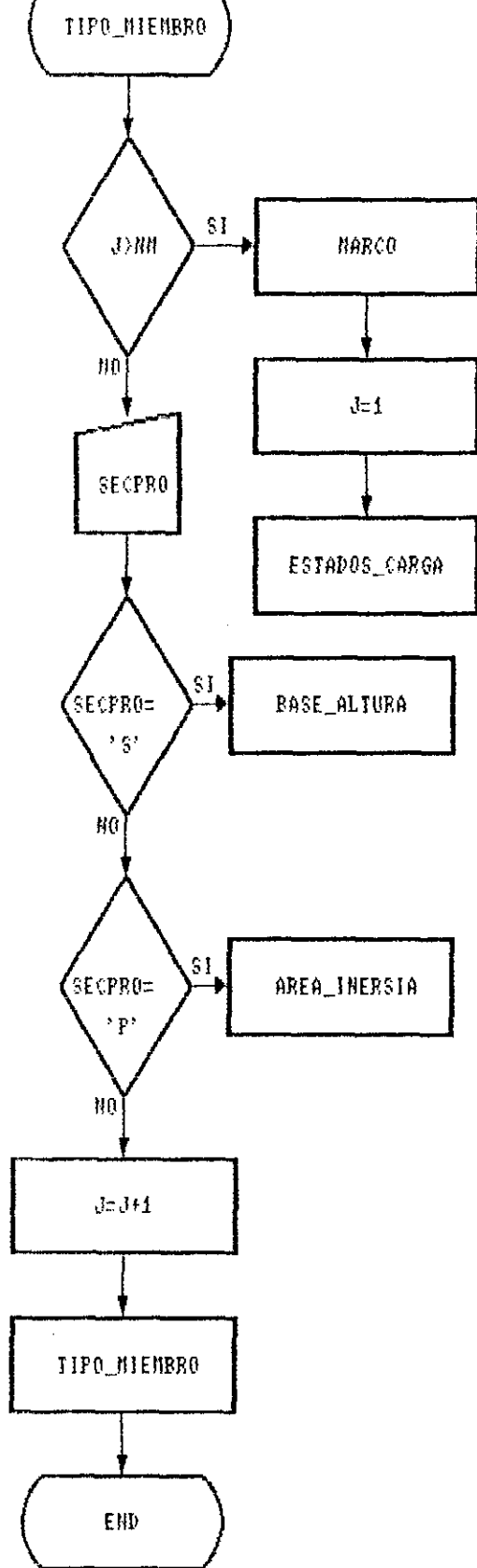
| | | |
|---------|--------|--|
| QIWAUX | String | en el archivo sin espacios en blanco |
| QIUPAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| QIUPAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| ALIAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| IAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| IBGAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| IBJAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| IBFAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |

| | | |
|-------|--------|--|
| HIAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| HPAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| HEAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| HPAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| HVAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| HPAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| CPAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |

| | | | |
|--------|--------|------------------------------------|--|
| PCAINZ | ST-ING | blanco | Ayuda para grabar en el archivo sin espacios en blanco |
| PCAINZ | ST-ING | blanco | Ayuda para grabar en el archivo sin espacios en blanco |
| PCAINZ | ST-ING | blanco | Ayuda para grabar en el archivo sin espacios en blanco |
| ADPH1 | ByLe | Para grabar | |
| ADPH2 | ByLe | Para grabar | |
| ADPH3 | ByLe | Para grabar | |
| ADPH4 | ByLe | Para grabar | |
| LDQ1 | ByLe | Para tomar un variable en espacios | |





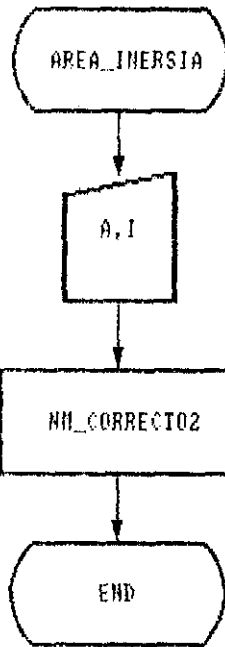


AREA_INERSIA

A, I

NI_CORRECTO2

END

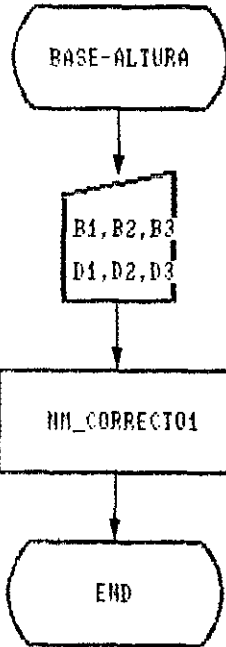


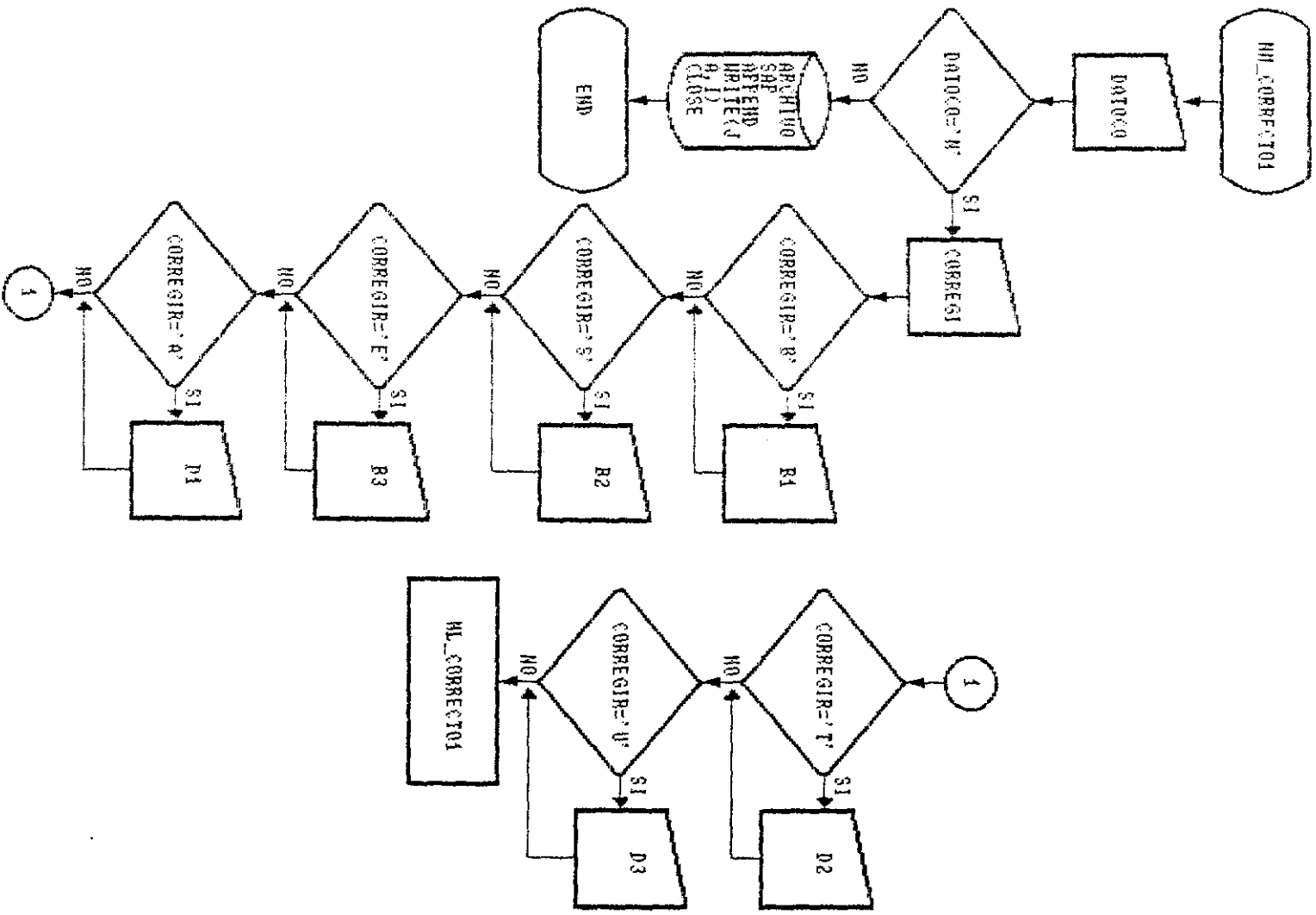
BASE-ALTURA

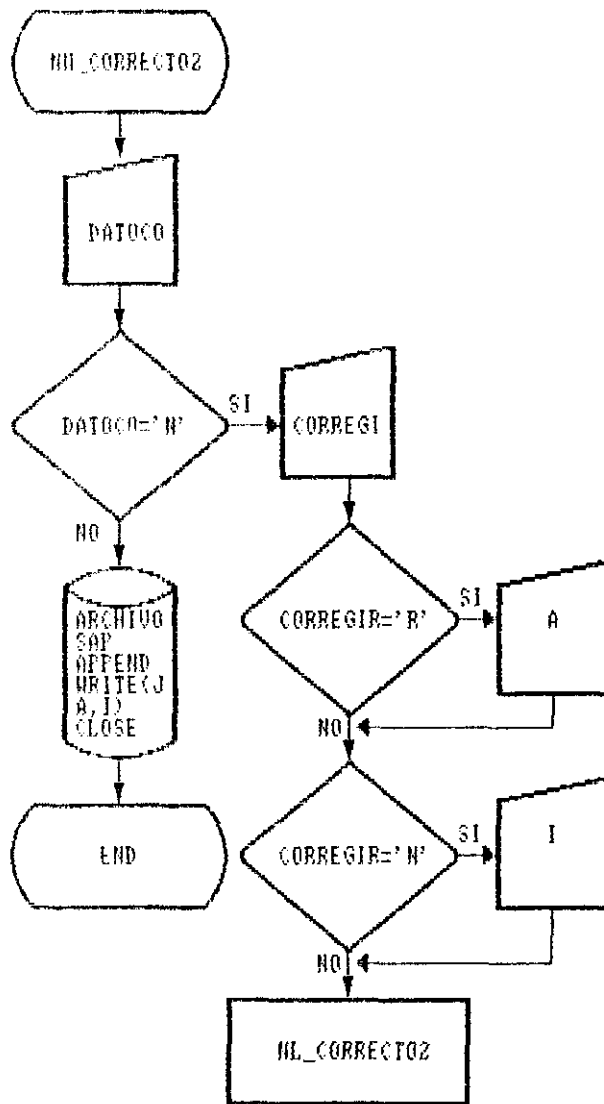
B1, B2, B3
D1, D2, D3

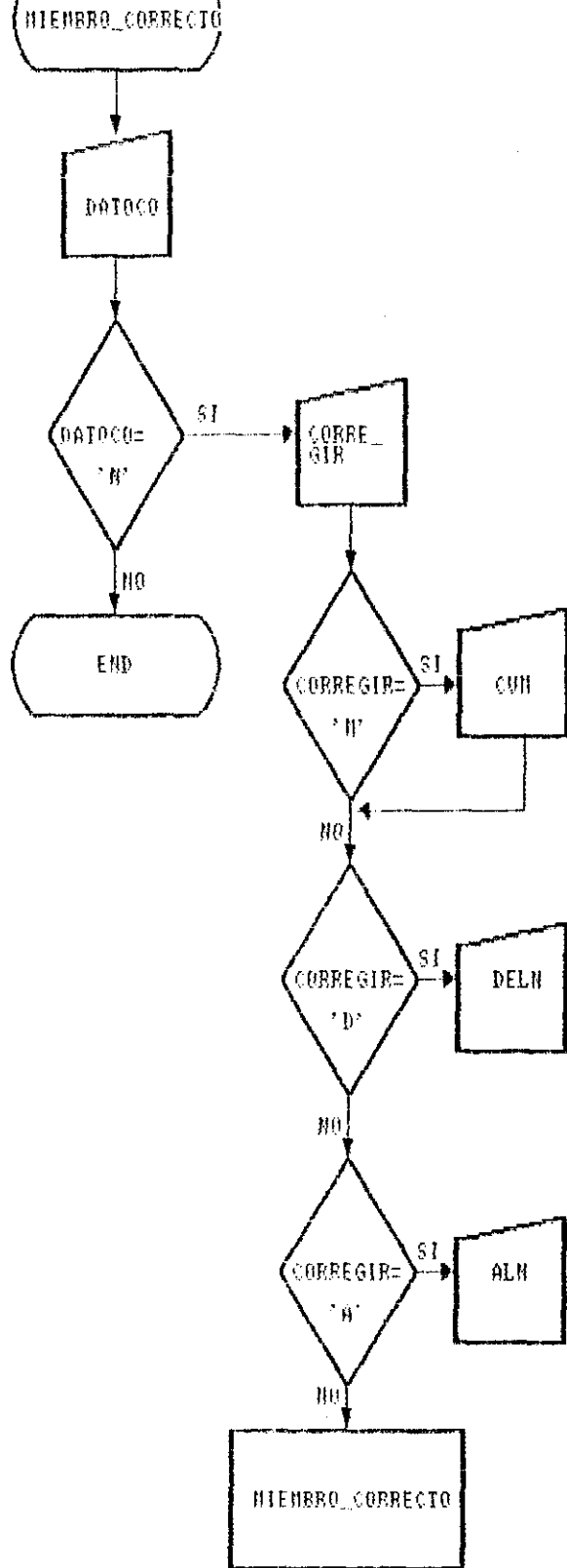
NN_CORRECTO1

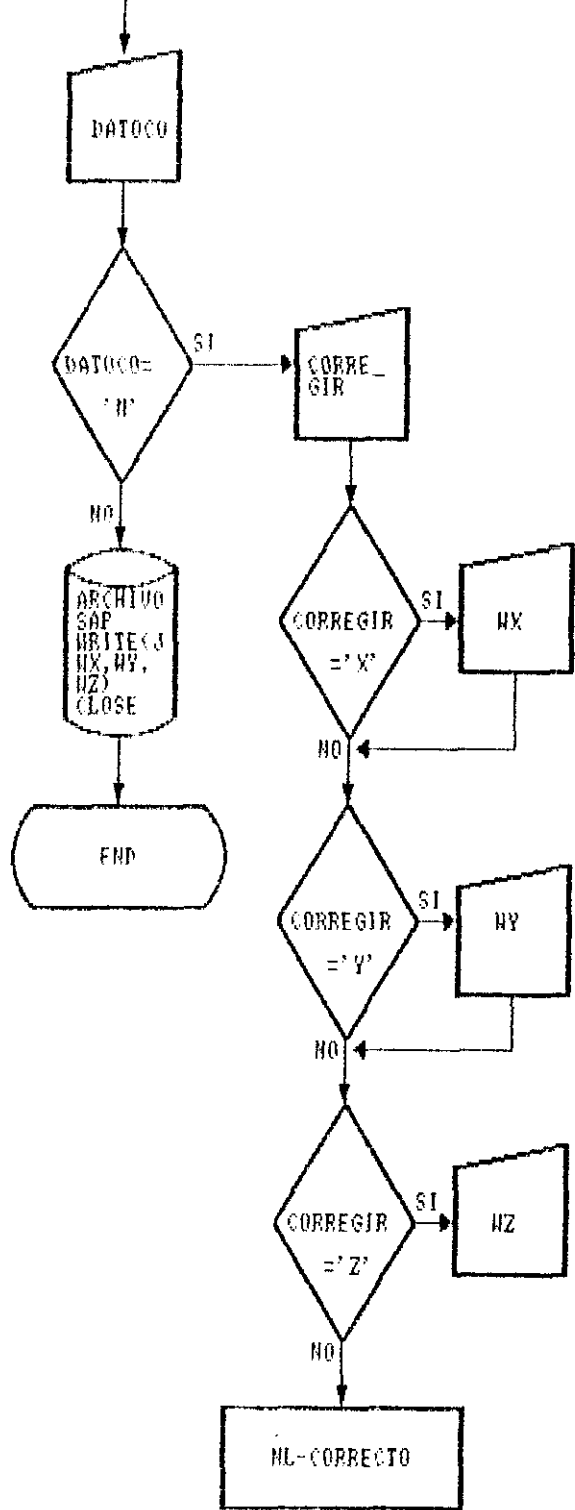
END

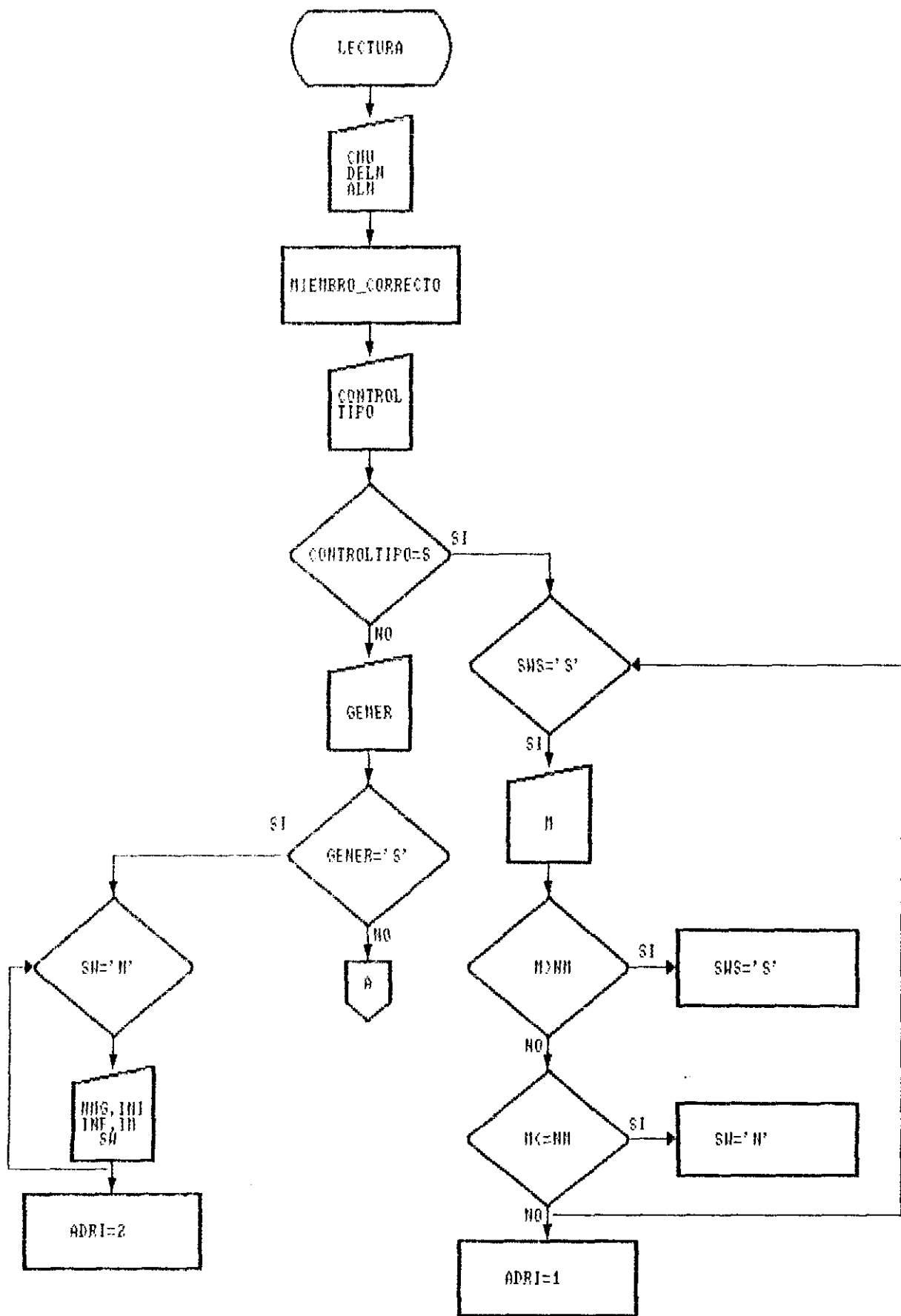


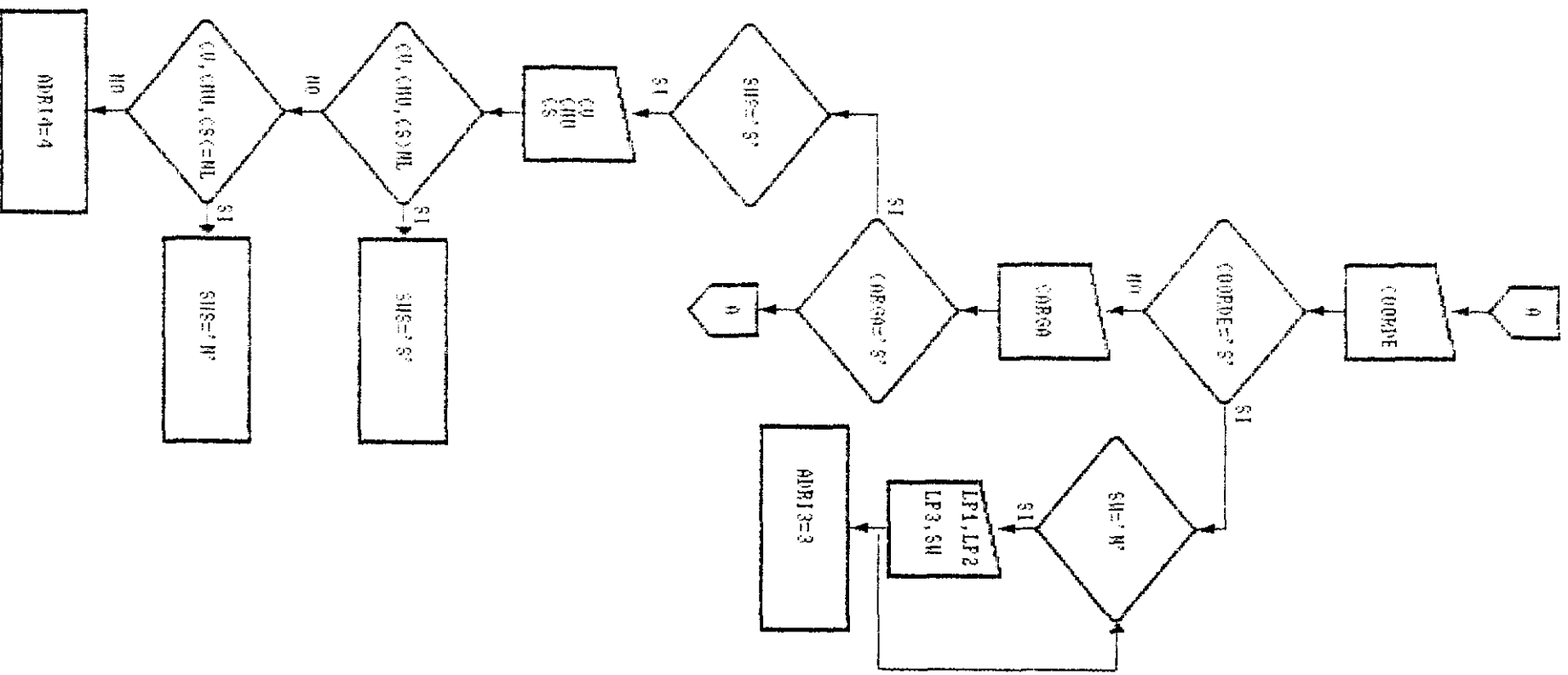


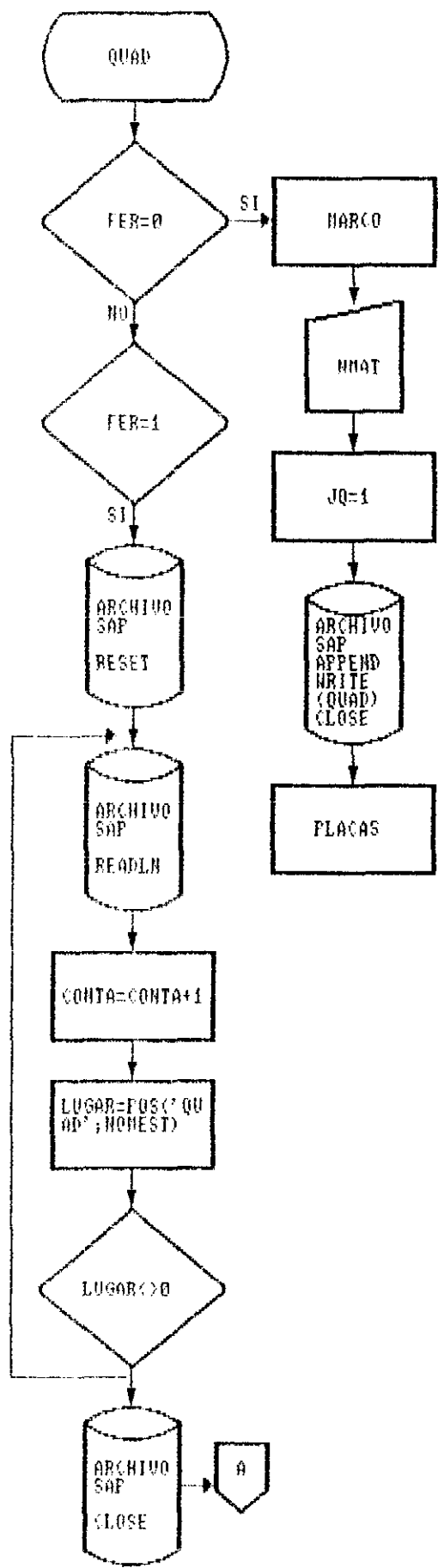


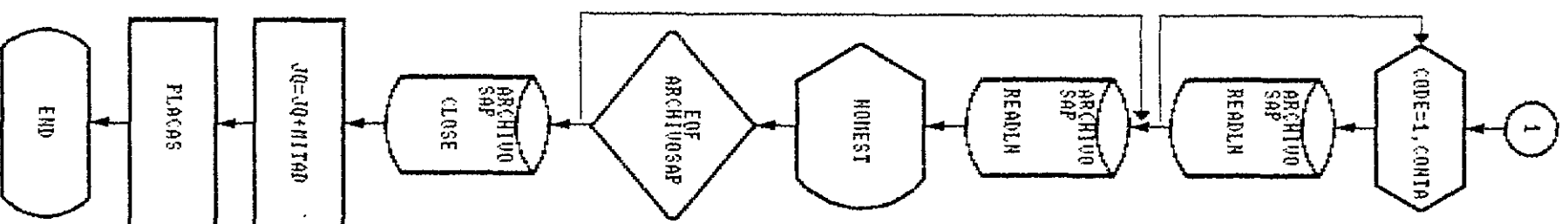
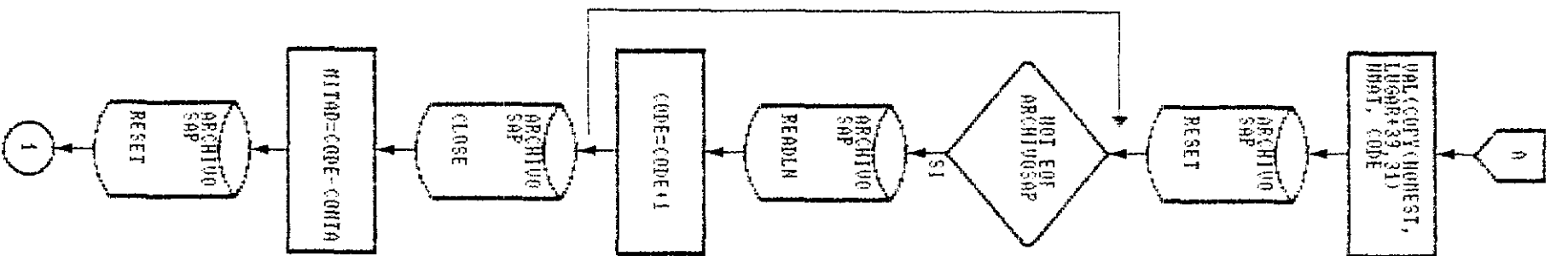


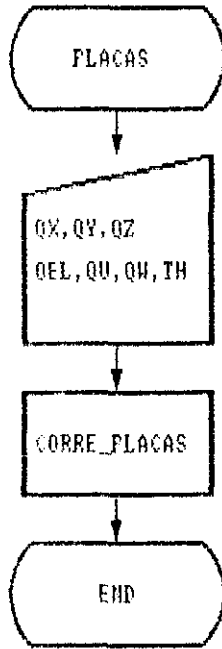


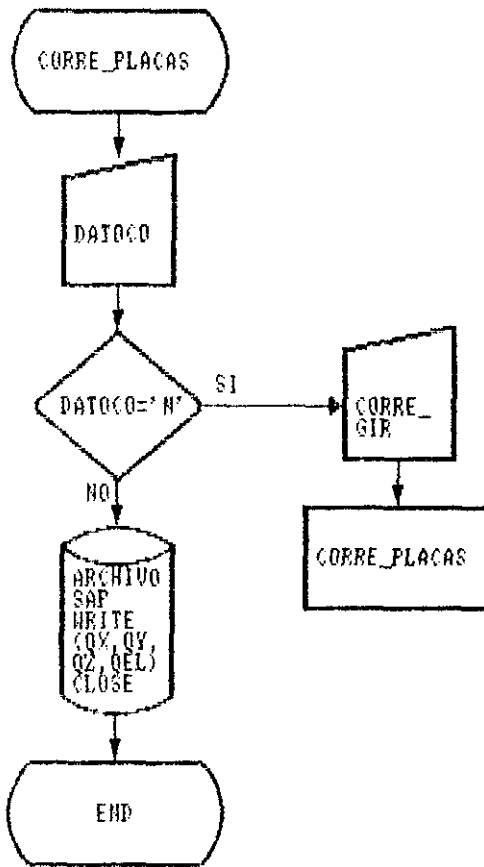












```
UNIT INGRESOS;
INTERFACE
Uses Cr1, UnitFun, Caratulas;
TYPE
```

```
  STR10=STRING(14);
  VAR
    POSITION,COLUMNA,N,COPE:INTEGER;
    CONTA,MIJAD,L:BYTE;
    NOME1,STRINGERBA;
    ARCHIYOSAP,TEXT;
    DATUCO,CORREGIR,SW,CHAR;
    LOGAR:BYTE;
```

```
PROCEDURE LOADS(NUMBER:STRING;VAR N:INTEGER;VAR L,FER:BYTE);
PROCEDURE FORCES(NUMBER:STRING;VAR N:INTEGER;VAR L,FER:BYTE);
PROCEDURE PLOT(NUMBER:STRING;VAR N:INTEGER;VAR L,FER:BYTE);
```

```
IMPLEMENTATION
```

```
PROCEDURE LOADS;
```

```
VAR
```

```
  PRIMM:INTEGER;
  SEGUN,IMUDO,ESTCAR,LON,LUOD,CONMAX:BYTE;
  FY,FY,F,MX,MY,MZ,DX,DY,DZ,RX,RY,RZ,REAL;
  PRJUNOAY,SEGUNOAY,IMUDNOAY,ESTCARAY:STRING(34);
  FYAY,FYAY,FZAY,MYAY,MZAY,MZAY:STRING(44);
  DYAY,DYAY,DZAY,RXAY,RYAY,RZAY:STRING(34);
```

```
PROCEDURE LEE_LOAD(POSITION:INTEGER;CONTA,L,OAD,L:BYTE); FORWARD;
```

```
PROCEDURE DATO_CORREL_LOAD(POSITION:INTEGER;CONTA,L:BYTE);
```

```
  Label 2;
```

```
  BEGIN
```

```
    GOTOXY(5,22);WRITE('
);
    GOTOXY(5,23);WRITE('
);
    GOTOXY(5,22);WRITE('Datos Correctos S/N: ');
    TEXTCLR(14);
    GOTOXY(21,22);WRITE('S');
    GOTOXY(23,22);WRITE('N');
    TEXTCLR(7);
    GOTOXY(26,22);DATUCO:=RENDEK;
    IF (DATUCO='N') OR (DATUCO='n') THEN
      BEGIN
        GOTOXY(5,22);WRITE('
);
        GOTOXY(5,22);WRITE('Corregir: 1a ó 2a Junta, Incremento, Estado, Fuer.Y, Fuer.Z,);
        GOTOXY(5,23);WRITE('Homen.X, Homen.Y, Homen.Z, Despl.X, Despl.Y, Despl.Z; Rota.: X, Y, Z; ');
        TEXTCLR(14);
        GOTOXY(15,22);WRITE('1');GOTOXY(20,22);WRITE('2');
        GOTOXY(30,22);WRITE('1');GOTOXY(42,22);WRITE('E');GOTOXY(50,22);WRITE('F');
        GOTOXY(59,22);WRITE('V');GOTOXY(69,22);WRITE('R');
        GOTOXY(5,23);  WRITE('N');  GOTOXY(15,23);WRITE('O');GOTOXY(27,23);WRITE('N');
        GOTOXY(32,23);WRITE('D');GOTOXY(43,23);WRITE('S');GOTOXY(53,23);WRITE('P');
        GOTOXY(66,23);WRITE('X');GOTOXY(69,23);WRITE('Y');GOTOXY(72,23);WRITE('Z');
        TEXTCLR(7);
        GOTOXY(76,23);CORREGIR:=RENDEK;
        LAST_CORREGIR:=
          1;  BEGIN
```

```
  COLUMNA:=3;  PRJUNM:= FunMudoFin(columna,position);
```

```
  IF PRJUNM=MaxInt Then
```

```
  BEGIN
```

```

        GOTO 2;
        END;
        END;
'2': BEGIN
        COLUMNA:=9; SEGIUN:= FunInEnt(columna, posicion);
        END;
'I', 'i': BEGIN
        COLUMNA:=14; INICIO:= FunInEnt(columna, posicion);
        END;
'E', 'e': BEGIN
        GOTOXY(22, POSICION); WRITE(' ');
        GOTOXY(22, POSICION); WRITE('L=');
        COLUMNA:=24; ESTCAR:= FunInEnt(columna, posicion);
        WHILE ESTCAR > L DO
        BEGIN
                COLUMNA:=24; ESTCAR:= FunInEnt(columna, posicion);
        END;
        END;
'F', 'f': BEGIN
        COLUMNA:=28; FY:=FunInDec2(columna, posicion);
        END;
'U', 'u': BEGIN
        COLUMNA:=33; FY:= FunInDec2(columna, posicion);
        END;
'R', 'r': BEGIN
        COLUMNA:=38; FZ:=FunInDec2(columna, posicion);
        END;
'M', 'm': BEGIN
        COLUMNA:=43; FY:=FunInDec2(columna, posicion);
        END;
'O', 'o': BEGIN
        COLUMNA:=48; HY:=FunInDec2(columna, posicion);
        END;
'N', 'n': BEGIN
        COLUMNA:=53; HZ:=FunInDec2(columna, posicion);
        END;
'D', 'd': BEGIN
        COLUMNA:=58; DX:=FunInDec1(columna, posicion);
        END;
'S', 's': BEGIN
        COLUMNA:=62; DY:=FunInDec1(columna, posicion);
        END;
'P', 'p': BEGIN
        COLUMNA:=66; DZ:=FunInDec1(columna, posicion);
        END;
'X', 'x': BEGIN
        COLUMNA:=70; RX:=FunInDec1(columna, posicion);
        END;
'Y', 'y': BEGIN
        COLUMNA:=74; RY:=FunInDec1(columna, posicion);
        END;
'Z', 'z': BEGIN
        COLUMNA:=78; RZ:=FunInDec1(columna, posicion);
        END;
END; *End_Case$
DATO_CORRELOAD(POSICION, CONTAX, L);
END *End_If Dato Correcto$
ELSE
BEGIN
        PRIJUNAU:= ' '; LON:=0;
        STR(PRIJUN, PRIJUNAU);
        LON:=LENGTH(PRIJUNAU);

```

```

WHILE LON<=3 DO
  BEGIN
    PRIJUNAU:=PRIJUNAU+' ';
    lon:=lon+1;
  END;
SEGJUNAU:=' ';LON:=0;
STR(SEGJUN,SEGJUNAU);
LON:=LENGTH(SEGJUNAU);
WHILE LON<=3 DO
  BEGIN
    SEGJUNAU:=SEGJUNAU+' ';
    lon:=lon+1;
  END;
INNUDOAU:=' ';LON:=0;
STR(INNUDO,INNUDOAU);
LON:=LENGTH(INNUDOAU);
WHILE LON<=3 DO
  BEGIN
    INNUDOAU:=INNUDOAU+' ';
    lon:=lon+1;
  END;
ESTCARAU:=' ';LON:=0;
STR(ESTCAR,ESTCARAU);
LON:=LENGTH(ESTCARAU);
WHILE LON<=3 DO
  BEGIN
    ESTCARAU:=ESTCARAU+' ';
    lon:=lon+1;
  END;
END;
STR(FX:1:1,FXAU);
STR(FY:1:1,FYAU);
STR(FZ:1:1,FZAU);
STR(MX:1:1,MXAU);
STR(MY:1:1,MYAU);
STR(MZ:1:1,MZAU);
STR(DX:1:1,DXAU);
STR(DY:1:1,DYAU);
STR(DZ:1:1,DZAU);
STR(RX:1:1,RXAU);
STR(RY:1:1,RYAU);
STR(RZ:1:1,RZAU);
ASSIGN(ARCHIVOSAP,NOMBRE);
APPEND(ARCHIVOSAP);
WRITELN(ARCHIVOSAP,PRIJUNAU,' ',SEGJUNAU,' ',INNUDOAU,' L=',ESTCARAU,' F=',FXAU,',',FYAU,',',FZAU,
        ' ',MXAU,',',MYAU,',',MZAU,' U=',DXAU,',',DYAU,',',DZAU,',',RXAU,',',RYAU,',',RZAU);
CLOSE(ARCHIVOSAP);
CONTAX:=CONTAX+1;
GOTOXY(5,23);WRITE(' ');
END; *End Dato Correcã
2: END; * Procedure Dato Correcã

```

```

PROCEDURE LEE_LOAD(POSICION:INTEGER;CONTAX,JLOAD,L:BYTE);
label 20,3;
BEGIN
  FOR POSICION:=POSICION TO 19 DO
    BEGIN
      IF JLOAD>CONTAX THEN
        BEGIN
          POSICION:=21;
          goto 20;
        END
      END
    END
  END

```

```

BEGIN
  GOTOXY(5,22);WRITE(' ');
  GOTOXY(5,23);WRITE(' ');
  GOTOXY(5,22);WRITE(' Presione Barra Espaciadora en Número de la Primera Junta');
  GOTOXY(5,23);WRITE(' y regresa al Menú Principal ...');
  TEXTCOLOR(14);
  GOTOXY(14,22);WRITE('BARRA ESPACIADORA');
  GOTOXY(18,23);WRITE('MENU PRINCIPAL ');
  TEXTCOLOR(7);
  COLUMNA:=3; PRIJUN:= FunNudoFin(columna,posicion);
  if PRIJUN=maxint then
  BEGIN
    FER:=1;
    GOTD 3;
  END;
  COLUMNA:=9; SEGJUN:= FunInEnt(columna,posicion);
  COLUMNA:=16; INNUDO:= FunInEnt(columna,posicion);
  GOTOXY(5,22);WRITE(' ');
  GOTOXY(5,23);WRITE(' ');
  GOTOXY(5,22);WRITE('L = Número de estado de carga');
  TEXTCOLOR(14);
  GOTOXY(5,22);WRITE('L = N');
  TEXTCOLOR(7);
  GOTOXY(22,POSICION);WRITE('L=');
  COLUMNA:=24; ESTCAR:= FunInEnt(columna,posicion);
  WHILE ESTCAR>L DO
  BEGIN
    COLUMNA:=24; ESTCAR:=FunInEnt(columna,posicion);
  END;
  GOTOXY(5,22);WRITE(' ');
  GOTOXY(5,23);WRITE(' ');
  GOTOXY(5,22);WRITE('F = Fuerzas aplicadas en las direcciones X, Y, Z');
  GOTOXY(5,23);WRITE('M = Momentos aplicados alrededor de los ejes X, Y, Z');
  TEXTCOLOR(14);
  GOTOXY(5,22);WRITE('f = F');
  GOTOXY(17,22);WRITE('A');GOTOXY(34,22);WRITE('D');GOTOXY(46,22);WRITE('X');
  GOTOXY(49,22);WRITE('Y');GOTOXY(52,22);WRITE('Z');
  GOTOXY(5,23);WRITE('m = M');GOTOXY(18,23);WRITE('A');
  GOTOXY(28,23);WRITE('A');GOTOXY(45,23);WRITE('E'); GOTOXY(58,23);WRITE('X');
  GOTOXY(53,23);WRITE('Y');GOTOXY(56,23);WRITE('Z');
  TEXTCOLOR(7);
  COLUMNA:=28; FX:=FunInDec2(columna,posicion);
  COLUMNA:=33; FY:=FunInDec2(columna,posicion);
  COLUMNA:=38; FZ:=FunInDec2(columna,posicion);
  COLUMNA:=43; MX:=FunInDec2(columna,posicion);
  COLUMNA:=48; MY:=FunInDec2(columna,posicion);
  COLUMNA:=53; MZ:=FunInDec2(columna,posicion);
  GOTOXY(5,22);WRITE(' ');
  GOTOXY(5,23);WRITE(' ');
  GOTOXY(5,22);WRITE('d = Desplazamientos aplicados en las direcciones X, Y, Z');
  GOTOXY(5,23);WRITE('r = Rotaciones aplicadas alrededor de los ejes X, Y, Z');
  TEXTCOLOR(14);
  GOTOXY(5,22);WRITE('d = D');
  GOTOXY(25,22);WRITE('A');GOTOXY(42,22);WRITE('D');
  GOTOXY(54,22);WRITE('X');GOTOXY(57,22);WRITE('Y');GOTOXY(60,22);WRITE('Z');
  GOTOXY(5,23);WRITE('r = R');
  GOTOXY(28,23);WRITE('A');GOTOXY(38,23);WRITE('A');GOTOXY(47,23);WRITE('E');
  GOTOXY(52,23);WRITE('X');GOTOXY(55,23);WRITE('Y');GOTOXY(58,23);WRITE('Z');
  TEXTCOLOR(7);
  COLUMNA:=58; DX:=FunInDec1(columna,posicion);
  COLUMNA:=62; DY:=FunInDec1(columna,posicion);

```

```

        COLUMNA:=66; DZ:=FunInDeci(columna,posicion);
        COLUMNA:=70; RX:=FunInDeci(columna,posicion);
        COLUMNA:=74; RY:=FunInDeci(columna,posicion);
        COLUMNA:=78; RZ:=FunInDeci(columna,posicion);
        DATO_CORRELOADA(POSICION,CONTAX,L);
        GOTOXY(5,23);WRITE(
        JLOAD:=JLOAD+1
    END; *End JLOAD:CONTAX
END; *DEL FORA
CLRSCL;
MARCO;
TEXTCOLOR(14);
GOTOXY(25,2);WRITE('CARGAS CONCENTRADAS EN LOS NUDOS');
GOTOXY(6,4);WRITE('Número de a declarar:',CONTAX);
GOTOXY(3,6);WRITE(' Juntas  Incr.  Est. |Fuerza direcc. |Mome.alreded. |Desp.direc. |Rot. al eje');
GOTOXY(3,7);WRITE(' 1º   2º          Carga | X   Y   Z | X   Y   Z | X   Y   Z | X   Y   Z');
TEXTCOLOR(7);
POSICION:=8;
LEE_LOAD(POSICION,CONTAX,JLOAD,L);
20: if posicion=21 then
    BEGIN
        ASSIGN(ARCHIVOSAP,NOMBRE);
        APPEND(ARCHIVOSAP);
        WRITELN(ARCHIVOSAP,'C FIN DE LOADS');
        CLOSE(ARCHIVOSAP);
        FER:=2;
    END;
3: END;

*PROGRAMA PRINCIPAL DE LOADS
BEGIN
    IF FER=0 THEN
        BEGIN
            CLRSCL;
            MARCO;
            TEXTCOLOR(14);
            GOTOXY(25,2);WRITE('CARGAS CONCENTRADAS EN LOS NUDOS');
            GOTOXY(6,4);WRITE('Ingrese la cantidad de cargas a declarar:');
            TEXTCOLOR(7);
            GOTOXY(5,22);WRITE('Ingrese 0 si no desea ingresar datos');
            TEXTCOLOR(14);GOTOXY(13,22);WRITE('0');TEXTCOLOR(7);
            CONTAX:=0;
            COLUMNA:=52;POSICION:=4;
            CONTAX:=FunInEnt(columna,posicion);
            TEXTCOLOR(14);
            GOTOXY(3,6);WRITE(' Juntas  Incr.  Est. |Fuerza direcc. |Mome.alreded. |Desp.direc. |Rot. al eje');
            GOTOXY(3,7);WRITE(' 1º   2º          Carga | X   Y   Z | X   Y   Z | X   Y   Z | X   Y   Z');
            TEXTCOLOR(7);
            JLOAD:=1;
            POSICION:=9;
            ASSIGN(ARCHIVOSAP,NOMBRE);
            APPEND(ARCHIVOSAP);
            WRITELN(ARCHIVOSAP,'LOADS : CARGAS CONCENTRADAS EN LOS NUDOS ',CONTAX);
            CLOSE(ARCHIVOSAP);
            LEE_LOAD(POSICION,CONTAX,JLOAD,L);
        END
    ELSE
        BEGIN
            IF FER=1 THEN
                BEGIN
                    ASSIGN(ARCHIVOSAP,NOMBRE);

```



```
CF1,CF2,CF3;REN1;  
JOB AUX;STRING$A;  
CF1AUX,CF2AUX,CF3AUX;STRING$AA;
```

```
PROCEDURE CORR1(POSITION:INTEGER;OF,JOFF:BYTE); FORWARD;
```

```
Label 2;  
BEGIN
```

```
  GOTXY(5,22);WRITE(  
    GOTXY(5,23);WRITE(  
    GOTXY(5,23);WRITE(' Sus Datos están Correctos S/M: '));  
  TEXTCOLOR(14);  
  GOTXY(31,23);WRITE('S');  
  GOTXY(33,23);WRITE('N');  
  TEXTCOLOR(7);  
  GOTXY(36,23);DATOCQ:=READY;  
  IF (DATOCQ='N') OR (DATOCQ='n') THEN  
    BEGIN
```

```
      GOTXY(5,23);WRITE(  
        GOTXY(5,22);WRITE(' Corregir: Factor de carga: 1, 2, 3: '));  
      TEXTCOLOR(14);  
      GOTXY(32,22);WRITE('1');  
      GOTXY(35,22);WRITE('2');  
      GOTXY(38,22);WRITE('3');  
      TEXTCOLOR(7);  
      GOTXY(32,22);CORREGIR:=READY;  
      CASE CORREGIR OF  
        '1': BEGIN
```

```
          COLUMNNA:=26; CF1:=FunIndex(columnna,position);  
          If cfl=Maxint then  
            BEGIN  
              FERR:=1;  
              ROTO 2;  
            END;
```

```
          END;  
          '2': BEGIN  
            COLUMNNA:=35; CF2:=FunIndex(columnna,position);  
          END;  
          '3': BEGIN  
            COLUMNNA:=44; CF3:=FunIndex(columnna,position);
```

```
          END;  
          CORRE_CORR1(POSITION,OF,JOFF);  
        END #Find_If_DatocoA  
      ELSE
```

```
        BEGIN  
          ASSIGN(ARCHIVO$AP,NOMBRE);  
          APPEND(ARCHIVO$AP);  
          JOB AUX:= ' ' ;LON:=0;  
          STR(JOF,JOFFAUX);  
          LON:=LENGTH(JOFFAUX);  
          WHILE LON<=3 DO  
            BEGIN  
              JOFFAUX:=JOFFAUX+' ';
```

```
              LON:=LON+1;  
            END;  
          WHILE LON<=3A  
            STR(CF1:1:2,CF1AUX);  
            WRITE(ARCHIVO$DIR,JOFFAUX,
```

```
              C=' ',CF1AUX,' ',CF2AUX,' ',CF3AUX);
```

```

CLOSE(ARCHIVOSAP);
END; *End_else_datocod
2: END; *End_ProcedureA

PROCEDURE COMPI(POSICION:INTEGER;OF,JOF:BYTE);
label 20,3;
BEGIN
  FOR POSICION:=POSICION TO 19 DO
    BEGIN
      IF JOF>OF THEN
        BEGIN
          POSICION:=21;
          Goto 20;
        END
      ELSE
        BEGIN
          TEXTCOLOR(14);
          GOTOXY(12,POSICION);WRITE(JOF);
          TEXTCOLOR(7);
          GOTOXY(5,22);WRITE(' ');
          GOTOXY(5,23);WRITE(' ');
          GOTOXY(5,22);WRITE('Presione Barra Espaciadora en Factor de Carga 1');
          GOTOXY(5,23);WRITE('y regresa al Menú Principal ....');
          TEXTCOLOR(14);
          GOTOXY(14,22);WRITE('BARRA ESPACIADORA');
          GOTOXY(18,23);WRITE('MENU PRINCIPAL ');
          TEXTCOLOR(7);
          COLUMNA:=26; CF1:=FunFinDec(columna, posicion);
          if cfl=maxint then
            BEGIN
              FER:=1;
              GOTO 3;
            END;
          COLUMNA:=35; CF2:=FunInDec(columna, posicion);
          COLUMNA:=44; CF3:=FunInDec(columna, posicion);
          CORRE_COMPI(POSICION,OF,JOF);
          JOF:=JOF+1;
        END; *End_ofA
      END; *End_ForA
    END;
  CLEAR;
  MARCO;
  TEXTCOLOR(14);
  GOTOXY(20,4);WRITE('COMBINACION DE CARGAS');
  GOTOXY(10,6);WRITE('Número Factor de carga para condición');
  GOTOXY(10,7);WRITE(' 1 2 3 ');
  TEXTCOLOR(7);
  POSICION:=8;
  COMPI(POSICION,OF,JOF);
20: if posicion=21 then
  BEGIN
    ASSIGN(ARCHIVOSAP,NOMBRE);
    APPEND(ARCHIVOSAP);
    WRITELN(ARCHIVOSAP,'C FIN DE FORCES');
    CLOSE (ARCHIVOSAP);
    FER:=2;
  END;
3: END; *End_ProcedureA

```

```

(***PROGRAMA PRINCIPAL DEL FORCES***)
BEGIN
  IF FER=0 THEN

```

```

REGIM
CLASCR:
MARCO:
TEXTCOLOR(14);
GOTOXY(18,2);WRITE(' C O M P I N A C I O N   D E   C A R G A S ');
GOTOXY(6,4);WRITE(' Ingrese la cantidad de combinaciones a declarar: ');
TEXTCLR(7);
GOTOXY(5,22);WRITE(' Ingrese 0 si no desea ingresar datos ');
TEXTCLR(14);
GOTOXY(13,22);WRITE(' 0 ');
TEXTCLR(7);
COLUMNA:=55;POSICION:=4;
PF:=funInferi(columna,position);
TEXTCLR(14);
GOTOXY(10,4);WRITE(' Numero      Factor de carga para condiclon ');
GOTOXY(10,7);WRITE('          1          2          3          ');
TEXTCLR(7);
JRF:=1;POSICION:=8;
ASSIGN(ARCHIVOSAP,'MONRE');
APPEND(ARCHIVOSAP);
WRITELN(ARCHIVOSAP,' FORCES : COMBINACIONES DE CARGAS ',JRF);
CLOSE(ARCHIVOSAP);
CORRU(POSICION,DF,JRF);
END
ELSE
REGIM
IF FER=1 THEN
REGIM
ASSIGN(ARCHIVOSAP,'MONRE');
RESET(ARCHIVOSAP);
CODE:=0;CUNTA:=R;
REPEAT
READLN(ARCHIVOSAP,'MONRE');
CUNTA:=CUNTA+1;
LUGAR:=PUB(' FORCES ',MONRE);
WRITE(LUGAR<>'M');
CLOSE(ARCHIVOSAP);
VAL(COPY(MONRE,1,LUGAR+3,3),DF,CODE);
ASSIGN(ARCHIVOSAP,'MONRE');
RESET(ARCHIVOSAP);
CODE:=0;
WHILE NOT EOF (ARCHIVOSAP) DO
REGIM
READLN(ARCHIVOSAP,'MONRE');
CODE:=CODE+1;
END;
CLOSE(ARCHIVOSAP);
MITAD:=0;
NTAD:=CODE-CUNTA;
CLASCR:
MARCO:
TEXTCLR(14);
GOTOXY(18,2);WRITE(' C O M P I N A C I O N   D E   C A R G A S ');
GOTOXY(6,4);WRITE(' Numero total de combinaciones a ingresar : ',DF);
GOTOXY(6,5);WRITE(' Numero de combinaciones ingresadas      : ',MITAD);
GOTOXY(3,6);WRITE('          Numero      Factor de carga ');
WRITELN;
TEXTCLR(7);
ASSIGN(ARCHIVOSAP,'MONRE');
RESET(ARCHIVOSAP);

```

```

        READLN(ARCHIVOSAP,NOMEST);
    REPEAT
        READLN(ARCHIVOSAP,NOMEST);
        WRITELN( ' ',NOMEST);
    UNTIL EOF(ARCHIVOSAP);
    WRITELN;WRITELN( ' PRESIONE <ENTER> PARA CONTINUAR EL INGRESO');
    READLN;
    CLOSE(ARCHIVOSAP);
    CLRSCL;
    MARCO;
    TEXTCOLOR(14);
    GOTOXY(18,2);WRITE('COMBINACION DE CARGAS');
    GOTOXY(6,4);WRITE('Número total de combinaciones a ingresar : ',QF);
    GOTOXY(6,5);WRITE('Número de combinaciones ingresadas : ',MITAD);
    GOTOXY(10,7);WRITE('Número Factor de carga para condición ');
    GOTOXY(10,8);WRITE('          1          2          3 ');
    TEXTCOLOR(7);
    JQF:=1;
    POSICION:=10;
    JQF:=JQF+MITAD;
    COMBI(POSICION,QF,JQF);
    END; #DEL IF FER=1A
    END; #DEL ELSE#
END; #del procedure FORCESS#

PROCEDURE PLOT;
VAR
    NV1,NV2,NH1,NH2:INTEGER;

PROCEDURE CORRE_INGRESO(POSICION:INTEGER);
LABEL 2;
BEGIN
    GOTOXY(5,21);WRITE(' ');
    GOTOXY(5,22);WRITE(' ');
    GOTOXY(5,23);WRITE(' ');
    GOTOXY(5,23);WRITE('Sus Datos están Correctos S/N: ');
    TEXTCOLOR(14);
    GOTOXY(31,23);WRITE('S');
    GOTOXY(33,23);WRITE('N');
    TEXTCOLOR(7);
    GOTOXY(36,23);DATOCD:=READKEY;
    IF (DATOCD='N') OR (DATOCD='n') THEN
        BEGIN
            GOTOXY(5,23);WRITE(' ');
            GOTOXY(5,22);WRITE('Corregir: Vector de vista para nudo: 1, 2; Nudo 1 ó Nudo 2');
            GOTOXY(5,23);WRITE(' del vector horizontal');
            TEXTCOLOR(14);
            GOTOXY(42,22);WRITE('1');GOTOXY(45,22);WRITE('2');GOTOXY(51,22);WRITE('N');GOTOXY(61,22);WRITE('U');
            TEXTCOLOR(7);
            GOTOXY(70,23);CORREGIR:=READKEY;
            CASE CORREGIR OF
                '1' : BEGIN
                    COLUMNA:=10;
                    NV1:=FunNudoFin(columna,posicion);
                    if NV1=maxint then GOTO 2;
                END;
                '2' : BEGIN
                    COLUMNA:=18;
                    NV2:=FunInEnt(columna,posicion);
                END;
                'N','n' : BEGIN

```

```

        COLUMNA:=41;
        NH1:=FunInEnt(columna, posicion);
        END;
    'H', 'u' : BEGIN
        COLUMNA:=58;
        NH2:=FunInEnt(columna, posicion);
        END;
    END; *End_Case$
    CORRE_INGRESO(POSICION);
END *End_If_Datoco$
ELSE
    BEGIN
    ASSIGN(ARCHIVOSAP, NOMBRE);
    APPEND(ARCHIVOSAP);
    WRITELN(ARCHIVOSAP, 'VP-', NV1, ', ', NV2, '   VH=', NH1, ', ', NH2);
    WRITELN(ARCHIVOSAP, 'C FIN DE PLOT');
    WRITELN(ARCHIVOSAP, 'END');
    CLOSE(ARCHIVOSAP);
    END; *End_else_dataco$
2: END; *End_Procedure$

```

```

PROCEDURE INGRESO(POSICION:INTEGER);
LABEL 3;
BEGIN
    GOTOXY(5,22);WRITE(' ');
    GOTOXY(5,23);WRITE(' ');
    GOTOXY(5,22);WRITE('Presione Barra Espaciadora en Vector de vista para Nudo 1');
    GOTOXY(5,23);WRITE('y regresa al Menú Principal ....');
    TEXTCOLOR(14);
    GOTOXY(14,22);WRITE('BARRA ESPACIADORA');
    GOTOXY(18,23);WRITE('MENU PRINCIPAL ');
    TEXTCOLOR(7);
    COLUMNA:=10;NV1:=FunNudoFin(columna, posicion);
    if NV1=maxint then GOTO 3;
    COLUMNA:=18;NV2:=FunInEnt(columna, posicion);
    COLUMNA:=41;NH1:=FunInEnt(columna, posicion);
    COLUMNA:=58;NH2:=FunInEnt(columna, posicion);
    CORRE_INGRESO(POSICION);
3: END; *End_Procedure$

```

```

(***PROGRAMA PRINCIPAL DEL PLOT***)
BEGIN
    CLASER;
    MARCO;
    TEXTCOLOR(14);
    GOTOXY(10,2);WRITE('GRAFICO DE LA ESTRUCTURA');
    GOTOXY(4,4);WRITE('Vectores de vista para nudo   Nudo 1 vector   Nudo 2 vector');
    GOTOXY(4,5);WRITE('      1      2           horizontal   horizontal');
    TEXTCOLOR(7);
    POSICION:=7;
    ASSIGN(ARCHIVOSAP, NOMBRE);
    APPEND(ARCHIVOSAP);
    WRITELN(ARCHIVOSAP, 'PLOT : GRAFICO DE LA ESTRUCTURA');
    CLOSE(ARCHIVOSAP);
    INGRESO(POSICION);
END;

END. *FIN DE LA UNIDAD INGRESO$

```

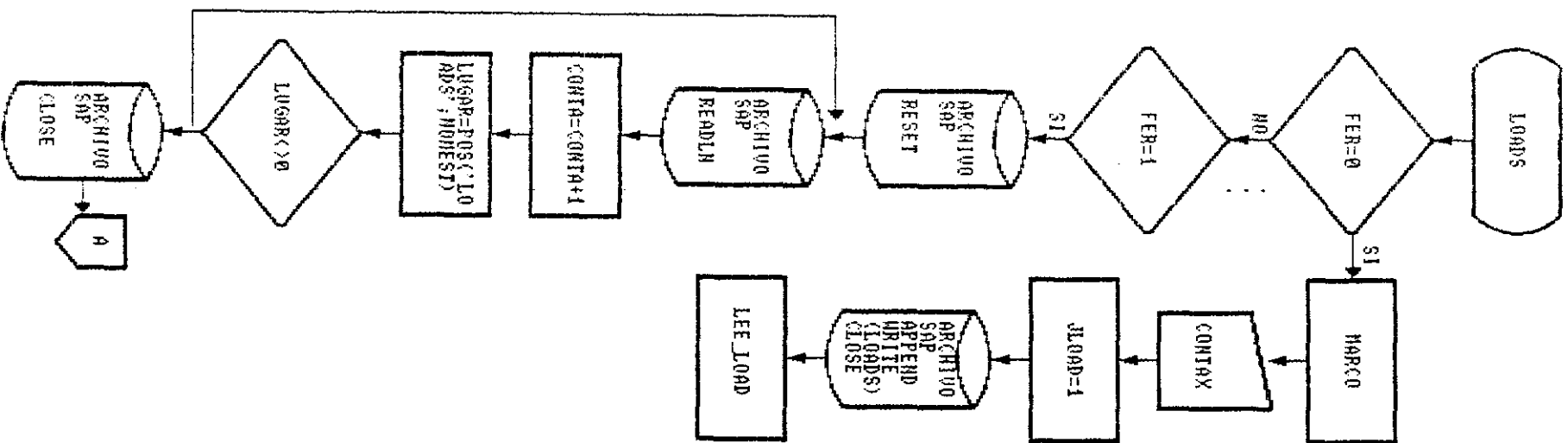
LISTA DE VARIABLES

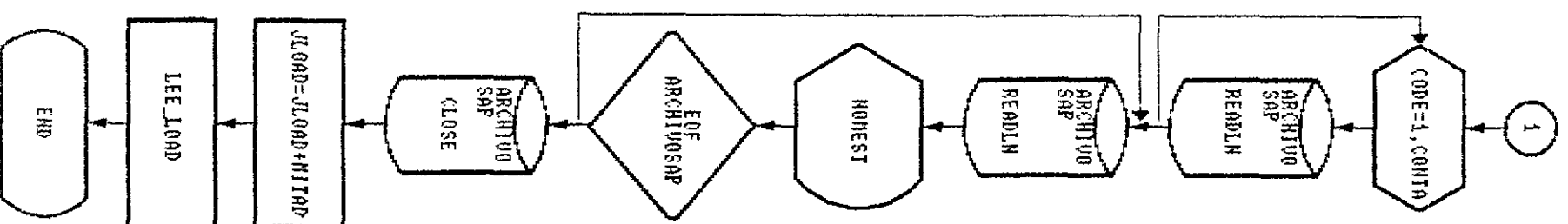
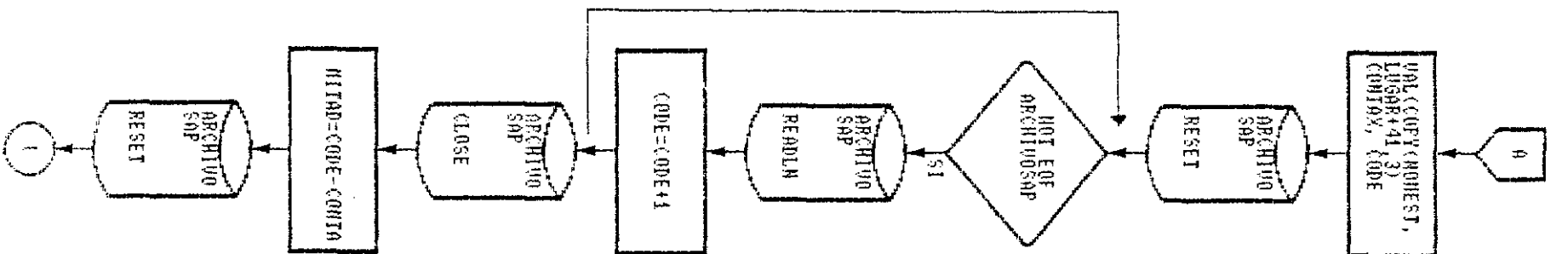
NOMBRE DEL PROGRAMA: LOAD

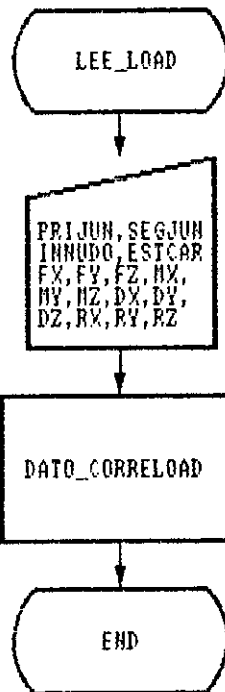
| NOMBRE | TIPO | DESCRIPCION |
|--------|------|--------------------------------------|
| PRIJUN | Byte | Número de la primera junta |
| SEGSUN | Byte | Número de la segunda junta |
| INNUJO | Byte | Incremento en el número de junta |
| ESTAR | Byte | Estado de carga |
| LOH | Byte | Para grabar sin espacios |
| JLOAD | Byte | Contador |
| CONTAX | Byte | Contador |
| FX | Real | Fuerza aplicada en dirección X |
| FY | Real | Fuerza aplicada en dirección Y |
| FZ | Real | Fuerza aplicada en dirección Z |
| MX | Real | Momento aplicado alrededor del eje X |
| MY | Real | Momento aplicado alrededor del eje Y |
| MZ | Real | Momento aplicado alrededor del eje Z |

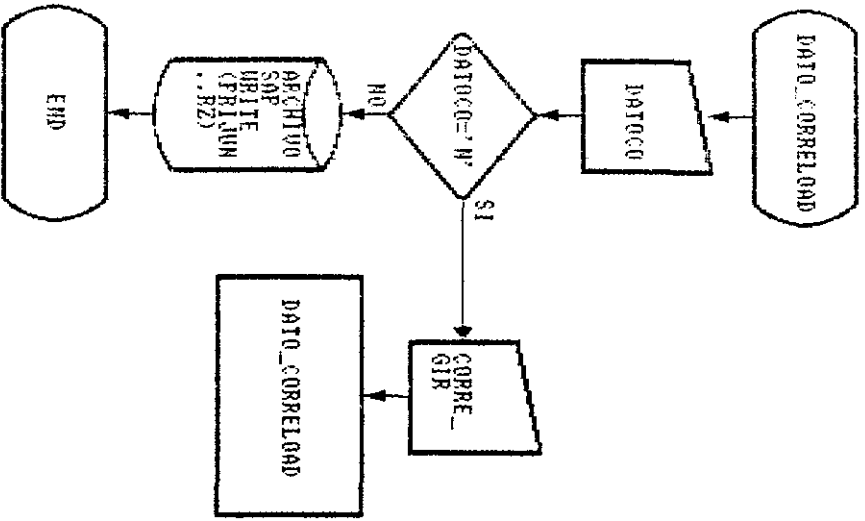
| | | |
|-----------|--------|--|
| | | aplicado en dirección X |
| DX | Real | Desplazamiento aplicado en dirección X |
| | | aplicado en dirección Y |
| DY | Real | Desplazamiento aplicado en dirección Y |
| | | aplicado en dirección Z |
| EX | Real | Rotación aplicada alrededor del eje X |
| EY | Real | Rotación aplicada alrededor del eje Y |
| EZ | Real | Rotación aplicada alrededor del eje Z |
| FILENAMEX | String | Para grabar |
| FILENAMEY | String | Para grabar |
| FILENAMEZ | String | Para grabar |
| FILENAMEX | String | Para grabar |
| FILEY | String | Para grabar |
| FILEZ | String | Para grabar |
| FILEX | String | Para grabar |
| FILEY | String | Para grabar |
| FILEZ | String | Para grabar |

| | | |
|-------|---------|--------------|
| ITANI | CHITING | Para Grahari |
| ITANI | CHITING | Para Grahari |
| ITANI | CHITING | Para Grahari |
| ITANI | CHITING | Para Grahari |
| ITANI | CHITING | Para Grahari |



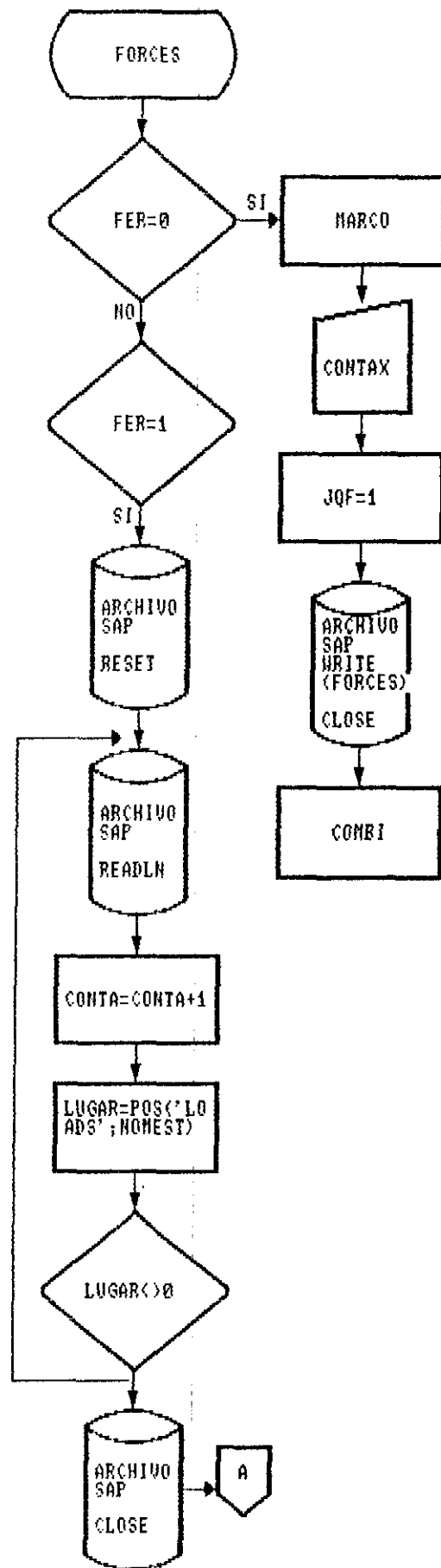


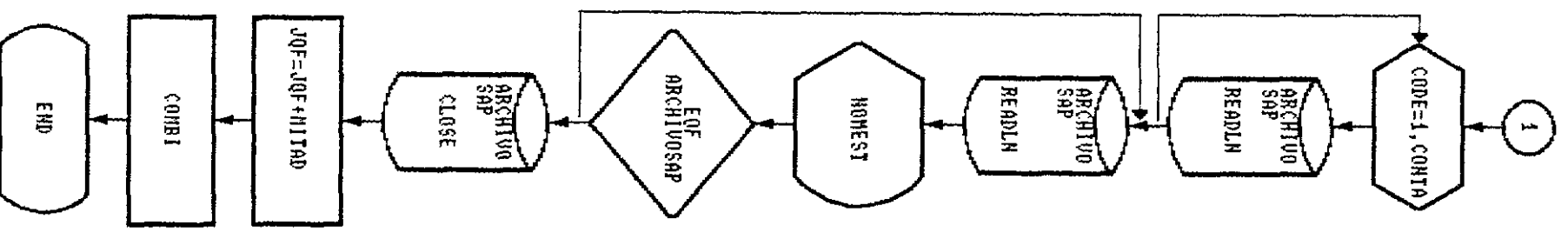
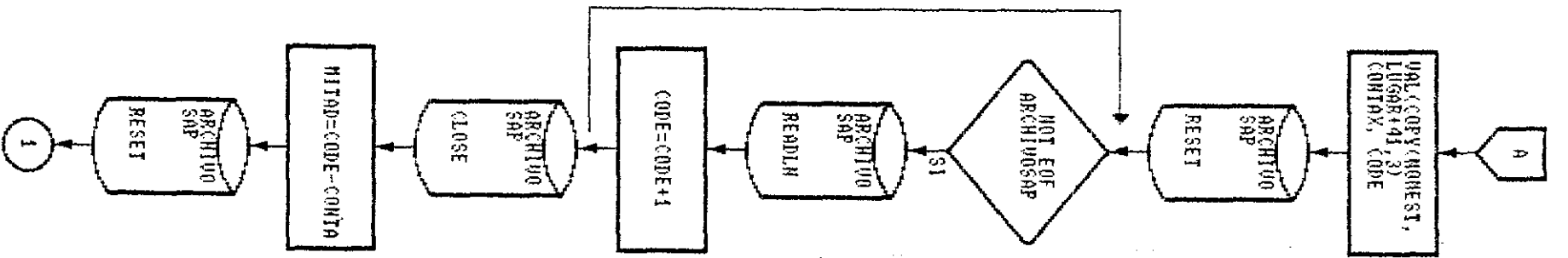


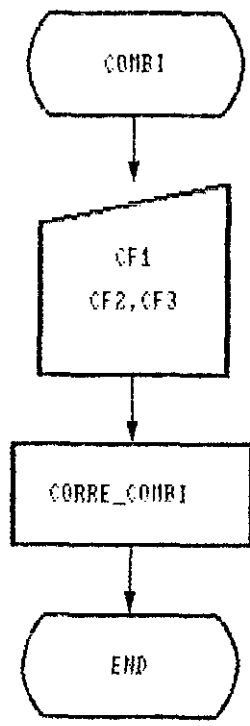


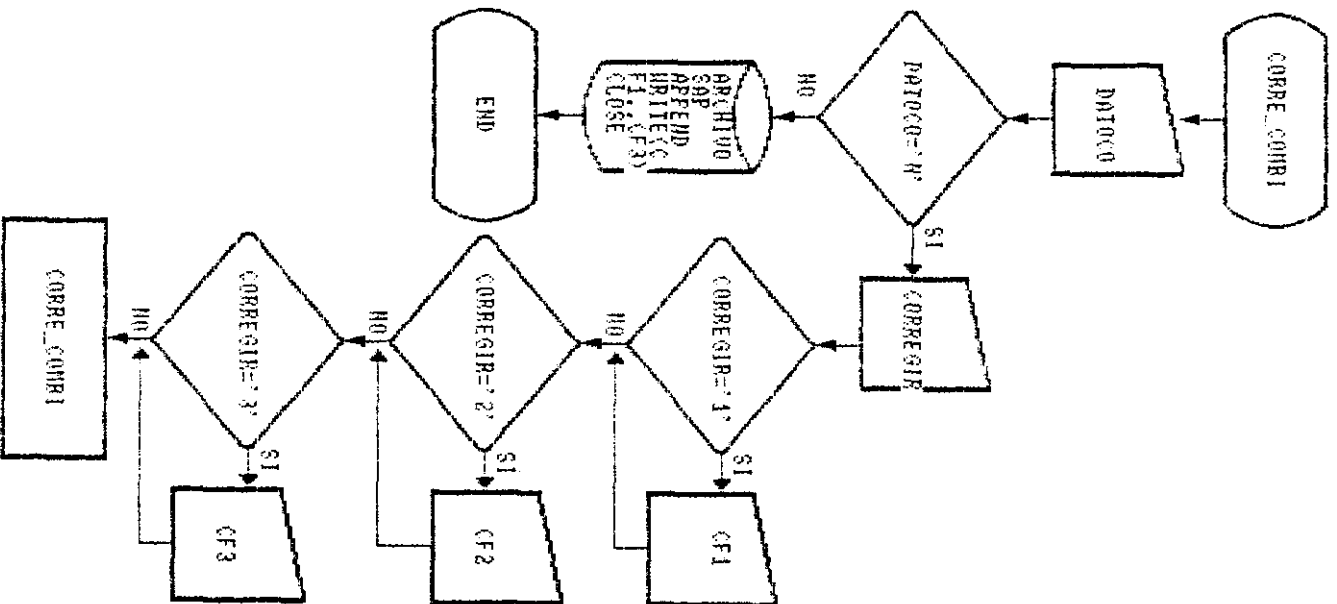
NOBRE DEL PROGRAMA: PUNTO

| NOBRE | TIPO | DESCRIPCION |
|--------|--------|---------------------------|
| QF | Byte | Cantidad de combinaciones |
| QOF | Byte | Centador |
| QOM | Byte | Para grabar sin espacios |
| QOI | Real | Factor de carga 1 |
| QOJ | Real | Factor de carga 2 |
| QOK | Real | Factor de carga 3 |
| QOFA1E | String | Para grabar |
| QOFA2E | String | Para grabar |
| QOFA3E | String | Para grabar |
| QOFA4E | String | Para grabar |



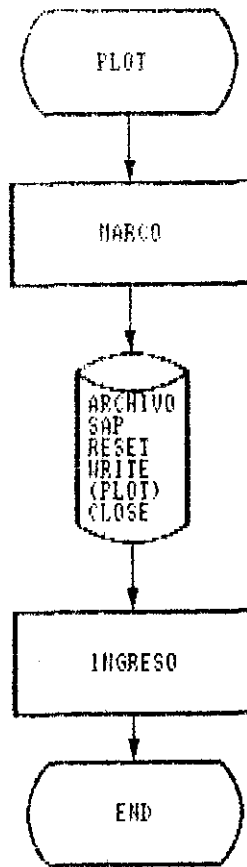


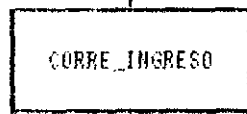
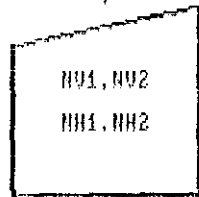


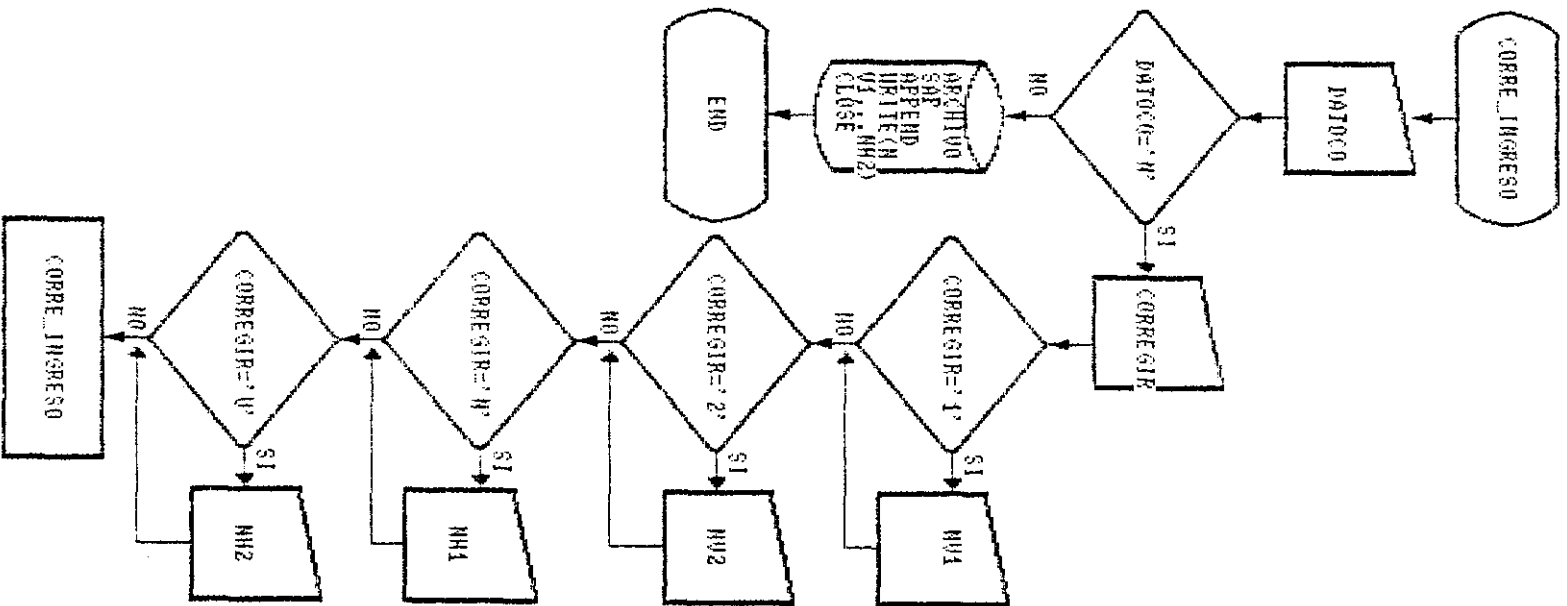


NOMBRE DEL PROGRAMA: E107

| NOMBRE | TIPO | DESCRIPCION |
|--------|---------|----------------------------|
| NV1 | Integer | Vector vista nudo1 |
| NVC | Integer | Vector vista nudo2 |
| NU1 | Integer | Vector horizontal nudo1 |
| NU2 | Integer | Vector horizontal nudo2 |







LISTA DE VARIABLES

VARIABLES GLOBALES DE CONSULTAS Y MODIFICACIONES

| NOMBRE | TIPO | DESCRIPCION |
|------------|---------|---|
| L | Byte | Valor del ordinal de Num1 |
| N | Integer | Número de nudos |
| POSICION | Integer | Para variar filas |
| COLUMNA | Integer | Para variar columna |
| STR10 | String | Para aceptar nombre del archivo |
| HOHEST | String | Para leer nombre del archivo |
| ARCHIVOSAP | Text | Variable de archivo |
| ARCHAUX | Text | Variable de archivo |
| DATOCO | Char | Para aceptar dato correcto S/N |
| CORREGIR | Char | El dato a corregir |
| SH | Char | Ayuda |
| LUGAR | Byte | Para la posición de lectura en el archivo |
| COPE | Byte | Variable para leer |
| COPEA | Byte | Variable para leer |

MEMBER

Byte

Leer

Variable

Page

Leer

MEMBER

Byte

Variable

Page

Leer

LISTA DE VARIABLES

| NOMBRE DEL PROGRAMA | TIPO | DESCRIPCION |
|---------------------|---------|-----------------------|
| NUMERO | Byte | Variable de control |
| OTRO | Integer | Variable de control |
| SW | Char | Switch de ayuda |
| REVER | Char | Variable de selección |
| OPCIONES | Char | Variable de opciones |
| CONSULTA | Char | Variable de consulta |
| NUMERO | Integer | Numero de nodos |
| NUMERO | Char | Nombre de archivo |

UNIT MENUCAMB;

INTERFACE

USES CRT,CARATULA,MENU,CAMBIOS1,CAMBIOS2;
PROCEDURE MENUCAMB1;

IMPLEMENTATION

PROCEDURE MENUCAMB1;

TYPE

 STR10:STRING#100;

VAR

 NOMBRE:STR10;

 CONS:CHAR;

 SITUA:INTEGER;

 SM,MUEVE,ELECCION:CHAR;

 FER:BYTE;

 N:INTEGER;

LABEL 50;

PROCEDURE OPCIONES;

BEGIN

 GOTOXY(22,10);WRITE(' Nombre de la Estructura');
 GOTOXY(22,11);WRITE(' Control de salida de Información');
 GOTOXY(22,12);WRITE(' Coordenadas de los Nudos');
 GOTOXY(22,13);WRITE(' Restricciones de los Nudos');
 GOTOXY(22,14);WRITE(' Apoyos Especiales');
 GOTOXY(22,15);WRITE(' Concatenación de Nudos');
 GOTOXY(22,16);WRITE(' Características de los Miembros');
 GOTOXY(22,17);WRITE(' B.Cáscaras, Placas y Membranas');
 GOTOXY(22,18);WRITE(' Cargas Concentradas en los Nudos');
 GOTOXY(22,19);WRITE(' Combinación de Cargas (fuerzas)');
 GOTOXY(22,20);WRITE(' Gráfico de la estructura');
 GOTOXY(22,21);WRITE(' Volver al Menú Principal');

END; *DEL PROCEDURE OPCIONES*

PROCEDURE CASE_OPCIONES;

BEGIN

 CASE SITUA OF

 10: BEGIN

 TEXTCOLOR(14);

 GOTOXY(22,10);WRITE(' Nombre de la Estructura');

 GOTOXY(16,23);WRITE('Consulta y/o Modifica el nombre de la estructura');

 TEXTCOLOR(7);

 END;

 11: BEGIN

 TEXTCOLOR(14);

 GOTOXY(22,11);WRITE(' Control de salida de Información');

 GOTOXY(17,23);WRITE('Consulta la forma de salida de la información');

 TEXTCOLOR(7);

 END;

 12: BEGIN

 TEXTCOLOR(14);

 GOTOXY(22,12);WRITE(' Coordenadas de los Nudos');

 GOTOXY(15,23);WRITE('Consulta y/o Modifica el ingreso de las coordenadas');

 TEXTCOLOR(7);

 END;

 13: BEGIN

 TEXTCOLOR(14);

 GOTOXY(22,13);WRITE(' Restricciones de los Nudos');

```

        GOTOXY(11,23);WRITE(' Consulta y/o Modifica las restricciones de los nudos deseados');
        TEXTCOLOR(7);
    END;
14: BEGIN
        TEXTCOLOR(14);
        GOTOXY(22,14);WRITE('      Apoyos Especiales');
        GOTOXY(10,23);WRITE(' Consulta y/o Modifica los datos ingresados en apoyos especiales');
        TEXTCOLOR(7);
    END;
15: BEGIN
        TEXTCOLOR(14);
        GOTOXY(22,15);WRITE('      Concatenación de Nudos');
        GOTOXY(13,23);WRITE(' Consulta y/o Modifica las concatenaciones de los nudos');
        TEXTCOLOR(7);
    END;
16: BEGIN
        TEXTCOLOR(14);
        GOTOXY(22,16);WRITE(' Características de los Miembros');
        GOTOXY(11,23);WRITE(' Consulta y/o Modifica las características dadas a los miembros');
        TEXTCOLOR(7);
    END;
17: BEGIN
        TEXTCOLOR(14);
        GOTOXY(22,17);WRITE('      B.Cáscaras, Placas y Membranas');
        GOTOXY(8,23);WRITE(' Consulta y/o Modifica las características de los materiales empleados');
        TEXTCOLOR(7);
    END;
18: BEGIN
        TEXTCOLOR(14);
        GOTOXY(22,18);WRITE(' Cargas Concentradas en los Nudos');
        GOTOXY(11,23);WRITE(' Consulta y/o Modifica la información de las cargas en los nudos');
        TEXTCOLOR(7);
    END;
19: BEGIN
        TEXTCOLOR(14);
        GOTOXY(22,19);WRITE('      Combinación de Cargas (fuerzas)');
        GOTOXY(16,23);WRITE(' Consulta y/o Modifica las fuerzas ingresadas');
        TEXTCOLOR(7);
    END;
20: BEGIN
        TEXTCOLOR(14);
        GOTOXY(22,20);WRITE('      Gráfico de la estructura');
        GOTOXY(10,23);WRITE(' Consulta y/o Modifica los datos para el gráfico de la estructura');
        TEXTCOLOR(7);
    END;
21: BEGIN
        TEXTCOLOR(14);
        GOTOXY(22,21);WRITE('      Volver al Menú Principal');
        GOTOXY(26,23);WRITE(' Retorna al Menú Principal');
        TEXTCOLOR(7);
    END;
END; *DEL CASES
END; *DEL PROCEDURE *CASE OPCIONES

PROCEDURE NOMBRE_ARCH(VAR NOMBRE:STR10);
VAR
    ARCHIVOSAP:TEXT;
    swl:char;
    NOMBRE1:STRING6BA;
    UNIDAD:CHAR;
BEGIN

```

```
CLRSR:
CWD90:
GOTOXY(15,9);WRITE(' Ingrese la Unidad de Disco: ');
GOTOXY(15,10);WRITE(' Nombre del Archivo      ');
TEXTCLR(14);
GOTOXY(15,8);WRITE(' ');GOTOXY(26,8);WRITE('U');GOTOXY(36,9);WRITE('D');
GOTOXY(15,10);WRITE(' ');GOTOXY(26,10);WRITE('A');
TEXTCLR(7);
SWI='N';
WHILE (SWI='N') OR (SWI='n') DO
  BEGIN
    REPEAT
      TEXTCLR(14);
      GOTOXY(49,8);WRITE(' ');
      TEXTCLR(7);
      GOTOXY(5,22);WRITE(' La unidad de disco puede ser: A; B; C; D; ');
      TEXTCLR(14);
      GOTOXY(8,22);WRITE('U');GOTOXY(18,22);WRITE('D');GOTOXY(35,22);WRITE('A; B; C; ');
      GOTOXY(50,22);WRITE('P; ');
      GOTOXY(49,8);UNIDAD:=RANDOMKEY;
      TEXTCLR(7);
      CASE UNIDAD OF
        'A','B','C','D':a,b,c,d;
      BEGIN
        TEXTCLR(14);
        GOTOXY(49,8);WRITE(UNIDAD);
        TEXTCLR(7);
      END;
    END;
  END;
  UNIT (UNIDAD='A') OR (UNIDAD='B') OR (UNIDAD='C') OR (UNIDAD='D') OR (UNIDAD='U') OR (UNIDAD='D') OR (UNIDAD='C') OR (UNIDAD='C');
  OR (UNIDAD='D') OR (UNIDAD='D');
  GOTOXY(5,22);CLRDL;
  GOTOXY(5,23);CLRDL;
  GOTOXY(25,23);WRITE(' Su Ingreso es Correcto S/N: ');
  TEXTCLR(14);
  GOTOXY(48,23);WRITE('S');GOTOXY(50,23);WRITE('N');
  GOTOXY(53,23);SWI:=RANDOMKEY;
  TEXTCLR(7);
  END;
  WRITE (SWI='N') OR (SWI='n') DO
  BEGIN
    GOTOXY(5,23);WRITE('El nombre del archivo debe tener máximo 8 caracteres');
    TEXTCLR(14);
    GOTOXY(49,10);WRITE(' ');
    GOTOXY(8,23);WRITE('N');GOTOXY(19,23);WRITE('A');GOTOXY(43,23);WRITE('B;C');
    GOTOXY(49,10);READLN(MOMPE1);
    TEXTCLR(7);
    GOTOXY(5,22);CLRDL;
    GOTOXY(5,23);CLRDL;
    GOTOXY(25,23);WRITE(' Su Ingreso es Correcto S/N: ');
    TEXTCLR(14);
    GOTOXY(48,23);WRITE('S');GOTOXY(50,23);WRITE('N');
    GOTOXY(53,23);SWI:=RANDOMKEY;
    TEXTCLR(7);
  END;
  WRITE (SWI='N');
  END;
  REPEAT(ARCHIVOSAP);
  $$$
```

```

IF (RESULT <> 0) THEN
  BEGIN
    REPEAT
      TEXTCOLOR(14);
      GOTXY(5,23);WRITE('ARCHIVO NO ENCONTRADO...PRESIONE UNA TECLA E INGRESE OTRO NOMBRE DE ARCHIVO');
      TEXTCOLOR(7);
      UNTIL KEYPRESSED;
      NOMBRE_ARCHIVO(NOMBRE);
    END;
  END;
endif;

BEGIN AND LANGUOSA
  WHILE TRUE DO
    BEGIN
      SW:=5;
      WHILE (SW='S') OR (SW='E') DO
        BEGIN
          CLEAR;
          TEXTCOLOR(14);
          FOR CO:=1 TO RO DO
            BEGIN
              GOTXY(CO,1); WRITE('='); GOTXY(CO,24);WRITE('=');
            END;
          FOR FI:=1 TO 24 DO
            BEGIN
              GOTXY(1,FI);WRITE(' '); GOTXY(RO,FI);WRITE(' ');
            END;
          GOTXY(1,1);WRITE(' '); GOTXY(RO,1);WRITE(' ');
          GOTXY(1,24);WRITE(' '); GOTXY(RO,24);WRITE(' ');
          GOTXY(1,2);WRITE('FONITELIA UNIVERSIDAD CATOLICA DEL ECUADOR');
          GOTXY(1,3);WRITE('PROGRAMAS ACADMICOS AMBATO');
          GOTXY(1,4);WRITE('ANALISIS DE ESTRUCTURAS');
          GOTXY(1,5);WRITE('MENU DE CAMBIOS Y CONSULTAS');
          TEXTCOLOR(7);
          FOR CO:=25 TO 35 DO
            BEGIN
              GOTXY(CO,6); WRITE('='); GOTXY(CO,8);WRITE('=');
            END;
          FOR FI:=6 TO 8 DO
            BEGIN
              GOTXY(25,FI);WRITE(' '); GOTXY(35,FI);WRITE(' ');
            END;
          GOTXY(25,6);WRITE(' '); GOTXY(35,6);WRITE(' ');
          GOTXY(25,8);WRITE(' '); GOTXY(35,8);WRITE(' ');
          TEXTCOLOR(14);
          TEXTCOLOR(LIGHTGRAY);GOTXY(33,7);WRITE('O P C I O N E S');
          TEXTCOLOR(14);
          FOR CO:=28 TO 60 DO
            BEGIN
              GOTXY(CO,9); WRITE('='); GOTXY(CO,22);WRITE('=');
            END;
          FOR FI:=9 TO 22 DO
            BEGIN
              GOTXY(28,FI);WRITE(' '); GOTXY(60,FI);WRITE(' ');
            END;
          GOTXY(28,9); WRITE(' '); GOTXY(60,9); WRITE(' ');
          GOTXY(29,22);WRITE(' '); GOTXY(60,22);WRITE(' ');
          TEXTCOLOR(7);
          CLEAR;
          STIM:=10;
          TEXTCOLOR(14);

```



```

CHR(13) : CONSUSIS(NOMBRE,N);
END; DEL CASE
ELECCION:='N';SW:='S';
END;
IF (SITUA=12) AND (MUEVE=CHR(013)) THEN
BEGIN
  NOMBRE_ARCH(NOMBRE);
  GOTOXY(5,23);CLREOL;
  GOTOXY(20,23);WRITE('DESEA CONSULTAR O MODIFICAR : ');
  TEXTCOLOR(14);
  GOTOXY(26,23);WRITE('C');
  GOTOXY(30,23);WRITE('M');
  TEXTCOLOR(7);
  GOTOXY(50,23);CONS := READKEY;
  fer:=0;
  CASE CONS OF
    'C','c' : BEGIN
      FER:=1;CONSJOINT(NOMBRE,FER,N);
      END;
    'M','m' : BEGIN
      FER:=2;CONSJOINT1(NOMBRE,FER,N);
      END;
  END; DEL CASE
  ELECCION:='N';SW:='S';
END;
IF (SITUA=13) AND (MUEVE=CHR(013)) THEN
BEGIN
  NOMBRE_ARCH(NOMBRE);
  GOTOXY(5,23);CLREOL;
  GOTOXY(20,23);WRITE('DESEA CONSULTAR O MODIFICAR : ');
  TEXTCOLOR(14);
  GOTOXY(26,23);WRITE('C');
  GOTOXY(30,23);WRITE('M');
  TEXTCOLOR(7);
  GOTOXY(50,23);CONS := READKEY;
  fer:=0;
  CASE CONS OF
    'C','c' : BEGIN
      FER:=1;CONSUREST(NOMBRE,FER,N);
      END;
    'M','m' : BEGIN
      FER:=2;CONSUREST(NOMBRE,FER,N);
      END;
  END; DEL CASE
  ELECCION:='N';SW:='S';
END;
IF (SITUA=14) AND (MUEVE=CHR(013)) THEN
BEGIN
  NOMBRE_ARCH(NOMBRE);
  GOTOXY(5,23);CLREOL;
  GOTOXY(20,23);WRITE('DESEA CONSULTAR O MODIFICAR : ');
  TEXTCOLOR(14);
  GOTOXY(26,23);WRITE('C');
  GOTOXY(30,23);WRITE('M');
  TEXTCOLOR(7);
  GOTOXY(50,23);CONS := READKEY;
  fer:=0;
  CASE CONS OF
    'C','c' : BEGIN
      FER:=1;CONSPRINGS(NOMBRE,FER);
      END;

```

```

'N', 'n' : BEGIN
    FER:=2;CONSFRINGS(NOMBRE,FER);
    END;
END; #DEL CASES
ELECTION:='N';SN:='9';
END;
IF (SITUA=15) AND (MUEVE=CHR(013)) THEN
BEGIN
    NOMBRE ARCH(NOMBRE);
    GOTOXY(5,23);CLREOL;
    GOTOXY(20,23);WRITE('DESEA CONSULTAR O MODIFICAR : █');
    TEXTCOLOR(14);
    GOTOXY(26,23);WRITE('C');
    GOTOXY(30,23);WRITE('N');
    TEXTCOLOR(7);
    GOTOXY(50,23);CONS := READKEY;
    (c:=A;
CASE CONS OF
    'C', 'c' : BEGIN
        FER:=1;NOCONSTRAINTS(NOMBRE,FER);
        END;
    'N', 'n' : BEGIN
        FER:=2;NOCONSTRAINTS(NOMBRE,FER);
        END;
END; #DEL CASES
ELECTION:='N';SN:='5';
END;
IF (SITUA=16) AND (MUEVE=CHR(013)) THEN
BEGIN
    NOMBRE ARCH(NOMBRE);
    GOTOXY(5,23);CLREOL;
    CLRCR;
    CUADRO;
    GOTOXY(20,8); WRITE('1.- PROPIEDADES Y CARGOS DE MIEMBROS');
    GOTOXY(20,10);WRITE('2.- CARACTERISTICAS DE LOS MIEMBROS');
    TEXTCOLOR(14);
    GOTOXY(20,8);WRITE('1');GOTOXY(20,10);WRITE('2');
    GOTOXY(23,20);WRITE('CONSULTA O MODIFICACION EN : █');
    TEXTCOLOR(7);
    GOTOXY(50,20);CONS:=READKEY;
CASE CONS OF
    '1' : BEGIN
        GOTOXY(23,23);WRITE('UD. SOLAMENTE PUEDE CONSULTAR : █');
        TEXTCOLOR(14);
        GOTOXY(43,23);WRITE('C');
        TEXTCOLOR(7);
        GOTOXY(53,23);CONS := READKEY;
CASE CONS OF
    'C', 'c' : CONSUMNL(NOMBRE);
    CHR(13) : CONSUMNL(NOMBRE);
END; #DEL CASES
END;
    '2' : BEGIN
        GOTOXY(20,23);WRITE('DESEA CONSULTAR O MODIFICAR : █');
        TEXTCOLOR(14);
        GOTOXY(26,23);WRITE('C');
        GOTOXY(30,23);WRITE('N');
        TEXTCOLOR(7);
        GOTOXY(50,23);CONS := READKEY;
CASE CONS OF
    'C', 'c' : BEGIN

```

```
FER:=1;CONSULFRAN(NOMBRE,FER,NM,NL);
```

```
END;
```

```
IF '0' > BEGIN
```

```
FER:=2;CONSULFRAN(NOMBRE,FER,NM,NL);
```

```
END;
```

```
END; *OCL CASE#
```

```
END;
```

```
IF (STIUM=17) AND (MOVE=CHR(013)) THEN
```

```
END;
```

```
IF (STIUM=17) AND (MOVE=CHR(013)) THEN  
BEGIN
```

```
  NOMBRE_ARCH(NOMBRE);
```

```
  GOTOXY(5,23);CLREOL;
```

```
  GOTOXY(20,23);WRITE('DESEA CONSULTAR O MODIFICAR : ');
```

```
  TEXTCOLOR(14);
```

```
  GOTOXY(26,23);WRITE('C');
```

```
  GOTOXY(38,23);WRITE('N');
```

```
  TEXTCOLOR(7);
```

```
  GOTOXY(50,23);CONS := READKEY;
```

```
  FER:=0;
```

```
  CASE CONS OF
```

```
    'C', 'c' : BEGIN
```

```
      FER:=1;CONSULQAD(NOMBRE,FER);
```

```
    END;
```

```
    'N', 'n' : BEGIN
```

```
      FER:=2;CONSULQAD(NOMBRE,FER);
```

```
    END;
```

```
  END; *OCL CASE#
```

```
  ELECTIOM:= 'N';SM:= 'S';
```

```
END;
```

```
IF (STIUM=18) AND (MOVE=CHR(013)) THEN
```

```
BEGIN
```

```
  NOMBRE_ARCH(NOMBRE);
```

```
  GOTOXY(5,23);CLREOL;
```

```
  GOTOXY(20,23);WRITE('DESEA CONSULTAR O MODIFICAR : ');
```

```
  TEXTCOLOR(14);
```

```
  GOTOXY(26,23);WRITE('C');
```

```
  GOTOXY(38,23);WRITE('N');
```

```
  TEXTCOLOR(7);
```

```
  GOTOXY(50,23);CONS := READKEY;
```

```
  FER:=0;
```

```
  CASE CONS OF
```

```
    'C', 'c' : BEGIN
```

```
      FER:=1;CONSULQAD(NOMBRE,FER);
```

```
    END;
```

```
    'N', 'n' : BEGIN
```

```
      FER:=2;CONSULQAD(NOMBRE,FER);
```

```
    END;
```

```
  END; *OCL CASE#
```

```
  ELECTIOM:= 'N';SM:= 'S';
```

```
END;
```

```
IF (STIUM=19) AND (MOVE=CHR(013)) THEN
```

```
BEGIN
```

```
  NOMBRE_ARCH(NOMBRE);
```

```
  GOTOXY(5,23);CLREOL;
```

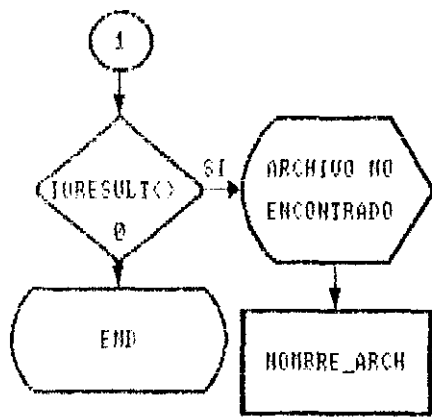
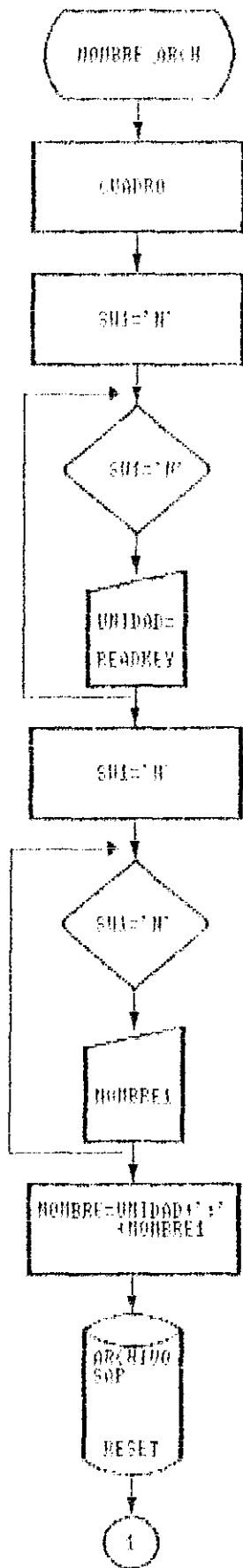
```
  GOTOXY(20,23);WRITE('DESEA CONSULTAR O MODIFICAR : ');
```

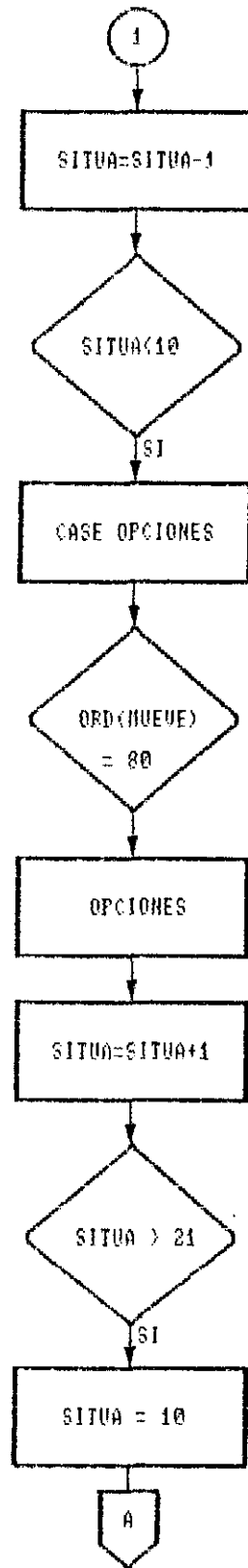
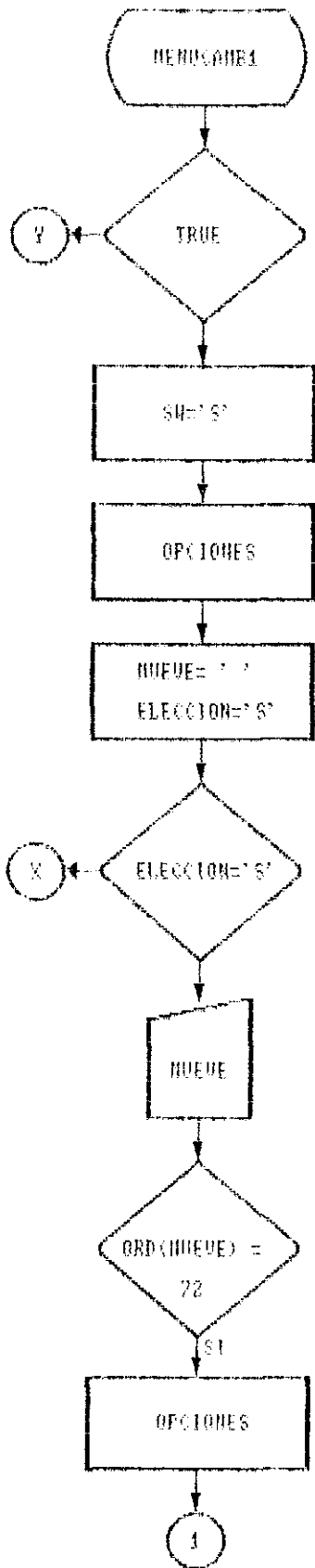
```
  TEXTCOLOR(14);
```

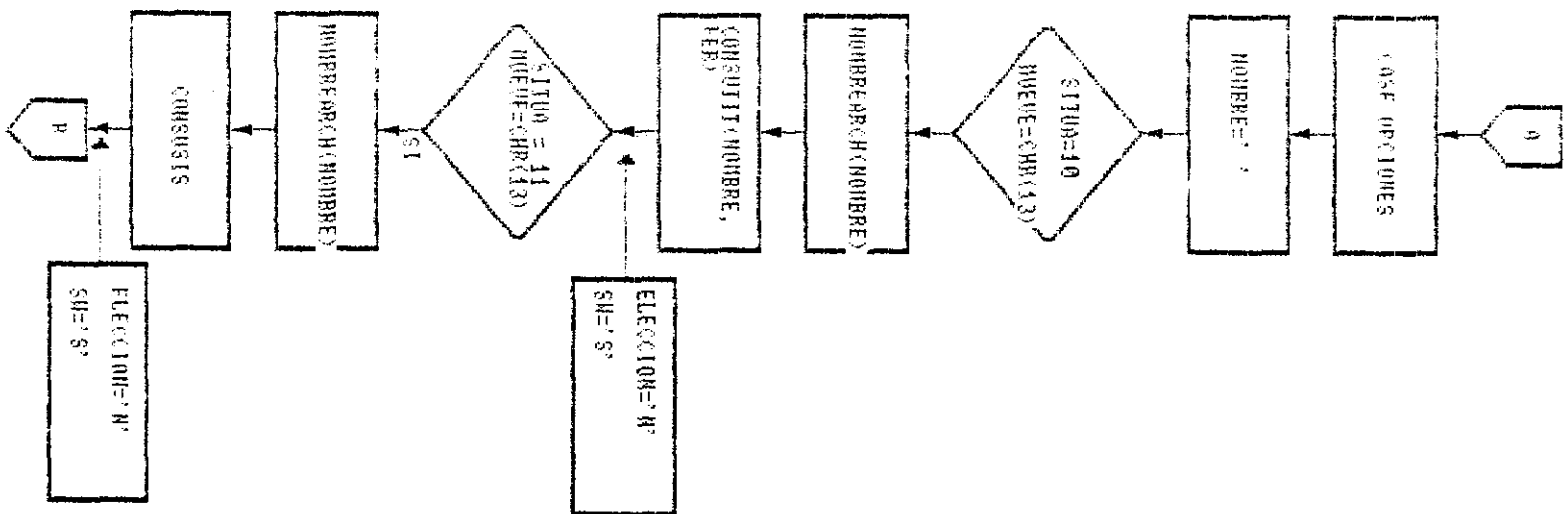
```
  GOTOXY(26,23);WRITE('C');
```

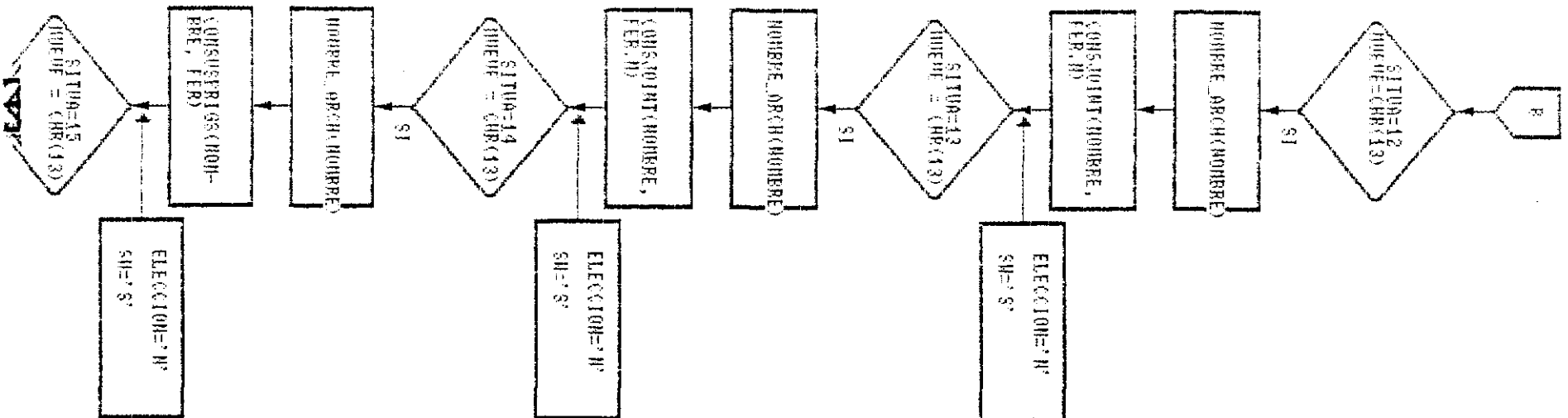
```
  GOTOXY(38,23);WRITE('N');
```

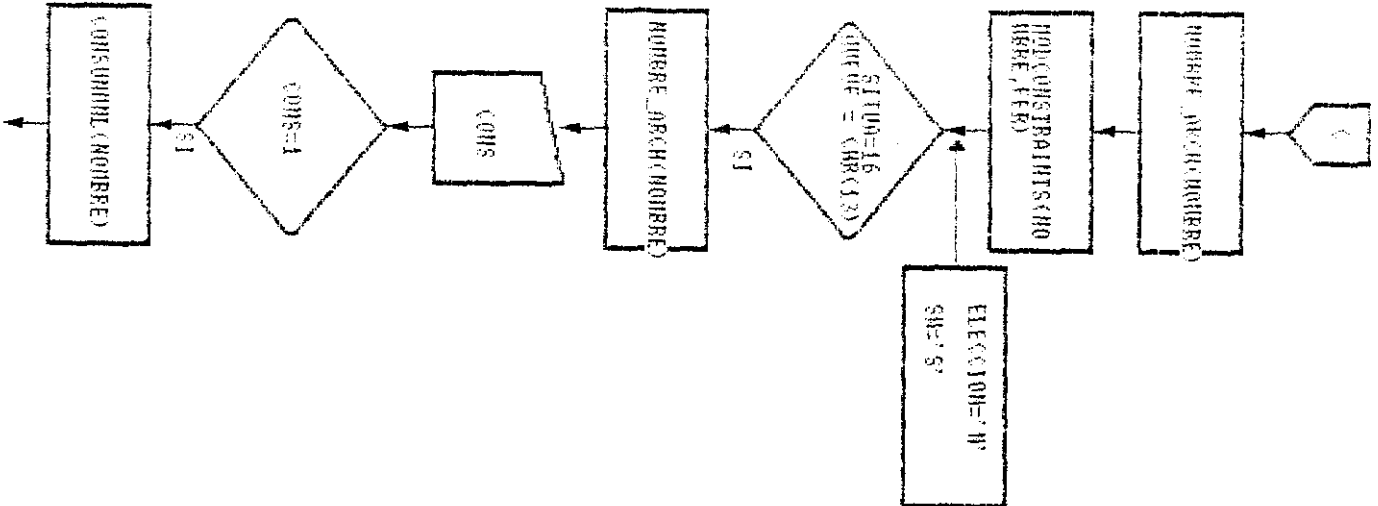
```
  TEXTCOLOR(7);
```

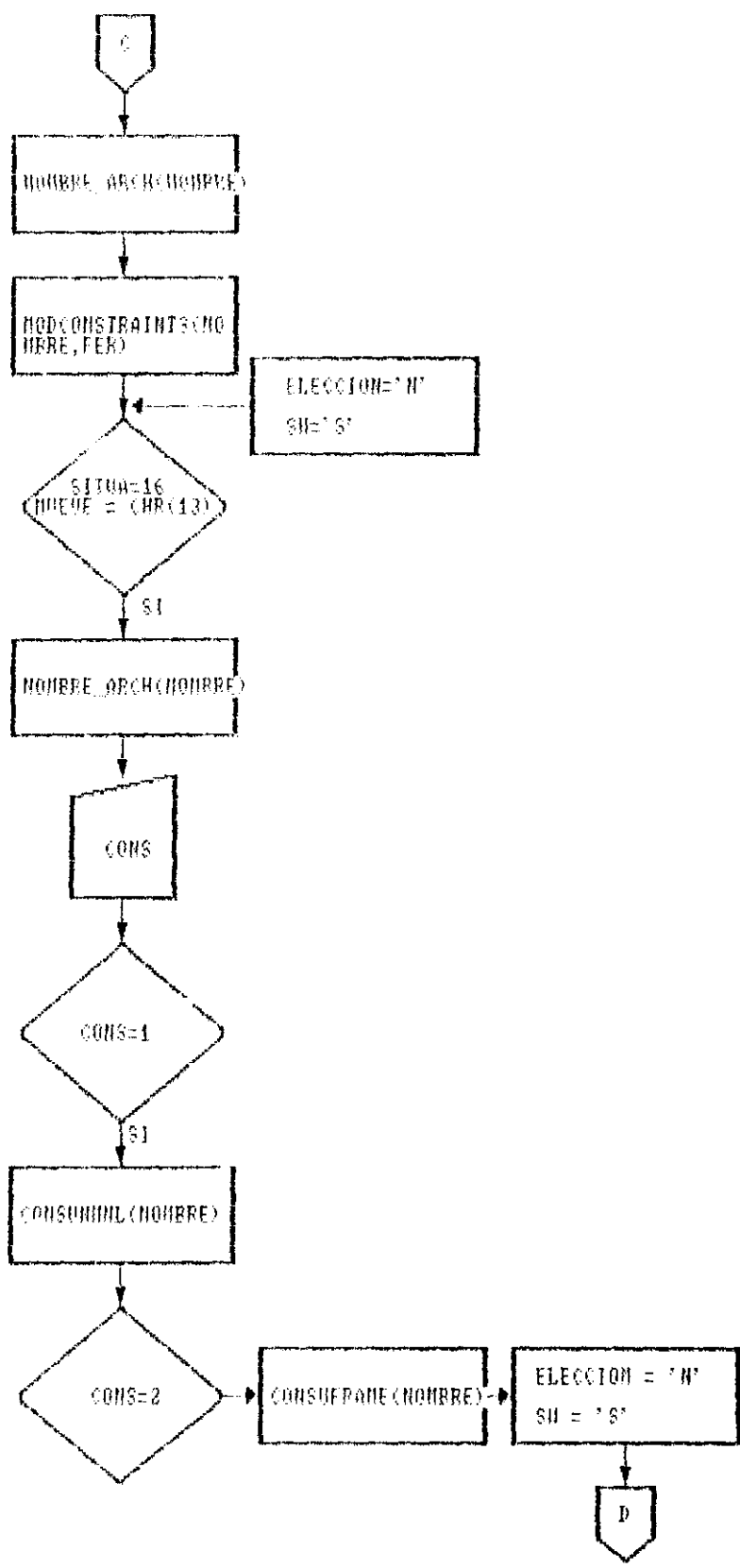



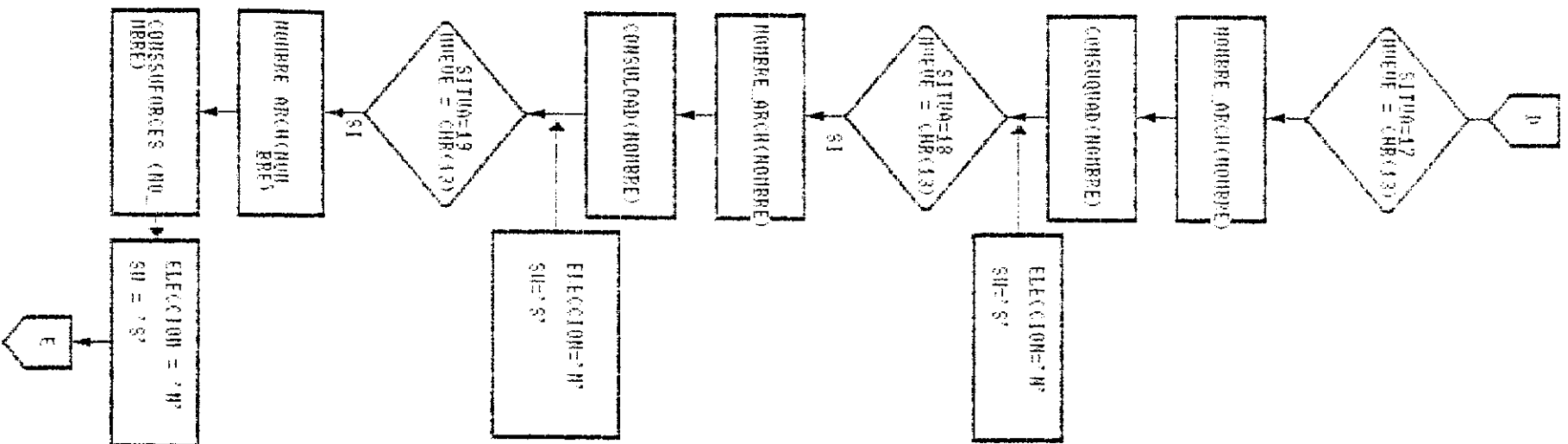


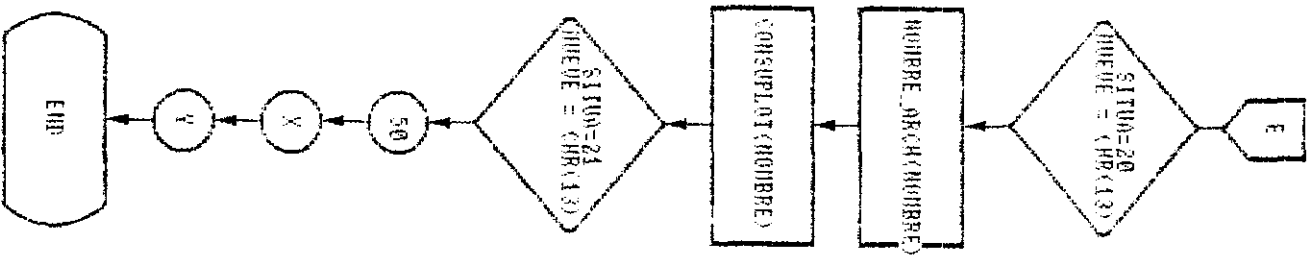












```
UNIT (AMPRES);
INTERFACE
Uses Crt, Uniton, Caratula;
TYPE
  *STR0=STRING(44);
VAR
```

```
  NUMERO, CODE, POSICION, COLUMNA: INTEGER;
  LUGAR: BYTE;
  NOMBRE: STRING(80);
  ARCHIVO: TEXT;
  DATUM: LOGEER(8, SW: 100);
  M1, A01, CONT, A001, A11, A0: BYTE;
  A0, H0, A1, T1, T2;
  M0, M1, M2, M3: STRING;
```

```
PROCEDURE LEER;
PROCEDURE ENTER;
PROCEDURE CONSULTA(MONTE:ESTR0;VAR M:ESTR0);
PROCEDURE CONSULTA(MONTE:ESTR0;VAR M:ESTR0);
CONSULTE (CONSULTA(MONTE:ESTR0;VAR M:ESTR0);
PROCEDURE CONSULTA(MONTE:ESTR0;VAR M:ESTR0);
PROCEDURE CONSULTA(MONTE:ESTR0;VAR M:ESTR0);
PROCEDURE CONSULTA(MONTE:ESTR0;VAR M:ESTR0);
CONSULTE (CONSULTA(MONTE:ESTR0;VAR M:ESTR0);
```

```
IMPLEMENTATION
```

```
PROCEDURE LEER;
BEGIN
  ASIGNAR(ARCHIVO, AUXILIO);
  REWRITER(ARCHIVO);
  LOGEER(CALDA);
END; *DEL PROCEDURE ENTER;
```

```
PROCEDURE ENTER;
BEGIN
  FECHA
  FECHA1
  (CONVY(18,23)WRITER( Fecha con letra y fecha al MENU ANTERIOR );
  TEXTO(OR(3));
  CONVY(27,23);WRITER(UNA FECHA);
  (CONVY(30,23);WRITER(MENU ANTERIOR);
  TEXTO(OR(7));
  WRITER(FEYESSECA);
```

```
END; *DEL PROCEDURE ENTER;
```

```
PROCEDURE CONSULTA;
VAR
```

```
  L1:LOGEER(304);
```

```
  SW:MMAR;
```

```
  PROC * De la Procedura L1, esta
```

```
  IF FECHA THEN
```

```
  BEGIN
```

```
    ASIGNAR(ARCHIVO, NOMBRE);
```

```
    RESETER(ARCHIVO);
```

```
    RESETER
```

```
    RESETER(ARCHIVO, NOMBRE);
```

```
    LUGAR:=FIRST(TITULO); *NOMBRE);
```

```
    (WRITER(LUGAR));
```

```
    (LOGEER(ARCHIVO));
```

```
    NOMBRE:=CONVY(NOMBRE, 10, SW);
```

```
    CUADRO;
```

```
    CONVY(13,23);WRITER(Numero de la estructura);
```

```
    TEXTO(OR(14));
```

11 FEB-2 AMB

RECIB

SW=77;

WRITE (SW=N) DP (SW=77) BR

RECIB

CLASDE: CUADRO

RECIBY(5,20)WRITE('Para el hombre de la Estructura debe ingresar opción 30 caracteres');

RECIBO(4,14);

RECIBY(1,20)WRITE('N');RECIBY(26,20)WRITE('E');;RECIBY(26,20)WRITE('0');

RECIBO(6,7);

RECIBY(1,14)WRITE('Nombre de la estructura');

RECIBY(15,14)WRITE('.....');

RECIBO(10,14);

RECIBY(30,2)WRITE('Estructura');

RECIBY(5,24)WRITE('Estructura ingreso es correcto S/N: ');

RECIBY(28,25)WRITE('E');

RECIBY(30,24)WRITE('E');

RECIBO(6,7);

RECIBY(35,24)WRITE('E');

END;

FFAB;

ASSIGN(ARCHIVAR,'ARCHIVAR');

DEFEND(ARCHIVAR);

WRITE(ARCHIVAR,'11-FEB-2, 11111101');

ASSIGN(ARCHIVOSAP,'ARCHIVOS');

RESET(ARCHIVOSAP);

READ(ARCHIVOSAP,'ARCHIVOS');

LEPAR;

IF SW=77(ARCHIVOSAP,'ARCHIVOS');

WRITE(ARCHIVOSAP,'E');

UNTIL EOF(ARCHIVOSAP);

CLOSE (ARCHIVOSAP);

CLOSE (ARCHIVAR);

ASSIGN(ARCHIVOSAP,'ARCHIVOS');

WRITE(ARCHIVOSAP);

CLOSE(ARCHIVOSAP);

APPEND(ARCHIVOSAP);

ASSIGN(ARCHIVOSAP,'ARCHIVOS');

RESET(ARCHIVOSAP);

WRITE(ARCHIVOSAP,'00');

RECIB;

READ(ARCHIVOSAP,'ARCHIVOS');

WRITE(ARCHIVOSAP,'ARCHIVOS');

END;

FFAB;

ASSIGN(ARCHIVOSAP);

DEFEND(ARCHIVOSAP);

END; *IF FER=2

ASSIGN(ARCHIVOSAP,'ARCHIVOS');

WRITE(ARCHIVOSAP);

CLOSE(ARCHIVOSAP);

END; *ELSE FER=1*

END; *OBTIENDE INFORMACION*

FFAB;

FFAB;

FFAB;

FFAB;

FFAB; *OBTIENDE INFORMACION*

FFAB;

FFAB;

FFAB;


```

END; *End Case#
DATA CORRECTO(FPBITON,ADRI);
END *End If Data Correcto#
ELSE
BEGIN
  ASSIGN(ARCHIVOS@,NOMBRE);
  APPEND(ARCHIVOS@);
  NU@O@UX:= ' ':LON:=@;
  STR(N@O,NU@O@UX);
  LON:=LENGTH(NU@O@UX);
  WHILE LON<=3 DO
  BEGIN
    NU@O@UX:=NU@O@UX' ':
    lon:=lon+1;
  END;
  X@UX:= ' ':LON:=@;
  STR(X:1,X@UX);
  LON:=LENGTH(X@UX);
  WHILE LON<=6 DO
  BEGIN
    X@UX:=X@UX' ':
    lon:=lon+1;
  END;
  Y@UX:= ' ':LON:=@;
  STR(Y:1,Y@UX);
  LON:=LENGTH(Y@UX);
  WHILE LON<=6 DO
  BEGIN
    Y@UX:=Y@UX' ':
    lon:=lon+1;
  END;
  Z@UX:= ' ':LON:=@;
  STR(Z:1,Z@UX);
  LON:=LENGTH(Z@UX);
  WHILE LON<=6 DO
  BEGIN
    Z@UX:=Z@UX' ':
    lon:=lon+1;
  END;
  ASSIGN(ARCH@UX,'AUXILISE');
  APPEND(ARCH@UX);
  if adri=@ then
    writeIn(ARCH@UX,NU@O@UX,' X=',X@UX,' Y=',Y@UX,' Z=',Z@UX);
  if adri=1 then
  begin
    writeIn(ARCH@UX,NU@O@UX,' X=',X@UX,' Y=',Y@UX,' Z=',Z@UX,' S=',S);
    adri:=@;
  end;
  if adri=2 then
  begin
    STR(NI:1,NI@UX);
    STR(NF:1,NF@UX);
    STR(II:1,II@UX);
    writeIn(ARCH@UX,NU@O@UX,' X=',X@UX,' Y=',Y@UX,' Z=',Z@UX,' B=',NI:3,',',NF:3,',',
    adri:=@;
  end;
  if adri=3 then
  begin
    STR(NHI:1,NH@UX);
    STR(NHE:1,NHE@UX);
    STR(NVI:1,NV@UX);

```



```

NAME=
LISTE=
UNTIL LISTR=0;
CALLA=CALLA1;
LISTR=POST(RESTAINTS ; ,NONE S);
CALLA=CALLA1;
UNTIL LISTR=0;

```

```

REPEAT
  RECALM(ARCHIVE, NONE S);
  LISTR=POST(RESTAINTS ; ,NONE S);
  UNTIL LISTR=0;
  LISTE(ARCHIVE S);
  VALUE V=NONE S; UNTIL (V=0) & (NONE);
  ARCHIVE(ARCHIVE S; ,NONE S);
  REST(ARCHIVE S);
  CALLA=0;
  REPEAT

```

```

  RECALM(ARCHIVE S; ,NONE S);
  LISTR=POST(RESTAINTS ; ,NONE S);
  UNTIL LISTR=0;
  LISTE(ARCHIVE S);
  VALUE V=NONE S; UNTIL (V=0) & (NONE);
  ARCHIVE(ARCHIVE S; ,NONE S);
  REST(ARCHIVE S);
  REPEAT

```

```

  RECALM(ARCHIVE S; ,NONE S);
  LISTR=POST(RESTAINTS ; ,NONE S);
  UNTIL LISTR=0;
  LISTE(ARCHIVE S);
  VALUE V=NONE S; UNTIL (V=0) & (NONE);
  ARCHIVE(ARCHIVE S; ,NONE S);
  REST(ARCHIVE S);
  REPEAT

```

```

  RECALM(ARCHIVE S; ,NONE S);
  LISTR=POST(RESTAINTS ; ,NONE S);
  UNTIL LISTR=0;
  LISTE(ARCHIVE S);
  VALUE V=NONE S; UNTIL (V=0) & (NONE);
  ARCHIVE(ARCHIVE S; ,NONE S);
  REST(ARCHIVE S);
  REPEAT

```

```

  RECALM(ARCHIVE S; ,NONE S);
  LISTR=POST(RESTAINTS ; ,NONE S);
  UNTIL LISTR=0;
  LISTE(ARCHIVE S);
  VALUE V=NONE S; UNTIL (V=0) & (NONE);
  ARCHIVE(ARCHIVE S; ,NONE S);
  REST(ARCHIVE S);
  REPEAT

```

```

  RECALM(ARCHIVE S; ,NONE S);
  LISTR=POST(RESTAINTS ; ,NONE S);
  UNTIL LISTR=0;
  LISTE(ARCHIVE S);
  VALUE V=NONE S; UNTIL (V=0) & (NONE);
  ARCHIVE(ARCHIVE S; ,NONE S);
  REST(ARCHIVE S);
  REPEAT

```

```

  RECALM(ARCHIVE S; ,NONE S);
  LISTR=POST(RESTAINTS ; ,NONE S);
  UNTIL LISTR=0;
  LISTE(ARCHIVE S);
  VALUE V=NONE S; UNTIL (V=0) & (NONE);
  ARCHIVE(ARCHIVE S; ,NONE S);
  REST(ARCHIVE S);
  REPEAT

```

```

  RECALM(ARCHIVE S; ,NONE S);
  LISTR=POST(RESTAINTS ; ,NONE S);
  UNTIL LISTR=0;
  LISTE(ARCHIVE S);
  VALUE V=NONE S; UNTIL (V=0) & (NONE);
  ARCHIVE(ARCHIVE S; ,NONE S);
  REST(ARCHIVE S);
  REPEAT

```

```

  RECALM(ARCHIVE S; ,NONE S);
  LISTR=POST(RESTAINTS ; ,NONE S);
  UNTIL LISTR=0;
  LISTE(ARCHIVE S);
  VALUE V=NONE S; UNTIL (V=0) & (NONE);
  ARCHIVE(ARCHIVE S; ,NONE S);
  REST(ARCHIVE S);
  REPEAT

```

```

  RECALM(ARCHIVE S; ,NONE S);
  LISTR=POST(RESTAINTS ; ,NONE S);
  UNTIL LISTR=0;
  LISTE(ARCHIVE S);
  VALUE V=NONE S; UNTIL (V=0) & (NONE);
  ARCHIVE(ARCHIVE S; ,NONE S);
  REST(ARCHIVE S);
  REPEAT

```

```

GO10YY(16,21):WRITE('E S I R I C I O N E S D E L O S M U D O S');
GO10YY(17,21):WRITE('NUMERO TOTAL DE MUDOS INGRESADOS :',N1);
GO10YY(18,21):WRITE('      12 29 INCRE. RESTRICCIONES);
GO10YY(19,21):WRITE('LINEA      MUDO MUDO');WRITELN;
TEXT(08(7));
RECFAT
READR(ARQUIVO)OVAR,NOMBRE);
MIDAD=MIDAD1;
WRITELN('      MIDAD,      ,NOMBRE);
LVAR=105(1,1)DE RESTRICIONES,NOMBRE);
WRITE(LVAR=98;
CLOSE(ARQUIVO)OVAR);
END;

PROCEDURE (ABRIR)O RESTRICIONES (MUDOS);
BEGIN
  GO10YY(2,21):WRITE(
  GO10YY(5,21):WRITE('Se Datos con Corchetes S/M; E');
  TEXT(08(14));
  GO10YY(29,21):WRITE('S);
  GO10YY(31,21):WRITE('N);
  TEXT(08(7));
  GO10YY(34,21):DA(ORIG=REGUARY;
  IF (ORIG= 'N') OR (ORIG= 'R') THEN
    BEGIN
      GO10YY(5,21):WRITE(
      GO10YY(2,21):WRITE('CORCHES:Mudo Inicial, Final; Incremento; Restricciones:R1,R2,R3,R4,R5,R6; E');
      TEXT(08(14));
      GO10YY(16,21):WRITE(');
      GO10YY(25,21):WRITE(');
      GO10YY(33,21):WRITE('R);
      GO10YY(59,21):WRITE('1);
      GO10YY(62,21):WRITE('2);
      GO10YY(65,21):WRITE('3);
      GO10YY(69,21):WRITE('4);
      GO10YY(71,21):WRITE('5);
      GO10YY(74,21):WRITE('6);
      TEXT(08(7));
      GO10YY(77,21):CORCHES=REGUARY;
      CASE CORCHES OF
        '1','Y': BEGIN
          CORUMMA:=10;11:=Fun]de]n((columna, posicion);
        END;
        'F','F': BEGIN
          CORUMMA:=30;12:=Fun]de]n((columna, posicion);
        END;
        'N','n': BEGIN
          CORUMMA:=11;13:=Fun]de]n]de]n((columna, posicion);
        END;
        '1': BEGIN
          CORUMMA:=35;14:=Fun]de]n]de]n((columna, posicion);
        END;
        '2': BEGIN
          CORUMMA:=57;15:=Fun]de]n]de]n((columna, posicion);
        END;
        '3': BEGIN
          CORUMMA:=59;16:=Fun]de]n]de]n((columna, posicion);
        END;
        '4': BEGIN
          CORUMMA:=61;17:=Fun]de]n]de]n((columna, posicion);
        END;

```

```

5.1 BEGIN
    COLINDEX:=6;R6:=FunRes(ColIndex,position);
    FRR:=
        BEGIN
            COLINDEX:=5;R5:=FunRes(ColIndex,position);

```

```

        END; #End Case4
        CORRECTO:=ESTADO(CORRECTO);
    END #End If Path Correcto
ELSE
    BEGIN

```

```

        J1AUX:=1;LON:=0;
        STR1(J1AUX);
        LON:=LENGTH(J1AUX);
        WHILE LON<=3 DO
            BEGIN
                J2AUX:=J1AUX+1;
                LON:=LON+1;
            END;

```

```

        J2AUX:=1;LON:=0;
        STR1(J2AUX);
        LON:=LENGTH(J2AUX);
        WHILE LON<=3 DO
            BEGIN

```

```

                J3AUX:=J2AUX+1;
                LON:=LON+1;
            END;
            J3AUX:=1;LON:=0;
            STR(FUNCIÓN(J3AUX));
            LON:=LENGTH(J3AUX);
            WHILE LON<=3 DO
                BEGIN

```

```

                    J3AUX:=J3AUX+1;
                    LON:=LON+1;
                END;
                STR(R1,R2);STR(R2,R3);STR(R3,R4);
                STR(R4,R5);STR(R5,R6);STR(R6,R7);
                ASSIGN(PROGRAM, AUX1(OR));
                OPEN(ARCHIVO);
                WRITE(ARCHIVO, J1AUX+1;J2AUX+1;J3AUX+1;
                    R1+R2+R3+R4+R5+R6+R7);
                CLOSE(ARCHIVO);
            END; #End DATOS

```

```

END; #End Procedure Paso Correcto

```

```

PROCEDURE RESTRICTION(POSITION:INTEGER);
BEGIN

```

```

    COLINDEX:=10; J1:=Fun1Fun(ColIndex,position);
    COLINDEX:=30; J2:=Fun1Fun(ColIndex,position);
    COLINDEX:=41; FUNRES(R)=Fun1Fun(ColIndex,position);
    TEXTOLINE(14);
    GOJOLY(53,POSITION);WRITE('R= ', , , , , );
    TEXTOLINE(7);

```

```

    COLINDEX:=55;R1:=FunRes(ColIndex,position);
    COLINDEX:=57;R2:=FunRes(ColIndex,position);
    COLINDEX:=59;R3:=FunRes(ColIndex,position);
    COLINDEX:=61;R4:=FunRes(ColIndex,position);
    COLINDEX:=63;R5:=FunRes(ColIndex,position);
    COLINDEX:=65;R6:=FunRes(ColIndex,position);
    CORRECTO:=ESTADO(CORRECTO);

```

```

END; #End Procedure Restricción

```

```

REGIM AFILIACIONAL DE CONDUCTISTAS
IF FF=1 THEN
  BEGIN
    LECTURA:
    ENTER:
  END
ELSE
  BEGIN
    CF FF=2 THEN
      BEGIN
        SA:=15;
        WHILE (SA=15) OR (SA=5) DO
          BEGIN
            LECTURA:
            CONTAS:=0;
            ASIGNA(ARCHIVOSAF,NUMERO);
            RESET(ARCHIVOSAF);
            REPEAT
              READLN(ARCHIVOSAF,NUMERO);
              USAR:=FOS(RESTRAINTS,NUMERO);
              UNTIL USAR < 0;
            REPEAT
              READLN(ARCHIVOSAF,NUMERO);
              USAR:=FOS('C' IN DE RESTRAINTS,NUMERO);
              CONTAS:=CONTAS+1;
              UNTIL USAR<0;
              CLOSE(ARCHIVOSAF);
              GO TO Y(5,7);
            GO TO Y(20,7); WRITE('QUE NO DE LINEA DESEA MODIFICAR : ');
              GO TO Y(54,7); READ(NUMERO);
              IF (NUMERO=0) OR (NUMERO)=(CONTAS) THEN
                BEGIN
                  REPEAT
                    TEXTOR(LETA);
                    GO TO Y(1,21); WRITE('NUMERO NO VALIDO .... PRECIONE UNA TECLA E INGRESE OTRO');
                    TEXTOR(LETA);
                    UNTIL KEYESCAPE;
                    GO TO 5;
                  END;
                  CONTAS:=0;
                  POSICION:=23;AOR:=0;N1;AOL:=0;N1AD:=0;
                  CREAR:
                  ASIGNA(ARCHIVOSAF,NUMERO);
                  RESET(ARCHIVOSAF);
                  REPEAT
                    READLN(ARCHIVOSAF,NUMERO);
                    N1AD:=N1AD+1;
                    USAR:=FOS(RESTRAINTS,NUMERO);
                    UNTIL USAR < 0;
                    CLOSE(ARCHIVOSAF);
                    N1AR:=N1AD+1;
                    ASIGNA(ARCHIVOSAF,NUMERO);
                    RESET(ARCHIVOSAF);
                    ASIGNA(ARCHIVOSAF,AUXILIO);
                    APTEND(ARCHIVOSAF);
                    CONTAS:=1;
                    WHILE CONTAS=N1AD DO
                      BEGIN
                        READLN(ARCHIVOSAF,NUMERO);
                        WRITELN(ARCHIVOSAF,NUMERO);
                      END
                    UNTIL N1AR<N1AD;
                  END
                END;
              END
            END
          END
        END
      END
    END
  END

```

```

CORP=CORP);
END;
CLOSE(ARCHIVO);
CLOSE(ARCHIVOS);
POSITION:=5;
TEXT:=OR(1);
GOLOGY(5,2);WRITE(
RESTRICTIONES);
GOLOGY(5,2);WRITE(
GOLOGY(5,2);WRITE(
ASSIGN(ARCHIVOS,'NOMBRE');
FESL(ARCHIVOS);
FOR CODE=1 TO NITDA DO
  READL(ARCHIVOS,'NOMBRE');
  ASSIGN(ARCHIVO,'ADV106');
  APPEND(ARCHIVO);
NEXT I
READL(ARCHIVOS,'NOMBRE');
WRITE(ARCHIVO,'NOMBRE');
UNTIL EOF(ARCHIVOS);
CLOSE(ARCHIVO);
CLOSE(ARCHIVOS);
WRITE(ARCHIVOS);
ASSIGN(ARCHIVOS,'NOMBRE');
WRITE(ARCHIVOS);
APPEND(ARCHIVOS);
ASSIGN(ARCHIVO,'ADV106');
RESET(ARCHIVO);
WRITE NOT EOF(ARCHIVO) DO
  BEGIN
    READL(ARCHIVO,'NOMBRE');
    WRITE(ARCHIVOS,'NOMBRE');
  END;
END;
CLOSE(ARCHIVOS);
CLOSE(ARCHIVO);
GOLOGY(5,2);WRITE(
GOLOGY(20,1);WRITE( DESA MODIFICAR OTRA LINEA S/N : );
GOLOGY(5,2);WRITE(
FNO: #MILE SM=199
ASSIGN(ARCHIVO,'ADV106');
WRITE(ARCHIVO);
CLOSE(ARCHIVO);
END; #1# FER=28
END; #del PROCEDURE CONSERVA;

PROCEDURE CONSERVA;
248
US,RESF,(5,(GR)RVC);
FN,UN:INTEGER;
LY,LYA,L,ARY,ART,REAL;
FNADY:STRING25;
LABEL 5;

PROCEDURE LECTURA;
RGLM;
ASSIGN(ARCHIVOS,'NOMBRE');
RESET(ARCHIVOS);
COUNT:=0;
REFER;
DEAR,MAQUINAR,POXCT);

```

```
USARI=POS('SPRINGS',MUNEST);
```

```
CONIA=CONIALL;
```

```
UNTIL LUGARCON;
```

```
CLASCR;
```

```
MARCON;
```

```
MTIAD=#;
```

```
TEXTCOLOR(14);
```

```
GOI0XY(22,21);WRITE(VALORVOS,ESPACIAL(5));
```

```
GOI0XY(5,21);WRITE('Numero de pesos ingresados: ');
```

```
GOI0XY(3,20);WRITE('M. Incr. Const. de resortes);
```

```
GOI0XY(3,20);WRITE('Lines 1 2 ');
```

```
TEXTCOLOR(7);
```

```
WRITELN;
```

```
REFERE!
```

```
REGON(MODULOVCOR,MUNEST);
```

```
MTIAD=MTIAD+1;
```

```
WRITELN('MUNEST',MUNEST);
```

```
LUGAR=POS('C.F.M DE SPRINGS',MUNEST);
```

```
UNTIL LUGARCON;
```

```
CLASE(ORBITVOSAP);
```

```
TEXTCOLOR(14);
```

```
GOI0XY(35,20);WRITE(MTIAD-1);
```

```
TEXTCOLOR(7);
```

```
END;
```

```
PROCEDURE DATO SPRINGS(POSICION:INTEGER);
```

```
BEGIN
```

```
GOI0XY(5,21);WRITE!
```

```
GOI0XY(5,21);WRITE('Sus Datos estan Correctos S/N: ');
```

```
TEXTCOLOR(14);
```

```
GOI0XY(31,21);WRITE('S');
```

```
GOI0XY(33,21);WRITE('N');
```

```
TEXTCOLOR(7);
```

```
GOI0XY(37,21);DATOCO:=READEY;
```

```
IF (DATOCO='N') OR (DATOCO='n') THEN
```

```
BEGIN
```

```
GOI0XY(3,21);WRITE('M.Incr. In.Incr. Cons.Ky;Cons.Ky;Cons.Kz;Const.Krx;Const.Kry;Const.Krz: ');
```

```
TEXTCOLOR(14);
```

```
GOI0XY(5,21);WRITE('1');GOI0XY(7,21);WRITE('F');GOI0XY(23,21);WRITE('M');GOI0XY(27,21);WRITE('O');
```

```
GOI0XY(36,21);WRITE('N');GOI0XY(45,21);WRITE('S');GOI0XY(54,21);WRITE('T');
```

```
GOI0XY(63,21);WRITE('A');GOI0XY(72,21);WRITE('E');
```

```
TEXTCOLOR(7);
```

```
GOI0XY(79,21);OPERORS:=READEY;
```

```
CASE CONJECION OF
```

```
'1','F': BEGIN
```

```
COLI0MNA:=7;FUN:=Fun[Inq](colI0MNA,POSICION);
```

```
END;
```

```
'F','F': BEGIN
```

```
COLI0MNA:=14;FUN:=Fun[Inq](colI0MNA,POSICION);
```

```
END;
```

```
'M','N': BEGIN
```

```
COLI0MNA:=30;FUN:=Fun[Inq](colI0MNA,POSICION);
```

```
END;
```

```
'O','O': BEGIN
```

```
COLI0MNA:=30;FUN:=Fun[Inq](colI0MNA,POSICION);
```

```
END;
```

```
'N','N': BEGIN
```

```
COLI0MNA:=41;FUN:=Fun[Inq](colI0MNA,POSICION);
```

```
END;
```

```
'S','S': BEGIN
```

```
COLI0MNA:=19;FUN:=Fun[Inq](colI0MNA,POSICION);
```



```

LECTURA:
CONTI:=0;
ASIGN(ARCHIVOSAP, JHORTEL);
ASESET(ARCHIVOSAP);
REPEAT
  READM(ARCHIVOSAP, MONEST);
  LUGAR:=POS('C' IN DE SPACES, MONEST);
  UNTIL LUGAR < 0;
  REPEAT
    READM(ARCHIVOSAP, MONEST);
    LUGAR:=POS('C' IN DE SPACES, MONEST);
    (CONTI:=CONTI+1);
  UNTIL LUGAR < 0;
  (LISE(ARCHIVOSAP);
  GOTXY(5,21);WRITE(
  GOTXY(20,21);WRITE('QUE NO DE LINEA DEBEA MODIFICAR : E');
  GOTXY(34,21);READM(RESOL);
  IF (NUMERO =0) OR (NUMERO =| QUITA) THEN
    BEGIN
      REPEAT
        TEXTOL(RESOL);
        GOTXY(12,21);WRITE('NUMERO NO VALIDO .... PRESIONE UNA TECLA E INGRESE OTRO');
        TEXTOL(RESOL);
        UNTIL KEYPRESSED;
        GOTD 5;
      END;
    CONTI:=0;
    POSICION:=23;NHR:=9;MITAD:=9;MITAD:=0;
  (CLEAR;
  ASIGN(ARCHIVOSAP, JHORTEL);
  RESET(ARCHIVOSAP);
  REPEAT
    READM(ARCHIVOSAP, MONEST);
    MITAD:=MITAD+1;
    LUGAR:=POS('S' IN DE SPACES, MONEST);
    UNTIL LUGAR < 0;
    (CLOSE(ARCHIVOSAP);
    MITAD:=MITAD+NUMERO-1;
    ASIGN(ARCHIVOSAP, JHORTEL);
    RESE(ARCHIVOSAP);
    ASIGN(ARCHIVOSAP, ABXILVAR);
    APPEND(ARCHIVOS);
    (CODE:=1;
    WHILE CODE<MITAD DO
      BEGIN
        READM(ARCHIVOSAP, MONEST);
        WRITE(ARCHIVOSAP, MONEST);
        (CODE:=CODE+1;
      END;
    (CLOSE(ARCHIVOS);
    (CLOSE(ARCHIVOSAP);
    (TEXTOL(RESOL);
    GOTXY(6,27);WRITE('N. Inic. N.Final Incremento Kx Ky Kz Kxx Kyy Kxz');
    (TEXTOL(RESOL);
    POSICION:=23;
    APOYO(POSICION);
    GOTXY(2,22);WRITE(
    GOTXY(2,27);WRITE(
    ASIGN(ARCHIVOSAP, JHORTEL);
    RESET(ARCHIVOSAP);
    (CODE:=1 TO MITAD); DO

```

```

        FEEDBACK(ARCHIVOSAF, NOMBRE);
        ASSIGN(ARCHWAY, 'ADUILLAR');
        APPEND(ARCHWAY);
        REPEAT
            REPEAT(ARCHIVOSAF, NOMBRE);
            WRITE(ARCHWAY, NOMBRE);
            UNTIL FOR(ARCHIVOSAF);
            CLOSE(ARCHWAY);
            ASSIGN(ARCHIVOSAF, NOMBRE);
            REWRITE(ARCHIVOSAF);
            APPEND(ARCHIVOSAF);
            ASSIGN(ARCHWAY, 'ADUILLAR');
            RESTART(ARCHWAY);
            WHILE NOT FOR(ARCHIVOSAF) DO
                BEGIN
                    FEEDBACK(ARCHWAY, NOMBRE);
                    RESTART(ARCHWAY, NOMBRE);
                END;
            END;
        CLOSE(ARCHIVOSAF);
        CLOSE(ARCHWAY);
        GOXY(15,21);WRITE;
        GOXY(12,21);WRITE('DESEA MONTEAR OTRA LINEA SIN :');
        GOXY(13,21);WRITE(ARCHWAY);
        END; WHILE SIN='S';
        ASSIGN(ARCHWAY, 'ADUILLAR');
        REWRITE(ARCHWAY);
        CLOSE(ARCHWAY);
        END; *IF FER=2;
        END; *DEL ELSE FER=1;
    END; *del procedure CONSTRINGSA

```

```

PROCEDURE MODCONSTRINGSA;
VAR
    MC, JUNTAS, IC, CI, CZ, O3, J1, J5, J6, J11, J2, J3, J4, J5, J6, LON: BYTE;
    PNC, LONC: INTEGER;
    FNCAY, STRINGESA:
        LABEL 5;

```

```

PROCEDURE LEOTURSA;
BEGIN
    ASSIGN(ARCHIVOSAF, NOMBRE);
    RESET(ARCHIVOSAF);
    LONCA:=0;
    REPEAT
        REAR(ARCHIVOSAF, NOMBRE);
        LOGAR:=POS('CONSTRAINTS', NOMBRE);
        CONTA:=CONTA+1;
        UNTIL LOGAR<=0;
        (1) REPR;
        MAPCO;
        AITAM:=0;
        TEXTOLR(1A);
        GOXY(12,2);WRITE('C O N C A T E N A C I O N   D E   N U D O S');
        GOXY(15,3);WRITE('Numero de concatenaciones ingresadas :');
        GOXY(13,5);WRITE(' #   No.   No. Inc.   C = Juntas Dependientes   I = Series de Nudos');
        GOXY(13,6);WRITE('Lin. 1   2');
        WRITEM;
        TEXTCOLR(17);
        REPRAL
        REAR(ARCHIVOSAF, NOMBRE);

```

```

    HITAD:=HITAD+1;
    WRITELN(' ',HITAD,' ',NOMEST);
    LUGAR:=FOS('C F/N DE CONSTRAINTS',NOMEST);
UNTIL LUGAR<0;
CLOSE(ARCHIVOSAP);
TEXTCOLOR(14);
GOTOXY(44,3);WRITE(NITAD-1);
TEXTCOLOR(7);
END;

```

```

PROCEDURE DATO_CONST(POSICION:INTEGER);

```

```

BEGIN

```

```

    GOTOXY(5,21);WRITE(

```

```

    GOTOXY(5,21);WRITE(' Su= Datos estan Correctos S/N: ');

```

```

    TEXTCOLOR(14);

```

```

    GOTOXY(31,21);WRITE(' S');

```

```

    GOTOXY(33,21);WRITE(' N');

```

```

    TEXTCOLOR(7);

```

```

    GOTOXY(36,21);DATULO:=READKEY;

```

```

    IF (DATULO='N') OR (DATULO='n') THEN

```

```

        BEGIN

```

```

            GOTOXY(2,21);WRITE(

```

```

            GOTOXY(2,21);WRITE(' N, In, N, Fi, Inc, Des1, Des2, Des3, Rot4, Rot5, Rot6, N. i1, N. i2, N. i3, N. i4, Ni5, N. i6: ');

```

```

            TEXTCOLOR(14);

```

```

            GOTOXY(4,21);WRITE(' F');GOTOXY(19,21);WRITE(' F');GOTOXY(14,21);WRITE(' C');GOTOXY(16,21);WRITE(' D');

```

```

            GOTOXY(22,21);WRITE(' E');GOTOXY(28,21);WRITE(' S');GOTOXY(31,21);WRITE(' R');GOTOXY(37,21);WRITE(' O');

```

```

            GOTOXY(43,21);WRITE(' I');GOTOXY(49,21);WRITE(' I');GOTOXY(54,21);WRITE(' 2');GOTOXY(59,21);WRITE(' 3');

```

```

            GOTOXY(64,21);WRITE(' 4');GOTOXY(68,21);WRITE(' 5');GOTOXY(73,21);WRITE(' 6');

```

```

            TEXTCOLOR(7);

```

```

            GOTOXY(76,21);DUMBE:=READKEY;

```

```

            CASE DUMBE OF

```

```

                'E', 'e' : BEGIN

```

```

                    GOTOXY(4, POSICION);WRITE(' ');

```

```

                    COLUMNA:=4; FNC:=FunInEnt(columna, posicion);

```

```

                END;

```

```

                'F', 'f' : BEGIN

```

```

                    GOTOXY(11, POSICION);WRITE(' ');

```

```

                    COLUMNA:=11; UNC:=FunInEnt(columna, posicion);

```

```

                END;

```

```

                'C', 'c' : BEGIN

```

```

                    GOTOXY(19, POSICION);WRITE(' ');

```

```

                    COLUMNA:=19; IC:=FunInEnt(columna, posicion);

```

```

                END;

```

```

                'D', 'd' : BEGIN

```

```

                    GOTOXY(26, POSICION);WRITE(' ');

```

```

                    COLUMNA:=26; DC:=FunInEnt(columna, posicion);

```

```

                END;

```

```

                'E', 'e' : BEGIN

```

```

                    GOTOXY(30, POSICION);WRITE(' ');

```

```

                    COLUMNA:=30; EC:=FunInEnt(columna, posicion);

```

```

                END;

```

```

                'S', 's' : BEGIN

```

```

                    GOTOXY(34, POSICION);WRITE(' ');

```

```

                    COLUMNA:=34; SC:=FunInEnt(columna, posicion);

```

```

                END;

```

```

                'R', 'r' : BEGIN

```

```

                    GOTOXY(38, POSICION);WRITE(' ');

```

```

                    COLUMNA:=38; RC:=FunInEnt(columna, posicion);

```

```

                END;

```

```

                'O', 'o' : BEGIN

```

```

                    GOTOXY(42, POSICION);WRITE(' ');

```



```

    COLUMNA:=46; E6:=FuncInEnt(columna, posicion);
    COLUMNA:=53; F6:=FuncInEnt(columna, posicion);
    COLUMNA:=57; I2:=FuncInEnt(columna, posicion);
    COLUMNA:=61; T7:=FuncInEnt(columna, posicion);
    COLUMNA:=65; I4:=FuncInEnt(columna, posicion);
    COLUMNA:=67; I5:=FuncInEnt(columna, posicion);
    COLUMNA:=71; I6:=FuncInEnt(columna, posicion);
    0470 CONTI=CONTI+1;

```

END:=End Procedure;

*PROGRAMA PRINCIPAL

BEGIN

IF P=1 THEN

BEGIN

LECTURA;

ENTRADA;

END

ELSE

BEGIN

IF P=2 THEN

BEGIN

SW:=0;

WHILE (SW=0) OR (SW=1) DO

BEGIN

LECTURA;

CONTA:=0;

ASSIGN(ARCHIVOSAP, NUMERO);

RESET(ARCHIVOSAP);

REPEAT

READLN(ARCHIVOSAP, NOMEST);

LUGAR:=POS('CONSTRAINTS ', NOMEST);

UNTIL LUGAR < 0;

REPEAT

READLN(ARCHIVOSAP, NOMEST);

LUGAR:=POS('FIN DE CONSTRAINTS ', NOMEST);

UNTIL LUGAR < 0;

UNTIL LUGAR < 0;

CLOSE(ARCHIVOSAP);

50 GOTOXY(5, 71);WRITE(' ');

GOTOXY(20, 71);WRITE('LUGAR NO DE LINEA DEBEA MODIFICAR : ');

GOTOXY(54, 71);READ(NUMERO);

IF (NUMERO < 0) OR (NUMERO > LONTA) THEN

BEGIN

REPEAT

TEXT(COLOR(14));

GOTOXY(17, 71);WRITE('NUMERO NO VALIDO ... PRESIONE UNA TECLA E INGRESE OTRO');

TEXT(COLOR(7));

UNTIL KEYPRESSED;

EXIT 5;

END;

CONTA:=0;

POSICION:=23;PORI:=0;MITAD1:=0;MITAD2:=0;

CREAR;

ASSIGN(ARCHIVOSAP, NUMERO);

RESET(ARCHIVOSAP);

REPEAT

READLN(ARCHIVOSAP, NOMEST);

MITAD1:=MITAD1+1;

LUGAR:=POS('CONSTRAINTS ', NOMEST);

UNTIL LUGAR < 0;

CLOSE(ARCHIVOSAP);

```

MITAD:=MITAD/MUNERO I;
ASIGN(ARCHIVOSAF,NOMBRE);
RESET(ARCHIVOSAF);
ASIGN(ARCHAUX,AUXILIAR I);
ATEND(ARCHAUX);
CODE:=1;
WHILE (CODE:=MITAD) DO
BEGIN
  READLN(ARCHIVOSAF,NOMEST);
  WRITELN(ARCHAUX,NOMEST);
  CODE:=7*CODE I;
END;
CLOSE(ARCHAUX);
CLOSE(ARCHIVOSAF);
TEXTI(UBOR(14);
GOTOXY(2,22);WRITE('N,Toic, N.Fin, Incr.  c1 c2 c3 c4 c5 c6  i1 i2 i3 i4 i5 i6');
TEXTI(UBOR(7);
POSITION:=13;
CONCATENAL(UN(POSITION);
GOTOXY(2,22);WRITE('
GOTOXY(2,23);WRITE('
ASIGN(ARCHIVOSAF,NOMBRE);
RESET(ARCHIVOSAF);
FOR CODE:=1 TO MITAD) DO
  READLN(ARCHIVOSAF,NOMEST);
ASIGN(ARCHAUX,'AUXILIAR');
APPEND(ARCHAUX);
REPEAT
  READLN(ARCHIVOSAF,NOMEST);
  WRITELN(ARCHAUX,NOMEST);
UNTIL EOF(ARCHIVOSAF);
CLOSE(ARCHAUX);
CLOSE(ARCHIVOSAF);
ASIGN(ARCHIVOSAF,NOMBRE);
WRITE(ARCHIVOSAF);
APPEND(ARCHIVOSAF);
ASIGN(ARCHAUX,'AUXILIAR');
RESET(ARCHAUX);
WHILE NOT EOF(ARCHAUX) DO
BEGIN
  READLN(ARCHAUX,NOMEST);
  WRITELN(ARCHIVOSAF,NOMEST);
END;
CLOSE(ARCHIVOSAF);
CLOSE(ARCHAUX);
GOTOXY(5,21);WRITE('
GOTOXY(20,21);WRITE('DESEA MODIFICAR OTRA LINEA S/N : ');
GOTOXY(5,21);S8:=READKEY;
END; *MILE S8= S'4
ASIGN(ARCHAUX,'AUXILIAR');
RESET(ARCHAUX);
CLOSE(ARCHAUX);
END; *I) FFR=24
END; *DEL FLESA
END; *DEL PROCEDURE MODCONTAINICA
END; *DE LA UNIDAD (ARQUIT)

```

LISTA DE VARIABLES

NUMERO DEL PROGRAMA OBJETIVO

NUMERO

TIPO

DESCRIPCION

10000

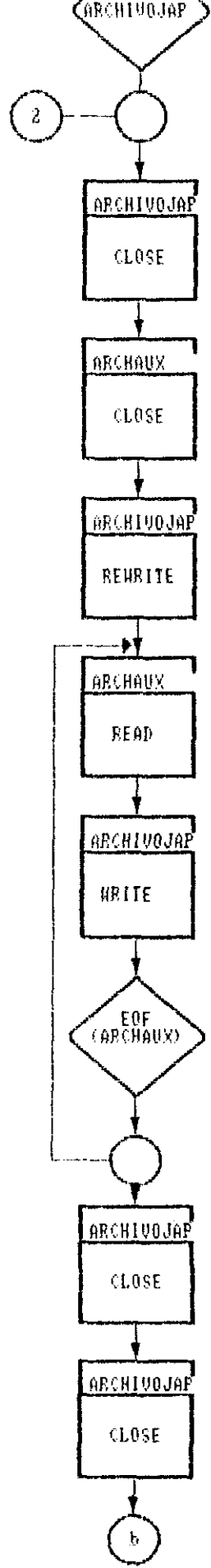
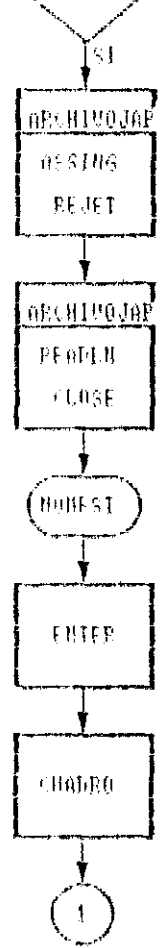
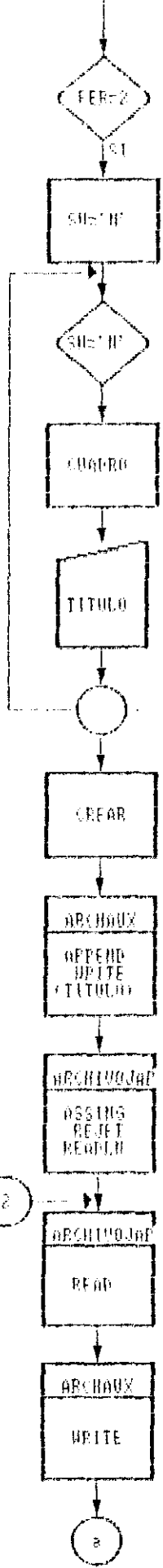
Ma. Ing.

Hambre de la estructura

100

2.1.1.1

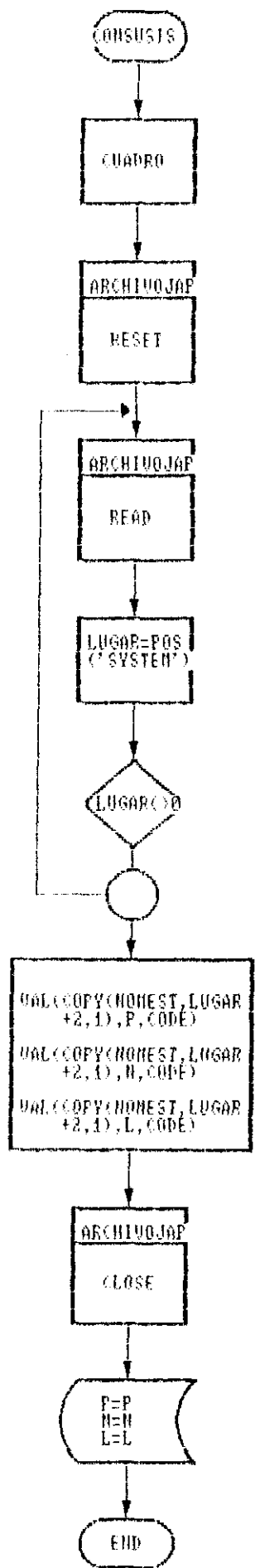
Switch de objeto



LISTA DE VARIABLES

NUMERO DEL PROGRAMA: 00000000

| NOMBRE | TIPO | DESCRIPCION |
|--------|------|------------------|
| F | Byte | Para consultas F |
| L | Byte | Para consultas L |

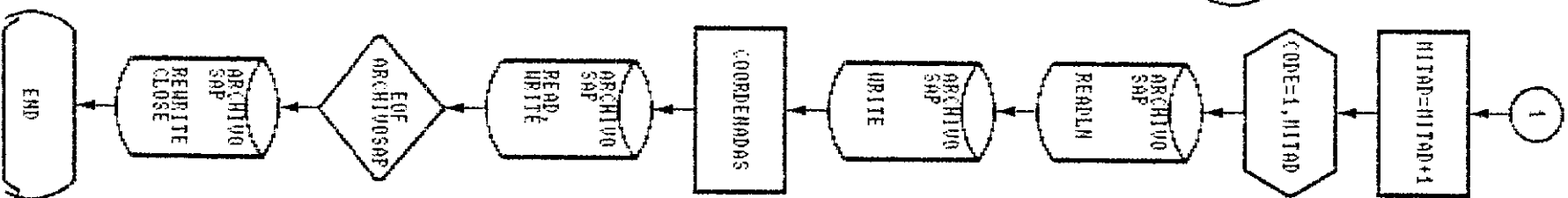
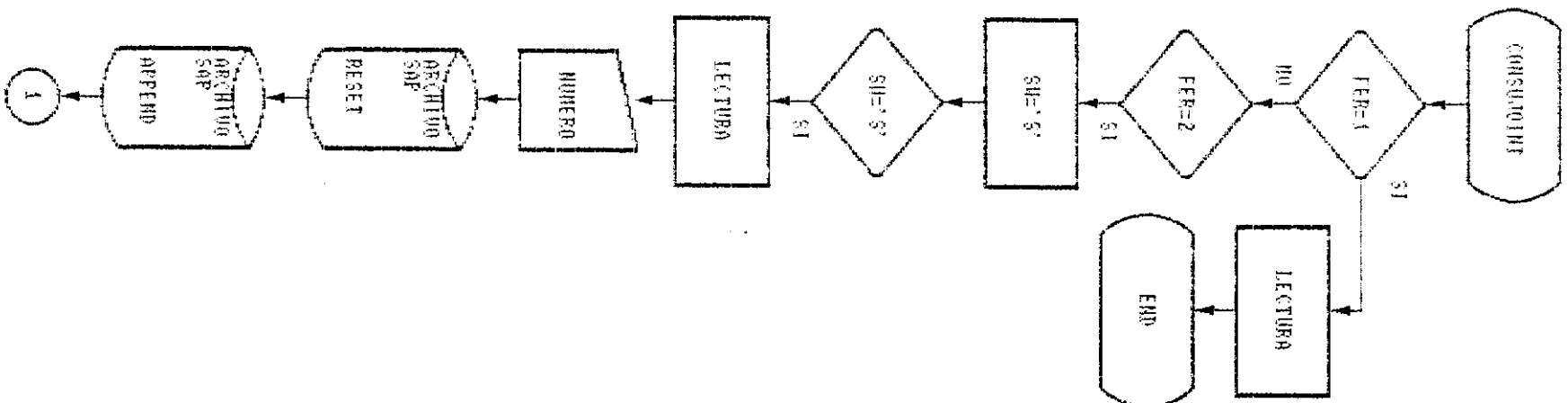


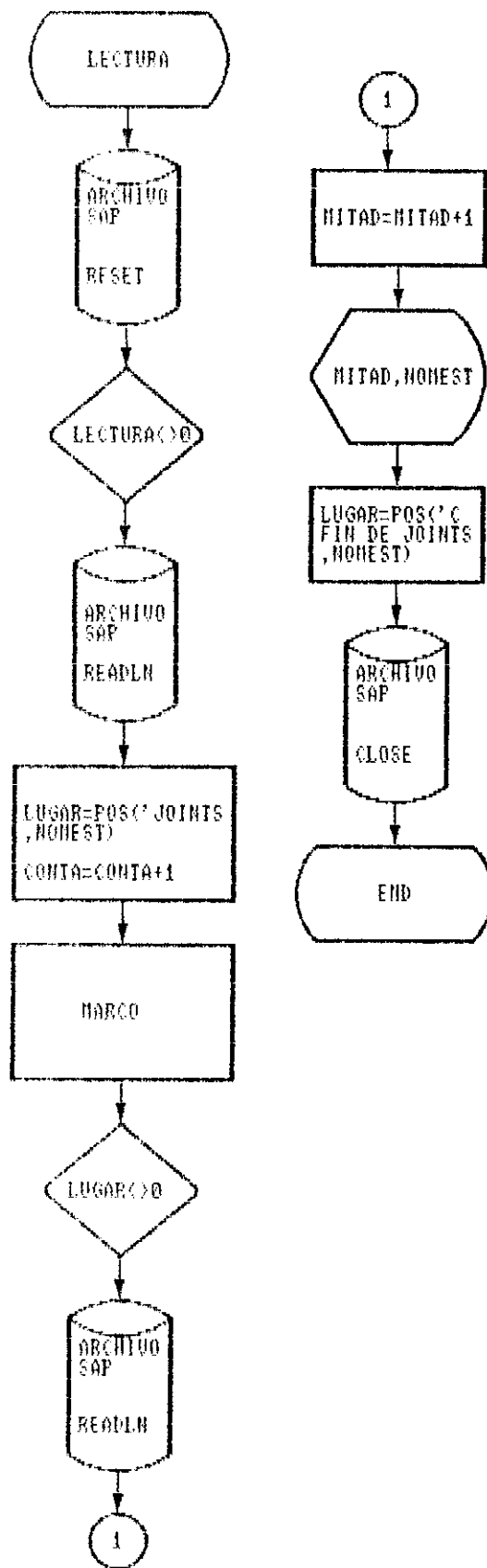
LISTA DE VARIABLES

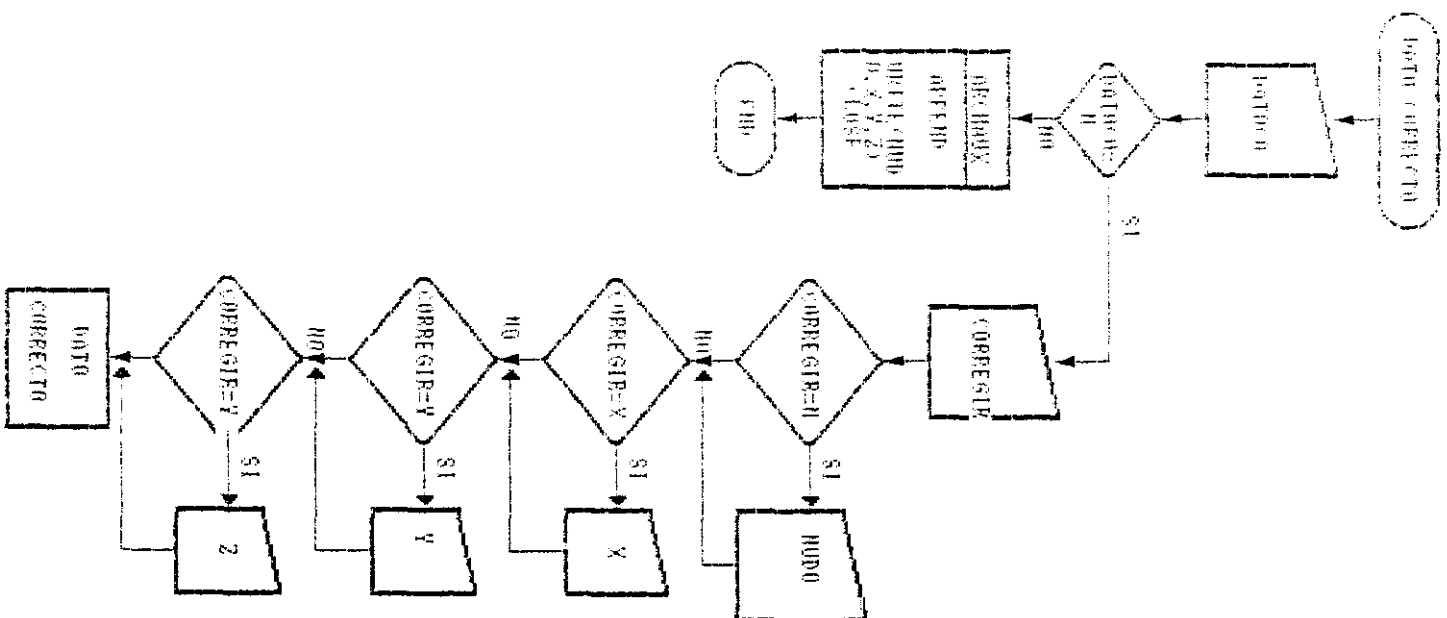
NOMBRE DEL PROGRAMA: COMJOINT

| NOMBRE | TIPO | DESCRIPCION |
|---------|---------|----------------------------|
| INDO | Integer | Número de nodo que ingresa |
| E | Byte | Escala |
| X | Real | Valor de X |
| Y | Real | Valor de Y |
| Z | Real | Valor de Z |
| II | Integer | Nudo Inicial |
| IF | Integer | Nudo Final |
| III | Integer | Nudo Horizontal Inicial |
| III' | Integer | Nudo Horizontal Final |
| IVI | Integer | Nudo Vertical Inicial |
| IV' | Integer | Nudo Vertical Final |
| II | Byte | Incremento Horizontal |
| IV | Byte | Incremento Vertical |
| IL | Byte | Incremento Lineal |
| III/AUX | String | Para guardar en el archivo |
| LOU | Byte | Para guardar en el archivo |
| III/AUX | String | Para guardar en el archivo |

| | | |
|---------|---------|----------------------------|
| HEFAUX | String | Para guardar en el archivo |
| HIHAUX | String | Para guardar en el archivo |
| HIIFAUX | String | Para guardar en el archivo |
| HIJHAUX | String | Para guardar en el archivo |
| HIKHAUX | String | Para guardar en el archivo |
| GENERAR | Char | Para las generaciones |
| IIHAUX | String | Para grabar |
| IJAUX | String | Para grabar |
| IHAUX | String | Para grabar |
| KAUX | String | Para grabar |
| LAUX | String | Para grabar en el archivo |
| MAUX | String | Para grabar en el archivo |
| NOPO | Integer | Para el control de nudos |







NOMBRE DEL PROGRAMA: COPLUNGET

| NOMBRE | TIPO | DESCRIPCION |
|------------|---------|----------------------------|
| J1 | Integer | Primer Nudo |
| J2 | Integer | Segundo Nudo |
| R1 | Byte | Restricciones de los nudos |
| R2 | Byte | Restricciones de los nudos |
| R3 | Byte | Restricciones de los nudos |
| R4 | Byte | Restricciones de los nudos |
| R5 | Byte | Restricciones de los nudos |
| R6 | Byte | Restricciones de los nudos |
| INCREMENTO | Byte | Incremento de los nudos |
| J1AUX | String | Para grabar |
| J2AUX | String | Para grabar |
| L01 | Byte | Para grabar |
| J3AUX | String | Para grabar incremento |
| R11 | String | Para grabar |
| R21 | String | Para grabar |
| R31 | String | Para grabar |
| R41 | String | Para grabar |

111.

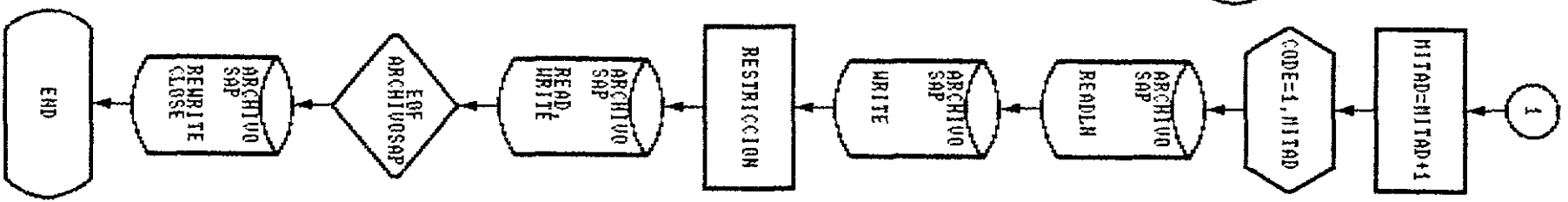
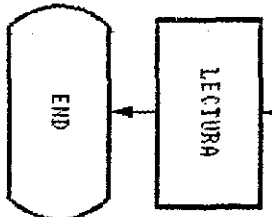
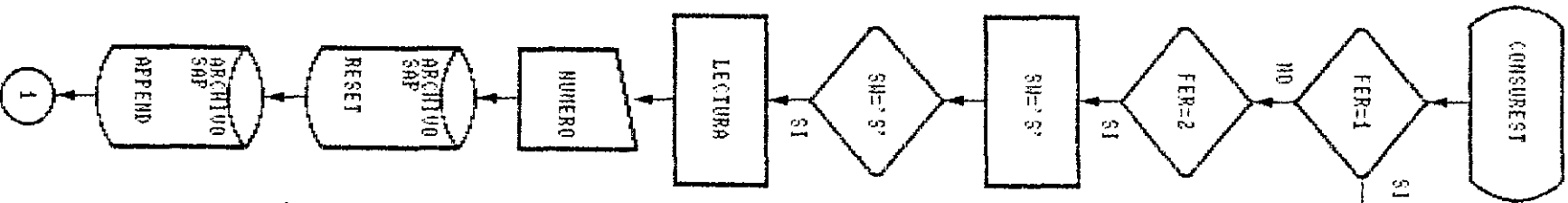
String

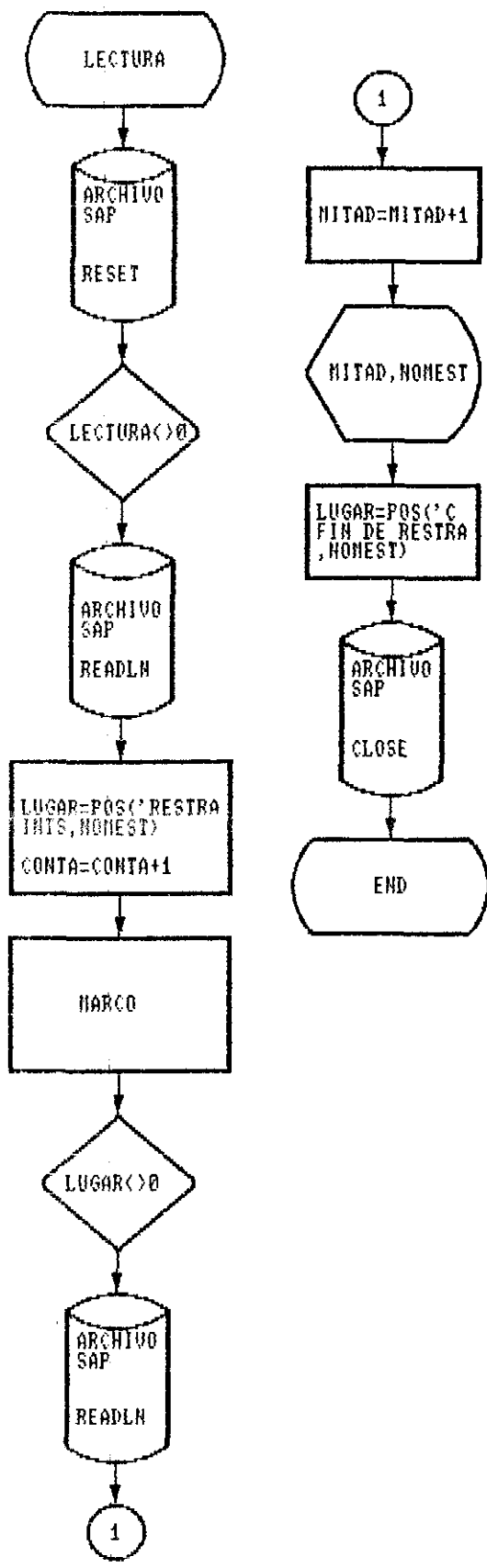
Paragraph

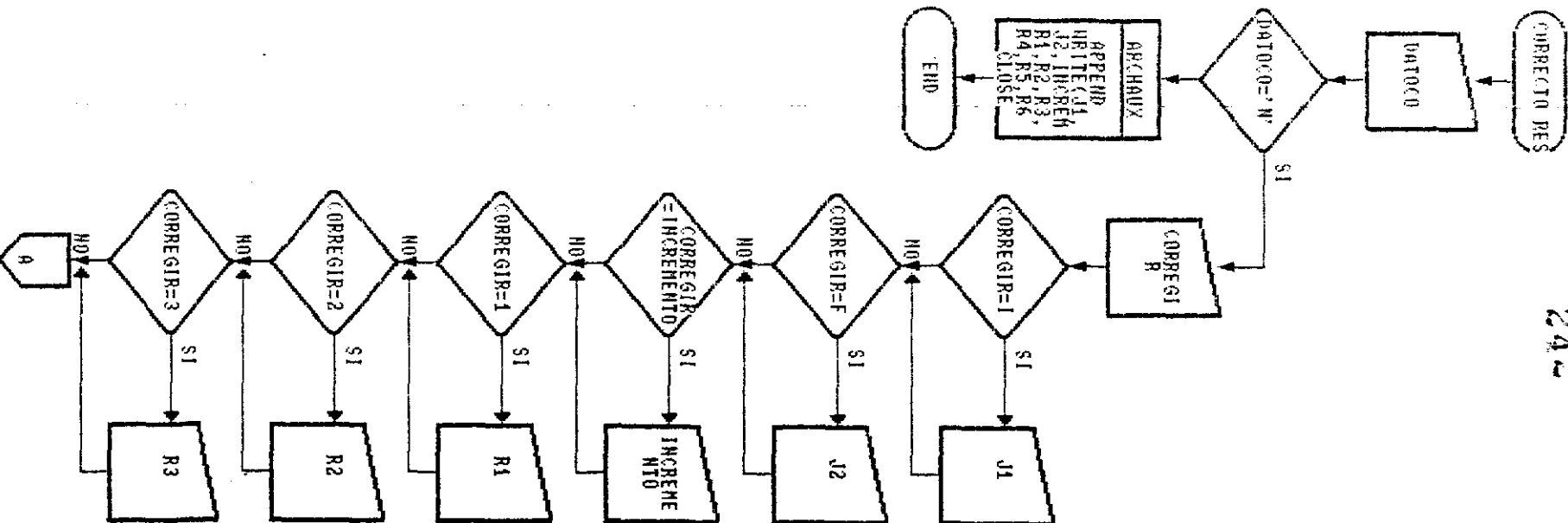
112.

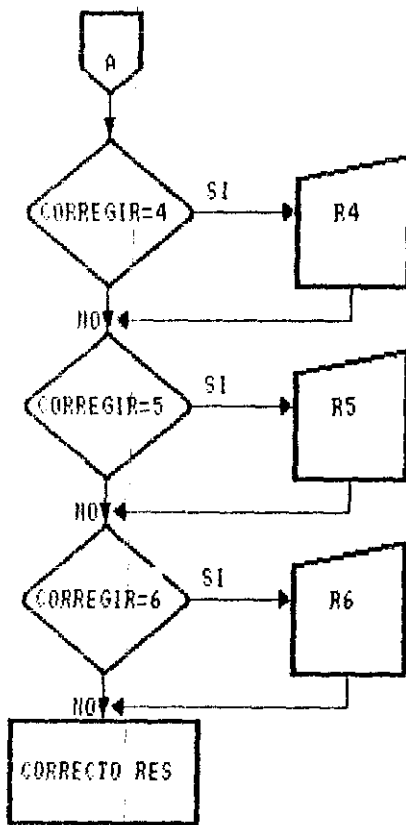
String

Paragraph



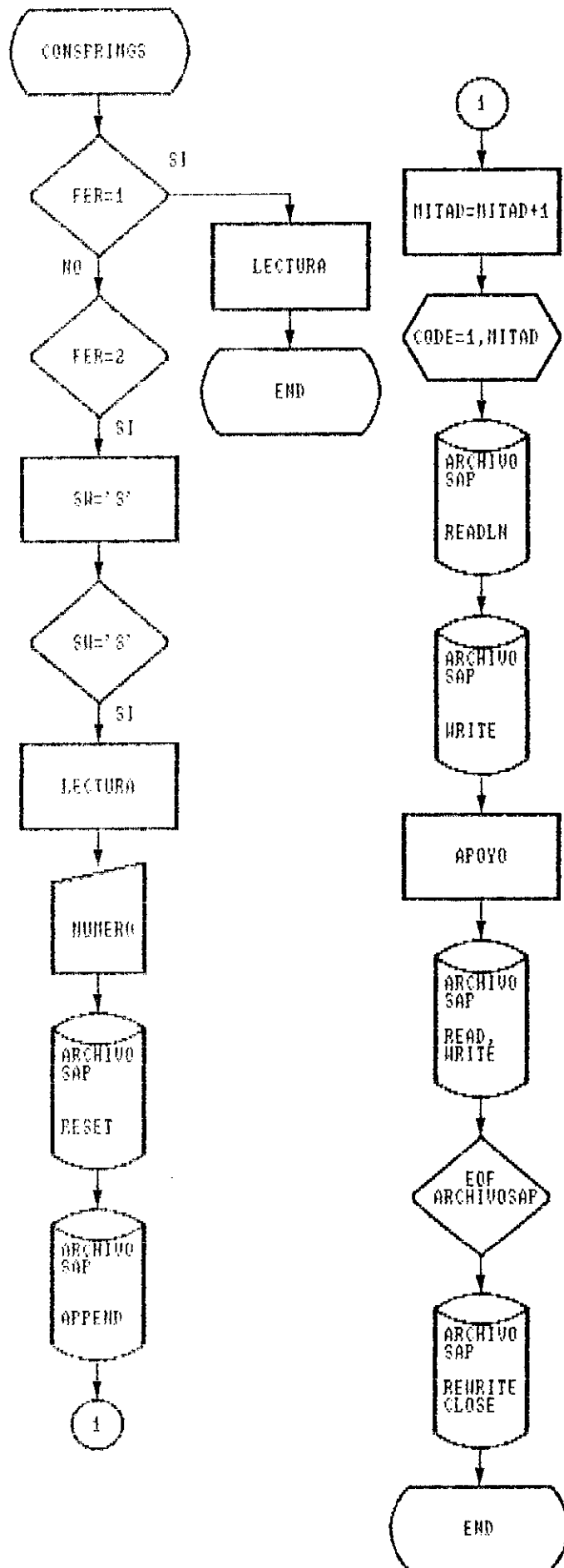


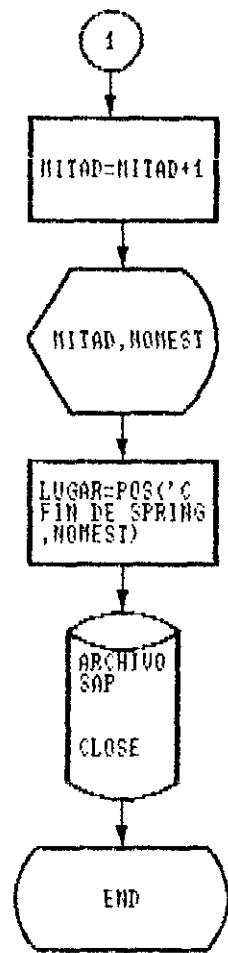
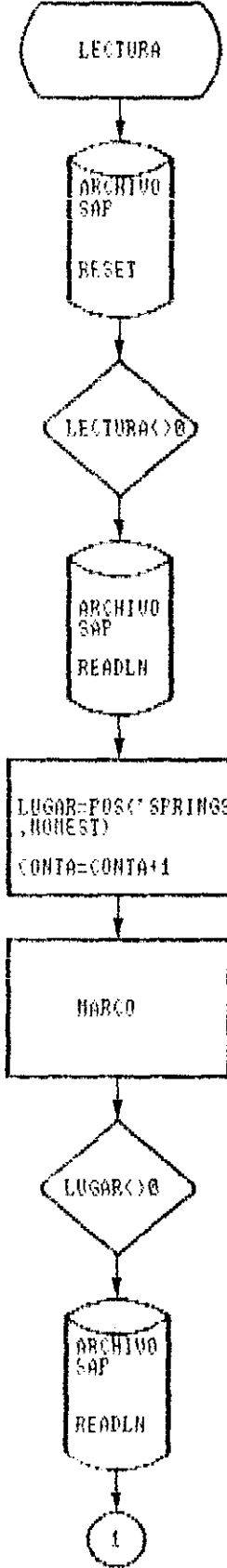


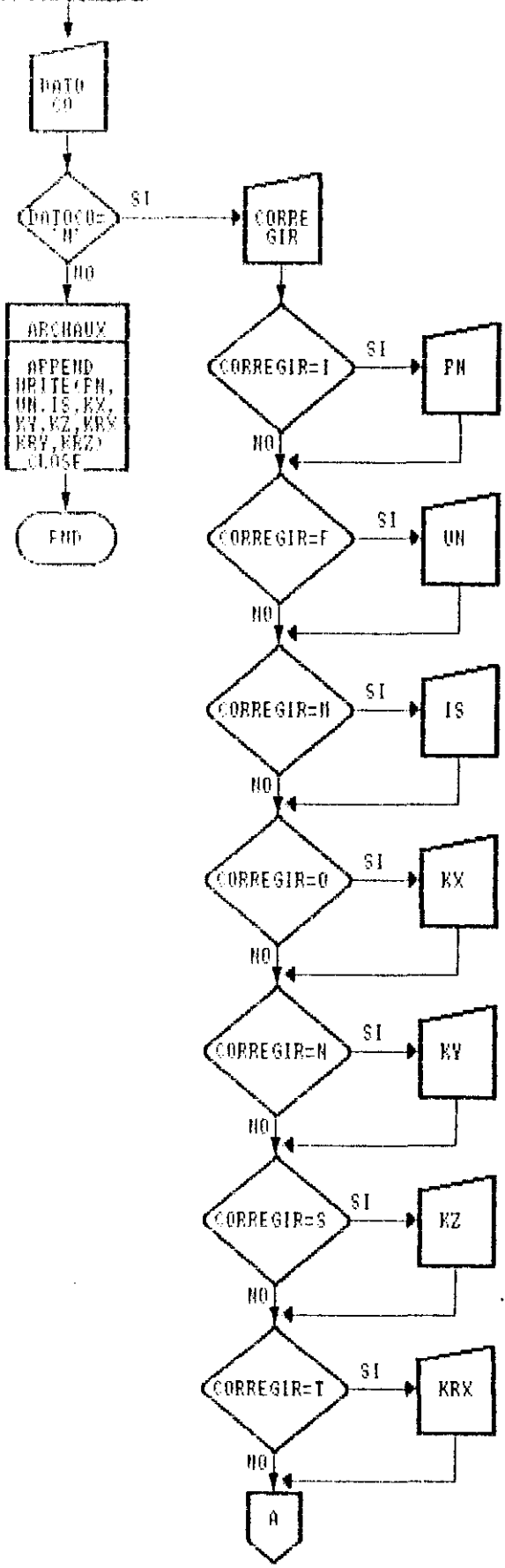


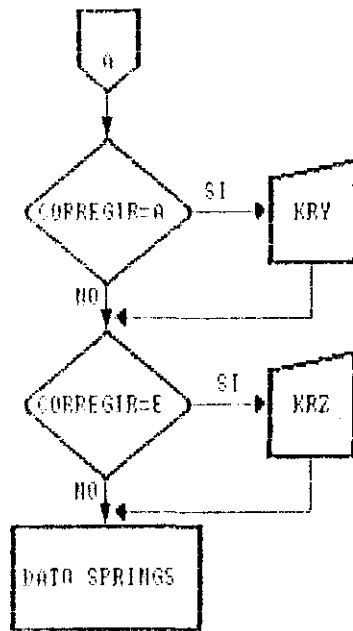
NOMBRE DEL PROGRAMA: CONSPRING

| NOMBRE | TIPO | DESCRIPCION |
|--------|---------|-----------------------------|
| NG | Byte | Número a declarar |
| JEEP | Byte | Contador |
| PH | Integer | Primer nodo |
| UH | Integer | Ultimo nodo |
| IS | Byte | Incremento |
| LOH | Byte | Para grabar sin espacios |
| KX | Real | Constante para traslación X |
| KY | Real | Constante para traslación Y |
| KZ | Real | Constante para traslación Z |
| KRX | Real | Constante para rotación X |
| KRY | Real | Constante para rotación Y |
| KRZ | Real | Constante para rotación Z |
| PNAUX | String | Para grabar |









NOME DA VARIÁVEL | Nº DE OBSERVAÇÕES

| NOME DA VARIÁVEL | TIPO | DESCRIÇÃO |
|------------------|-------|----------------------------|
| 01 | Nº de | Número de dias de trabalho |
| 02 | Nº de | Contratos |
| 03 | Nº de | Unidades de trabalho |
| 04 | Nº de | Unidades de trabalho |
| 05 | Nº de | Unidades de trabalho |
| 06 | Nº de | Unidades de trabalho |
| 07 | Nº de | Unidades de trabalho |
| 08 | Nº de | Unidades de trabalho |
| 09 | Nº de | Unidades de trabalho |
| 10 | Nº de | Unidades de trabalho |
| 11 | Nº de | Unidades de trabalho |
| 12 | Nº de | Unidades de trabalho |
| 13 | Nº de | Unidades de trabalho |
| 14 | Nº de | Unidades de trabalho |
| 15 | Nº de | Unidades de trabalho |
| 16 | Nº de | Unidades de trabalho |
| 17 | Nº de | Unidades de trabalho |
| 18 | Nº de | Unidades de trabalho |
| 19 | Nº de | Unidades de trabalho |
| 20 | Nº de | Unidades de trabalho |
| 21 | Nº de | Unidades de trabalho |
| 22 | Nº de | Unidades de trabalho |
| 23 | Nº de | Unidades de trabalho |
| 24 | Nº de | Unidades de trabalho |
| 25 | Nº de | Unidades de trabalho |
| 26 | Nº de | Unidades de trabalho |
| 27 | Nº de | Unidades de trabalho |
| 28 | Nº de | Unidades de trabalho |
| 29 | Nº de | Unidades de trabalho |
| 30 | Nº de | Unidades de trabalho |
| 31 | Nº de | Unidades de trabalho |
| 32 | Nº de | Unidades de trabalho |
| 33 | Nº de | Unidades de trabalho |
| 34 | Nº de | Unidades de trabalho |
| 35 | Nº de | Unidades de trabalho |
| 36 | Nº de | Unidades de trabalho |
| 37 | Nº de | Unidades de trabalho |
| 38 | Nº de | Unidades de trabalho |
| 39 | Nº de | Unidades de trabalho |
| 40 | Nº de | Unidades de trabalho |
| 41 | Nº de | Unidades de trabalho |
| 42 | Nº de | Unidades de trabalho |
| 43 | Nº de | Unidades de trabalho |
| 44 | Nº de | Unidades de trabalho |
| 45 | Nº de | Unidades de trabalho |
| 46 | Nº de | Unidades de trabalho |
| 47 | Nº de | Unidades de trabalho |
| 48 | Nº de | Unidades de trabalho |
| 49 | Nº de | Unidades de trabalho |
| 50 | Nº de | Unidades de trabalho |
| 51 | Nº de | Unidades de trabalho |
| 52 | Nº de | Unidades de trabalho |
| 53 | Nº de | Unidades de trabalho |
| 54 | Nº de | Unidades de trabalho |
| 55 | Nº de | Unidades de trabalho |
| 56 | Nº de | Unidades de trabalho |
| 57 | Nº de | Unidades de trabalho |
| 58 | Nº de | Unidades de trabalho |
| 59 | Nº de | Unidades de trabalho |
| 60 | Nº de | Unidades de trabalho |
| 61 | Nº de | Unidades de trabalho |
| 62 | Nº de | Unidades de trabalho |
| 63 | Nº de | Unidades de trabalho |
| 64 | Nº de | Unidades de trabalho |
| 65 | Nº de | Unidades de trabalho |
| 66 | Nº de | Unidades de trabalho |
| 67 | Nº de | Unidades de trabalho |
| 68 | Nº de | Unidades de trabalho |
| 69 | Nº de | Unidades de trabalho |
| 70 | Nº de | Unidades de trabalho |
| 71 | Nº de | Unidades de trabalho |
| 72 | Nº de | Unidades de trabalho |
| 73 | Nº de | Unidades de trabalho |
| 74 | Nº de | Unidades de trabalho |
| 75 | Nº de | Unidades de trabalho |
| 76 | Nº de | Unidades de trabalho |
| 77 | Nº de | Unidades de trabalho |
| 78 | Nº de | Unidades de trabalho |
| 79 | Nº de | Unidades de trabalho |
| 80 | Nº de | Unidades de trabalho |
| 81 | Nº de | Unidades de trabalho |
| 82 | Nº de | Unidades de trabalho |
| 83 | Nº de | Unidades de trabalho |
| 84 | Nº de | Unidades de trabalho |
| 85 | Nº de | Unidades de trabalho |
| 86 | Nº de | Unidades de trabalho |
| 87 | Nº de | Unidades de trabalho |
| 88 | Nº de | Unidades de trabalho |
| 89 | Nº de | Unidades de trabalho |
| 90 | Nº de | Unidades de trabalho |
| 91 | Nº de | Unidades de trabalho |
| 92 | Nº de | Unidades de trabalho |
| 93 | Nº de | Unidades de trabalho |
| 94 | Nº de | Unidades de trabalho |
| 95 | Nº de | Unidades de trabalho |
| 96 | Nº de | Unidades de trabalho |
| 97 | Nº de | Unidades de trabalho |
| 98 | Nº de | Unidades de trabalho |
| 99 | Nº de | Unidades de trabalho |
| 100 | Nº de | Unidades de trabalho |

13

Byte

número de nudo
para el giro C3
Incremento en el
número de nudo
para el giro C3

14

Byte

Incremento en el
número de nudo
para el giro C4

15

Byte

Incremento en el
número de nudo
para el giro C5

16

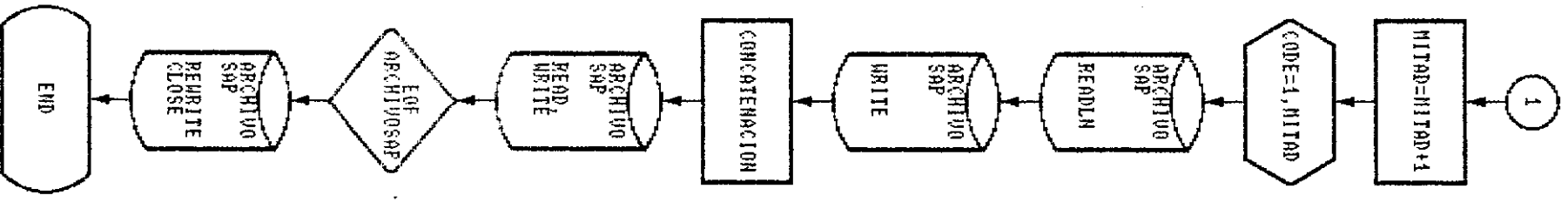
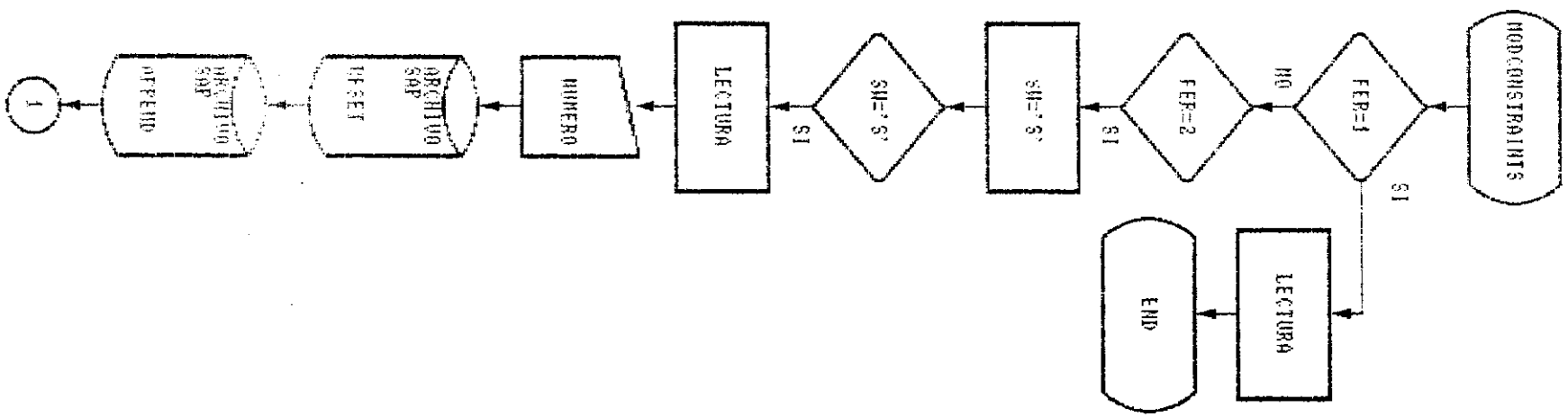
Byte

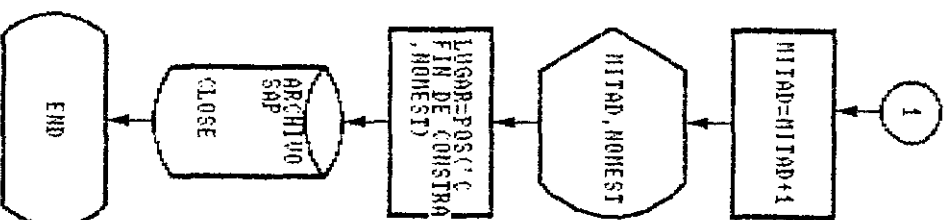
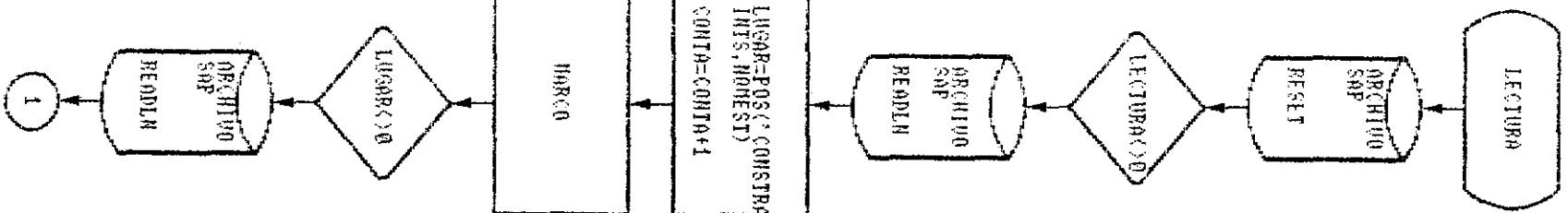
Incremento en el
número de nudo
para el giro C6

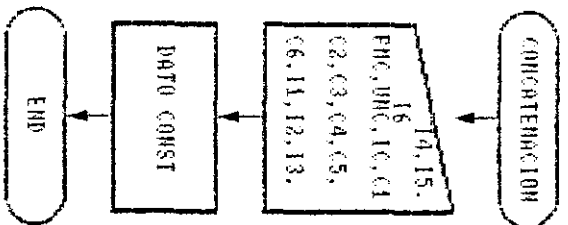
PHCAUZ

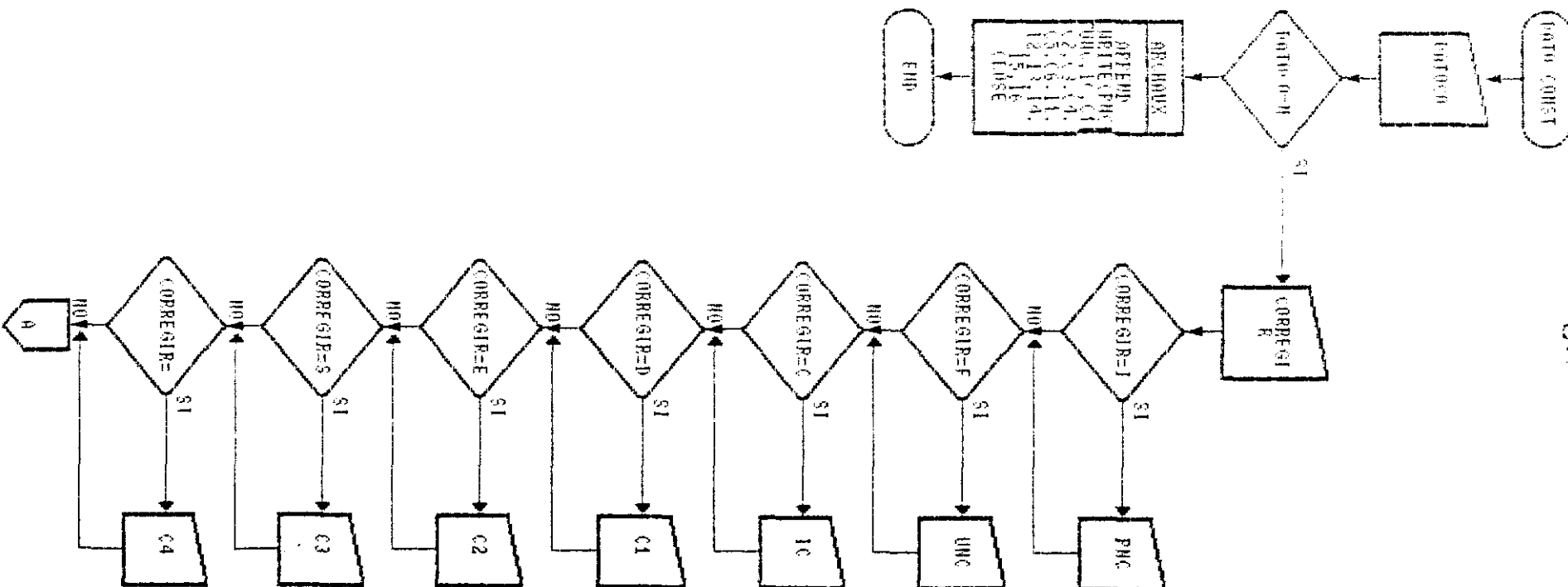
String

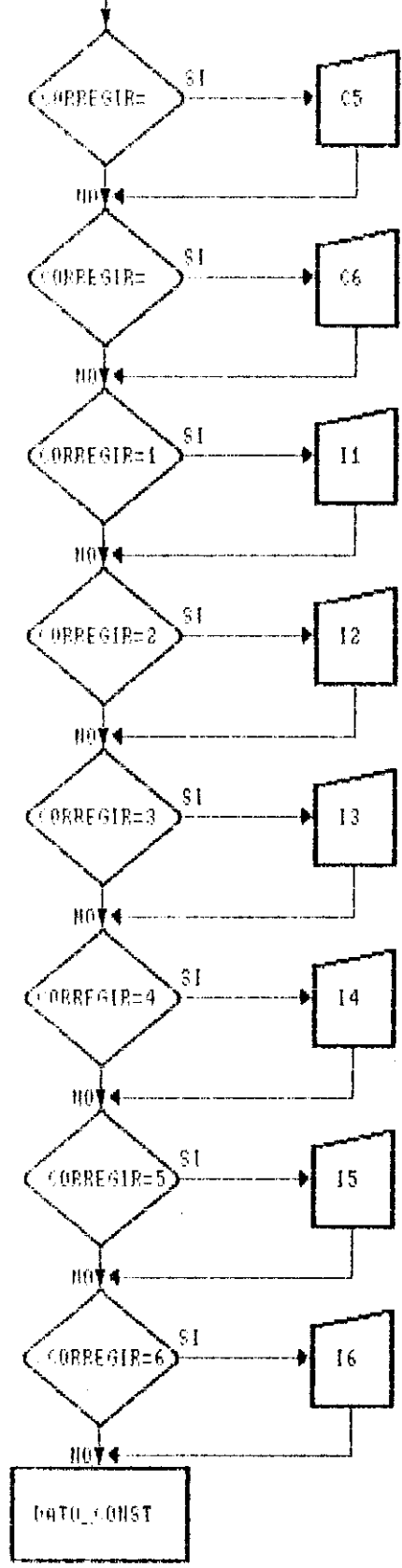
Para grabar











PROGRAM: CIPHER2

BEGIN

 COUNT := 1; ON := 0;

 FORWARD (COUNT);

 FORWARD (COUNT);

 WHILE (ON = 0)

 BEGIN

 FORWARD (COUNT);

 COUNT := COUNT + 1;

 END;

 IF COUNT = 1 THEN ON := 0;

 START (COUNT);

 FORWARD (COUNT);

 WHILE (ON = 0)

 BEGIN

 FORWARD (COUNT);

 COUNT := COUNT + 1;

 END;

 COUNT := COUNT + 1;

 START (COUNT);

 FORWARD (COUNT);

 WHILE (ON = 0)

 BEGIN

 FORWARD (COUNT);

 COUNT := COUNT + 1;

 END;

PROGRAM: OPTION1;

PROGRAM: OPTION1;

BEGIN

 COUNT := 1; ON := 0;

 START (COUNT);

 FORWARD (COUNT);

 WHILE (ON = 0)

 BEGIN

 FORWARD (COUNT);

 COUNT := COUNT + 1;

 END;

END;

PROGRAM: OPTION2;

PROGRAM: OPTION2;

BEGIN

 FORWARD (COUNT);

 START (COUNT);

 FORWARD (COUNT);

 START (COUNT);

END;

PROGRAM: OPTION3;

PROGRAM: OPTION3;

BEGIN

 START (COUNT);

 FORWARD (COUNT);

END;

PROGRAM: OPTION4;

PROGRAM: OPTION4;

BEGIN

 START (COUNT);

 FORWARD (COUNT);

END;

PROGRAM: procedure_1 section 25

FORWARD := 1; ON := 0; FORWARD (COUNT);

FORWARD := 1; ON := 0; FORWARD (COUNT);


```

add14:=1;
END; #if CONDOR
GOXY(59,1081,109) # add4=0; # if 1;
add14:=0;
if (add4 = 0) # (add5 = 0) THEN
  BEGIN

```

```

    add5 :=

```

```

      add5 + (add7 - add6) # (add7 - add6);

```

```

  BEGIN

```

```

    EXTRACT(1);

```

```

    GOXY(59,1081,109) # add5 = 0; # if 1;

```

```

    EXTRACT(2);

```

```

    GOXY(59,1081,109) # add5 = 0;

```

```

    while add5 > 0

```

```

    BEGIN

```

```

      EXTRACT(1) # (add5 > 0);

```

```

      if (add5 > 0) then

```

```

        if (add5 > 0) then

```

```

          end; while add5;

```

```

          GOXY(59,1081,109);

```

```

          while add5 > 0

```

```

          BEGIN

```

```

            EXTRACT(1) # (add5 > 0);

```

```

            if (add5 > 0) then

```

```

              if (add5 > 0) then

```

```

                end; while add5;

```

```

                GOXY(59,1081,109);

```

```

                while add5 > 0

```

```

                BEGIN

```

```

                  EXTRACT(1) # (add5 > 0);

```

```

                  if (add5 > 0) then

```

```

                    if (add5 > 0) then

```

```

                      end; while add5;

```

```

                      end;

```

```

                    GOXY(59,1081,109);

```

```

                    GOXY(59,1081,109) # (add5 > 0);

```

```

                    EXTRACT(1);

```

```

                    GOXY(59,1081,109) # (add5 > 0);

```

```

                    GOXY(59,1081,109) # (add5 > 0);

```

```

                    EXTRACT(2);

```

```

                    end; while (add5 > 0);

```

```

                    add5 := 0;

```

```

                END; #if CONDOR

```

```

                EXTRACT(1) # (add5 > 0);

```

```

                EXTRACT(2);

```

```

            if (add5 > 0) and (add7 > 0) and (add7 > 0) and (add7 > 0) THEN

```

```

            BEGIN

```

```

              EXTRACT(1) # (add5 > 0);

```

```

              EXTRACT(2) # (add5 > 0);

```

```

              EXTRACT(3) # (add5 > 0);

```

```

              EXTRACT(4) # (add5 > 0);

```

```

            if (add5 > 0) and (add7 > 0) and (add7 > 0) THEN

```

```

            BEGIN

```

```

              EXTRACT(1);

```

```

              EXTRACT(2) # (add5 > 0);

```

```

              EXTRACT(3) # (add5 > 0);

```

```

              EXTRACT(4) # (add5 > 0);

```

```

            END;

```

```

        if (add5 > 0) and (add7 > 0) and (add7 > 0) THEN

```

```

        BEGIN

```

```

WRITELN(ARCNAUX,ENVAUX,' ',DELNAUX,' ',ALNAUX,'      M=',MAUX,'      G=',NMG AUX,' ',INIAUX,' ',INFAUX,' ',
        ENAUX,'      NSU=',EVAUX,' ',CMUAUX,' ',CSAUX);
goto 28;
END;
IF ((adr1=1) and (adr2=2) and (adr3=3)) THEN
BEGIN
STEPPE;OPTIOM1;OPTIOM2;OPTIOM3;
WRITELN(ARCNAUX,ENVAUX,' ',DELNAUX,' ',ALNAUX,'      M=',MAUX,'      G=',NMG AUX,' ',INIAUX,' ',INFAUX,' ',
        ENAUX,'      NSU=',EVAUX,' ',CMUAUX,' ',CSAUX);
        LP=',LP1AUX,' ',LP2AUX,' ',LP3AUX);
goto 28;
END;
IF ((adr1=1) and (adr2=3)) THEN
BEGIN
STEPPE;OPTIOM1;OPTIOM2;
WRITELN(ARCNAUX,ENVAUX,' ',DELNAUX,' ',ALNAUX,'      M=',MAUX,'      G=',NMG AUX,' ',INIAUX,' ',INFAUX,' ',
        ENAUX,'      NSU=',EVAUX,' ',CMUAUX,' ',CSAUX);
        LP=',LP1AUX,' ',LP2AUX,' ',LP3AUX);
goto 28;
END;
IF ((adr1=1) and (adr2=2)) THEN
BEGIN
STEPPE;OPTIOM1;OPTIOM2;
WRITELN(ARCNAUX,ENVAUX,' ',DELNAUX,' ',ALNAUX,'      M=',MAUX,'      G=',NMG AUX,' ',INIAUX,' ',INFAUX,' ',
        ENAUX,'      NSU=',EVAUX,' ',CMUAUX,' ',CSAUX);
        LP=',LP1AUX,' ',LP2AUX,' ',LP3AUX);
goto 28;
END;
IF ((adr1=1) and (adr2=3)) THEN
BEGIN
STEPPE;OPTIOM1;OPTIOM2;
WRITELN(ARCNAUX,ENVAUX,' ',DELNAUX,' ',ALNAUX,'      M=',MAUX,'      G=',NMG AUX,' ',INIAUX,' ',INFAUX,' ',
        ENAUX,'      NSU=',EVAUX,' ',CMUAUX,' ',CSAUX);
        LP=',LP1AUX,' ',LP2AUX,' ',LP3AUX);
goto 28;
END;
IF ((adr1=2) and (adr2=3)) THEN
BEGIN
STEPPE;OPTIOM2;OPTIOM3;
WRITELN(ARCNAUX,ENVAUX,' ',DELNAUX,' ',ALNAUX,'      M=',MAUX,'      G=',NMG AUX,' ',INIAUX,' ',INFAUX,' ',
        ENAUX,'      NSU=',EVAUX,' ',CMUAUX,' ',CSAUX);
        LP=',LP1AUX,' ',LP2AUX,' ',LP3AUX);
goto 28;
END;
IF ((adr1=2) and (adr3=4)) THEN
BEGIN
STEPPE;OPTIOM2;OPTIOM4;
WRITELN(ARCNAUX,ENVAUX,' ',DELNAUX,' ',ALNAUX,'      M=',MAUX,'      G=',NMG AUX,' ',INIAUX,' ',INFAUX,' ',
        ENAUX,'      NSU=',EVAUX,' ',CMUAUX,' ',CSAUX);
        LP=',LP1AUX,' ',LP2AUX,' ',LP3AUX);
goto 28;
END;
IF ((adr1=3) and (adr2=4)) THEN
BEGIN
STEPPE;OPTIOM3;OPTIOM4;
WRITELN(ARCNAUX,ENVAUX,' ',DELNAUX,' ',ALNAUX,'      M=',MAUX,'      G=',NMG AUX,' ',INIAUX,' ',INFAUX,' ',
        ENAUX,'      NSU=',EVAUX,' ',CMUAUX,' ',CSAUX);
        LP=',LP1AUX,' ',LP2AUX,' ',LP3AUX);
goto 28;
END;
IF ((adr1=4)) THEN
BEGIN

```

```

SIEMPRE;OFF (ON);
MOTIEN(ARCHAUX,UNVAUX,'',DELNAUX,'',ALNAUX,'',
      PS1='UNVAUX','',LHVAUX,'',CSAUX);
goto Z0;
END;
IF (a0=17=7) THEN
REPTI
SIEMPRE;OFF (ON);
MOTIEN(ARCHAUX,UNVAUX,'',DELNAUX,'',ALNAUX,'',
      LF='LF1AUX','',LF2AUX,'',LF3AUX);
goto Z0;
END;
IF a0+2=7 THEN
REPTI
SIEMPRE;OFF (ON);
MOTIEN(ARCHAUX,UNVAUX,'',DELNAUX,'',ALNAUX,'',
      G='HGGAUX','',INIAUX,'',INFAUX,'',IM);
goto Z0;
END;
IF (ACT)=1 THEN
REPTI
SIEMPRE;OFF (ON);
MOTIEN(ARCHAUX,UNVAUX,'',DELNAUX,'',ALNAUX,'',
      H='MAUX');
goto Z0;
END;
IF (ADP)=0 THEN
REPTI
SIEMPRE;
MOTIEN(ARCHAUX,UNVAUX,'',DELNAUX,'',ALNAUX);
goto Z0;
END;
Z0: CLOSE (ARCHAUX);
END; >Del del Procedure?

BEGIN #DEL PROCEDURE CONSEQUENCES
IF FEV=1 THEN
BEGIN
LEF ARCH;
EPIE;
END
ELSE
BEGIN
IF FEV=2 THEN
BEGIN
SW='S';
WHILE (SW='S') OR (SW='e') DO
BEGIN
LEF ARCH;
CONTA:=0;
ASSIGN(ARCHIVOSAF,NUMER);
RESET(ARCHIVOSAF);
REPEAT
READLN(ARCHIVOSAF,MMEST);
LUGAR:=POS('e NUMERO',MMEST);
UNTIL LUGAR < 0;
REPEAT
READLN(ARCHIVOSAF,MMEST);
LUGAR:=POS('e FIN DE FRASE',MMEST);
CONTA:=CONTA+1;
UNTIL LUGAR > 0;
CLOSE (ARCHIVOSAF);
WRITE(5,21)WRITE(1);

```

```

GOTRUE(4,23)WRITE(OUT NO DE LINEA PESSA MODIFICAR : E);
GOTOXY(4,11)PAUSE(1000);
IF (NUMERO =0) GO (NUMERO =CONTA) THEN
RESIB
LEFEAT
WRITE(OUT);
GOTOXY(2,11)WRITE(NUMERO NO VALIDO .... PRESIUME UMA TELA E INGRESE OUTRO);
GOTO(2,7);
WRITE(4,23)S:0;
GOTO 5;
END;
(OBIA=0;
POSICION:=23;ABR1=-0;M1001=0;M1002=0;
OBRAR;
ASSIGN(ARCIIVOSAP,NUMERE);
RESET(ARCIIVOSAP);
REPEAT
READ(ARCIIVOSAP,NUMERE);
M1001:=M1001+1;
M1002:=M1002+1;
WRITE(OUT, 'A');
LIDE(ARCIIVOSAP);
M1003:=M1003+NUMERO+1;
ASSIGN(ARCIIVOSAP,NUMERE);
RESET(ARCIIVOSAP);
ASSIGN(ARIPARA, AUXILIAR);
APPEND(ARCIIVAP);
OBRAR:=1;
WHILE CONT =M1003 DO
BEGIN
READ(ARCIIVOSAP,NUMERE);
WRITE(ARCIIVAP,NUMERE);
OBRAR:=OBRAR+1;
END;
ELSE(ARCIIVAP);
OBRAR(ARCIIVOSAP);
TEXT(OR(1,1));
GOTOXY(4,22)WRITE(ITEM, No.1 No.2 Tipo NUm.S/N Gener.S/N Coor.S/N Est.Carga S/N
TEXT(OR(2));
POSICION:=23;
LECHURA(POSICION,NO,NO,ABR1);
GOTOXY(2,22)WRITE(
GOTOXY(2,23)WRITE(
ASSIGN(ARCIIVOSAP,NUMERE);
RESET(ARCIIVOSAP);
FOR CODE:=1 TO M1001 DO
READ(ARCIIVOSAP,NUMERE);
ASSIGN(ARCIIVAP, AUXILIAR);
APPEND(ARCIIVAP);
REPEAT
READ(ARCIIVOSAP,NUMERE);
WRITE(ARCIIVAP,NUMERE);
CLOSE(ARCIIVOSAP);
LIDE(ARCIIVOSAP);
ASSIGN(ARCIIVOSAP,NUMERE);
RESET(ARCIIVOSAP);
AFEM(ARCIIVOSAP);
ASSIGN(ARCIIVAP, AUXILIAR);
RESET(ARCIIVAP);
WHILE NOT EOF(ARCIIVAP) DO

```

```

BEGIN
  READLN(ARCHIVO,'.NOMES1');
  WRITELN(ARCHIVO,'.NOMES1');
END;

CLOSE(ARCHIVOAP);
CLOSE(ORLWAL);
GOJUY(5,2);WRITE('
GOJUY(20,2);WRITE('SESA PROFICAR OTRA LINEA/N : ');
GOJUY(5,2);WRITE('
END; #WHILE SW='S'
ASIGN(ARCHIVO,'AUXILIAR');
DEWRITE(ARCHIVO);
CLOSE(ARCHIVO);
END; #IF FER=28
END; #DEL ELSE
END; #DEL PROCEDURE CONSUEBARE;

PROCEDURE CONSUMANL;
VAR
  CONTAL:BYTE;
PROCEDURE LECTURA;
BEGIN
  ASIGN(ARCHIVOAP,'.NOMES1');
  RESET(ARCHIVOAP);
  CONTAL:=0;
  REPEAT
    READLN(ARCHIVOAP,'.NOMES1');
    LUGAR:=POS('C NUMERO DE MIEMBROS',.NOMES1);
    CONTAL:=CONTAL+1;
  UNTIL LUGAR < 0;
  CLOSE(ARCHIVOAP);
  ASIGN(ARCHIVOAP,'.NOMES1');
  RESET(ARCHIVOAP);
  CONTAL:=0;
  REPEAT
    READLN(ARCHIVOAP,'.NOMES1');
    LUGAR:=POS('C NUMERO DE MIEMBROS',.NOMES1);
    CONTAL:=CONTAL+1;
  UNTIL LUGAR < 0;
  CLOSE(ARCHIVOAP);
  MIAD:=CONTAL-CONTAL;
  ASIGN(ARCHIVOAP,'.NOMES1');
  RESET(ARCHIVOAP);
  FOR CODE:=1 TO CONTAL DO
    READLN(ARCHIVOAP);
  CLSER;
  MADO;
  TEXIDU(0,14);
  GOJUY(20,2);WRITE('EMPLEADOS Y CARGAS DE LOS MIEMBROS');
  WRITELN;WRITELN;
  TEXIDU(0,7);
  FOR CODE:=1 TO MIAD DO
  BEGIN
    READLN(ARCHIVOAP,'.NOMES1');
    WRITELN(' ',.NOMES1);
  END;
  CLOSE(ARCHIVOAP);
END; #DEL PROCEDURE LECTURAS
BEGIN #DEL PROCEDURE CONSUMANA

```

```

LECTURA:
ENTER:
END: #DEL PROCEDURE CONSUMO

```

```

PROCEDURE CONSUMO:
VAR
  NOMB, LON: BYTE;
  DEL, QU, QW: INTEGER;
  AX, AY, AZ, TH: REAL;
  QXAX, QYAX, QZAX, THAX: SET OF INTEGER;
  QELAX, QOAX, QVAX: SET OF REAL;
LABEL 5:

```

```

PROCEDURE LECTURA:
BEGIN
  ASSIGN(ARCHIVOSAP, NOMBRE);
  RESET(ARCHIVOSAP);
  FON(A):=0;
  REPEAT
    READLN(ARCHIVOSAP, NOMBRE);
    LUGAR:=POS('QUAD', NOMBRE);
    CONT:=CONT+1;
  UNTIL LUGAR<>0;
  CLOSE;
  MARCO;
  MITAD:=0;
  TEXTCOLOR(14);
  GOTOXY(10,2);WRITE('C A S C A R A S, P L A C A S Y M E M B R A N A S');
  GOTOXY(6,3);WRITE('Numero de tipos de materiales declarados: ');
  GOTOXY(2,3);WRITE(' # Factor Gravitacional Módulo Módulo Peso/ Espesor');
  GOTOXY(2,4);WRITE('Linea X Y Z Elastic. Poiss. Volú. ');
  TEXTCOLOR(7);
  WRITELN;
  REPEAT
    READLN(ARCHIVOSAP, NOMBRE);
    MITAD:=MITAD+1;
    WRITELN('|| ', MITAD, ' ', NOMBRE);
    LUGAR:=POS('C FIN DE QUAD', NOMBRE);
  UNTIL LUGAR<>0;
  CLOSE(ARCHIVOSAP);
  TEXTCOLOR(14);
  GOTOXY(49,1);WRITE(MITAD);
  TEXTCOLOR(7);
END;

```

```

PROCEDURE CORRE_PLACAS(POSICION:INTEGER);
BEGIN
  GOTOXY(5,21);WRITE(' ');
  GOTOXY(5,21);WRITE('Sus Datos están Correctos S/N: ');
  TEXTCOLOR(14);
  GOTOXY(31,21);WRITE('S');
  GOTOXY(33,21);WRITE('N');
  TEXTCOLOR(7);
  GOTOXY(36,21);DATOCO:=READKEY;
  IF (DATOCO='N') OR (DATOCO='n') THEN
    BEGIN
      GOTOXY(5,21);WRITE(' ');
      GOTOXY(3,21);WRITE(' Fact. gravit: X, Y, Z; Módulos: Elastic. ó Poisson; Peso/Volumen; Espesor: ');
      TEXTCOLOR(14);
      GOTOXY(15,21);WRITE('X'); GOTOXY(18,21);WRITE('Y'); GOTOXY(21,21);WRITE('Z'); GOTOXY(32,21);WRITE('E');
      GOTOXY(43,21);WRITE('P'); GOTOXY(57,21);WRITE('V'); GOTOXY(67,21);WRITE('S');
    END;
  END;

```

```

TEXT(COL02/2);
GO10XY175:211 (CONTRACT=1)AN1;
CAST CORRESPR OF
'Y',Y'; BEGIN
    COL00MA=0;
    QX:=funInObj(columa,position);
END;
'Y',Y'; BEGIN
    COL00MA=1;
    QY:=funInObj(columa,position);
END;
I,'Y'; BEGIN
    COL00MA=2;
    QZ:=funInObj(columa,position);
END;
'E,'e'; BEGIN
    COL00MA=3;
    QG:=funInObj(columa,position);
END;
'P','p'; BEGIN
    COL00MA=4;
    QH:=funInObj(columa,position);
END;
'V','v'; BEGIN
    COL00MA=5;
    QI:=funInObj(columa,position);
END;
'S','s'; BEGIN
    COL00MA=6;
    TH:=funInObj(columa,position);
END;
END; #End Case#
CORRE_FLAGS(POSITION);
END #End If_Datoc#
ELSE
BEGIN
    ACSTH(ARCHWAY,'MILLAR');
    AFFEND(ARCHWAY);
    QY00X:='';L0M:=0;
    STR(QY);;2,QY00X);
    STR(QY);;2,QY00X);
    L0M:=LENGTH(QY00X);
    WHILE L0M<=6 DO
    BEGIN
        QY00D:=QY00X+' ';
        L0M:=L0M+1;
    END;
    QZ00X:='';L0M:=0;
    STR(QZ);;2,QZ00X);
    L0M:=LENGTH(QZ00X);
    WHILE L0M<=6 DO
    BEGIN
        QZ00X:=QZ00X+' ';
        L0M:=L0M+1;
    END;

```

```

END;
DELAUX:= ' ' * LON:=0;
STR(BEL,DELAUX);
LON:=LENGTH(DELAUX);
WHILE LON<=3 DO
BEGIN
    BELAUX:=DELAUX+' ' *
        LON:=LON+1;
END;
QUAUX:= ' ' * LON:=0;
STR(QW,QUAUX);
LON:=LENGTH(QUAUX);
WHILE LON<=3 DO
BEGIN
    QUAUX:=QUAUX+' ' *
        LON:=LON+1;
END;
QUAUX:= ' ' * LON:=0;
STR(QW,QUAUX);
LON:=LENGTH(QUAUX);
WHILE LON<=3 DO
BEGIN
    QUAUX:=QUAUX+' ' *
        LON:=LON+1;
END;
THAUX:= ' ' * LON:=0;
STR(TR,THAUX);
LON:=LENGTH(THAUX);
WHILE LON<=6 DO
BEGIN
    THAUX:=THAUX+' ' *
        LON:=LON+1;
END;
WRITELN(ARQUAUX, 'X=',QUAUX, ' Y=',QVAUX, ' Z=',QZVAUX, ' E=',DELAUX, ' U=',QUAUX, ' W=',QWVAUX, ' TH=
',CLOSE(ARQUAUX);
END;end else datorec
END; end Procedure4

PROCEDURE PLACA(POS:LINEA;MIBERE):
BEGIN
    COUMMA:=5;QY:=funIndBel(columa,position);
    COUMMA:=13;QY:=funIndBel(columa,position);
    COUMMA:=21;QZ:=funIndBel(columa,position);
    COUMMA:=32;QEL:=funInd(columa,position);
    COUMMA:=44;QV:=funInd(columa,position);
    COUMMA:=57;QW:=funInd(columa,position);
    COUMMA:=66;TH:=funIndBel(columa,position);
    CORRE_PLACA(POS);
END;end Procedure4

(*PROGRAMA PRINCIPAL*)
BEGIN
    IF FER=1 THEN
        BEGIN
            LEATURA;
            ENTER;
        END
    ELSE
        BEGIN
            IF FER=2 THEN
                BEGIN

```

END=0;

WHILE (STR=?) AND (W=?) DO

READ

 EQUINA;

 FARM=0;

 ASSIGNOVAL JORNADA JORNADA 1;

 RESELECAO JORNADA 1;

 REPRO;

 READ (VALOR,VALOR,PROJETO);

 LIGAS=PROJ (1) DE QUAD. PROJETO;

 END IF (LIGAS = 0);

 REPRO;

 READ (VALOR,VALOR,PROJETO);

 LIGAS=PROJ (1) DE QUAD. PROJETO;

 REPRO=VALOR;

 WRITE (LIGAS, ' ');

 LIGAS=VALOR*VALOR;

 CONTINUA=?) RELEVE;

 CONTINUA=?) RELEVE: CONTINUA=?) RELEVE RESEA PRODUCCAO: E?);

 CONTINUA=?) RELEVE RESEA PRODUCCAO;

 IF (LIGAS=0) GO TO (LIGAS=?) ONTA) THEN

 CONTINUA

 REPRO;

 LEITURAS(1);

 CONTINUA=?) RELEVE RESEA PRODUCCAO NO VALOR PRESSIONE UNA TECLA E INGRESSE DIRIG);

 CONTINUA=?) RELEVE;

 WRITE (LIGAS, ' ');

 END IF

END;

 CONTINUA=?)

 CONTINUA=?) RELEVE RESEA PRODUCCAO NO VALOR PRESSIONE UNA TECLA E INGRESSE DIRIG);

 REPRO;

 ASSIGNOVAL JORNADA JORNADA 1;

 RESELECAO JORNADA 1;

 REPRO;

 READ (VALOR,VALOR,PROJETO);

 LIGAS=PROJ (1) DE QUAD. PROJETO;

 END IF (LIGAS = 0);

 REPRO;

 WRITE (LIGAS, ' ');

 LIGAS=VALOR*VALOR;

 CONTINUA=?) RELEVE;

 CONTINUA=?) RELEVE: CONTINUA=?) RELEVE RESEA PRODUCCAO:

 CONTINUA=?) RELEVE RESEA PRODUCCAO;

 CONTINUA

 LIGAS=VALOR;

 WRITE (LIGAS, ' ');

 REPRO;

 READ (VALOR,VALOR,PROJETO);

 LIGAS=PROJ (1) DE QUAD. PROJETO;

 END IF (LIGAS = 0);

 REPRO;

 LEITURAS(1);

 CONTINUA=?) RELEVE;

 CONTINUA=?) RELEVE: CONTINUA=?) RELEVE RESEA PRODUCCAO NO VALOR PRESSIONE UNA TECLA E INGRESSE DIRIG);

 CONTINUA=?) RELEVE;

 WRITE (LIGAS, ' ');

 LIGAS=VALOR*VALOR;

 CONTINUA=?) RELEVE;

 CONTINUA=?) RELEVE: CONTINUA=?) RELEVE RESEA PRODUCCAO:

 CONTINUA=?) RELEVE RESEA PRODUCCAO;

```

        ASSIGN(ARCHIVOSAF,NUMERO);
        RESET(ARCHIVOSAF);
        FOR CODE:=1 TO NITAD-1 DO
            READLN(ARCHIVOSAF,NUMEST);
        ASSIGN(ARCHIVOSAF,'AUXILIAR');
        APPEND(ARCHIVOSAF);
        RESET;
            READLN(ARCHIVOSAF,NUMEST);
            WRITELN(CANTIDAD,NUMEST);
        UNTIL EOF(ARCHIVOSAF);
        CLOSE(ARCHIVOSAF);
        CLOSE(ARCHIVOSAF);
        ASSIGN(ARCHIVOSAF,NUMERO);
        REMOVE(ARCHIVOSAF);
        APPEND(ARCHIVOSAF);
        ASSIGN(ARCHIVOSAF,'AUXILIAR');
        RESET(ARCHIVOSAF);
        WHILE NOT EOF(ARCHIVOSAF) DO
            BEGIN
                READLN(ARCHIVOSAF,NUMEST);
                WRITELN(ARCHIVOSAF,NUMEST);
            END;
        CLOSE(ARCHIVOSAF);
        CLOSE(ARCHIVOSAF);
        GOTOXY(5,21);WRITE(' ');
        GOTOXY(20,21);WRITE(' RESEA MODIFICAR OTRA LINEA S/N : ');
        GOTOXY(53,21);SM:=REARKEY;
    END; *WHILE SM= 'A'
    ASSIGN(ARCHIVOSAF,'AUXILIAR');
    REMOVE(ARCHIVOSAF);
    CLOSE(ARCHIVOSAF);
END; *IF CER=25
END; *OCT ELSE A
END; *DEL PROCEDURE CONSUMIDOR

```

PROCEDURE CONSUMIDOR;

VAR

```

    FRUJUN:INTEGER;
    G:GMIN,IMBUDU,F:SECAE,F:ON,AL:DDO,C:CONTA:BYTE;
    EX,EY,EZ,FX,FY,FZ,OX,OY,OZ,OX,OX,RZ:REAL;
    FR:FRUJUN,SE:SE:FRUJUN,IMBUDU:F,SE:SE:FRUJUN:STRING64;
    EX:OX,EY:OY,EZ:OZ,FX:OX,FY:OY,FZ:OZ,OX:OX,RZ:OX:STRING64;
    OX:OX,OY:OY,OZ:OZ,OX:OX,OY:OY,RZ:OX,RZ:OX:STRING64;

```

LABEL 3;

PROCEDURE LECLUB;

BEGIN

```

    ASSIGN(ARCHIVOSAF,NUMEST);
    RESET(ARCHIVOSAF);
    CONTA:=0;
    RESET;
        READLN(ARCHIVOSAF,NUMEST);
        LUGAR:=POS('LOADE',NUMEST);
        CONTA:=CONTA+1;
    UNTIL LUGAR<0;
    LUGAR:=0;
    NUMEST:=0;
    NITAD:=0;
    TEXTCOLOR(14);

```

```

    GOTOXY(25,2);WRITE(' CARGAS CONSUMIDAS EN LOS MUOS ');
    GOTOXY(5,5);WRITE(' Cantidad de cargas declaradas ');

```



```

E7,77: BEGIN
      COLUMN:=70; E7:=FunIndex2(column,position);
      END;
E8,88: BEGIN
      COLUMN:=80; E8:=FunIndex2(column,position);
      END;
E9,99: BEGIN
      COLUMN:=90; E9:=FunIndex2(column,position);
      END;
M,11: BEGIN
      COLUMN:=10; M:=FunIndex2(column,position);
      END;
P1,11: BEGIN
      COLUMN:=10; P1:=FunIndex1(column,position);
      END;
P2,22: BEGIN
      COLUMN:=20; P2:=FunIndex1(column,position);
      END;
P3,33: BEGIN
      COLUMN:=30; P3:=FunIndex1(column,position);
      END;
X1,11: BEGIN
      COLUMN:=70; X1:=FunIndex1(column,position);
      END;
X2,22: BEGIN
      COLUMN:=74; X2:=FunIndex1(column,position);
      END;
X3,33: BEGIN
      COLUMN:=78; X3:=FunIndex1(column,position);
      END;
END; *End_Case#
CALL CONSOLEOP(POSITION,1);
END; *End_of_Data_Collection#
ELSE
PRINT
PRJUNDAY:= ' '; LON:=0;
STR(PRJUN,PRJUNDAY);
LON:=LENGTH(PRJUNDAY);
WHILE LON<=3 DO
BEGIN
  PRJUNDAY:=PRJUNDAY+' ';
  LON:=LON+1;
END;
SECONDAY:= ' '; LON:=0;
STR(SECON,SECONDAY);
LON:=LENGTH(SECONDAY);
WHILE LON<=3 DO
BEGIN
  SECONDAY:=SECONDAY+' ';
  LON:=LON+1;
END;
THURSDAY:= ' '; LON:=0;
STR(THURD,THURSDAY);
LON:=LENGTH(THURSDAY);
WHILE LON<=3 DO
BEGIN
  THURSDAY:=THURSDAY+' ';
  LON:=LON+1;
END;
ESTABDAY:= ' '; LON:=0;
STR(ESTAB,ESTABDAY);

```

```

END:=LEND;
WHILE LEND<=20
BEGIN
    ESTCARAUX:=ESTCARAUX+1;
    Ion:=Ion+1;
END;
SIP(FX:=1,FXAUX);
SIP(FY:=1,FYAUX);
SIP(FZ:=1,FZAUX);
SIP(MX:=1,MYAUX);
SIP(MY:=1,MYAUX);
SIP(MZ:=1,MZAUX);
SIP(OX:=1,OXAUX);
SIP(OY:=1,OYAUX);
SIP(OZ:=1,OZAUX);
SIP(PX:=1,PXAUX);
SIP(PY:=1,PYAUX);
SIP(PZ:=1,PZAUX);
ASIGN(ARCHIVO, 'AUX(1.AE)');
OPEN(ARCHAUX);
NEXT(ENCARAUX,ENLUNO, ' ',SEGLUNO, ' ',INMUNO, ' L=',ESTCARAUX, ' F=',FXAUX, ',',FYAUX, ',',FZAUX,
    MYAUX, ',',MYAUX, ',',MZAUX, ' U=',OXAUX, ',',OYAUX, ',',OZAUX, ',',PXAUX, ',',PYAUX, ',',PZAUX);
LIST(ARCHAUX);
END;
# Procedure Data Formateo

```

```

PROCEDURE (EE, LEAD(POSICION:=INUSER;L:BYE);

```

```

BEGIN
    COLUMNA:=3; FRIUN:= FunInEn(columna, posicion);
    COLUMNA:=9; SEGLUN:= FunInEn(columna, posicion);
    COLUMNA:=16; INMUN:= FunInEn(columna, posicion);
    COLUMNA:=24; ESTCAR:= FunInEn(columna, posicion);
    WHILE ESTCAR<=20
    BEGIN
        COLUMNA:=34; ESTCAR:=FunInEn(columna, posicion);
    END;
    COLUMNA:=28; FX:=FunInDec(columna, posicion);
    COLUMNA:=33; FY:=FunInDec(columna, posicion);
    COLUMNA:=38; FZ:=FunInDec(columna, posicion);
    COLUMNA:=43; MY:=FunInDec(columna, posicion);
    COLUMNA:=48; MY:=FunInDec(columna, posicion);
    COLUMNA:=53; MZ:=FunInDec(columna, posicion);
    COLUMNA:=58; OX:=FunInDec(columna, posicion);
    COLUMNA:=63; OY:=FunInDec(columna, posicion);
    COLUMNA:=68; OZ:=FunInDec(columna, posicion);
    COLUMNA:=73; PX:=FunInDec(columna, posicion);
    COLUMNA:=78; PY:=FunInDec(columna, posicion);
    COLUMNA:=83; PZ:=FunInDec(columna, posicion);
    DATA:=GET(ARCHIVO);
END;

```

```

#PROGRAMA PRINCIPAL DE LUNOS

```

```

BEGIN
    IF EEP=1 THEN
        BEGIN
            LIST(PSA);
            ENLUN;
        END
    ELSE
        BEGIN
            ASIGN(ARCHIVO, NUMBRE);
        END
    END

```



```

CLOSE (ARCHIVO);
CLOSE (ARCHIVOSAP);
TEXTOLONG(1);
GOBACK(3,2);WRITE('CON. 200. Inc. Est. Fuerzas y Momentos Desplazam. y Rotaciones
TEXTOLONG(1);
POSITION:=25;
LEFT(100);POSITION,1);
GOBACK(2,2);WRITE('
GOBACK(2,2);WRITE('
ASSIGN(ARCHIVOSAP,NOMBRE1);
RESET(ARCHIVOSAP);
FOR (LINE)=1 TO 1000 DO
    READLN(ARCHIVOSAP,NOMBRE1);
    ASSIGN(ARCHIVO,'AUXILIAR');
    APPEND(ARCHIVO);
    REWRITE
        REWRITE(ARCHIVOSAP,NOMBRE1);
        WRITELN(ARCHIVO,NOMBRE1);
    ON 11 TO 100 DIV 50;
CLOSE (ARCHIVO);
CLOSE (ARCHIVOSAP);
ASSIGN(ARCHIVOSAP,NOMBRE1);
REWRITE (ARCHIVOSAP);
APPEND(ARCHIVOSAP);
ASSIGN(ARCHIVO,'AUXILIAR');
RESET(ARCHIVO);
WRITE NOT EOL (ARCHIVO) DO
    BEGIN
        READLN(ARCHIVO,NOMBRE1);
        WRITELN(ARCHIVOSAP,NOMBRE1);
    END;
CLOSE (ARCHIVOSAP);
CLOSE (ARCHIVO);
GOBACK(5,2);WRITE('
GOBACK(20,2);WRITE('DESEA MODIFICAR OTRA LINEA S/N : ');
GOBACK(53,2);SM:=REARKEY;
END; *WHILE SM= 'S'
ASSIGN(ARCHIVO,'AUXILIAR');
REWRITE (ARCHIVO);
CLOSE (ARCHIVO);
END; *IF FFF=28
END; *DEL L1 SE2
END; *DEL PROCEDURE CONSULTADA

```

```

PROCEDURE CONSULTARFORCES;
VAR
    JDE,DE,CON:BYTE;
    CF1,CF2,CF3:BLM;
    JDEaux:STRING43A;
    CF1aux,CF2aux,CF3aux:STRING46A;
LABEL 5;

```

```

PROCEDURE LECTURA;
BEGIN
    ASSIGN(ARCHIVOSAP,NOMBRE1);
    RESET(ARCHIVOSAP);
    CONTA:=0;
    REPEAT
        READLN(ARCHIVOSAP,NOMBRE1);
        LUBAR:=POS('FORCES',NOMBRE1);
        CONTA:=CONTA+1;
    UNTIL

```



```

P
CONTI=0;
POSITION=2;ADP=0;MTAD=0;MTAD=0;
CONTI=0;
ASIG(ARCHIVUSOP,NUMBE);
PSE(ARCHIVUSOP);
MATE)
  READ(ARCHIVUSOP,NUMEST);
  MTAD=MTAD+1;
  CONTI=CONTI+1;
  WRITE(ARCHIVUSOP,CONTI);
  CONTI=CONTI+1;
  CLOSE(ARCHIVUSOP);
  MTAD=MTAD+NUMBE;
  ASIG(ARCHIVUSOP,NUMBE);
  PSE(ARCHIVUSOP);
  POSITION=ARCHIVUSOP+ARCHIVUSOP;
  APPEND(ARCHIVUSOP);
  CONTI=1;
  WHILE (CONTI=MTAD) DO
  BEGIN
    READ(ARCHIVUSOP,NUMEST);
    WRITE(ARCHIVUSOP,NUMEST);
    CONTI=CONTI+1;
  END;
  CLOSE(ARCHIVUSOP);
  CLOSE(ARCHIVUSOP);
  WRITE(ARCHIVUSOP);
  GOXY(10,2);WRITE('Numero Factor: 1 2 3 4');
  IF (CONTI=1);
  POSITION=2;
  CONTI(ARCHIVUSOP);
  GOXY(2,2);WRITE('
  GOXY(2,23);WRITE('
  ASIG(ARCHIVUSOP,NUMBE);
  PSE(ARCHIVUSOP);
  FOR (CONTI=1 TO MTAD) DO
    READ(ARCHIVUSOP,NUMEST);
  ASSIGN(ARCHIVUSOP,ARCHIVUSOP);
  APPEND(ARCHIVUSOP);
  REPEAT
    READ(ARCHIVUSOP,NUMEST);
    WRITE(ARCHIVUSOP,NUMEST);
  UNTIL (ARCHIVUSOP);
  CLOSE(ARCHIVUSOP);
  CLOSE(ARCHIVUSOP);
  ASIG(ARCHIVUSOP,NUMBE);
  PSE(ARCHIVUSOP);
  APPEND(ARCHIVUSOP);
  POSITION(ARCHIVUSOP,ARCHIVUSOP);
  PSE(ARCHIVUSOP);
  WRITE(ARCHIVUSOP);
  REPEAT(ARCHIVUSOP,NUMEST);
  WRITE(ARCHIVUSOP,NUMEST);
  END;
  CLOSE(ARCHIVUSOP);
  CLOSE(ARCHIVUSOP);
  GOXY(10,2);WRITE('BASEA MODIFICAR LINEA S/N : 11');
  GOXY(10,2);WRITE('
  END;=WRITE(

```

```

        WRITE(ARCHIVO, ADELTA);
        WRITE(ARCHIVO);
        CLOSE(ARCHIVO);
    END; *IF TER=2;
END; *DEL FINE
END; *DEL PROCEDURE CONTINUA;

PROCEDURE LECTURAS;
VAR
    NV1,NV2,NV3,NV4:INTEGER;

PROCEDURE LECTURAS;
BEGIN
    READLN(ARCHIVO,NUMERO);
    READLN(ARCHIVO,NO);
    CONT:=0;
    READLN
        READLN(ARCHIVO,NOBEST);
        LIGAR:=POSICION : ,NOBEST);
    READLN(ARCHIVO);
    TER:=0;
    READLN
    TEXTCOLOR(14);
    GOTOXY(28,2);WRITE(' TITULO DE LA ESTRUCTURA');
    GOTOXY(27,4);WRITE(' Vec.de vista = N. 1 y 2 de');
    GOTOXY(27,5);WRITE(' Nudos 1 y 2 = vect.horiz. ');
    WRITE('N');
    TEXTCOLOR(7);
    READLN(ARCHIVO,NOBEST);
    GOTOXY(1,7);WRITE(' ');
    GOTOXY(2,7);WRITE('NOBEST');
    CLOSE(ARCHIVO);
END; *DEL PROCEDURE LECTURAS

PROCEDURE CORRE_INGRESO(POSICION:INTEGER);
BEGIN
    GOTOXY(5,21);WRITE(' ');
    GOTOXY(5,21);WRITE(' Sus Datos estan Correctos S/N: ');
    TEXTCOLOR(14);
    GOTOXY(31,21);WRITE(' S');
    GOTOXY(33,21);WRITE(' N');
    TEXTCOLOR(7);
    GOTOXY(36,21);DATAID:=NOBEST;
    IF (DATAID= 'N') OR (DATAID='n') THEN
        BEGIN
            GOTOXY(2,21);WRITE(' ');
            GOTOXY(2,21);WRITE(' Vector de vista para nudo: 1, 2; Nudo 1 ó Nudo 2 del vector horizontal');
            TEXTCOLOR(14);
            GOTOXY(29,21);WRITE('1');GOTOXY(32,21);WRITE('2');GOTOXY(35,21);WRITE('N');GOTOXY(45,21);WRITE('W');
            TEXTCOLOR(7);
            GOTOXY(78,23);CONT:=READLN;
            CASE CONTREGIO OF
                1 : BEGIN
                    COLUMNA:=10;
                    NV1:=Fun1(Fun1(columna,posicion));
                END;
                2 : BEGIN
                    COLUMNA:=18;
                    NV2:=Fun1(Fun1(columna,posicion));
                END;
            END;
            NV1,NV2 : BEGIN

```


2024-10-08 11

File:

11051100100001

11051100100001

ES1(0000001);

getenv("20")-Verbores de Vista para modo Modo 1 vector

getenv("1,20")-Horiz 1 horizontal Modo 2 vector

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

horizontal);

NOMBRE DEL PROGRAMA: CANTONCIL

| NOMBRE | TIPO | DESCRIPCION |
|--------|---------|------------------------------|
| E | Integer | Módulo de Elasticidad |
| CM | Byte | Cantidad de miembros |
| CHI | Byte | Contador |
| CHV | Byte | Número de miembros |
| DELI | Integer | Nudo Inicial |
| ALFI | Integer | Nudo Final |
| H | Byte | Estado de carga |
| NMG | Byte | Número de miembros a generar |
| MI | Byte | Incremento al nudo inicial |
| MF | Byte | Incremento al nudo final |
| IB | Byte | Incremento de miembros |
| LPI | Byte | Coordenada en X |
| LPI | Byte | Coordenada en Y |
| LPI | Byte | Coordenada en Z |
| CV | Integer | Carga viva |
| CMU | Integer | Carga muerta |
| CS | Integer | Carga Sísmica |
| MI | Byte | Tipo de miembro |

| | | |
|-------------|------|--------------------------------|
| B | Byte | Base |
| D | Byte | Altura |
| WX | Byte | Carga distribuida en X |
| WY | Byte | Carga distribuida en Y |
| WZ | Byte | Carga distribuida en Z |
| B1 | Real | Base 1 |
| B2 | Real | Base 2 |
| B3 | Real | Base 3 |
| D1 | Real | Altura 1 |
| D2 | Real | Altura 2 |
| D3 | Real | Altura 3 |
| A | Real | Area |
| I | Real | Inersia |
| J | Byte | Contador |
| SWC | Char | Para aceptar datos de ayuda |
| SW | Char | Para aceptar datos de ayuda |
| SECFRO | Char | Para aceptar datos de ayuda |
| CONTROLLEPO | Char | Ayuda |
| GENEL | Char | Ayuda |
| COORDE | Char | Ayuda |
| CAISTA | Char | Ayuda |

CEVAUX

String

Ayuda para grabar
en el archivo sin
espacios en
blanco

Ayuda para grabar
en el archivo sin
espacios en
blanco

DEVAUX

String

Ayuda para grabar
en el archivo sin
espacios en
blanco

Ayuda para grabar
en el archivo sin
espacios en
blanco

ALVAUX

String

Ayuda para grabar
en el archivo sin
espacios en
blanco

Ayuda para grabar
en el archivo sin
espacios en
blanco

HAUX

String

Ayuda para grabar
en el archivo sin
espacios en
blanco

Ayuda para grabar
en el archivo sin
espacios en
blanco

BEVAUX

String

HEVAUX

String

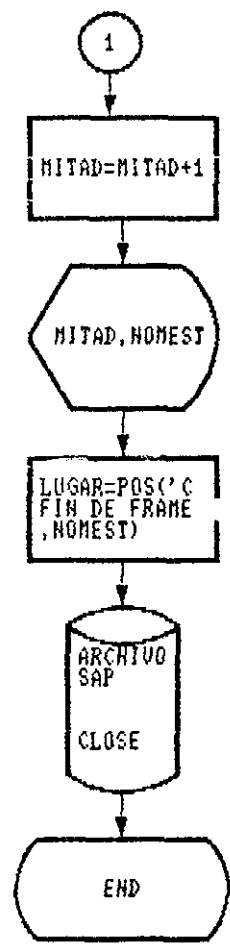
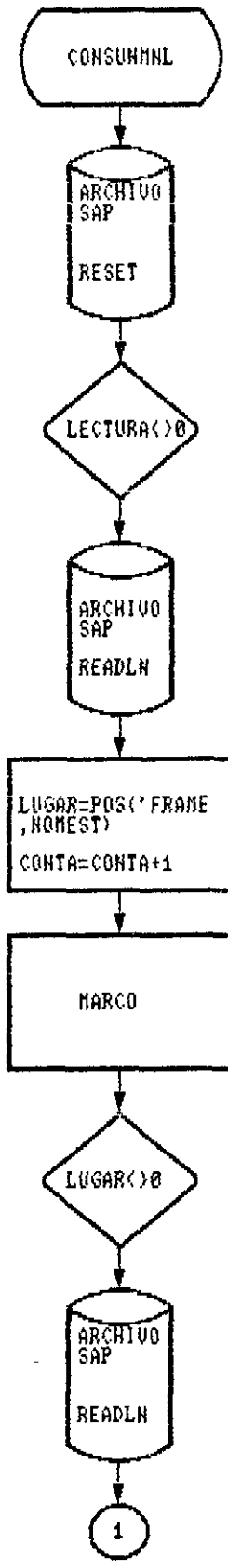
HEVAUX

String

| | | | |
|--------|--------|-------------------|----|
| | | espacios | en |
| | | blanco | |
| IMAUX | String | Ayuda para grabar | |
| | | en el archivo sin | |
| | | espacios | en |
| | | blanco | |
| LP1AUX | String | Ayuda para grabar | |
| | | en el archivo sin | |
| | | espacios | en |
| | | blanco | |
| LP2AUX | String | Ayuda para grabar | |
| | | en el archivo sin | |
| | | espacios | en |
| | | blanco | |
| LP3AUX | String | Ayuda para grabar | |
| | | en el archivo sin | |
| | | espacios | en |
| | | blanco | |
| CVAUX | String | Ayuda para grabar | |
| | | en el archivo sin | |
| | | espacios | en |
| | | blanco | |
| CMUAUX | String | Ayuda para grabar | |
| | | en el archivo sin | |
| | | espacios | en |
| | | blanco | |
| CSAUX | String | Ayuda para grabar | |
| | | en el archivo sin | |

| | | | |
|-------|--------|-------------------|----|
| | | espacios | en |
| | | blanco | |
| WXAUX | String | Ayuda para grabar | |
| | | en el archivo sin | |
| | | espacios | en |
| | | blanco | |
| WYAUX | String | Ayuda para grabar | |
| | | en el archivo sin | |
| | | espacios | en |
| | | blanco | |
| WZAUX | String | Ayuda para grabar | |
| | | en el archivo sin | |
| | | espacios | en |
| | | blanco | |
| AAUX | String | Ayuda para grabar | |
| | | en el archivo sin | |
| | | espacios | en |
| | | blanco | |
| IAUX | String | Ayuda para grabar | |
| | | en el archivo sin | |
| | | espacios | en |
| | | blanco | |
| B1AUX | String | Ayuda para grabar | |
| | | en el archivo sin | |
| | | espacios | en |
| | | blanco | |
| B2AUX | String | Ayuda para grabar | |
| | | en el archivo sin | |
| | | espacios | en |

| | | |
|-------|--------|---|
| B3AUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| D1AUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| D2AUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| D3AUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| ADRI1 | Byte | Para grabar |
| ADRI2 | Byte | Para grabar |
| ADRI3 | Byte | Para grabar |
| ADRI4 | Byte | Para grabar |
| LON | Byte | Para tomar una variable sin espacios |



| NOMBRE | TIPO | DESCRIPCION |
|--------|---------|----------------------------|
| E | Integer | Módulo de elasticidad |
| CM | Byte | Cantidad de miembros |
| CM1 | Byte | Contador |
| CMV | Byte | Número de miembros |
| EELN | Integer | Nudo Inicial |
| ALN | Integer | Nudo Final |
| M | Byte | Estado de carga |
| MM1 | Byte | Número de miembros |
| | | generar |
| INI | Byte | Incremento al nudo inicial |
| INF | Byte | Incremento al nudo final |
| IH | Byte | Incremento de miembros |
| LP1 | Byte | Coordenada en X |
| LP2 | Byte | Coordenada en Y |
| LPS | Byte | Coordenada en Z |
| CV | Integer | Carga viva |
| CM1 | Integer | Carga muerta |
| CS | Integer | Carga Sismica |
| NM | Byte | Tipo de miembro |

| | | |
|-------------|------|--------------------------------|
| NL | Byte | Estado de carga |
| B | Byte | Base |
| D | Byte | Altura |
| WX | Byte | Carga distribuida en X |
| WY | Byte | Carga distribuida en Y |
| WZ | Byte | Carga distribuida en Z |
| B1 | Real | Base 1 |
| B2 | Real | Base 2 |
| B3 | Real | Base 3 |
| D1 | Real | Altura 1 |
| D2 | Real | Altura 2 |
| D3 | Real | Altura 3 |
| A | Real | Area |
| I | Real | Inersia |
| J | Byte | Contador |
| SWS | Char | Para aceptar datos de ayuda |
| SW | Char | Para aceptar datos de ayuda |
| SECPRO | Char | Para aceptar datos de ayuda |
| CONTROLTIPO | Char | Ayuda |
| GENER | Char | Ayuda |
| COORDE | Char | Ayuda |
| CARGA | Char | Ayuda |

| | | |
|---------|--------|---|
| | | en el archivo sin espacios en blanco |
| CFVAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| DELNAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| ALNAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| MAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| NHGAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| INIAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| INFAUX | String | Ayuda para grabar |

| | | |
|--------|--------|---|
| | | espacios en |
| | | blanco |
| IMAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| LP1AUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| LP2AUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| LP3AUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| CVAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| CMUAUX | String | Ayuda para grabar en el archivo sin espacios en blanco |
| CSAUX | String | Ayuda para grabar en el archivo sin |

espacios en
blanco

WXDX String Ayuda para grabar
en el archivo sin
espacios en
blanco

WYDX String Ayuda para grabar
en el archivo sin
espacios en
blanco

WZDX String Ayuda para grabar
en el archivo sin
espacios en
blanco

AXDX String Ayuda para grabar
en el archivo sin
espacios en
blanco

IAUX String Ayuda para grabar
en el archivo sin
espacios en
blanco

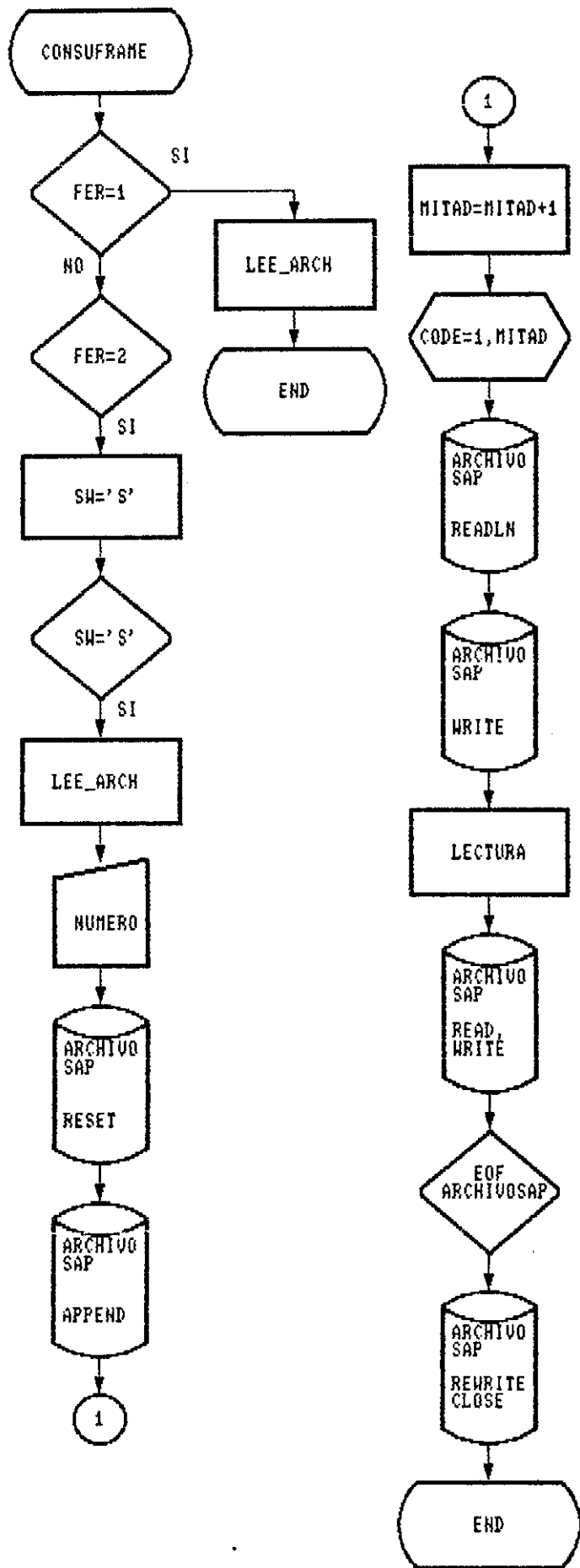
BIAUX String Ayuda para grabar
en el archivo sin
espacios en
blanco

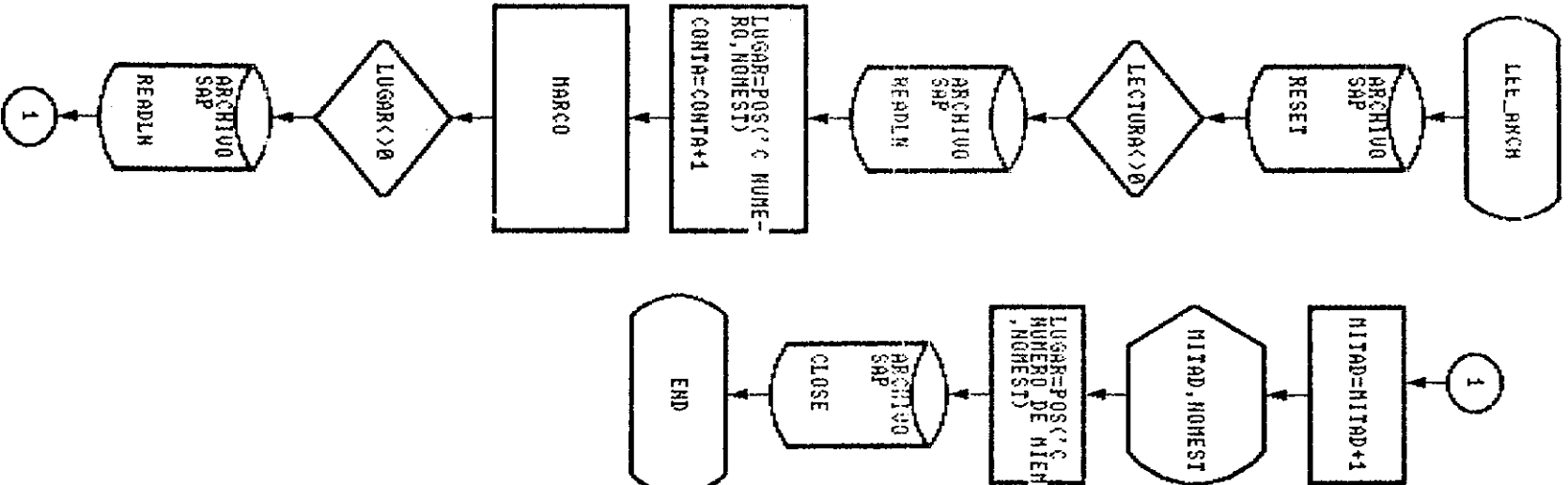
BEAUX String Ayuda para grabar
en el archivo sin
espacios en

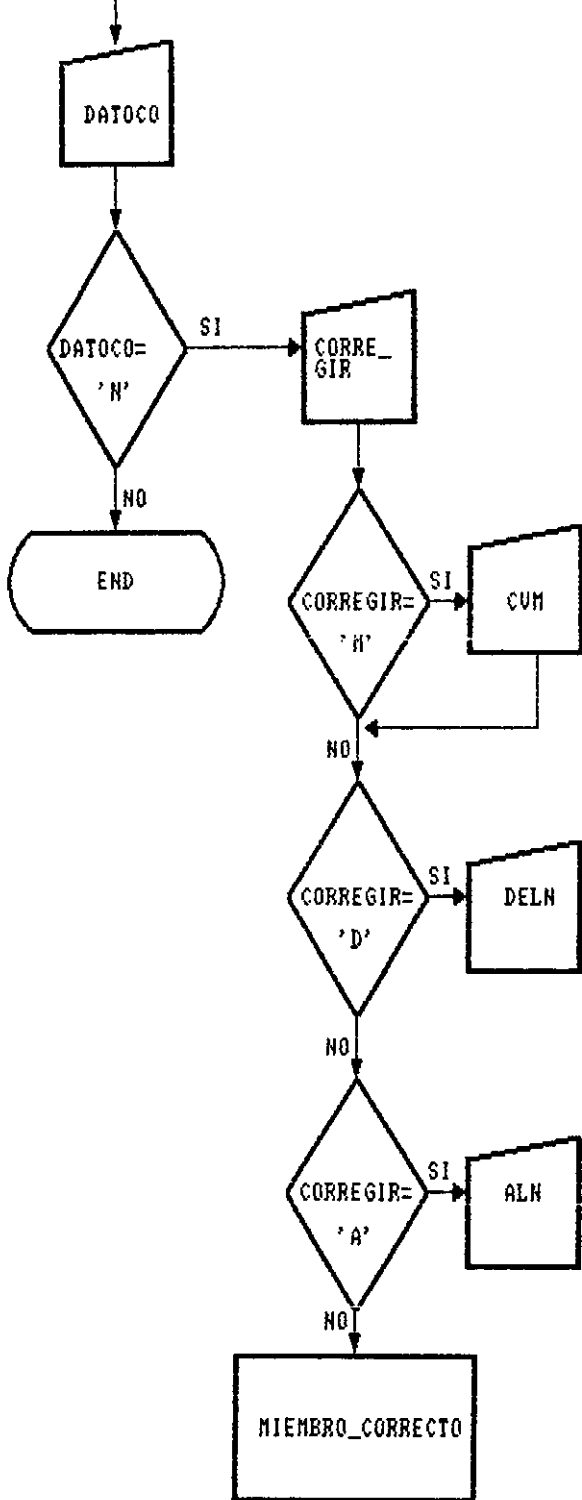
| | | | |
|-------|--------|---|--|
| | | blanco | |
| ESAOX | String | Ayuda para grabar en el archivo sin espacios en blanco | |
| PIAOX | String | Ayuda para grabar en el archivo sin espacios en blanco | |
| D2AOX | String | Ayuda para grabar en el archivo sin espacios en blanco | |
| D3AOX | String | Ayuda para grabar en el archivo sin espacios en blanco | |
| APRI1 | Byte | Para grabar | |
| APRI2 | Byte | Para grabar | |
| APRI3 | Byte | Para grabar | |
| APRI4 | Byte | Para grabar | |
| ION | Byte | Para tomar una variable sin espacios | |

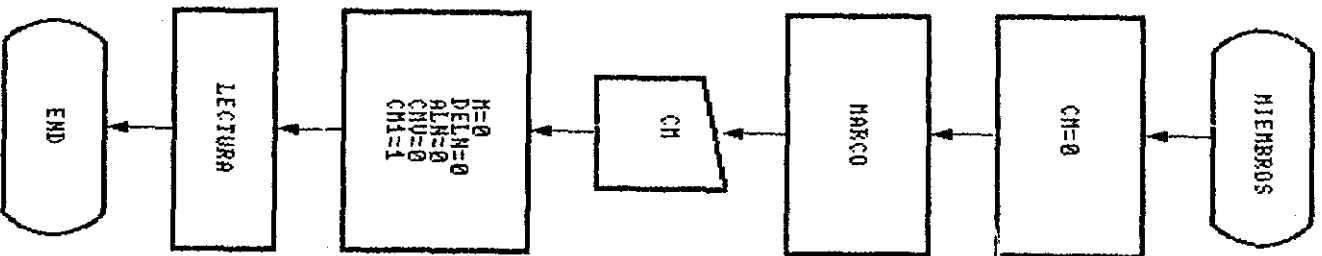
NOMBRE DEL PROGRAMA: REPORTE

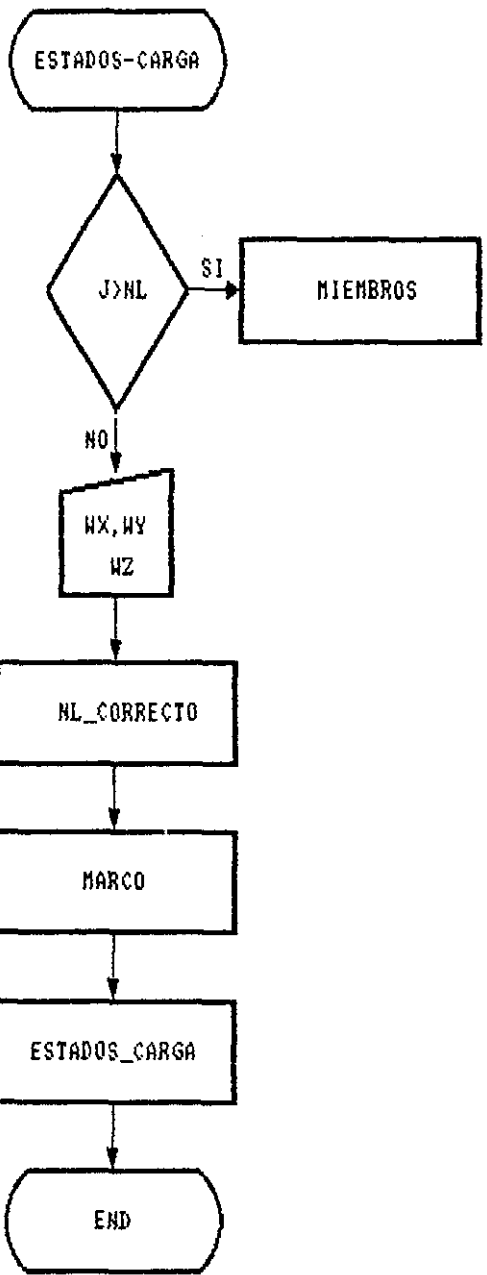
| NOMBRE | TIPO | DESCRIPCION |
|------------|--------|-------------------------|
| SW | Char | Switch de ayuda |
| CONS | Char | Para consultar |
| NOMEST | String | Para leer archivo |
| ARCHIVOCAF | Text | Variable de archivo. |





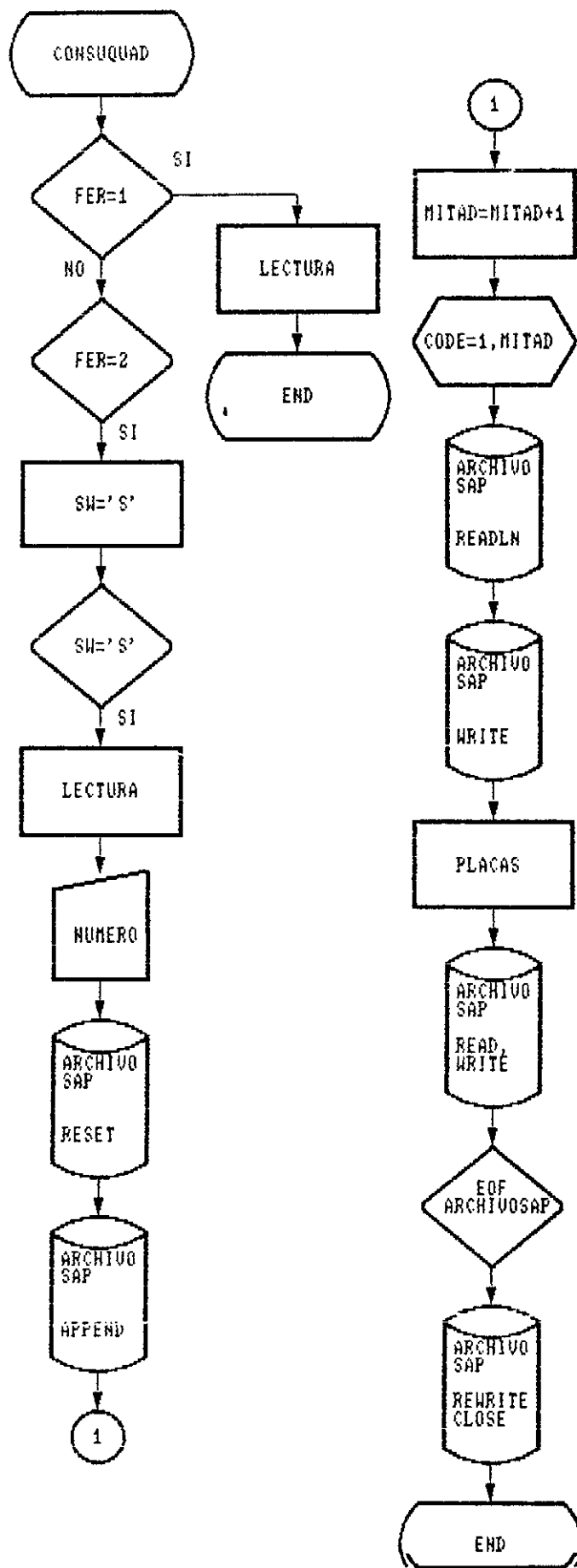


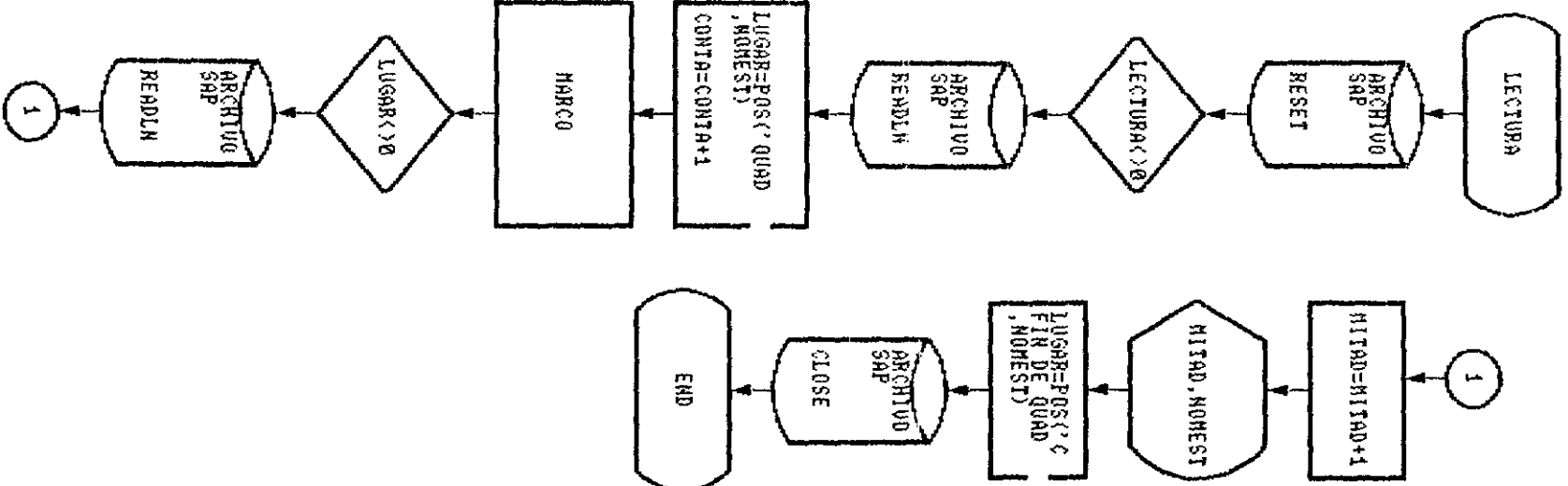


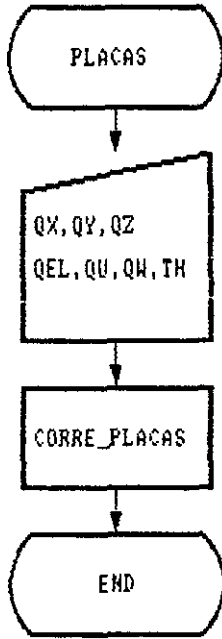


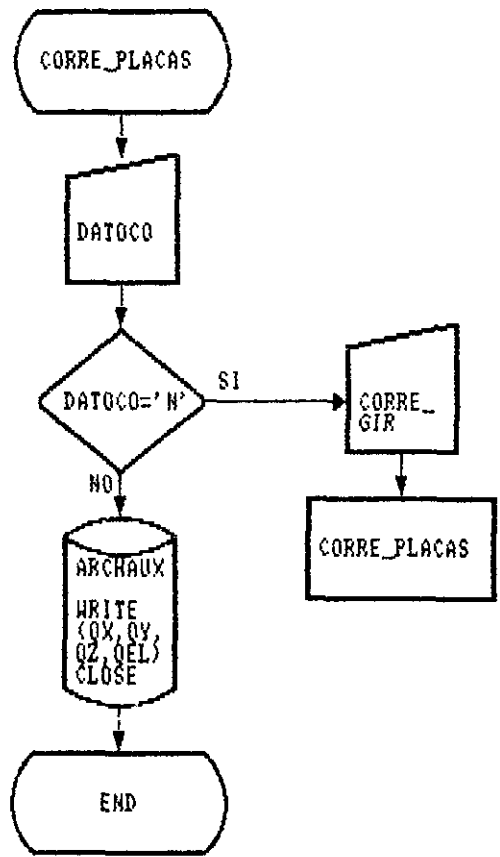
NOMBRE DEL PROGRAMA: CONSUCUAD

| NOMBRE | TIPO | DESCRIPCION |
|--------|---------|------------------------------|
| NMAT | Byte | Número materiales a declarar |
| LON | Byte | Para grabar sin espacios |
| QEL | Integer | Módulo de elasticidad |
| QU | Integer | Módulo de Poisson |
| QW | Integer | Peso, volumen |
| QX | Real | Factor gravitacional X |
| QY | Real | Factor gravitacional Y |
| QZ | Real | Factor gravitacional Z |
| TH | Real | Espesor |
| QXAUX | String | Para grabar |
| QYAUX | String | Para grabar |
| QZAUX | String | Para grabar |
| THAUX | String | Para grabar |
| QELAUX | String | Para grabar |
| QUAUX | String | Para grabar |
| QWAUX | String | Para grabar |









LISTA DE VARIABLES

NOMBRE DEL PROGRAMA: CONEULCAD

| NOMBRE | TIPO | DESCRIPCION |
|--------|---------|--------------------------------------|
| PRIJUN | Integer | Número de la primera junta |
| SEGJUN | Byte | Número de la segunda junta |
| INNUDO | Byte | Incremento en el número de junta |
| ESTAR | Byte | Estado de carga |
| LON | Byte | Para grabar sin espacios |
| JLOAD | Byte | Contador |
| CONTAX | Byte | Contador |
| FX | Real | Fuerza aplicada en dirección X |
| FY | Real | Fuerza aplicada en dirección Y |
| FZ | Real | Fuerza aplicada en dirección Z |
| MX | Real | Momento aplicado alrededor del eje X |
| MY | Real | Momento aplicado alrededor del eje Y |
| MZ | Real | Momento aplicado alrededor del eje Z |

| | | | |
|----------|--------|-------------------|----|
| | | aplicado | en |
| | | dirección X | |
| BY | Real | Desplazamiento | |
| | | aplicado | en |
| | | dirección Y | |
| BZ | Real | Desplazamiento | |
| | | aplicado | en |
| | | dirección Z | |
| EX | Real | Rotación aplicada | |
| | | alrededor del eje | |
| | | X | |
| RY | Real | Rotación aplicada | |
| | | alrededor del eje | |
| | | Y | |
| RZ | Real | Rotación aplicada | |
| | | alrededor del eje | |
| | | Z | |
| FRIJUNAU | String | Para grabar | |
| SEGJUNAU | String | Para grabar | |
| INNUDCAU | String | Para grabar | |
| ESTCARAU | String | Para grabar | |
| FXAU | String | Para grabar | |
| FYAU | String | Para grabar | |
| FZAU | String | Para grabar | |
| MXAU | String | Para grabar | |
| MYAU | String | Para grabar | |
| MZAU | String | Para grabar | |
| DXAU | String | Para grabar | |

DZAUX

String

Para grabar

KXAUX

String

Para grabar

RYAUX

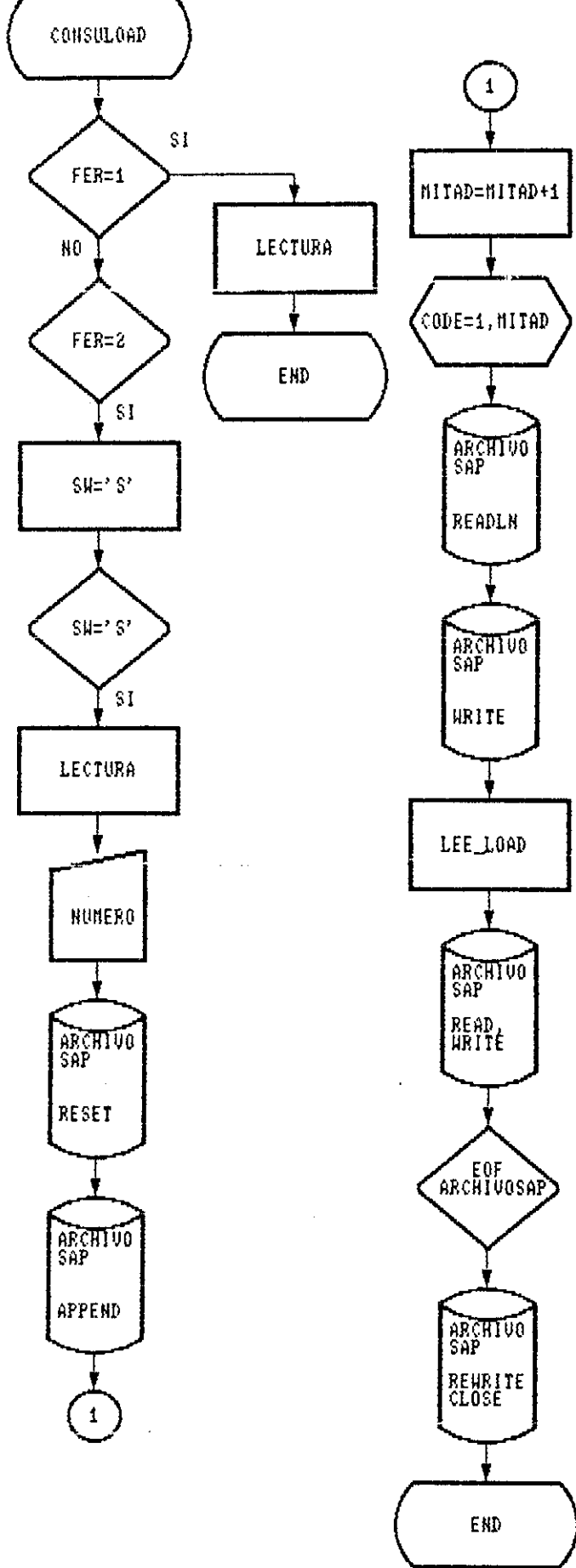
String

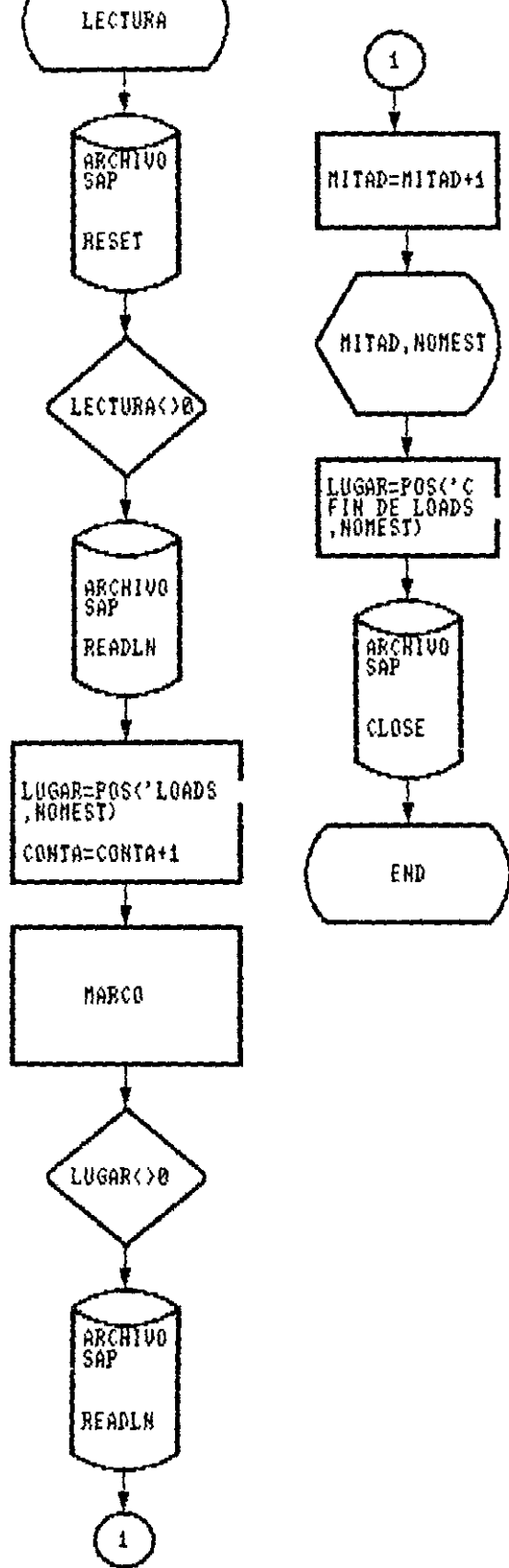
Para grabar

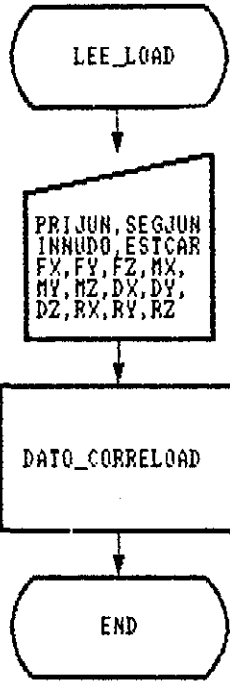
RZAUX

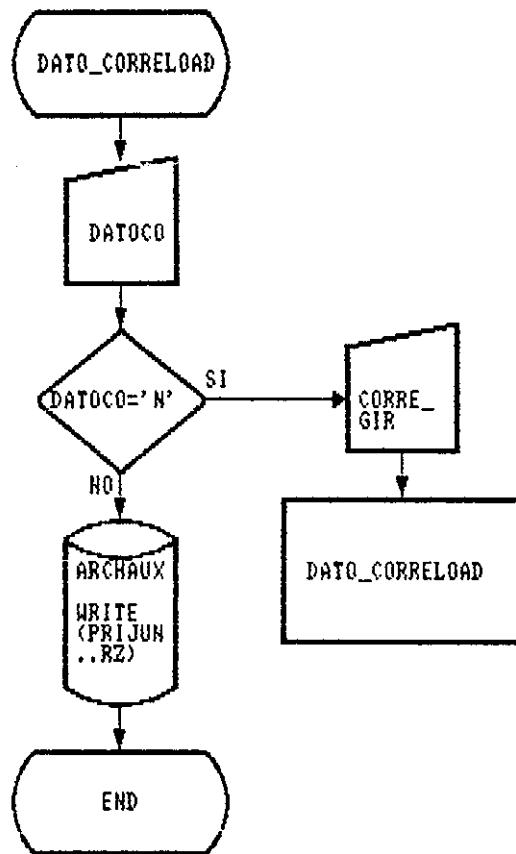
String

Para grabar





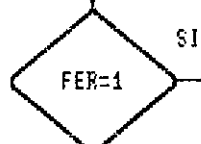
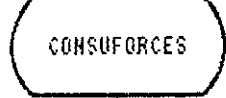




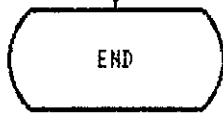
LISTA DE VARIABLES

NOMBRE DEL PROGRAMA: OPERACIONES

| NOMBRE | TIPO | DESCRIPCION |
|---------|--------|---------------------------|
| OP | Byte | Cantidad de combinaciones |
| CONT | Byte | Contador |
| LDN | Byte | Para grabar sin espacio |
| OP1 | Real | Factor de carga 1 |
| OP2 | Real | Factor de carga 2 |
| OP3 | Real | Factor de carga 3 |
| J1TABX | String | Para grabar |
| OP1TABX | String | Para grabar |
| OP2TABX | String | Para grabar |
| OP3TABX | String | Para grabar |



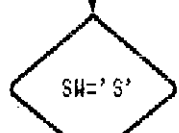
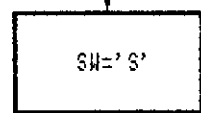
SI



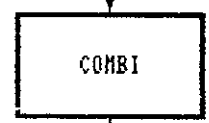
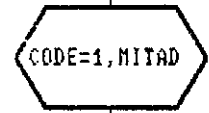
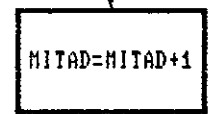
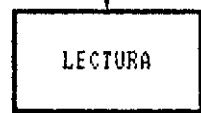
NO

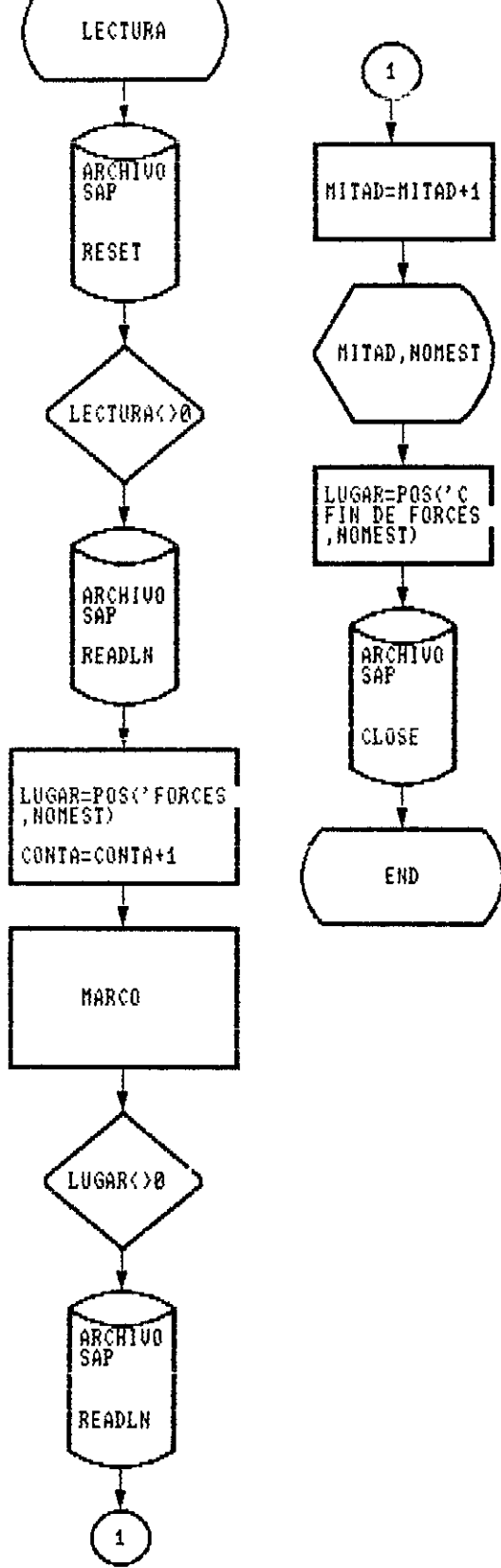


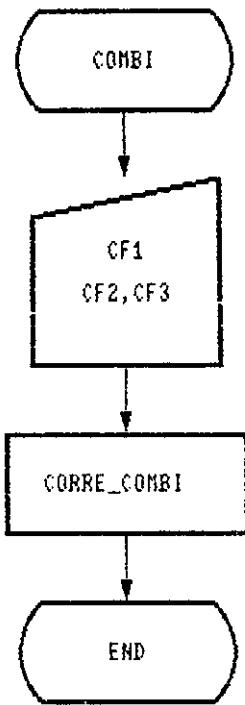
SI

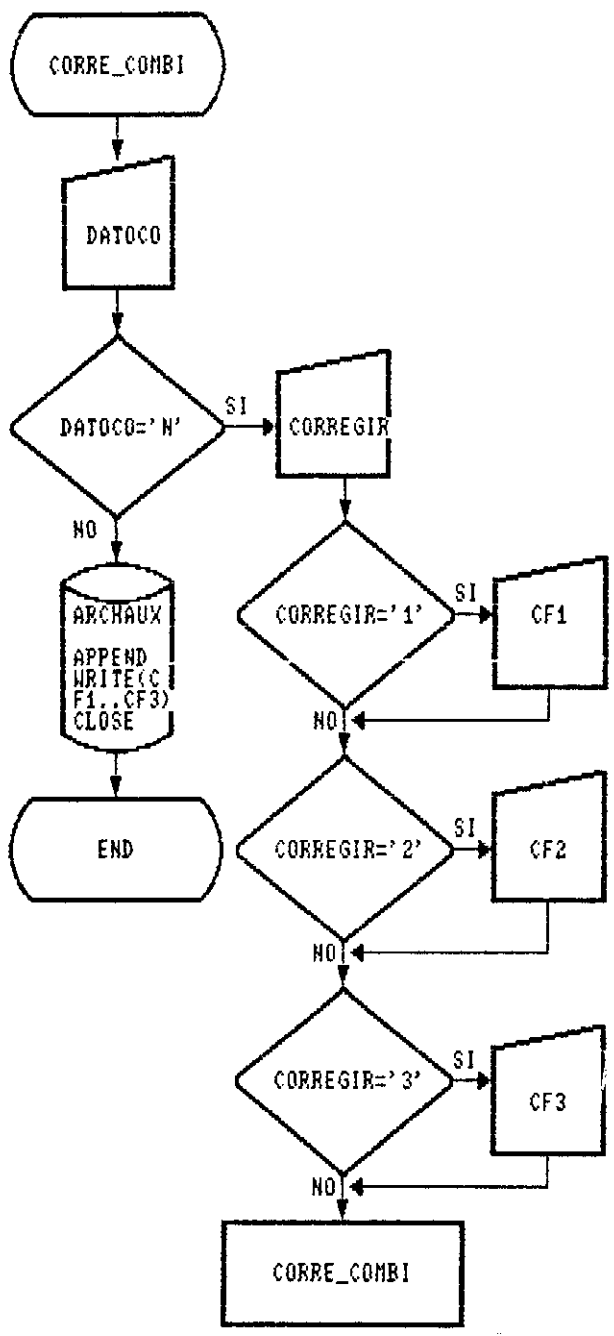


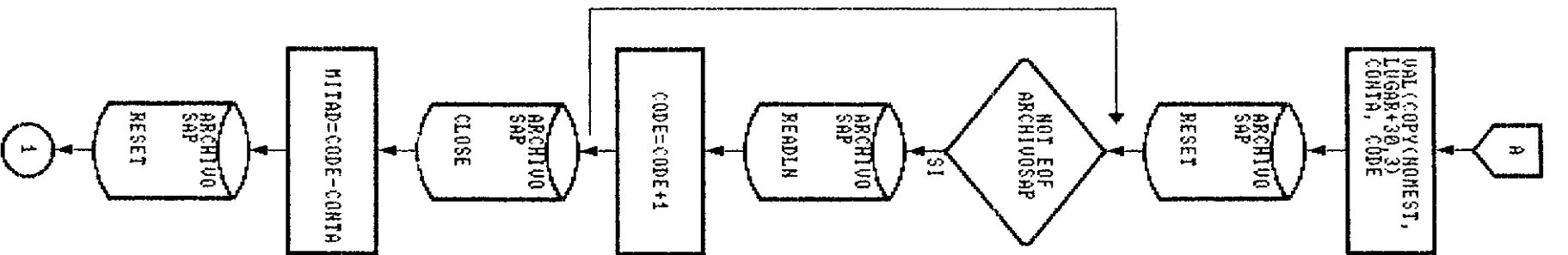
SI



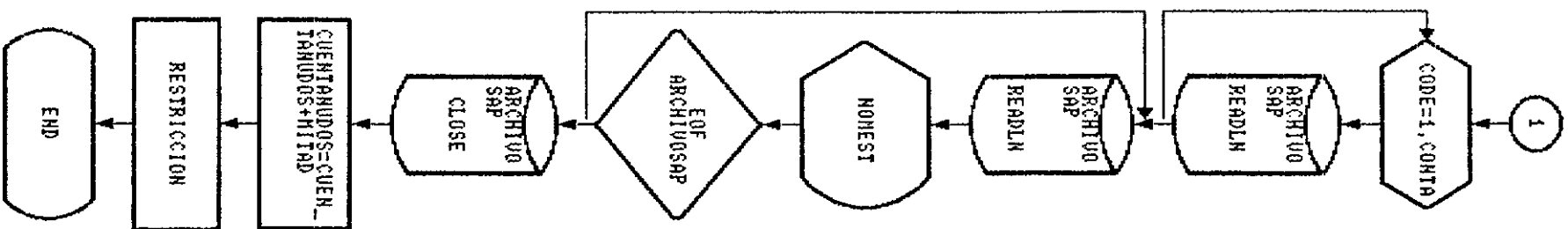
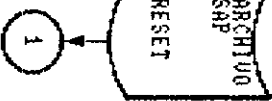






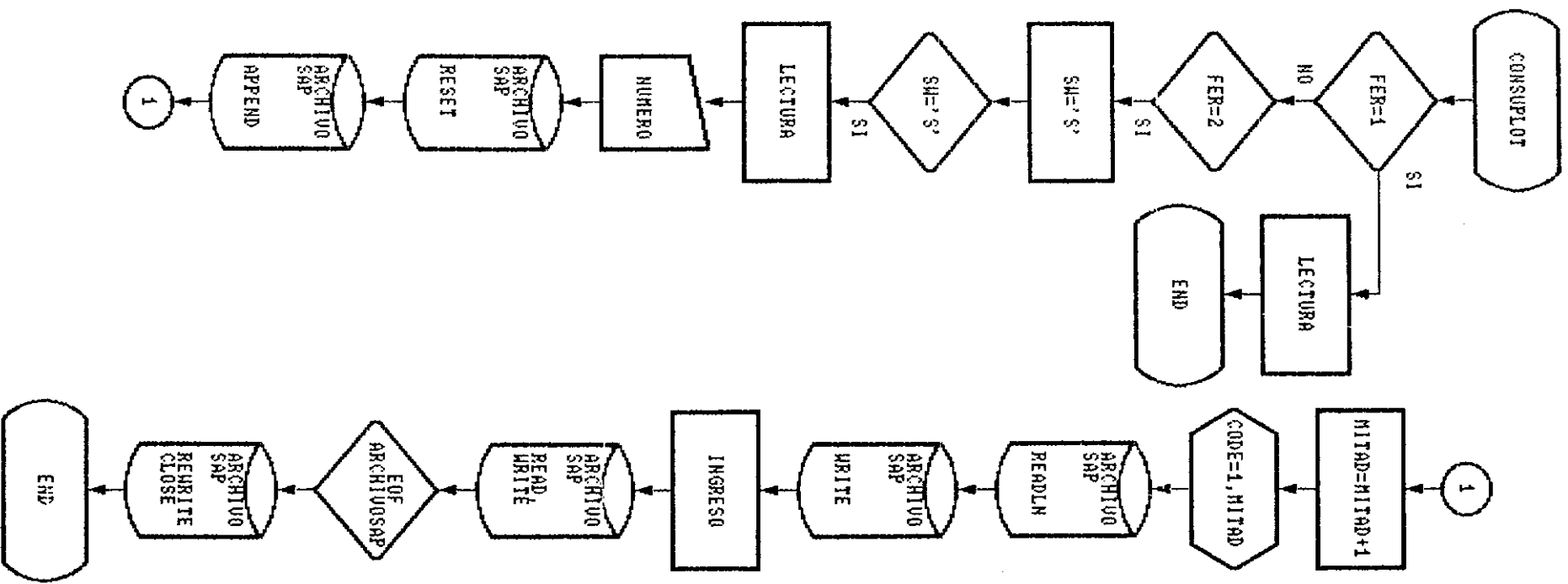


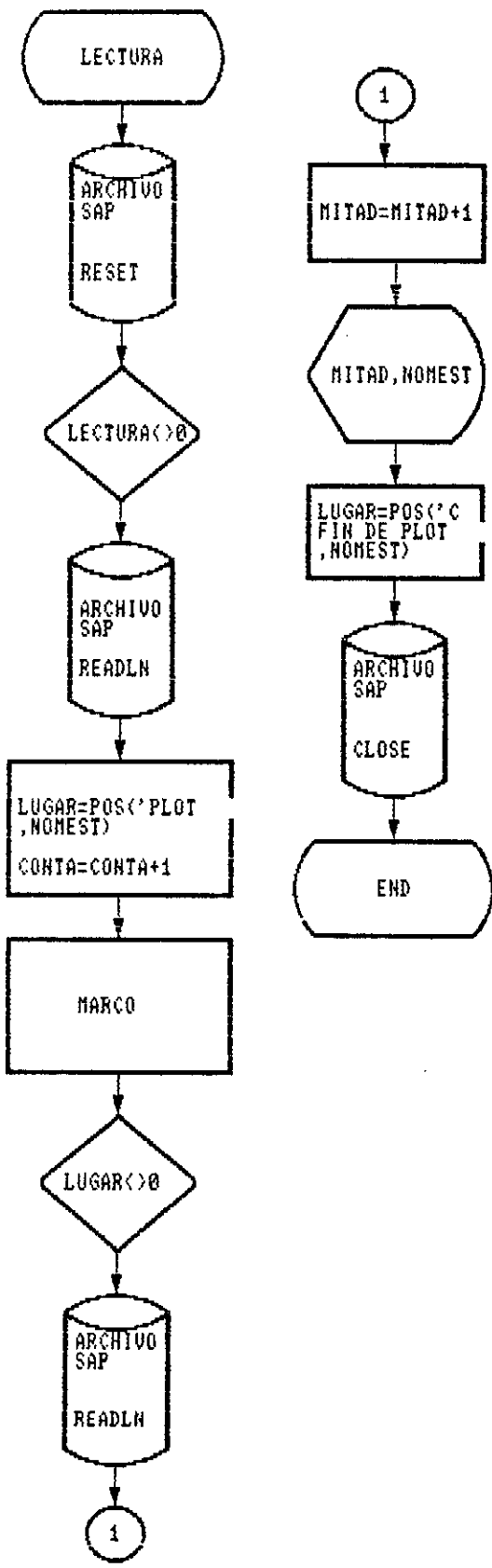
MITAD=CODE-CONTA

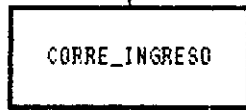
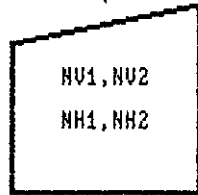


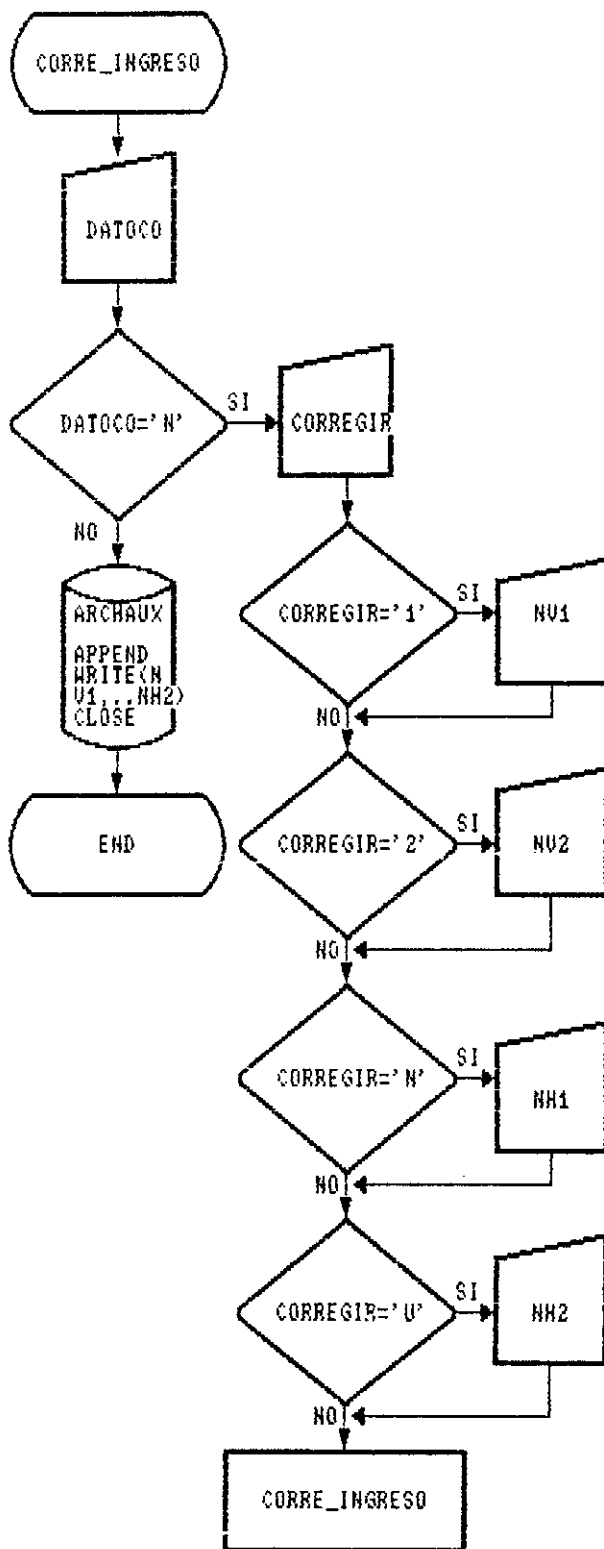
NOMBRE DEL PROGRAMA: CONSUELDT

| NOMBRE | TIPO | DESCRIPCION |
|--------|---------|----------------------------|
| NV1 | Integer | Vector vista nudo1 |
| NV2 | Integer | Vector vista nudo2 |
| NH1 | Integer | Vector horizontal nudo1 |
| NH2 | Integer | Vector horizontal nudo2 |









```
UNIT IMPRESION;
INTERFACE
USES CRT,PRINTER,CARATULA;
```

```
TYPE
```

```
  STR10=STRING#14A;
VAR
  NOMBRE:SIK10;
  SW,CONS:CHAR;
  NOME1:STRING#80A;
  ARCHIVOSAP:TEXT;
```

```
PROCEDURE REPORTE;
```

```
IMPLEMENTATION
```

```
PROCEDURE REPORTE;
```

```
PROCEDURE NOMBRE_ARCH(VAR NOMBRE:STR10);
```

```
VAR
```

```
  ARCHIVOSAP:TEXT;
  sw1:char;
  NOMBRE1:STRING#80A;
  UNIDAD:CHAR;
```

```
BEGIN
```

```
  CLRSCR;
  CUADRO;
  GOTOXY(15,8); WRITE('Ingrese la Unidad de Disco: ==> ');
  GOTOXY(15,10);WRITE('Nombre del Archivo      : ==> ');
  TEXTCOLOR(14);
  GOTOXY(15,8); WRITE('I');GOTOXY(26,8);WRITE('U');GOTOXY(36,8); WRITE('D');
  GOTOXY(15,10);WRITE('N');GOTOXY(26,10);WRITE('A');
  TEXTCOLOR(7);
  SW1:='N';
  WHILE (SW1='N') OR (SW1='n') DO
  BEGIN
    REPEAT
      TEXTCOLOR(14);
      GOTOXY(49,8); WRITE('■');
      TEXTCOLOR(7);
      GOTOXY(5,22);WRITE('La unidad de disco puede ser: A: B: C: ó D:');
      TEXTCOLOR(14);
      GOTOXY(8,22);WRITE('U');GOTOXY(18,22);WRITE('D'); GOTOXY(35,22);WRITE('A: B: C:');
      GOTOXY(50,22);WRITE('D:');
      GOTOXY(49,8);UNIDAD:=READKEY;
      TEXTCOLOR(7);
      CASE UNIDAD OF
        'A','B','C','D','a','b','c','d':
          BEGIN
            TEXTCOLOR(14);
            GOTOXY(49,8);WRITE(UNIDAD);
            TEXTCOLOR(7);
          END;
        #Del Case#
      UNTIL (UNIDAD='A') OR (UNIDAD='a') OR (UNIDAD='B') OR (UNIDAD='b') OR (UNIDAD='C') OR (UNIDAD='c')
        OR (UNIDAD='D') OR (UNIDAD='d');
      GOTOXY(5,22);CLREOL;
      GOTOXY(5,23);CLREOL;
      GOTOXY(25,23);WRITE('Su Ingreso es Correcto S/N: ■');
      GOTOXY(48,23);WRITE('S');GOTOXY(50,23);WRITE('N');
```

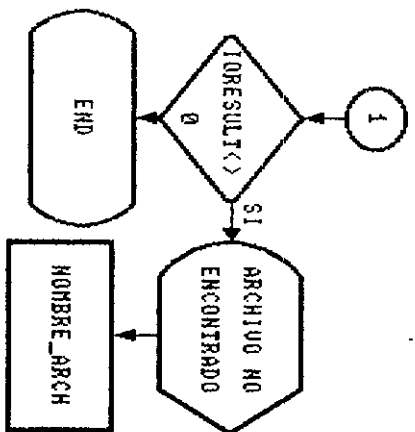
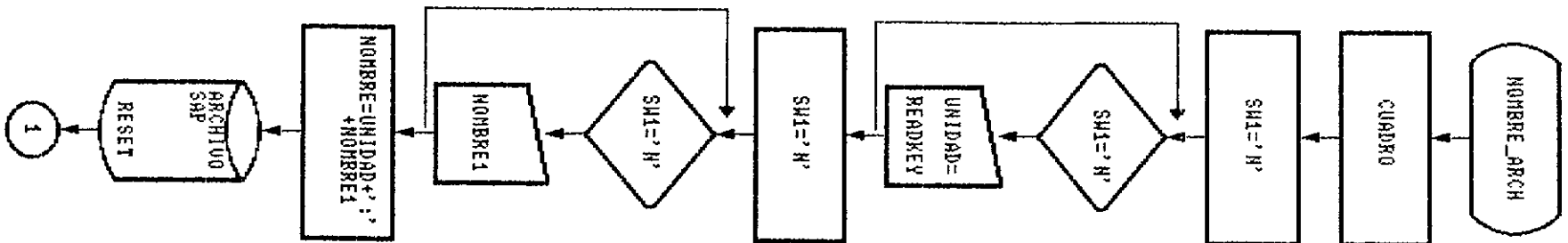
```

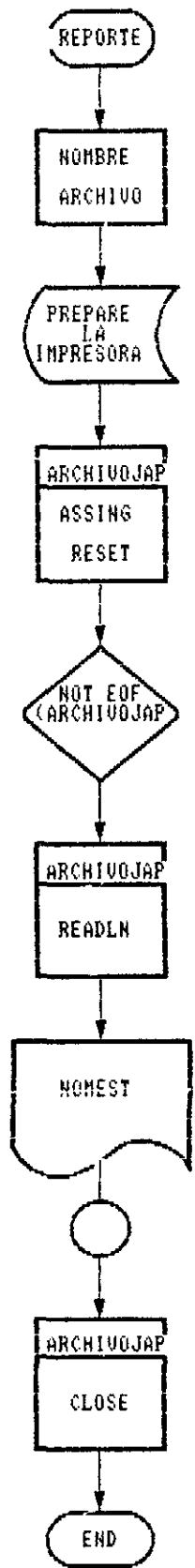
TEXTCOLOR(7);
END; aWHILE (SW1='N')a
SW1:='N';
WHILE (SW1='N') OR (SW1='n') DO
BEGIN
GOTOXY(5,23);WRITE('El nombre del archivo debe tener máximo 8 caracteres');
TEXTCOLOR(14);
GOTOXY(49,10);WRITE('#####');
GOTOXY(8,23);WRITE('N');GOTOXY(19,23);WRITE('A');GOTOXY(45,23);WRITE('B C');
GOTOXY(49,10);READLN(NOMBRE1);
TEXTCOLOR(7);
GOTOXY(5,22);CLREOL;
GOTOXY(5,23);CLREOL;
GOTOXY(25,23);WRITE('Su Ingreso es Correcto S/N: #');
TEXTCOLOR(14);
GOTOXY(40,23);WRITE('S');GOTOXY(50,23);WRITE('N');
GOTOXY(53,23);SW1:=READKEY;
TEXTCOLOR(7);
END; aWHILE (SW1='N')a
NOMBRE:=UNIDAD+' '+NOMBRE1;
ASSIGN(ARCHIVOSAP,NOMBRE);
a$I-a
RESET(ARCHIVOSAP);
a$I+a
IF IORESULT <>0 THEN
BEGIN
REPEAT
TEXTCOLOR(14);
GOTOXY(5,23);WRITE('ARCHIVO NO ENCONTRADO...PRESIONE UNA TECLA E INGRESE OTRO NOMBRE DE ARCHIVO');
TEXTCOLOR(7);
UNTIL KEYPRESSED;
NOMBRE_ARCH(NOMBRE);
END;
end; aDEL PROCEDURE Nombre_archa

BEGIN
NOMBRE_ARCH(NOMBRE);
GOTOXY(5,23);WRITE(' ');
REPEAT
TEXTCOLOR(14);
GOTOXY(10,23);WRITE('PREPARE LA IMPRESORA Y PRESIONE UNA TECLA PARA CONTINUAR');
TEXTCOLOR(7);
UNTIL KEYPRESSED;
ASSIGN(ARCHIVOSAP,NOMBRE);
RESET(ARCHIVOSAP);
WHILE NOT EOF(ARCHIVOSAP) DO
BEGIN
READLN(ARCHIVOSAP,NOMEST);
WRITELN(LST,NOMEST);
END;
CLOSE(ARCHIVOSAP);

END;aDEL PROCEDUREa
END. aDE LA UNIDADa

```





NOMBRE DEL PROGRAMA: UNIFUN

| NOMBRE | TIPO | DESCRIPCION |
|---------|----------|--------------------------------|
| ENT | Byte | Aceptación de enteros |
| DEC | Byte | Aceptacion de decimales |
| I | Byte | Contador |
| BANDERA | Byte | Contador |
| SIG | Shortint | Contador |
| TEMP | Longint | Contador |
| NUM | Integer | Almacena número a ser devuelto |
| CH | Char | Contador auxiliar |

Unit Unifun;

Interface

uses crt;

var

ent,dec,i,bandera:byte;
sig:shortint;
temp:longint;
num:integer;
ch:char;

procedure WriteXY(x,y:integer; mensaje:string);
function FunInEnt(columna,posicion:integer):integer;
function FunInEnt1(columna,posicion:integer):integer;
function FunInDec(columna,posicion:integer):real;
function FunInDec1(columna,posicion:integer):real;
function FunInDec2(columna,posicion:integer):real;
function FunRes(columna,posicion:integer):integer;
function FunFinDec(columna,posicion:integer):real;
function FunKudoFin(columna,posicion:integer):integer;

Implementation

procedure WriteXY;

begin

if (x in #1..#80A) and (y in #1..#25A) then
begin
gotoxy(x,y); write(mensaje)
end
else
writexy(1,25,'Coordenadas mal dadas')

end; aFin del procedimiento WriteXY

function FunInEnt(columna,posicion:integer):integer;

begin

num:=0; i:=1; ent:=1; bandera:=0; temp:=1; sig:=1;
gotoxy(columna,posicion); write('#####'); gotoxy(columna,posicion);
if ch=chr(13) then writexy(columna,posicion,'0');
while ch<>chr(13) do

begin

ch:=readkey;

case ch of

'0'..'9': begin

gotoxy(columna-1+i,posicion); write(ch);

num:=num*10+ord(ch)-48;

i:=i+1; ent:=ent+1;

end;

chr(8) : begin

i:=i-1; ent:=ent-1;

if i<>0 then

begin

gotoxy(columna-1+i,posicion); write(' ',ch);

end;

if ent>1 then

num:=trunc(num/10)

else

begin

ent:=1; i:=1; sig:=1; num:=0;

sound(100); delay(100); nosound; gotoxy(columna,posicion);

end; zif

end;

end; aEnd_cased

```

end; endwhile
num:=num*sig;
FunInEnt:=num;
if se quiere imprimir NUMa
ch:= ' ';
end; if FunInEnta

function FunInEnt1(columna,posicion:integer):integer;
begin
num:=0; i:=1; ent:=1; bandera:=0; temp:=1; sig:=1;
gotoxy(columna,posicion);write('#####');gotoxy(columna,posicion);
while ch<>chr(13) do
begin
ch:=readkey;
case ch of
'0'..'9': begin
gotoxy(columna-1+i,posicion); write(ch);
num:=num*10+ord(ch)-48;
i:=i+1; ent:=ent+1;
end;
chr(8) : begin
i:=i-1; ent:=ent-1;
if i<>0 then
begin
gotoxy(columna-1+i,posicion); write(' ',ch);
end;
if ent>1 then
num:=trunc(num/10)
else
begin
ent:=1; i:=1; sig:=1; num:=0;
sound(100); delay(100); nosound; gotoxy(columna,posicion);
end; ifa
end;
end; if End_casea
end; endwhile
num:=num*sig;
FunInEnt1:=num;
ch:= ' ';
end; if FunInEnt1a

```

```

function FunInDec(columna,posicion:integer):real;
var
num:real;
begin
num:=0; i:=1; ent:=1; bandera:=0; temp:=1; sig:=1;
gotoxy(columna,posicion); write('#####'); gotoxy(columna,posicion);
while ch<>chr(13) do
begin
ch:=readkey;
if bandera=0 then
begin
case ch of
'-': begin
if i<=1 then
begin
gotoxy(columna-1+i,posicion); write(ch);
i:=i+1; ent:=ent+1;
sig:=-1;
end; ifa
end;
end;

```



```

begin
    num:=0; i:=1; ent:=1; bandera:=0; temp:=1; sig:=1;
    gotoxy(columna,position); write(' '); gotoxy(columna,position);
    while ch<>chr(13) do
        begin
            ch:=readkey;
            if bandera=0 then
                begin
                    case ch of
                        '-': begin
                            if i<=1 then
                                begin
                                    gotoxy(columna-1+i,position); write(ch);
                                    i:=i+1; ent:=ent+1;
                                    sig:=-1;
                                end;
                            zifa
                        end;
                        '0'..'9': begin
                                    gotoxy(columna-1+i,position); write(ch);
                                    num:=num*10+ord(ch)-48;
                                    i:=i+1; ent:=ent+1;
                                end;
                        chr(8) : begin
                                    i:=i-1; ent:=ent-1;
                                    if i<>0 then
                                        begin
                                            gotoxy(columna-1+i,position); write(' '); ch);
                                        end;
                                    if ent>1 then
                                        num:=trunc(num/10)
                                    else
                                        begin
                                            ent:=1; i:=1; sig:=1; num:=0;
                                            sound(100); delay(100); nosound; gotoxy(columna,position);
                                            end;
                                        zifa
                                    end;
                                ' ': begin
                                    bandera:=1;
                                    gotoxy(columna-1+i,position); write(ch); i:=i+1; dec:=1;
                                end;
                            end_end_case
                        end_end_ifa
                    else
                        begin
                            case ch of
                                '0'..'9': begin
                                    gotoxy(columna-1+i,position); write(ch);
                                    temp:=temp*10;
                                    num:=num+(ord(ch)-48)/temp;
                                    i:=i+1; dec:=dec+1;
                                end;
                                chr(8) : begin
                                    i:=i-1; dec:=dec-1;
                                    gotoxy(columna-1+i,position); write(' '); ch);
                                    if dec<=0 then
                                        begin
                                            bandera:=0;
                                            sig:=1;
                                        end
                                    end
                                else

```

```

begin
    temp:=temp div 10;
    num:=trunc(num*temp); num:=num/temp;
end zifá

end;
end; aEnd_casá
end; aelseá
end; awhileá
num:=num*sig;
FunInDec1:=num;
ch:=' ';
end; afunction FunInDec1á

function FunInDec2(columna, posicion:integer):real;
var
    num:real;
begin
    num:=0; i:=1; ent:=1; bandera:=0; temp:=1; sig:=1;
    gotoxy(columna, posicion); write('#####'); gotoxy(columna, posicion);
    while ch<>chr(13) do
        begin
            ch:=readkey;
            if bandera=0 then
                begin
                    case ch of
                        '-': begin
                            if i<=1 then
                                begin
                                    gotoxy(columna-1+i, posicion); write(ch);
                                    i:=i+1; ent:=ent+1;
                                    sig:=-1;
                                end; zifá
                            end;
                        '0'..'9': begin
                            gotoxy(columna-1+i, posicion); write(ch);
                            num:=num*10+ord(ch)-48;
                            i:=i+1; ent:=ent+1;
                        end;
                        chr(8) : begin
                            i:=i-1; ent:=ent-1;
                            if i<>0 then
                                begin
                                    gotoxy(columna-1+i, posicion); write('#####', ch);
                                end;
                            if ent>1 then
                                num:=trunc(num/10)
                            end;
                        else
                            begin
                                ent:=1; i:=1; sig:=1; num:=0;
                                sound(100); delay(100); nosound; gotoxy(columna, posicion);
                            end; zifá
                        end;
                    end;
                end;
            bandera:=1;
            gotoxy(columna-1+i, posicion); write(ch); i:=i+1; dec:=1;
        end;
    end aEnd_casá
end
else
begin

```

```

        end; xdel If else Banderaa
    end; xdel If ch= ' ' labela
end; xwhilea
num:=num+sig;
20:FunFinDec:=num;
ch:= ' ';
end; xProgramaa

```

```

function FunNudoFin(columna, posicion: integer): integer;

```

```

label l0;

```

```

begin

```

```

    num:=0; i:=1;

```

```

    gotoxy(columna, posicion); write('███'); gotoxy(columna, posicion);

```

```

    while ch(<>chr(13)) do

```

```

        begin

```

```

            ch:=readkey;

```

```

            if ((ch= ' ') and (i=1)) then

```

```

                begin

```

```

                    num:=maxint;

```

```

                    goto l0;

```

```

                end

```

```

            else

```

```

                begin

```

```

                    case ch of

```

```

                        '0'..'9': begin

```

```

                            gotoxy(columna-1+i, posicion); write(ch);

```

```

                            num:=num*10+ord(ch)-48;

```

```

                            i:=i+1;

```

```

                        end;

```

```

                        chr(8) : begin

```

```

                            i:=i-1; ent:=ent-1;

```

```

                            if i(<>0) then

```

```

                                begin

```

```

                                    gotoxy(columna-1+i, posicion); write('█', ch);

```

```

                                end;

```

```

                                if ent>1 then

```

```

                                    num:=trunc(num/10)

```

```

                                else

```

```

                                    begin

```

```

                                        ent:=1; i:=1; sig:=1; num:=0;

```

```

                                        sound(100); delay(100); nosound; gotoxy(columna, posicion);

```

```

                                    end; xifa

```

```

                                end;

```

```

                            end; xDel Casea

```

```

                        end; xend_if elsea

```

```

                    end; xDel Whilea

```

```

l0: FunNudoFin:=num;

```

```

    ch:= ' ';

```

```

end; xfunction FunNudoFina

```

```

end. xFin de la Unidad

```

```

case ch of
  '0'..'9': begin
    gotoxy(columna-1+i, posicion); write(ch);
    temp:=temp*10;
    num:=num+(ord(ch)-48)/temp;
    i:=i+1; dec:=dec+1;
  end;
chr(8) : begin
  i:=i-1; dec:=dec-1;
  gotoxy(columna-1+i, posicion); write(' ', ch);
  if dec<=0 then
    begin
      bandera:=0;
      sig:=1;
    end
  else
    begin
      temp:=temp div 10;
      num:=trunc(num#temp); num:=num/temp;
    end #if#
  end;
end; #End_case#
end; #while#
num:=num#sig;
FunInDec2:=num;
ch:=' ';
end; #function FunInDec2#

```

```

function FunRes(columna, posicion:integer):integer;
begin
  num:=0;
  repeat
    gotoxy(columna, posicion); ch:=readkey;
    case ch of
      '0', '1': begin
        gotoxy(columna, posicion); write(ch);
        num:=ord(ch)-48;
      end;
    end; #Del Case#
  until (ch='0') or (ch='1');
  FunRes:=num;
end; #function FunRes#

```

```

function FunFinDec(columna, posicion:integer):real;
label 20;
var
  num:real;
begin
  num:=0; i:=1; ent:=1; bandera:=0; temp:=1; sig:=1;
  gotoxy(columna, posicion); write(' '); gotoxy(columna, posicion);
  while ch<>chr(13) do
    begin
      ch:=readkey;
      if ((ch=' ') and (i=1)) then
        begin
          num:=maxint;
          goto 20;
        end
      else

```

```

begin
if bandera=0 then
begin
case ch of
'-':begin
if i<=1 then
begin
gotoxy(columna-1+i, posicion); write(ch);
i:=i+1; ent:=ent+1;
sig:=-1;
end; wif;
end;
'0'..'9':begin
gotoxy(columna-1+i, posicion); write(ch);
num:=num*10+ord(ch)-48;
i:=i+1; ent:=ent+1;
end;
chr(8) :begin
i:=i-1; ent:=ent-1;
if i<>0 then
begin
gotoxy(columna-1+i, posicion); write(' ', ch);
end;
if ent>1 then
num:=trunc(num/10)
else
begin
ent:=1; i:=1; sig:=1; num:=0;
sound(100); delay(100); nosound; gotoxy(columna, posicion);
end; wif;
end;
'.' :begin
bandera:=1;
gotoxy(columna-1+i, posicion); write(ch); i:=i+1; dec:=1;
end;
end; wEnd_case;
end wif bandera;
else
begin
case ch of
'0'..'9':begin
gotoxy(columna-1+i, posicion); write(ch);
temp:=temp*10;
num:=num+(ord(ch)-48)/temp;
i:=i+1; dec:=dec+1;
end;
chr(8) :begin
i:=i-1; dec:=dec-1;
gotoxy(columna-1+i, posicion); write(' ', ch);
if dec<=0 then
begin
bandera:=0;
sig:=1;
end
else
begin
temp:=temp div 10;
num:=trunc(num/temp); num:=num/temp;
end wif;
end;
end; wEnd_case;

```

RECOMENDACIONES

Se recomienda a los profesionales y alumnos que empiezan a utilizar SAP80 para el cálculo de estructuras, utilizar el presente sistema, ya que éste va armando el archivo de texto, de acuerdo a los ingresos del usuario; es más conveniente realizar este archivo de texto de esta manera, ya que la forma de editarlo como hasta ahora se lo ha venido haciendo en cualquier editor de archivos de texto, es muy complejo, ya que existen muchas equivocaciones para aquellos que empiezan a trabajar por primera vez con SAP80.