



**PONTIFICIA
UNIVERSIDAD
CATOLICA
DEL ECUADOR**
SEDE AMBATO

UNIDAD DE INGENIERIA DE SISTEMAS

Disertación de grado previa a la obtención del Título de
Ingeniero en Sistemas

“Estudio sobre el Administrador de Base de Datos Sybase SQL aplicado
a la Automatización de Información Financiera del Fideicomiso en la
EEASA. RCN SA.utilizando Power Builder”

Autor:

PATRICIA JANNETH CONDO CARRANZA

Asesor:

Ing. Victor Chuncha

Ambato – Ecuador

Julio 2007



SECRETARIA
ESCUELA DE INGENIERIA
DE SISTEMAS

20-12-07

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
SEDE AMBATO**

HOJA DE APROBACIÓN

Disertación de grado previa a la obtención del Título de
Ingeniero en Sistemas

“Estudio sobre el Administrador de Base de Datos Sybase SQL aplicado
a la Automatización de Información Financiera del Fideicomiso en la
EEASA. RCN SA. utilizando Power Builder”

Autor:

PATRICIA JANNETH CONDO CARRANZA

Victor Chuncha, Ing.
DIRECTOR DE DISERTACIÓN

f.  _____

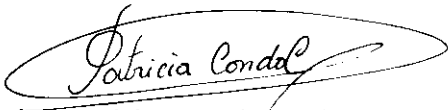
Ambato – Ecuador

Julio 2007

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Patricia Janneth Condo Carranza, portador de la cédula de ciudadanía No. 180246719-9 declaro que los resultados obtenidos en la investigación que presento como informe final, previo la obtención del título de Ingeniero en Sistemas son absolutamente originales, auténticos y personales.

En tal virtud, declaro que el contenido, las conclusiones y los efectos legales y académicos que se desprenden del trabajo propuesto de investigación y luego de la redacción de este documento son y serán de mi sola y exclusiva responsabilidad legal y académica.

A handwritten signature in black ink, reading "Patricia Condo Carranza", is enclosed within a hand-drawn oval. The signature is fluid and cursive.

Patricia Janneth Condo Carranza

CI. 180246719-9

INTRODUCCION

En los últimos años la tecnología informática ha cambiado sustancialmente, a tal punto que ha transformado la vida de los seres humanos, convirtiéndose en una herramienta indispensable en tareas desde las más simples hasta las complejas.

El presente proyecto es un ejemplo claro de cómo un programa computacional puede ayudar en la gestión y toma de decisiones en la Empresa Eléctrica Ambato.

El disponer de la información en forma ágil y clara proporcionada por el Departamento Financiero y almacenada en la base de datos, con el propósito de consulta y actualización constituye el soporte de conocimiento y elaboración de reportes e informes económicos acerca del movimiento mensual del Fideicomiso de la Empresa Eléctrica Ambato S.A.

En el primer capítulo se encuentra una breve reseña histórica de la EEASA RRCN S.A., su evolución y estructura administrativa.

En su contenido se presenta el Estudio sobre el Administrador de la base de Datos SYBASE SQL, con la aplicación realizada para El Control del Fideicomiso de la EEASA RCN S.A. utilizando como Front End Power Builder y con la Herramienta de Diseño Power Designer.

DEDICATORIA

El presente trabajo está dedicado con mucho cariño y respeto a mis padres por todo su sacrificio que han hecho durante toda su vida para darme la educación y formación dentro de un ambiente de amor y respeto, a mi esposo Marco y a mis hijos Erika, Daniela y Christopher quienes fueron mi aliento para continuar con mi carrera, sacrificando el tiempo de estar juntos, pero siempre me esperaban con mucho cariño en el calor de nuestro hogar.

A mis hermanos por ser el complemento de mi hogar, quienes han estado junto a mí en los más duros momentos de mi vida brindándome apoyo incondicional y manteniéndonos siempre juntos,

AGRADECIMIENTO

Doy infinitas gracia a Dios por todas sus bendiciones derramadas en mi vida, porque todos mis logros se los debo a él, gracias a mis padres y mi esposo por haber confiado en mí y me brindaron su apoyo incondicional, al Ing. Victor Chuncha, Ing. Patricio Medina, Ing. Wilgberto Sánchez y al Ing Oscar Armas, quienes me brindaron su respaldo en el desarrollo del presente trabajo.

INDICE

CAPITULO I

GENERALIDADES DE LA EMPRESA ELECTRICA AMBATO RCN S.A

1.1	RESEÑA HISTÓRICA.....	1
1.2	BASE LEGAL.....	3
1.2.1	EVOLUCIÓN LEGAL.....	3
1.2.2	DISPOSICIONES LEGALES.....	4
1.3	OBJETIVO.....	4
1.4	ESTRUCTURA ADMINISTRATIVA Y FINANCIERA.....	5

CAPITULO II

BASE DE DATOS SYBASE SQL

2.1.-	INTRODUCCIÓN.....	9
2.2.-	DEFINICIONES DE SYBASE SQL ANYWHERE.....	10
2.2.1.-	VENTAJAS DE SQL ANYWHERE.....	12
2.2.2.-	TERMINOLOGÍA.....	13
2.2.3.-	CONCEPTOS USADOS EN SYBASE SQL ANYWHERE.....	16
2.2.3.1.-	MOTOR (ENGINE) SYBASE SQL ANYWHERE.....	16
2.2.3.2.-	COMPONENTES DE STANDALONE SQL ANYWHERE.....	18
2.2.3.3.-	EJECUCIÓN DE UNA APLICACIÓN EN SERVIDOR DE SQL ANYWHERE EN RED.....	19
2.2.3.3.1	SYBASE SQL ANYWHERE SERVER.....	21
2.2.4	DIFERENCIAS ENTRE SYBASE SQL ANYWHERE ENGINE Y SYBASE SQL ANYWHERE SERVER.....	24
2.2.5	SYBASE SQL ANYWHERE SOBRE MÚLTIPLES PLATAFORMAS.....	24
2.2.6	INTERFACES DE PROGRAMACIÓN PARA SYBASE SQL ANYWHERE.....	28
2.3.	COMPONENTES DE SYBASE SQL ANYWHERE.....	31
2.3.1.	COMPONENTES PROPORCIONADOS POR SQL ANYWHERE.....	31
2.3.2.	VARIABLES DE ENTORNO (ENVIRONMENT VARIABLES).....	32
2.4.	PROTOCOLOS SOPORTADOS POR SYBASE SQL ANYWHERE.....	35
2.5.	CARACTERÍSTICAS DE SYBASE SQL ANYWHERE.....	36
2.5.1	DISEÑO DE LA BASE DE DATOS.....	36
2.5.2	PLANIFICACIÓN DE LA BASE DE DATOS.....	39
2.5.3	CONEXIÓN CON LA BASE DE DATOS.....	47

CAPITULO III

INTRODUCCIÓN A WINDOWS NT SERVER, POWER DESIGNER Y POWER BUILDER

3.1.	WINDOWS NT SERVER.....	53
3.1.1.	DESCRIPCIÓN GENERAL DE WINDOWS NT SERVER.....	53
3.2.	POWER DESIGNER.....	55
3.2.1.	TÉRMINOS USADOS EN EL DISEÑO DE BASES DE DATOS.....	55
3.2.2.	POWER DESIGNER.....	57
3.2.3.	DEFINIENDO LA IMPLEMENTACIÓN FÍSICA.....	57
3.2.4.	DEFINIENDO UNA REFERENCIA Y UNA INTEGRIDAD REFERENCIAL.....	58
3.2.5.	CREAR VISTAS.....	59
3.2.6.	USAR LOS TRIGGERS Y PROCEDURES.....	59
3.2.7.	GENERAR UN DATABASE SCRIPT.....	60
3.3.	POWER BUILDER.....	60
3.3.1.	CARACTERÍSTICAS DE POWER BUILDER.....	60
3.3.2.	AMBIENTE DE POWERBUILDER.....	62

CAPITULO IV

ANALISIS Y DESARROLLO DEL SISTEMA DE AUTOMATIZACION DEL CONTROL DEL FIDEICOMISO DE LA EMPRESA ELECTRICA AMBATO RCN. S.A.

4.	ANÁLISIS Y DESARROLLO	69
4.1.	REQUERIMIENTO DEL NUEVO SISTEMA	69
4.2.	NECESIDADES Y LIMITACIONES	70
4.3.	DIAGRAMAS.....	71
4.4.	DISEÑO	72
4.5.	DISEÑO DE ENTRADA.....	73
4.6.	BARRA DE HERRAMIENTAS DE PANTALLAS DE ENTRADA	73
4.7.	PANTALLAS DE ENTRADA DE DATOS	74
4.8.	DISEÑO DE SALIDAS	77
4.9.	BARRA DE HERRAMIENTAS DE PANTALLAS DE SALIDA	77
4.10.	PANTALLAS DE SALIDA.....	78
4.11.	DISEÑO DE LA BASE DE DATOS.....	80
4.12.	DIAGRAMA DEL SISTEMA.....	82
4.13.	DIAGRAMA DE ITERACIÓN	83
4.14.	DIAGRAMA DE SECUENCIA	83
4.15.	DICCIONARIO DE DATOS.....	85

CAPITULO V

IMPLEMENTACION

5.1.	ESTÁNDARES DE CODIFICACIÓN	92
5.2.	CONVENCIONES SOBRE LOS NOMBRES DE OBJETOS Y CONTROLES.....	94
5.3.	PRUEBAS	96
5.3.1.	PRUEBAS DE INTEGRIDAD	96
5.3.2.	PRUEBAS DE FUNCIONAMIENTO	96
5.4.	IMPLEMENTACION	97
5.5.	BENEFICIOS QUE SE OBTIENEN DEL SISTEMA DEL FIDEICOMISO.....	97

CONCLUSIONES.....	99
RECOMENDACIONES.....	101
MANUAL DEL USUARIO.....	102
ANEXOS.....	122

CAPITULO I

GENERALIDADES

DE LA EMPRESA ELECTRICA AMBATO RCN S.A

1.1 Reseña histórica

Ya sea en un negocio de compra - venta de mercaderías o en una industria donde se transforman las materias en productos, o en cualquier institución pública, el disponer del servicio de energía eléctrica, constituye una necesidad fundamental para la vida misma de la empresa, y prácticamente todo el desarrollo tecnológico que pueda alcanzar un país se basa en las aplicaciones de la electricidad como fuente primaria.

En nuestra región, el disponer de energía eléctrica ha sido una aspiración permanente desde la instalación en 1915 de una turbina PELTON de 120 kilovatios en la central Miraflores, que aún funciona, habiendo sido parte de la acción normal del Cabildo de nuestra ciudad el preocuparse de que este servicio pudiese abastecer los requerimientos de una ciudad en constante desarrollo.

El crecimiento de la provincia y las complicaciones de orden económico y administrativo que esto implicaba en una institución municipal, llevó a nuestras autoridades a crear un organismo que, constituido como sociedad anónima, pudiese generar y distribuir el servicio eléctrico en una forma eficiente y fue así como en el año 1959 mediante escritura del 29 de abril hecha efectiva el 2 de julio de ese año, inició su operación la Empresa Eléctrica Ambato S.A.

En esos años, la ciudad acusaba un considerable déficit energético, debido a la limitada capacidad de producción de la única fuente de generación existente al momento, la central hidroeléctrica Miraflores, que contaba con apenas 1.450 kilovatios y el mal estado en los sistemas de distribución; en consecuencia, la



primera gran tarea que afrontó la administración de la Empresa, fue la consecución de fondos para la construcción de la Central La Península, obra que había sido iniciada por el Ilustre Municipio de Ambato y fue concluida en su primera etapa en Octubre de 1961, con una potencia instalada de 1.500 kilovatios.

En el año 1962, se amplió la capacidad de la Central La Península en 1.500 kilovatios, instalando un nuevo grupo generador que venía a superar el déficit energético que tenía el sistema.

En el año de 1964, la primera interconexión cantonal a través de la línea Ambato-Pelileo a 13.800 voltios, daba la pauta que la empresa se proyectaba hacia el resto de la provincia.

En el año 1967 inicia su funcionamiento la central Térmica El Batán, la misma que es ampliada en el año 1969, incorporando hasta ese momento a los cantones de Patate y Quero. En el año 1973 se realiza la interconexión con las ciudades de Pillaro y Baños, para el año 1975 se realizó una nueva ampliación de la central El Batán.

La crisis energética que atravesaba esos días el sistema obligó a la construcción y puesta en funcionamiento en el año 1976 de la segunda central Térmica denominada Lligua.

La construcción de la central Agoyán y la distribución de la energía a través del Sistema Interconectado a partir del año 1979, vino a solucionar en gran parte la creciente necesidad de la nuestra provincia. Siendo la distribución de energía por medio de la línea de interconexión Baños-Puyo a 69 y 138 kilovoltios, obra que fue construida en el año de 1983, la que dio lugar a la expansión del servicio para el Oriente Ecuatoriano, administrando el Sistema Eléctrico Pastaza por delegación del INECEL.



Como aspecto relevante en la vida de la Empresa, es la construcción del edificio institucional, el mismo que fué inaugurado en el año 1982. Han transcurrido treinta y nueve años y en este lapso la Empresa ha ido incorporando paulatinamente las diferentes cabeceras cantonales de la provincia de Tungurahua, incentivando el desarrollo de la electrificación rural y finalmente aceptó el reto de la integración eléctrica regional con las provincias de Pastaza y Morona Santiago, todo ello enmarcado dentro de una búsqueda permanente de eficiencia en el servicio a través de mejorar tecnológicamente y económicamente.

1.2 Base legal

1.2.1 Evolución legal

El I. Municipio de Ambato y la H. Junta de Reconstrucción de Tungurahua, luego de innumerables gestiones encaminadas a constituir la Empresa Eléctrica Ambato como Sociedad Anónima el 11 de abril de 1959 se efectuó la primera reunión de la Junta Promotora, la segunda Junta Promotora se reunió el 27 de abril, habiéndose aprobado y determinado todos los aspectos estatutarios y de conformación de la nueva Empresa.

El 29 de abril de 1959, ante el Notario Segundo José Villalba, y los testigos de Ley, comparecieron por el I.Municipio el Dr. Ruperto Camacho, Alcalde Cantonal, y el Dr. Luis Molina Procurador Síndico Municipal y por la H. Junta de Reconstrucción de Tungurahua, El Dr. Germánico Holguín Presidente y el Ing. Carlos Padilla Gerente, presentaron la minuta correspondiente para realizar la Escritura de Constitución de la Empresa Eléctrica Ambato S.A.

El día 2 de Julio de 1959, en acto especial y con la asistencia de los Señores Representantes de los Accionistas Dr. Ruperto Camacho, por el I. Municipio de Ambato; Dr. Germánico Holguín por la Junta de Reconstrucción de Tungurahua, se procedió a la entrega formal de los bienes de la Ex-Empresa Eléctrica Municipal a favor de la nueva Empresa Eléctrica S.A.

1.2.2 Disposiciones legales

El entorno legal en el que desenvuelve la Empresa sus actividades esta basado en las siguientes disposiciones:

- La Ley Básica de Electrificación, esta ley se puso en vigencia el 10 de septiembre de 1973.
- Ley de Compañías, este normativo se expidió como primera codificación de la ley de Compañías mediante el Registro Oficial No. 424 del 19 de Julio de 1968.
- Código de Trabajo.
- Sistema Uniforme de Cuentas para el Sector Eléctrico, en el Art. 31 literal h) de la Ley Básica de Electrificación se determina la obligatoriedad de llevar la contabilidad de acuerdo con éste sistema.
- Estatutos de la Empresa, en la Escritura de Constitución del 29 de Abril de 1959 se aprobó y determinó los aspectos estatutarios y de conformación de la EEASA, la misma que fue inscrita en el Registro de la Propiedad No. 2514 de Septiembre 24 del mismo año.

1.3 Objetivo

La Institución tiene por objetivo el servicio público, desarrollo de la industria y comercio, así como la realización técnica, económica y financiera de la Empresa Eléctrica, para beneficio social y comercial de la ciudad de Ambato y del Cantón, debido al desarrollo que ha tenido la Empresa desde el año de su fundación en 1959, este objetivo se ha ido modificando en relación con la incorporación de nuevos centros poblados aumentando su ámbito de acción y procurando estar a la par con el avance de la tecnología, manteniendo su misión prioritaria que es el servicio de electricidad en las mejores condiciones de eficiencia y calidad.

1.4 Estructura Administrativa y Financiera

1.4.1 Nivel Ejecutivo

Presidencia Ejecutiva.- El Presidente Ejecutivo es el representante legal de la Empresa y actúa como secretario de la Junta de Accionistas y Directorio.

1.4.2 Nivel De Control

Comisarios.- Son los encargados de efectuar exámenes independientes sobre las operaciones administrativas, financieras y técnicas de la Empresa.

Auditoría Interna.- Es la unidad que se encarga de controlar internamente, las actividades administrativas, financieras y técnicas de la Empresa. Sus funciones y atribuciones se determinan en el Estatuto.

1.4.3 Nivel De Apoyo

Planificación.- La división de Planificación es el organismo encargado de realizar anualmente los planes de expansión globales de la Empresa, la misma que consta de dos secciones:

Asesoría Jurídica.- Es el organismo asesor encargado de receptor y resolver todas las consultar de tipo legal. Sus funciones están determinadas en las Leyes pertinentes de la materia, como son la Ley de Compañías, Ley Básica de Electrificación y demás Leyes a la que está sujeta la Empresa.

Relaciones Industriales.- Es el organismo encargado de la administración de los asuntos con el personal, procurando las mejores condiciones de trabajo, precautelando sus salud y creando un ambiente propicio para su bienestar y desarrollo, contando para ello con dos secciones:

- La sección de Servicios con el Personal, es la encargada de la captación, desarrollo y control de los recursos humanos necesarios para el cumplimiento de los objetivos de la Empresa.
- La Sección de Servicios Generales, es la encargada del mantenimiento de los edificios, equipos de oficina, reproducción de documentos, y vigilancia. El servicio de transporte merece especial atención, para lo cual se encarga de suministrar los combustibles, lubricantes, reparaciones y llevar estadísticas relacionadas con el parque automotor de la Empresa.

1.4.4 Nivel Auxiliar

Secretaria.- Es la unidad encargada de velar por el control y trámite oportuno de toda la correspondencia que se produce en la Empresa, como función principal tiene la de redactar y suscribir las Actas de las Sesiones de Juntas de Accionistas y de Directorios. Atendiendo el diariamente el despacho de los asuntos resueltos por la Presidencia Ejecutiva.

Archivo.- Es la unidad encargada de la custodia de todos los documentos de la Empresa, manteniéndolos debidamente ordenados y clasificados, de manera que permita su ubicación inmediata.

1.4.5 Nivel Operacional

Departamento de Diseño y Construcción.- El Departamento de Diseño y Construcción, es el organismo operacional encargado de ejecutar las obras contempladas dentro de las inversiones que debe efectuar la Empresa para cubrir con la demanda de potencia del Sistema Eléctrico atendido por la Empresa, buscando la mayor eficiencia técnico-económica.

Departamento de Operación y Mantenimiento.- Agrupa a todas las actividades inherentes con la Generación, transporte y distribución de energía eléctrica, en las mejores condiciones posibles de calidad y continuidad del servicio para los usuarios, y mayores rendimientos técnicos económicos para la Empresa, en su área de concesión.

Departamento Comercial.- El Departamento Comercial es el organismo operacional encargado de relacionar a la Empresa con los abonados, promoviendo la concesión de los servicios prestados y asegurando la recaudación oportuna de los valores correspondientes.

Departamento Financiero

El Departamento Financiero, es el organismo operacional encargado de planificar, dirigir, coordinar y controlar las actividades de la Empresa.

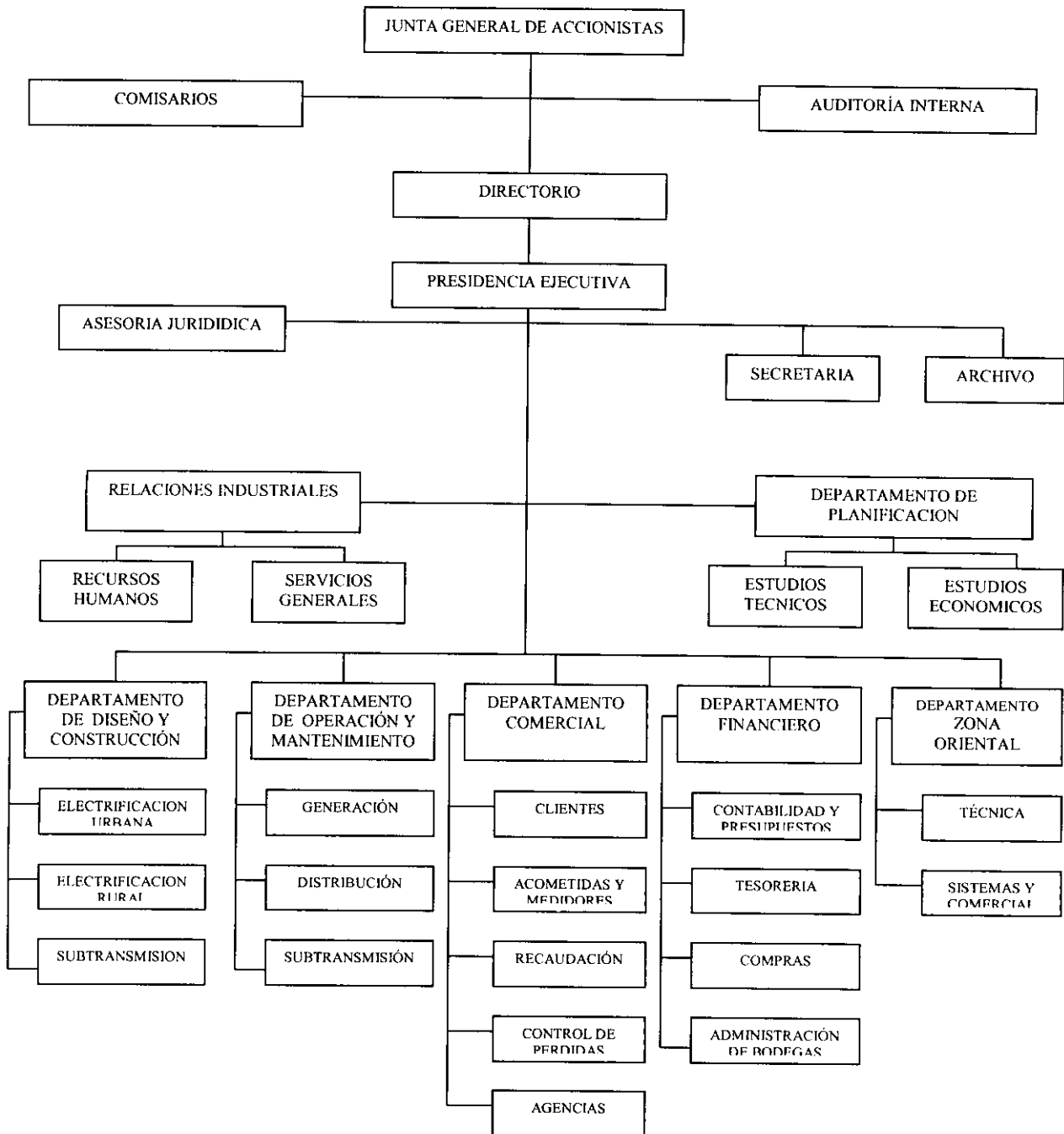
El Departamento Financiero esta formado por cuatro secciones: Contabilidad y Presupuesto, Tesorería, Compras e Importaciones y Bodegas.

Departamento De La Zona Oriental

Es el organismo descentralizado encargado de la construcción de las obras, contempladas dentro del plan anual de actividades que realice la Empresa, así como también la distribución y comercialización de la energía en las mejores condiciones de calidad y continuidad del servicio para los usuarios.

Orgánicamente este departamento esta conformado por tres sistemas físicamente separados, lo que ha obligado a otorgarles independencia administrativa, Sistema Eléctrico Pastaza, Sistema Eléctrico Palora, Sistema Eléctrico Tena.

ORGANIGRAMA DE LA EMPRESA ELECTRICA AMBATO RCN S.A.



CAPITULO II

BASE DE DATOS SYBASE SQL

2.1.- Introducción.

En la actualidad grandes y pequeñas empresas generan una gran cantidad de datos, siendo estos necesarios para su funcionamiento; desde una lista de Proveedores hasta un reporte de los bienes o servicios que han sido utilizados durante cierto período. Naciendo así la necesidad de almacenarlos y automatizar el proceso del manejo de esta información.

En nuestro medio se puede encontrar una gran variedad de Software como son las Bases de Datos, herramientas de Diseño de Bases de Datos, lenguajes de programación, Sistemas Operativos, Protocolos de Comunicación y otros; gracias a la integración de cada uno de ellos, se puede desarrollar un Software (Aplicaciones) que satisfaga los requerimientos más exigentes en cualquier empresa.

Los Sistemas Informáticos personalizados a una empresa les resultan complejos de implantar debido a la infraestructura y costos requeridos, por lo que hay que buscar una tecnología de punta, que sin ser extremadamente costosa, brinde las seguridades y aproveche de mejor manera los recursos ya existentes en las organizaciones.

Una de estas herramientas constituyen las Bases de Datos Relacionales, que permiten un manejo y administración confiable de información; la más accesible es Sybase SQL Anywhere, este administrador de base de datos va a satisfacer los requerimientos de pequeñas, medianas y grandes empresas en el manejo de banco de datos, según sus necesidades. Esta Base de Datos está disponible para los distintos Sistemas Operativos como Windows NT, Windows 95, Windows 3.x, OS/S, Netware, Solaris/Sparc, HP-UX, AIX, DOS y QNX.

Sybase SQL Anywhere utiliza otras herramientas de trabajo como los Diseñadores de Bases de Datos, para realizar un diseño confiable y sobre todo amigable para el usuario. Esta es la parte esencial, de la cual dependerá el éxito de un Proyecto de Software. Dentro de estas herramientas se puede citar a Power Designer, Erwin y otros.

Power Builder es una herramienta para el desarrollo del Software de Aplicación con múltiples ventajas tales como :

- ❑ Programación Orientada a Objetos
- ❑ Presentación en modo gráfico
- ❑ Interfaz amigable para el usuario
- ❑ Manejo de archivos con diferentes formatos
- ❑ Facilidad de migración de datos
- ❑ Dispone de un ambiente Cliente/Servidor
- ❑ Poder construir aplicaciones que puedan ser ejecutadas bajo distintas plataformas de Sistemas Operativos
- ❑ Interoperatividad con las principales Bases de Datos y otras aplicaciones existentes en el mercado, etc.

2.2.- Definiciones de Sybase Sql Anywhere

Sybase SQL Anywhere es un completo sistema manejador de Bases de Datos SQL.

Incluye dos versiones ejecutables para el manejo de bases de datos:

- ❑ Multiusuario, para el manejo de bases de datos en un ambiente cliente servidor en red.
- ❑ Monousuario, permite el manejo del motor de Base de Datos en un computador en modo standard.

Sybase SQL Anywhere fue diseñado para su manejo en computadores personales, siendo un Sistema manejador de Bases de Datos con completas características para transacciones y procesos, el cual posee un alto rendimiento; mientras que el requerimiento de recursos es menor (velocidad de CPU, espacio de memoria y disco) que en otros manejadores de Bases de Datos.

SQL Anywhere es un paquete completo que proporciona tecnologías de administración e intercambio de datos, las cuales facilitan un rápido desarrollo e implantación de aplicaciones que trabajan con bases de datos. SQL Anywhere proporciona bases de datos de tipo empresarial que pueden usarse en servidores de 64 bits con miles de usuarios y acceder desde incluso dispositivos de bolsillo. Sus tecnologías de intercambio de datos amplían la información contenida en las aplicaciones y sistemas corporativos con información contenida en bases de datos que se ejecutan en entornos de misión crítica en la primera línea. Las herramientas de diseño y administración de SQL Anywhere permiten a los desarrolladores implementar e instalar aplicaciones en la primera línea, y a los administradores de equipos, administrarlos y ofrecer asistencia de forma sencilla.

Sybase SQL Anywhere está disponible para sistemas operativos como Windows NT, Windows 95, Windows 3.x, OS/2, NetWare, Solaris/Sparc, HP-UX, AIX, DOS, y QNX.

Sybase SQL Anywhere puede ser usado como un sistema manejador de Bases de Datos standard monousuario (standalone) o en un ambiente cliente/servidor.

Los archivos de Bases de Datos son compatibles entre versiones y en todos los sistemas operativos. El sistema operativo en el cual ejecute SQL Anywhere depende de sus licencias que posee.

SQL Anywhere es una rápida y eficiente Base de Datos para varios ambientes, desde los computadores portátiles hasta los grandes servidores con un gran número de concurrencias de usuarios. Esta es una gran solución para la diversidad de necesidades de hoy en día permitiendo la escalabilidad.

Con un eficaz ODBC driver (y ODBC lleno 2.1 nivelado 2 apoyo), tan satisfactorio como otras interfaces, SQL Anywhere es también una Base de Datos para diseñadores. Si usa lenguaje C, Power Builder, u otras herramientas para diseñar aplicaciones, y si diseña aplicaciones monousuario o cliente/servidor, SQL Anywhere provee de todas las capacidades que espera de un manejador de Bases de Datos completo.

2.2.1.- Ventajas de SQL Anywhere.

Fácil de utilizar y de administrar: las funciones de administración y personalización automáticas, así como la compatibilidad remota lo convierten en una elección ideal para entornos que no requieran administración y que estén implantados de forma masiva.

Rendimiento y escalabilidad de tipo empresarial: es posible escalarlo para que lo utilicen miles de usuarios y contenga cientos de gigabytes de información, incorporando funciones tales como la compatibilidad con multiprocesadores simétricos (SMP), paralelismo intra-consulta e inter-consula, vistas materializadas y aislamiento de instantáneas.

Seguridad de extremo a extremo: seguridad validada por la Federal Information Processing Standards (FIPS) 140-2, certificación de Common Criteria, así como algoritmos y protocolos de encriptación estándares del mercado, incluyendo AES, ECC, RSA y SSL, lo que garantiza la seguridad de extremo a extremo para todo tipo de sectores.

Compatibilidad multiplataforma: admite una amplia variedad de sistemas operativos, desde los servidores UNIX de 64 bits, los Embedded XP o las plataformas móviles más populares, con el fin de proporcionar la máxima flexibilidad en la implantación de la base de datos SQL.

API abierta y soporte para herramientas: admite numerosos estándares para un acceso sencillo y de alto rendimiento a los datos. SQL Anywhere permite aprovechar las inversiones existentes en conocimientos y herramientas de desarrollo.

Completo soporte para los servicios Web: simplifica el desarrollo de aplicaciones de servicios Web con bases de datos, la integración con sistemas existentes y la implementación de aplicaciones en entornos de TI diversos.

Admite dispositivos de bolsillo y dispositivos inteligentes: las base de datos UltraLite y el mecanismo de sincronización proporcionan una solución ideal para aquellos entornos con limitaciones en cuanto a memoria, como pueden ser los entornos con dispositivos de bolsillo.

2.2.2.- Terminología

Con una Base de Datos monousuario ejecutándose en un motor de Base de Datos monousuario, la configuración por defecto crea el nombre del motor, el nombre de la Base de Datos, y el archivo de Base de Datos, por sí solo, separadamente de la ruta y extensión asociada con el archivo. En esta situación hay una pequeña ambigüedad cuando se habla acerca de "La Base de Datos".

En entornos con múltiples Bases de Datos, múltiples archivos, y varios motores de Bases de Datos operando simultáneamente, es importante distinguir entre los diferentes componentes que se crean al ejecutar el sistema de manejo de Base de Datos SQL Anywhere.

A continuación se presenta una lista de términos usados comúnmente:

- ❑ **Base de Datos.-** Una Base de Datos es una recopilación de información que está relacionada con un tema o con un propósito en particular. La clave para una recuperación y almacenamiento eficiente está en el proceso de planificación. En primer lugar se debe identificar la tarea que se desea que realice la Base de Datos por el usuario, siendo capaces de crear un diseño práctico que ofrezca como resultado una herramienta de gestión de datos más precisa y más rápida.
- ❑ Cuando se diseña nuestra propia Base de Datos, el primer paso en el proceso de planificación es identificar la tarea que la Base de Datos debe realizar y la información que se necesita gestionar. Antes de diseñar nuestra propia Base de Datos se debe considerar las siguientes cuestiones:

Qué información se desea obtener de nuestra Base de Datos?

Sobre qué áreas de temas independientes se necesita almacenar los hechos?

Cómo están relacionados estos temas entre sí?

Qué hechos se necesita almacenar sobre cada tema?

SQL Anywhere nos ayuda a gestionar nuestra Base de Datos proporcionando una estructura eficiente para almacenar y recuperar la información. El lugar en el que se fusiona la información sobre cada tema que se necesita se denomina **tabla**, y cada tipo de hechos recogidos en nuestras tablas se denomina **campo**.

- ❑ **Primary key (Clave primaria).-** Una clave primaria es un grupo de columnas que identifican como única a cada fila en una tabla.
- ❑ Cada tabla en una Base de Datos relacional tiene una **clave primaria**. La clave primaria es una columna, o grupo de columnas, que permiten que cada fila en la tabla sea identificada como única. Dos filas no pueden tener el mismo valor en la clave primaria.

- ❑ **Foreign Key (clave foránea).**- Una **clave foránea (foreign key)** es una columna o grupo de columnas que contienen valores de clave primaria de otras tablas. La clave foránea permite tener acceso a datos de más de una tabla al mismo tiempo.
- ❑ **Referential Integrity (integridad referencial).**- Una **referencia** simboliza el enlace entre dos tablas.
- ❑ Se puede crear una referencia entre dos tablas. Hace esto cuando se quiere una columna en referencia a otra. Cuando se crea una referencia, la clave primaria en la tabla padre migra como una clave foránea en la tabla hija.
- ❑ La **integridad referencial** determina qué sucede en una columna de clave foránea en una tabla hija cuando se actualiza o elimina el valor correspondiente en la columna de clave primaria de la tabla padre.
- ❑ **Archivo de Base de Datos (Database File).**- Aunque las tablas de una Base de Datos pueden estar en varios archivos en diferentes unidades de disco (drives), cada Base de Datos es identificada por un único archivo raíz (root file). En adelante, cuando se hable del archivo de Base de Datos (database file), se refiere al archivo raíz o root file.
- ❑ **Nombre o alias de la Base de Datos (Database name or Alias).**- Un motor o servidor de Base de Datos SQL Anywhere puede ejecutar varias Bases de Datos simultáneamente, manejando accesos a cada una de ellas. Cuando una Base de Datos es inicializada en un motor de Base de Datos, a esta se le asigna un nombre de Base de Datos, también llamado alias de la Base de Datos. Si el nombre de la Base de Datos no es explícitamente asignado, la Base de Datos recibe el nombre del archivo raíz (root file) omitiendo el path y extensión.
- ❑ **Nombre del servidor o motor.**- Cuando un motor o servidor de Base de Datos es inicializado, se le asigna un nombre de motor o servidor (engine name, server

name). El nombre del servidor o motor es completamente distinto del nombre de los programas del motor de Base de Datos. Por default, el server name es el primer nombre de la Base de Datos, por ejemplo: si un motor de Base de Datos es inicializado con una Base de Datos C:\PB\EX\PSDEMODB.DB y no se especifica el nombre, entonces el nombre del motor es PSDEMODB.

- **Clientes y servidores.**- Los dos, cliente y servidor (side) de una configuración SQL Anywhere client/server constan de varios componentes. Los términos cliente (client) y servidor (server) son comúnmente usados para describir no solamente el computador en el cual cada lado (side) de la configuración se sitúa, sino también los programas que están comunicados, y también la colección de componentes de software en cada computador.

2.2.3.- Conceptos usados en Sybase SQL Anywhere

2.2.3.1.- Motor (engine) Sybase SQL Anywhere.

Está destinado para el manejo de bases de datos en computadores personales en modo standard (aplicación cliente).

SQL Anywhere Server y SQL Anywhere Engine manejan bases de datos exactamente de la misma manera y son totalmente compatibles. Pero, SQL Anywhere Engine no permite la comunicación y funcionamiento en red.

La figura muestra la arquitectura de una instalación standard de SQL Anywhere, funcionando el motor de base de datos singleo y trabajando con una Base de Datos singlea. Toda instalación complicada funciona en base a este esquema básico, así, se debe entender cómo funciona esta estructura básica, aun cuando se realice una instalación para un entorno multiusuario en aplicaciones cliente/servidor.



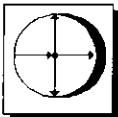



SIMBOLOGIA PARA LAS FIGURAS	
Símbolo	Significado
	Un servidor de base de datos, tal como Sybase SQL de Server o Sybase SQL Anywhere.
	Un motor de base de datos SQL Anywhere, o un Cliente SQL Anywhere y servidor de red. Todas las aplicaciones que trabajan con un motor SQL Anywhere standalone trabajan idénticamente con un servidor de red SQL Anywhere, por medio de un Cliente SQL Anywhere.
	Un Servidor de Repetición Sybase.
	Un Agente de Repetición. Un agente de repetición requiere que una base de datos actúe como un sitio primario de datos en la instalación del Servidor de Repetición Sybase.
	Un Registro Manejador de Transferencia SQL Anywhere.
	Una aplicación cliente.

FIGURA No. 2

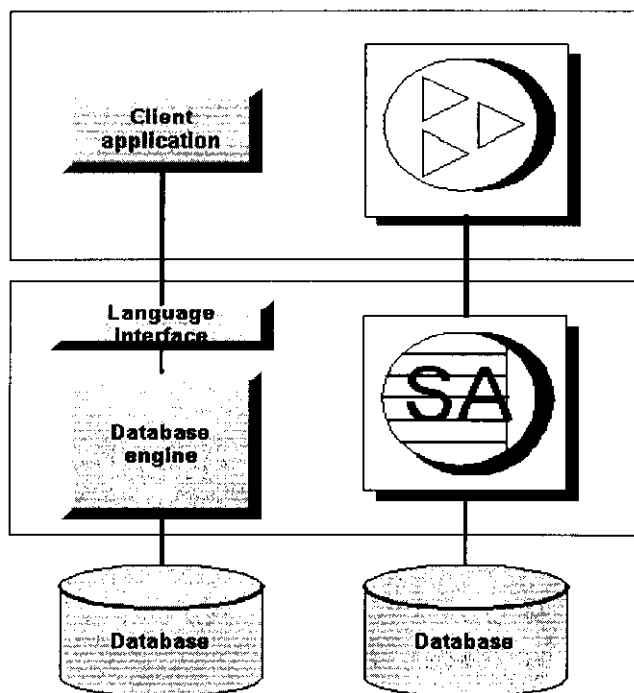


FIGURA No. 3

2.2.3.2.- Componentes de Standalone SQL Anywhere.

Los componentes para la configuración básica de Standalone SQL Anywhere son:

- ❑ La aplicación cliente
- ❑ La capa de interface de SQL Anywhere
- ❑ El motor de la base de datos de SQL Anywhere
- ❑ La base de datos

Los usuarios de Bases de Datos no manipulan directamente los archivos de Bases de Datos. En cambio, su aplicación cliente se comunica con el motor de Base de Datos, usando una interface de programación que soporte SQL Anywhere, y el motor de Base de Datos se encarga de toda manipulación de la Base de Datos actual.

La aplicación Cliente.

Los usuarios de Base de Datos trabajan con una base de datos de aplicación, o una aplicación cliente.

Una aplicación cliente comunicada con el motor de base de Datos SQL Anywhere debe usar una **interface de programación** que soportada por SQL Anywhere. La aplicación cliente llama funciones desde una de las interfaces de programación de SQL Anywhere. La aplicación cliente junto con la capa de interface forma el **client side** de la instalación.

El motor de Base de Datos de SQL Anywhere.

El motor de Base de Datos junto con la base de datos forman el **server side** de la instalación. Una aplicación cliente manipula una Base de Datos una aplicación cliente manipula una Base de Datos para enviarle sus requerimientos al motor de Base de Datos.

Declaraciones SQL.

La comunicación entre una aplicación cliente y un motor de Base de Datos se realiza a través de declaraciones (sentencias) SQL (Structured Query Language). Por ejemplo una sentencia SELECT, o query es usada para obtener información de la Base de Datos. Una declaración UPDATE se usa para modificar el contenido de una de las tablas de la Base de Datos.

La aplicación cliente envía las declaraciones SQL y el motor de Base de Datos las procesa y devuelve los resultados a la aplicación.

2.2.3.3.- Ejecución de una aplicación en servidor de SQL Anywhere en red.

Para esto generalmente se requiere un componente extra para manejar las comunicaciones en red desde el computador cliente. De cualquier modo, para aplicaciones en el mismo computador el servidor de la base de datos puede ser ejecutado exactamente de la misma manera como un motor de Base de Datos Standalone, que ya se ha descrito aquí usando una conexión directa al servidor de la red.

Si un CLIENTE de SQL Anywhere está ejecutando y el servidor de la red está ejecutándose en la misma máquina, el CLIENTE no será usado por aplicaciones cliente que estén en la misma máquina, la conexión será directa.

SQL Anywhere es un sistema de Base de Datos relacional. La Base de Datos se asigna ella misma una o más unidades de discos, y consta de los siguientes objetos:

- **Tablas.**- Mantienen la información en la Base de Datos.
- **Claves (Keys).**- Relaciona la información de una tabla con otra.
- **Indices.**- Permiten el acceso rápido a la información de la Base de Datos.
- **Vistas (Views).**- Son tablas calculadas (computadas).
- **Stored Procedures.**- Contiene comandos que pueden ser ejecutados por cualquier aplicación cliente (los Stored Procedures no están disponibles en el sistema SQL Anywhere Desktop Runtime).
- **Triggers.**- Ayudan en el mantenimiento de la integridad de la información en la Base de Datos (los Triggers no están disponibles en el sistema SQL Anywhere Desktop Runtime). Un trigger es un procedimiento asignado en la Base de Datos que se ejecuta automáticamente cada vez que la información en una tabla especificada es alterada
- **Tablas del sistema.**- Contienen información acerca de la estructura de la Base de Datos.

Múltiples Bases de Datos en un motor de Base de Datos single

Un motor de Base de Datos simple SQL Anywhere puede manipular el acceso a varias Bases de Datos simultáneamente. Se puede inicializar y detener Bases de Datos desde las herramientas para administración de Bases de Datos o desde aplicaciones cliente, y puede conectarse a cualquiera de las Bases de Datos que se estén usando en un motor de Base de Datos.

Tan lejano como el usuario de la Base de Datos se encuentre, la interacción con el motor de Base de Datos siempre es por una **conexión**. Cada vez que los usuarios se conectan a la Base de Datos, ingresando su identificación de usuario y su Password, se conectan a una base de datos específica en un motor de Base de Datos específico. Una vez que la conexión es establecida, ella provee de un canal por el cual se efectúan todas las comunicaciones. La conexión aísla al usuario de los otros componentes de un sistema de Base de Datos en ejecución tal como una sesión en red e interprocesa los mecanismos de comunicación.

Base de Datos multi archivos.

SQL Anywhere soporta múltiples Bases de Datos. Cuando una Base de Datos SQL Anywhere está inicializada, este está compuesto de un archivo (file), el **root file** (archivo de la raíz). Como tablas y otros objetos de la Base de Datos son añadidos a la Base de Datos, pero de cualquier modo ellos necesitan ser asignados a diferentes archivos, los cuales estarían en unidades de disco diferentes al root file.

Los usuarios de la Base de Datos (que no sea el administrador) no necesitan estar informados de la localización física de los archivos de la Base de Datos. El motor de Base de Datos controla todo acceso a los archivos y maneja esta complejidad desde usuario.

2.2.3.3.1. Sybase Sql Anywhere Server.

Sybase SQL Anywhere Server permite conexiones de muchos usuarios al mismo tiempo, en una red. En este caso, el motor de la Base de Datos se ejecuta en un computador (el servidor de la Base de Datos), mientras que las aplicaciones cliente se ejecutan en otros computadores (computadores cliente).

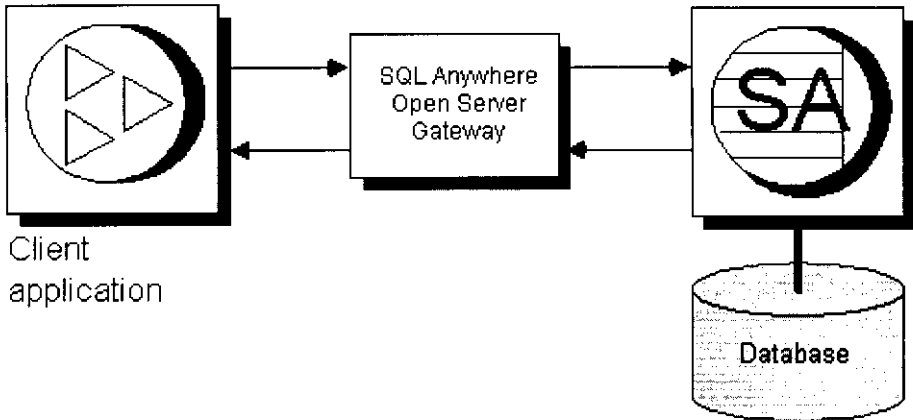


FIGURA No. 4

Sybase SQL Anywhere Server permite a múltiples usuarios en red el acceso a SQL Anywhere. El motor de SQL Anywhere Standalone no permite al acceso múltiple ni la comunicación en red.

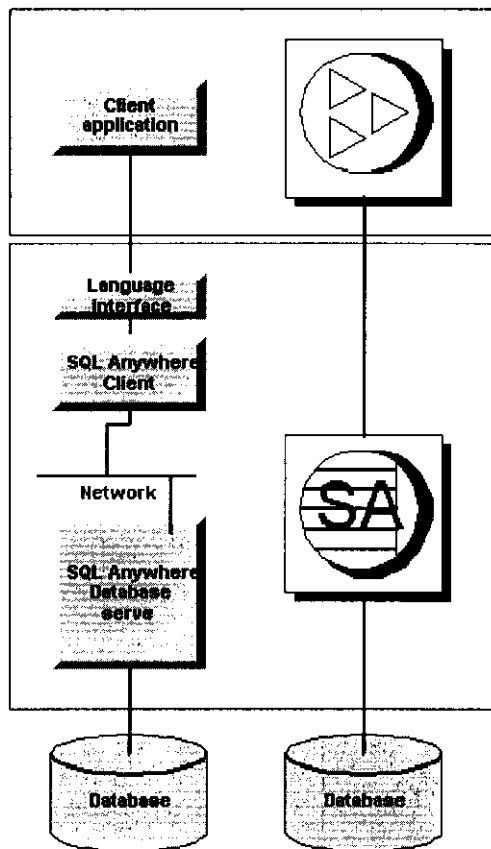


FIGURA No. 5

Servidor de datos SQL Anywhere.

La parte de cliente (client side) de SQL Anywhere configura las consultas SQL (queries) y los comandos de red se encuentran en la parte de servidor (server side), que lleva las ordenes y devuelve los resultados de regreso al cliente.

SQL Anywhere Client.

El SQL Anywhere Client es un programa que maneja las comunicaciones en red con el servidor de la Base de Datos de SQL Anywhere. El SQL Anywhere Client es un programa llamado DBCLIENT (en la versión para Windows 3.x se llama DBCLIENW). SQL Anywhere Client se comunica con el servidor de red de SQL Anywhere. El motor standard (standalone) no maneja comunicaciones desde SQL Anywhere Client.

Para QNX, la Base de Datos cliente es diferente de otros sistemas operativos. En lugar de ser un ejecutable aparte, SQL Anywhere Client es una librería llamada dbclient, que se carga dinámicamente con aplicaciones cliente.

Configuración standalone versus configuración de red.

Desde el punto de vista de las aplicaciones cliente, no hay diferencia entre la configuración standalone y la configuración de red. En la configuración de usuario single (single user), una aplicación cliente envía los requerimientos y comandos al motor de Base de Datos. En cambio, en la configuración multiusuario, esos requerimientos son enviados al Cliente SQL Anywhere. En cada caso, la aplicación cliente tiene solo un punto de contacto con el DBMS. La complejidad adicional del manejo de requerimientos en un entorno multiusuario de red, está oculta desde la aplicación cliente.

Aplicaciones standalone trabajan con el servidor de red SQL Anywhere.

Una vez que una aplicación cliente es desarrollada y trabaja en una configuración standalone SQL Anywhere en un computador single, no es necesario realizar cambios en la aplicación para que trabaje como una aplicación cliente en un entorno de red en contraste con un servidor de SQL Anywhere.

Opciones para el funcionamiento del servidor.

Todo lo tratado en la guía de usuario SQL Anywhere, se aplica tanto al motor SQL Anywhere standalone como al servidor de red SQL Anywhere. Una aplicación de Base de Datos no puede determinar las diferencias entre el motor standalone ejecutándose en la misma máquina, y el servidor de red ejecutándose en un nodo diferente en su red.

Como se ejecute el servidor de Base de Datos depende del sistema operativo que esté usando.

2.2.4 Diferencias entre Sybase SQL Anywhere Engine y Sybase SQL Anywhere Server.

Desde el punto de vista de la aplicación del cliente, no hay ninguna diferencia entre la configuración (setup) standalone y la configuración en red. En la configuración para monousuario, una aplicación cliente envía sus requerimientos y comandos al motor de la Base de Datos. En cambio, en la configuración multiusuario estos requerimientos se envían a SQL Anywhere Client. En cada caso, la aplicación cliente tiene un punto de contacto con el DBMS. La complejidad adicional de requerimientos de manipulación en multiusuario y el entorno de red son ocultados por la aplicación cliente.

Aplicaciones Standalone trabajando con el servidor de SQL Anywhere.

Una vez que se desarrolla una aplicación del cliente y funciona en la configuración de SQL Standalone en monousuario, no se requiere ningún cambio en la aplicación para que trabaje como una aplicación del cliente en un ambiente de la red con un servidor SQL en cualquier parte.

2.2.5 Sybase SQL Anywhere sobre múltiples plataformas.

El motor de Base de Datos de SQL Anywhere Standalone (standard) está disponible para sistemas operativos Windows 3.x, Windows 95 y Windows NT, OS/2, y DOS. La Base de Datos SQL Anywhere Server está disponible para sistemas operativos Novell NetWare, Windows 95 y Windows NT Server, OS/2, Windows 3.x, DOS, y QNX.

SQL Anywhere está disponible para los siguientes sistemas operativos:

□ **Windows y Windows NT.**

El motor de Base de Datos standalone (standard) y servidor multiusuario están disponibles para Windows 3.x, Windows 95 y Windows NT. La utilidad de administración de Sybase Central está disponible para Windows 95 y Windows NT solamente.

□ **OS/2.**

El motor de Base de Datos Standalone y servidor de red multiusuario están disponibles para OS/2. Esto incluye un Administrador de Presentación versión de ISQL, y utilidades de administración de comandos en línea. Sybase Central no está disponible para OS/2.

□ **Netware.**

El servidor de red está disponible como un Netware Loadable Module (NLM). La utilidad ISQL también está disponible como un NLM.

□ **UNIX.**

□ El motor standalone está disponible para Sun Microsystems Solaris/Sparc, Hewlett Packard HPUNIX, e IBM AIX. Esto incluye las utilidades character-mode ISQL y administrador de comandos en línea.

□ **QNX.**

Solamente el servidor de red está disponible para QNX. Esto incluye las utilidades character-mode ISQL y administrador de comandos en línea.

□ **DOS.**

El motor de Base de Datos standalone y servidor de red multiusuario están disponibles para DOS. Esto incluye las utilidades character-mode ISQL y administrador de comandos en línea.

Una Base de Datos SQL Anywhere Server puede soportar múltiples clientes operando en diferentes sistemas operativos, comunicándose mediante diferentes protocolos de comunicación de red.

Clientes y Servidores.

El lado del cliente y el lado del servidor de una configuración SQL Anywhere cliente/servidor constan de varios componentes. Los términos CLIENTE y SERVIDOR son comúnmente usados para describir no solamente los computadores en los cuales cada lado (bando) de la configuración (setup) se sitúa, sino también a los programas que están comunicándose, y también a la colección de componentes de software en cada uno de los computadores. Por todas partes, los términos cliente y servidor se utilizan siempre que sea posible especificar cuál de estos significados se invoca.

Funcionamiento sistemas operativos mixtos en una computadora single.

SQL Anywhere soporta situaciones en las cuales la aplicación cliente y el motor de Base de Datos se ejecutan en diferentes sistemas operativos en el mismo computador. Este soporte requiere un Cliente SQL Anywhere (DBCLIENT) para el sistema operativo de la aplicación cliente. Puede pensar en la configuración mixta del sistema operativo como un arreglo cliente/servidor en red, con ambas partes (sides) cliente y servidor residiendo en la misma computadora.

Las situaciones donde puede ocurrir esto son:

- Aplicaciones cliente en DOS o Windows con un motor OS/2.
- Aplicaciones cliente con DOS o Windows 3.x con un motor Windows 95 o Windows NT.

Los sistemas operativos mixtos pueden ocurrir cuando ejecuta aplicaciones DOS o Windows 3.x con un motor Windows 95 o Windows NT.

Aplicaciones Cliente DOS o Windows en OS/2.

El motor OS/2 Standalone o servidor de red puede ser accesado desde aplicaciones cliente DOS o WIN-OS/2 en la misma máquina.

La aplicación cliente se comunica con el SQL Anywhere Client por el sistema operativo de la aplicación cliente. (SQL Anywhere Client DOS y Windows 3.x son instalados con SQL Anywhere para OS-2 o como una instalación separada de DESKTOP RUNTIME SYSTEM FOR OS/2). El SQL Anywhere Client maneja la comunicación con el motor de Base de Datos OS/2 o con el servidor usando su nombre (pipes) o DDE.

Líneas de comando para clientes DOS o Windows 3.x

La línea de comando para ejecutar el cliente DOS es:

dbclient engine-name

La línea de comando para ejecutar el cliente Windows 3.x es:

dbclienw engine-name

Otros servidores.

Además SQL Anywhere mantiene acceso a servidores de red en otra parte en la vía de red simultáneamente una transmisión distinta enlazada con el SQL Anywhere Client.

Clientes DOS o Windows 3.x en Windows 95 o NT.

El motor o servidor de Base de Datos de Windows 95 o Windows NT puede ser accesado desde una aplicación cliente DOS o Windows 3.x en la misma máquina.

La aplicación cliente se comunica con el SQL Anywhere Client por el sistema operativo de la aplicación cliente (DOS y Windows SQL Anywhere Clients son instalados con SQL Anywhere). El SQL Anywhere Client manipula la comunicación con el motor o servidor de Base de Datos de Windows 95 o Windows NT Server, usando su nombre (pipes) (Windows NT) o DDE (Windows 95).

Líneas de comando para clientes DOS o Windows 3.x en SQL Anywhere.

La línea de comando para ejecutar SQL Anywhere Client en DOS es:

dbclient engine-name

La línea de comando para ejecutar SQL Anywhere Client en Windows 3.x es:

dbclientw engine-name

Otros servidores.

Además SQL Anywhere mantiene acceso a servidores de red en otra parte en la vía de red simultáneamente una transmisión distinta enlazada con el SQL Anywhere Client.

2.2.6 Interfaces de programación para Sybase SQL Anywhere

La conversación entre una aplicación cliente y un motor o servidor de Base de Datos ocurre gracias a la interface de programación de SQL Anywhere. Las interfaces de programación disponibles son:

- Interfaces de alto nivel en sistemas de desarrollo de aplicaciones ODBC-enabled tales como Powerbuilder de Powersoft y en aplicaciones con lenguajes de macroinstrucción ODBC-enabled.
- Interfaces de bajo nivel full-functions: ODBC y SQL incluido.
- Interfaces de alto nivel: WSQL DDE y WSQL HLI.

A continuación se describe las principales características de las interfaces de alto y bajo nivel:

Interfaces de programación de bajo nivel.

El ODBC y SQL incluido proveen las interfaces de bajo nivel a SQL Anywhere.

El ODBC es mantenido (soportado) por una amplia gama de DBMS y es implementado por Windows, Windows NT, y OS/2 como un controlador de Base de Datos: una librería de vínculo dinámico (DLL) que una aplicación invoca para ganar acceso a las Bases de Datos SQL Anywhere. En otras plataformas, la interface ODBC es implementada como una librería estática.

SQL incluido es un poco más rápido que el ODBC y tiene un grupo más amplio de funciones. SQL incluido se implementa tal como un lenguaje preprocesador C/C++, el cual traduce las declaraciones SQL incluidas en su codificación en llamados a la interface. Se ejecuta la interface tal como un DLL para Windows, Windows NT, y OS/2, y como una librería en las otras plataformas.

La interface ODBC.

La interface ODBC es la más usada comúnmente para SQL Anywhere.

SQL Anywhere soporta la interface ODBC (Open Database Connectivity) de Microsoft no solamente en ambientes Windows y Windows NT Server, pero también en DOS, OS/2, Macintosh, y QNX. Esta interface de bajo nivel provee de casi todas las funcionalidades incluidas con SQL, (en la implementación SQL Anywhere), solamente una pequeña diferencia de rendimiento, las aplicaciones cliente que usan la interface ODBC pueden trabajar con muchas RDBMS diferentes.

SQL Anywhere soporta todas las versiones de ODBC Versión 2.1 funciones API (core, level 1 and level 2).

Interfaces incluidas en SQL.

La interface de programación original de SQL Anywhere está incluida en SQL. SQL Anywhere viene con librerías de vínculo dinámicas (librerías normales en DOS, QNX, y Netware) y un preprocesador que habilita el desarrollo de aplicaciones C y C++ usando SQL incluido.

Interfaces de programación de alto nivel.

SQL Anywhere también provee dos interfaces de programación de alto nivel, DDE (para Windows y Windows NT Server solamente), y HLI (Windows, OS/2, y Windows NT Server solamente). Las interfaces de alto nivel de SQL Anywhere son también proporcionadas por el sistema ODBC de desarrollo de aplicaciones y las aplicaciones.

Interface DDE.

El servidor DDE es una aplicación Windows que activa su acceso y altera los datos en la Base de Datos SQL Anywhere usando el **intercambio dinámico de datos** (DDE - dynamic data exchange). Muchas aplicaciones Windows, incluso hojas de cálculo principales y procesadores de palabras, soportan el protocolo DDE.

Interface HLI.

El HLI es proporcionado como un DLL para Windows, OS/2, y Windows NT Server, y puede ser usado desde muchas aplicaciones o ambientes que pueden llamar a los DLLs. Esto es más simple que usar el ODBC, pero también es lento y tiene menos funcionalidad.

Los programas SQL Anywhere.

SQL Anywhere incluye un grupo de herramientas de administración de Bases de Datos, tan satisfactorio como el mismo motor de base de datos de SQL Anywhere.

Cada herramienta de la Base de Datos es una aplicación cliente y se comunica con el motor de la Base de Datos usando la interface SQL Anywhere incluida.

La versión Windows 3.x de muchos de los programas tiene un nombre ligeramente diferente (termina con una **W**), para que las aplicaciones Windows 3.x puedan coexistir en el path (camino del sistema) con aplicaciones Windows 95, DOS, OS/2, o Windows NT.

2.3. Componentes de Sybase Sql Anywhere.

2.3.1. Componentes proporcionados por Sql Anywhere.

Sybase SQL Anywhere es un sistema administrador de Bases de Datos multi-plataforma que está disponible en las ediciones standard y profesional, y las cuales están incluidas en una parte de otros varios productos, incluyendo Powersoft Powerbuilder, Optima++, y S-Designer. Dependiendo del sistema operativo que esté usando, y qué producto en particular ha comprado, tiene acceso a algunas o a todas las características descritas en este trabajo de investigación.

Los componentes disponibles dependen del sistema operativo en el que esté trabajando, y en particular del producto que ha comprado. A continuación se describirá los contenidos de la Edición Standard de SQL Anywhere. Herramientas adicionales y componentes son incluidos en SQL Anywhere Profesional. Si recibe SQL Anywhere como parte de otro producto, debe consultar la documentación y ver qué componentes están disponibles.

Componentes de SQL Anywhere.

El CD-ROM de la Edición Standard de SQL Anywhere incluye los siguientes componentes:

- Standalone Database Engine - Motor de Base de Datos Standard .- Para que tenga acceso a la base de datos de la misma máquina.
- Network Server and Client - Servidor y cliente de red .- Para tener acceso a Bases de Datos a través de una red.
- SQL Remote - SQL Remoto .- Tecnología basada en replicación de mensajes.
- Sybase Central .- Utilidad de administración gráfica.

- ISQL .- Utilidad interactiva SQL.
- Command-line administration utilities - Utilidad de administración de comandos en línea .- Para administración tareas con de Bases de Datos.

2.3.2. Variables de Entorno (Environment Variables).

La siguiente es una lista de las variables de entorno que son usadas por SQL Anywhere y una descripción de dónde éstas son usadas.

Variable de entorno SQLANY.

SQLANY = path

Esta variable es usada para almacenar el directorio dónde SQL Anywhere está instalado. El directorio de instalación por defecto es C:\SQLANY50. El proceso de instalación automática añade la variable de entorno SQLANY a su entorno de inicio. La variable SQLANY es usada por los archivos batch o archivos de comando que forman parte de los ejemplos incluidos en SQL.

En QNX, SQL Anywhere es instalado en un directorio fijo y la variable SQLANY no es requerida.

Variable de entorno SQLCONNECT.

SQLCONNECT = keyword(palabra clave) = valor ; ...

SQLCONNECT = keyword(palabra clave)# = valor ; ...

La variable de entorno SQLCONNECT especifica los parámetros de conexión que son usados por varias de las herramientas de Base de Datos para conectarse a un motor de Base de Datos o servidor de red. Esta es una lista de parámetros de configuración de la forma KEYWORD = value, delimitado por punto y coma. El signo de número “#” es una alternativa para emparejar signos, y debe ser usado cuando se configura los parámetros de conexión listados para la variable de entorno SQLCONNECT, como usando “=” dentro de una configuración de variable de entorno es un error de sintaxis.

Verbose Keyword (verbo clave)	Forma corta
Userid (identificación de usuario)	UID
Password (clave de usuario)	PWD
ConnectionName (nombre de conexión)	CON
EngineName (nombre de motor)	ENG
DatabaseName (nombre de Base de Datos)	DBN
DatabaseFile (archivo de Base de Datos)	DBF
DatabaseSwitches (switches de BDD)	DBS
AutoStop	AutoStop
Start (inicio)	Start
Unconditional (incondicional)	UNC
DataSourceName (nombre de origen de datos)	DSN

Variable de entorno SQLPATH.

SQLPATH = path (vía); ...

PATH = path (vía); ...

ISQL busca en SQLPATH los archivos de comandos y archivos de ayuda antes de buscar en el path del sistema.

Variable de entorno SQLREMOTE.

SQLREMOTE = path

Direcciona la conexión para el archivo de mensaje en la replicación de SQL remoto son subdirectorios de la variable de entorno SQLREMOTE. Esta variable debe señalar a un directorio compartido.

Variable de entorno SQLSTART.

SQLSTART = start-line(línea de inicio)

SQLSTARTW = start-line(línea de inicio)

La información sobre la variable de entorno SQLSTART viene añadida a la variable de entorno SQLCONNECT como el parámetro **Start**.

Variable de entorno TMP.

TMP = directorio

TMPDIR = directorio

TEMP = directorio

El motor de Base de Datos crea archivos temporales para varias operaciones tales como escoger y llevar a cabo uniones. Esos archivos temporales serán ubicados en el directorio especificado por las variables de entorno TMP, TMPDIR, o TEMP (el motor de Base de Datos toma el primero de los tres que encuentre). Si ninguna de las variables de entorno es definida, los archivos temporales son ubicados en el directorio actual.

Códigos de respuesta de los componentes del Software.

Los componentes de Base de Datos usan los códigos de la siguiente lista. El archivo SQLDEF.H tiene constantes definidas para esos códigos.

Código	Explicación
0	Éxito, correcto
1	Fallo general
2	Formato de archivo no válido
3	Archivo no existe, imposible abrir
4	Fuera de memoria
5	Terminado por usuario
6	Comunicación fallida
7	Falta nombre de Base de Datos requerido
8	Protocolo Cliente/Servidor desequilibrado
9	Imposible conectar al motor de Base de Datos
10	No se ejecuta el motor de Base de Datos
11	Servidor de BDD no existe
254	Extender tiempo de detención
255	Parámetro no válido en línea de comando

2.4. Protocolos soportados por Sybase Sql Anywhere.

Configurando apropiadamente, SQL Anywhere se ejecuta bajo las siguientes redes y protocolos:

- El servidor de Base de Datos NetWare se ejecuta en todas las redes Novell usando los protocolos IPX y TCP/IP.
- El servidor de Base de Datos Windows NT y Windows 95 se ejecutan en redes usando los protocolos NetBIOS, TCP/IP o IPX.
- El servidor de Base de Datos OS/2 se ejecuta en redes usando los protocolos NetBIOS, TCP/IP o IPX.
- El servidor de Base de Datos DOS se ejecuta en redes usando los protocolos NetBIOS o IPX.
- El servidor de Base de Datos QNX se ejecuta en redes usando mensajes QNX o TCP/IP.
- El servidor de Base de Datos OS/2 y Windows NT usan el protocolo de nivel de transporte Named Pipes para comunicación con aplicaciones cliente DOS y Windows 3.x ejecutándose en la misma máquina que el motor de Base de Datos. Named Pipes no son usados para comunicaciones de red.
- El servidor de Base de Datos Windows 3.x usa un transporte basado en mensajes Windows para comunicarse con un cliente ejecutándose en la misma máquina.

El SQL Anywhere Client para cada plataforma soporta o permite los mismos protocolos que el correspondiente servidor, para que SQL Anywhere se ejecute apropiadamente, el protocolo agrupado en el cliente y servidor deben ser compatibles con cada capa.

2.5. Características de Sybase Sql Anywhere.

2.5.1 Diseño de la Base De Datos.

SQL Anywhere es un sistema administrador de Base de Datos relacionales. Para ello se presenta algunos conceptos básicos y se explica paso a paso el diseño de la Base de Datos.

Diseñar una Base de Datos no es una tarea difícil, por pequeño o mediano que sea el tamaño de la Base de Datos. Un mal diseño de la Base de Datos puede conducir a un ineficiente y posiblemente poco confiable sistema de Base de Datos. Como las aplicaciones cliente están creadas para trabajar en partes específicas de una Base de Datos, y contar con el diseño de la Base de Datos, un mal diseño puede dificultar la revisión.

Conceptos de Bases de Datos Relacionales.

Tablas de Bases de Datos.

En una Base de Datos relacional, todos los datos se mantienen en **tablas**, las cuales están formadas de **filas** (rows) y **columnas** (columns).

Cada tabla tiene una o más columnas, y cada columna tiene asignado un tipo de datos específico, tal como un número entero, una cadena de caracteres (para texto), o una fecha. Cada fila de la tabla tiene un valor para cada columna.

Un típico fragmento de una tabla conteniendo información de empleados puede verse como esto:

Emp_id	Emp_apellido	Emp_nombre	Emp_phone
10057	Huong	Zhang	1096
10693	Donaldson	Anne	7821

Características de las tablas de Bases de Datos.

Las tablas de una Base de Datos relacional tienen algunas características importantes:

- No tiene ninguna importancia el orden de las filas o columnas.
- Cada fila contiene uno y solamente un valor para cada columna.
- No existen dos filas que contengan el mismo grupo de valores.
- Cada valor para una columna dada tiene el mismo tipo.

El siguiente cuadro lista algunos de los términos formales e informales usados en las Base de Datos relacionales describiendo tablas y sus contenidos, junto con sus equivalencias en otras Bases de Datos no relacionales. A continuación los términos informales.

Formal	Informal	Equivalencia
Término relacional	Término relacional	Término no relacional
Relación	Tabla	Archivo
Atributo	Columna	Campo
Tupla (tuple)	Fila	Registro

Claves en Bases de Datos relacionales.

Las claves primaria y foránea permiten identificar a cada fila en las tablas de la Base de Datos, y permiten relaciones entre las tablas que han sido definidas. Esas claves definen la estructura de la relación de una Base de Datos.

Cada tabla tiene una clave primaria.

Cada tabla en una Base de Datos relacional tiene una **clave primaria**. La clave primaria es una columna, o grupo de columnas, que permiten que cada fila en la tabla sea identificada como única. Dos filas no pueden tener el mismo valor en la clave primaria.

Ejemplo:

En una tabla con información acerca de los empleados de una empresa, la clave primaria puede ser el número de identificación asignado a cada empleado.

Las tablas están relacionadas por medio de las claves foráneas o externas (foreign keys)

La información de una tabla está relacionada con otras tablas por medio de la clave foránea (foreign key).

Ejemplo:

Una Base de Datos tiene una tabla con los datos de los empleados y una tabla con los datos de los departamentos. La tabla departamento tiene las siguientes columnas:

- Dept_id (un número de identificación para el departamento que es la clave primaria para la tabla)
- Dept_nombre (contiene el nombre del departamento)

Para buscar al empleado de un departamento, no es necesario poner el nombre del departamento del empleado en la tabla de empleados. En cambio, la tabla de empleados contiene una columna que guarda la identificación del departamento al cual pertenece el empleado. Esta es llamada Foreign Key para la tabla de departamentos. Una clave foránea se refiere a una fila en particular en la tabla que contiene su correspondiente clave primaria.

En este ejemplo, la tabla empleados (la cual contiene la clave foránea en la relación) es llamada la tabla foránea (externa) o tabla referencial. La tabla departamento (la cual contiene la clave primaria) es llamada la tabla primaria o la tabla referenciada.

Si una clave primaria no está asignada, la combinación de todas las columnas en la tabla hará de clave primaria. Esto puede conducir a un muy extenso registro de transacciones.

Otros objetos de la Base de Datos.

Una Base de Datos relacional posee más que un grupo de tablas relacionadas. Entre algunos objetos que crea una Base de Datos relacional de SQL se presenta:

- Los índices.- Los índices permiten acelerar la visualización de la información.
- Los views (vistas-consultas).- Son tablas calculadas.
- Los stored procedures y los triggers.- Estas son rutinas que posee la misma Base de Datos que actúan sobre la información de la Base de Datos.

Todos estos objetos están diseñados con varios objetivos sobre las tablas base que mantienen la información. Una **tabla base** es una tabla que está presente permanentemente en la Base de Datos.

2.5.2 Planificación de la Base De Datos.

Para el diseño de una Base de Datos, se planifica qué tablas requiere y qué datos va a contener cada una, también debe determinar como serán relacionadas las tablas.

Se necesita determinar a cerca de qué cosas quiere grabar información (cada una de estas cosas es una **entidad**) y cómo esta información estará relacionada (**relationships**). Una técnica muy recomendable en su diseño de Base de Datos es dibujar trazar un dibujo o esquema de sus tablas. Este diseño gráfico de una Base de Datos es llamado **diagrama entidad-relación** (E-R). Usualmente, cada cuadro en un diagrama E-R corresponde a una tabla en una Base de Datos relacional, y cada línea del diagrama corresponde a una clave foránea.

Diseño Entidad-Relación.

El diseño entidad-relación es un ejemplo de diseño de Base de Datos top-down. Actualmente hay métodos sofisticados y herramientas disponibles para realizar este tipo de diseños de Base de Datos en forma muy detallada.

Entidad.

Cada tabla de la Base de Datos describe una **entidad**, que equivale a un sustantivo. Empleados, ventas, departamentos y productos son ejemplos de entidades representadas por una tabla en una Base de Datos. Las entidades que crea dentro de la Base de Datos surgen de las actividades para las cuales usará su Base de Datos, para registrar ventas, para llevar información de sus empleados, o alguna otra actividad.

Relaciones (relationship).

Una relación entre entidades de una Base de Datos, equivale a un verbo. Un empleado es asociado con un departamento, o una oficina es localizada en una ciudad. Las relaciones

en una Base de Datos aparecen como claves foráneas entre tablas relacionadas, o aparecen como tablas separadas.

Las relaciones en la Base de Datos son una codificación de reglas o practicas que gobiernan los datos en las tablas. Si cada departamento tiene un departamento principal, entonces una columna simple puede ser construida dentro de la tabla departamento para guardar el nombre del departamento principal. Cuando estas reglas están dentro de la estructura de la Base de Datos, no hay provisión para excepciones: en ninguna parte podrá poner un segundo departamento principal, y duplicar el valor de entrada en departamento significará duplicar la identificación (clave) del departamento, la cual es la clave primaria. Esto no es permitido.

Relaciones entre tablas.

Hay tres géneros o tipos de relaciones entre tablas:

- o Relación uno a muchos.



- o Relación uno a uno.



- o Relación muchos a muchos.



El proceso de diseño.

Hay cinco pasos principales en el proceso de diseño:

- Paso 1: identificar las entidades y relaciones.
- Paso 2: identificar los datos requeridos.
- Paso 3: normalizar los datos.
- Paso 4: establecer las relaciones entre tablas.
- Paso 5: verificar el diseño.

Paso 1: Identificar las entidades y relaciones.

Identificar las entidades en su diseño y sus relaciones:

- Definir las actividades de alto nivel (high-level activities). Identificar las actividades generales para las que se use esta Base de Datos. Por ejemplo, desea guardar información acerca de los empleados.
- Identificar las entidades. Para la lista de actividades, las áreas sobre las cuales necesita mantener información. Estas llegarán a ser tablas. Por ejemplo, empleados contratados, asignados a un departamento, etc.
- Identificar las relaciones. Observe las actividades y determine qué relaciones existen entre las tablas. Por ejemplo, hay una relación entre departamentos y empleados. Se da un nombre a esta relación.
- Descomponer las actividades. empieza con las actividades de alto nivel. Ahora examine esas actividades más cuidadosamente para ver si ellas pueden ser descompuestas dentro de las actividades de alto nivel. Por ejemplo, una actividad de alto nivel tal como mantener información de los empleados puede ser descompuesta en:
 - Añadir nuevos empleados.
 - Cambiar la información de los empleados existentes.
 - Eliminar determinados empleados.

- Identificar las reglas de su empresa o negocio. Observe la descripción de su empresa y vea qué reglas tiene. Por ejemplo, una regla puede ser que un departamento tiene uno y solamente un departamento principal (head). Estas reglas estarán estipuladas en la estructura de su Base de Datos.

Paso 2: Identificar los datos requeridos.

Identificar los datos requeridos:

- Identificar los datos que se manejan.
- Listar todos los datos que necesitará guardar y buscar. Los datos que se describen en la tabla (sujeto) responden a las siguientes interrogantes: quién, qué, dónde, cuándo, y por qué.
- Organice los datos para cada tabla.
- Liste los datos disponibles para cada tabla como parece más apropiado.
- Organice los datos para cada relación entre tablas.
- Liste los datos que se utilizan para cada relación (si existen).

Paso 3: Normalizar los datos.

La normalización es una serie de pruebas que se usa para eliminar las redundancias en los datos y asegurarse que los datos están asociados con la tabla o relación correcta. Hay cinco pruebas. Aquí se describiré tres de las más usadas.

Formas normales.

Las formas normales son los tests (pruebas) que generalmente se usan para normalizar los datos. Cuando sus datos pasan por el primer test, se considera que está en primera forma normal, cuando pasan el segundo test, están en segunda forma normal, y cuando pasan el tercer test, están en tercera forma normal.

Normalizar los datos:

- Liste los datos.
- Identifique por lo menos una clave para cada tabla. Cada tabla necesita tener una clave primaria (primary key).
- Identifique las claves para las relaciones. Las claves para las relaciones son las claves de las dos tablas a relacionarse.
- Verifique los datos calculados en su listado. Los datos calculados normalmente no están grabados en su Base de Datos.
- Ponga sus datos en primera forma normal.
- Remueva los datos repetitivos de las tablas y relaciones.
- Cree una o más tablas y relaciones con los datos que se remueve (remove).
- Ponga los datos en segunda forma normal.
- Identifique las tablas y relaciones con más de una clave.
- Remueva los datos que dependen solamente en una parte de la clave.
- Cree una o más tablas y relaciones con los datos que se remueve (remove).
- Ponga los datos en tercera forma normal.
- Remueva los datos que dependen de otros datos de la tabla o relación y no de la clave.
- Cree una o más tablas y relaciones con los datos que se remueve (remove).

Datos y claves.

Antes de que empiece a normalizar (hacer los tests con sus datos), simplemente liste los datos e identifique una única clave primaria para cada tabla. La clave se puede crear de un fragmento de dato (columna) o varios fragmentos o columnas (clave concatenada).

La clave primaria es el grupo de columnas que identifican como únicas a las filas en una tabla.

Paso 4: Resolver las relaciones.**Resolver relaciones que acarrear datos.**

Cuando finaliza el proceso de normalización, su diseño está casi completo. Todo lo que se necesita hacer es resolver las relaciones.

Algunas de estas relaciones pueden acarrear datos. Esta situación a menudo ocurre en las relaciones muchos a muchos.

Cuando este es el caso, cambie la relación a una tabla. La clave para la nueva tabla queda igual como era para la relación.

Resolver relaciones que no acarrear datos.

Para implementar relaciones que no acarrear datos, se necesita definir claves foráneas. Una **clave foránea (foreign key)** es una columna o grupo de columnas que contienen valores de clave primaria de otras tablas. La clave foránea permite tener acceso a datos de más de una tabla al mismo tiempo.

Hay varias reglas básicas que ayudan a decidir dónde poner las claves.

Una a muchas.- En una relación una a muchas, la clave primaria en la una es acarreada (transportada) en las muchas.

Una a una.- En la relación una a una, la clave foránea puede ir en cualquier tabla. Si hay un lado obligatorio (mandatory), y no hay otro, la clave debe ir en el lado obligatorio.

Muchos a muchos.- En una relación muchos a muchos, una nueva tabla es creada con las dos claves foráneas. Las tablas existentes están ahora relacionadas una con otra por medio de esta nueva tabla.

Paso 5: Verificar el diseño.

Antes de implementar su diseño, necesita estar seguro que cumplirá con sus necesidades. Examine las actividades que se identificó al empezar el proceso de diseño y asegúrese que puede acceder a todos los datos en las actividades requeridas:

- Puede encontrar una ruta para obtener toda la información que necesita?
- El diseño cumple con sus requerimientos?
- Están disponibles todos los datos requeridos?

Si puede contestar que sí a todas estas preguntas, está listo para implementar su diseño.

Diseñando las propiedades de las tablas de la Base de Datos.

El diseño de la Base de Datos especifica qué tablas tiene y qué columnas contiene cada una. Para cada columna, necesita decidir el nombre correspondiente, el tipo de datos y tamaño o amplitud, si se permiten o no valores nulos, y si quiere que la Base de Datos condicione o limite los valores permitidos en la columna.

Escoger los nombres de las columnas.

Los nombres de las columnas pueden ser una combinación de letras, números o símbolos. De cualquier modo, si el nombre de la columna contiene otros caracteres que no sean letras o números, guiones de subrayado (_), o si no empieza con una letra, o este es una palabra clave entonces en cualquier momento que el nombre de la columna sea usado, necesitará estar encerrado entre doble comillas.

Escoger tipos de datos para las columnas

Los tipos de datos permitidos por SQL Anywhere incluyen:

- Tipo de datos integer o entero (int, integer, smallint)
- Tipo de datos decimal (decimal, numeric)
- Tipo de datos punto flotante (float, double)
- Tipo de datos character o caracter (char, varchar, long varchar)
- Tipo de datos binario (binary, long binary)
- Tipo de datos date/time o fecha/hora (date, time and timestamp)
- Tipo de datos definido por el usuario o user defined

El tipo de datos afecta al tamaño máximo de la columna. Por ejemplo, si especifica SMALLINT, una columna puede contener un valor máximo de 32,767. Si especifica INTEGER, el máximo valor es 2,147,483,647. En el caso de CHAR, la máxima extensión de un valor en la columna necesita ser especificada.

El tipo de datos long binary puede ser usado para mantener en su Base de Datos, información tal como imágenes (por ejemplo, guardados como bitmaps o mapas de bits) o documentos procesados en Word.

NULO y NO NULO (NULL and NOT NULL).

Cuando el valor de la columna es obligatorio (mandatory) para un registro, define la columna como NO NULO - NOT NULL. Por otra parte, la columna está permitida a contener el valor nulo el cual represente a ningún valor. Por default en SQL está permitido el valor NULO; se debe explícitamente declarar las columnas que serán NO NULAS a menos que haya una buena razón para permitir valores nulos.

Escoger constraints (restricciones).

Aún cuando el tipo de datos de una columna restrinja los valores permitidos en dicha columna (por ejemplo, solamente números o solamente fechas), se puede restringir aún más los valores permitidos.

Puede restringir los valores de cualquier columna especificando un CHECK CONSTRAINT. Se puede usar cualquier condición válida que pueda aparecer en la cláusula WHERE para restringir los valores permitidos, aunque la mayoría de CHECK CONSTRAINTS usan las condiciones BETWEEN o IN.

Escoger claves primarias y foráneas.

La clave primaria es la columna o columnas que identifican como únicas a las filas de una tabla. Si su tabla está propiamente normalizada, una clave primaria debe estar definida como parte del diseño de la Base de Datos.

Una clave foránea es una columna o grupo de columnas que contienen los valores de clave primaria de otras tablas. Las claves foráneas crean relaciones una a una y una a muchas en su Base de Datos. Si su diseño está propiamente normalizado, las claves foráneas deben ser definidas como parte de su diseño de Base de Datos.

2.5.3 Conexión con la Base De Datos.

Toda aplicación cliente que usa una Base de Datos necesita establecer una **conexión** a esa Base de Datos antes de realizar cualquier trabajo. Mientras se solicita al usuario introducir una identificación de usuario (user ID), una contraseña (password) y otros parámetros, la conexión está establecida por la aplicación cliente por medio de una de las interfaces de programación de SQL Anywhere.

Una vez establecida la conexión, se forma el canal a través del cual todas las actividades que se ordene desde la aplicación cliente se realizan. Por ejemplo, los permisos que se tienen para manipular la Base de Datos son determinados por el user ID, y el motor de Base de Datos está informado de su user ID porque ésta forma parte de la petición para establecer la conexión.

Parámetros de conexión con Base de Datos.

Cuando una aplicación se conecta a la base de datos, ésta usa un grupo de **parámetros de conexión**, que funcionan conjuntamente con un **connection string** la conexión, para definir la conexión. Por ejemplo, connection string especifica una identificación de usuario (user ID) y su password (contraseña), tan satisfactoriamente como otra información opcional; todas las conexiones que usan el ODBC deben suministrar un nombre de Fuente de los Datos (Data Source).

Parámetros de conexión usados con Bases de Datos SQL Anywhere.

Una gran clase de aplicaciones cliente, usan un grupo similar de parámetros de conexión cuando intentan conectarse a una Base de Datos SQL Anywhere. Estos son:

- Todas las aplicaciones conectadas a Bases de Datos SQL Anywhere a través de la interface usando la función SQLDriverConnect. Esta incluye aplicaciones desarrolladas usando esos sistemas.

- ❑ Las utilidades de Bases de Datos SQL Anywhere como ISQL, DBBACKUP, DBWATCH, DBUNLOAD, y DBVALID.
- ❑ Todas las demás aplicaciones SQL incluidas se conectan usando la función `db_string_connect`, la cual es recomendada para conectarse a una Base de Datos incluida dentro de SQL.

Métodos usados para especificar los parámetros de conexión.

La manera que se especifique los parámetros de conexión depende de la aplicación cliente en particular en la que se trabaje. Las aplicaciones pueden usar uno de los siguientes métodos:

- ❑ Aplicaciones con ODBC activado (enabled) usan una configuración Origen de Datos (data source), a la cual se puede entrar usando el programa Administrador de ODBC (ODBC Administrator).
- ❑ Se puede sugerir cómo llenar los campos en la caja de diálogo.
- ❑ Se puede requerir ingresar los argumentos en la línea de comando.
- ❑ La aplicación puede mostrar los archivos pre-especificados para buscar los valores de los parámetros.
- ❑ La aplicación puede usar grupos de variables de entorno.
- ❑ La aplicación puede tener un grupo fijo de parámetros establecidos permanentemente.

Conexión de las aplicaciones cliente a una Base de Datos.

Una vez que se ha especificado el parámetro de conexión, usando cualquier método que su aplicación cliente requiera, la aplicación intenta conectarse a la Base de Datos. El procedimiento del cliente listado a continuación es exactamente el mismo para cada uno de los dos importantes grupos de aplicaciones cliente:

- ❑ Cualquier aplicación cliente ODBC activado (enabled). Muchos sistemas de desarrollo de aplicaciones, tales como Powersoft Powerbuilder, pertenecen a esta clase de aplicaciones.

- Cualquier aplicación cliente usando SQL incluido y usando la función recomendada para la conexión a una Base de Datos (db_string_connect). Todas las herramientas (tools) de Bases de Datos SQL Anywhere, incluyendo ISQL, son una parte de este grupo.

Conectándose a una Base de Datos.

La conexión a una Base de Datos es fundamentalmente importante: no puede trabajar en ninguna Base de Datos sin antes haberse conectado a ella. Por esta razón, el proceso siguiente para aplicaciones cliente, se detalla a continuación:

1. La aplicación intenta encontrar el motor o servidor de Base de Datos apropiado.
 - Si el nombre del motor (EngineName) está especificado, la aplicación despliega el nombre de un motor de Base de Datos local, y entonces para un cliente SQL Anywhere (DBCLIENT) con su server name (nombre del servidor). Para QNX, la aplicación despliega en la red para un motor o servidor.
 - Si el nombre del motor no está especificado, la aplicación intenta conectarse al motor local por default (si hay solamente un motor ejecutándose, este es el default, de otra forma, el default escogido es el sistema operativo especificado).
2. Si ningún motor de Base de Datos local es encontrado, y el Cliente SQL Anywhere no está ejecutándose, la aplicación inicializa un motor de Base de Datos o SQL Anywhere Client usando el parámetro **Start**.
 - Si se especifica el parámetro Start, pero DatabaseFile (Archivo de Base de Datos) es especificado, la aplicación intenta inicializar un motor de Base de Datos en el archivo nombrado (named file), usando el parámetros Start por default (omisión).
3. Si la aplicación encuentra con éxito o inicializa una Base de Datos, la aplicación intenta conectarse con la Base de Datos (si ningún motor ha sido encontrado o inicializado aún, el intento de conexión falla en este punto).

- Si ningún Nombre de Base de Datos (DatabaseName) ni ningún Archivo de Base de Datos (DatabaseFile) es conocido o está dado, la aplicación intenta conectarse a la Base de Datos por omisión (default) en el motor, usando los parámetros especificados: Userid, Password, y ConnectionName.
- Si la Base de Datos nombrada con DatabaseName está en uso, la aplicación intenta conectarse a la Base de Datos usando los parámetros especificados: Userid, Password, y ConnectionName.
- Si DatabaseName no está especificado, pero DatabaseFile sí, la aplicación intenta conectarse a la Base de Datos cuyo nombre es la raíz de DatabaseFile.
- Si no encuentra ninguna Base de Datos en funcionamiento, pero una Base de Datos especificada por la raíz de DatabaseFile está en funcionamiento, la aplicación intenta conectarse a la Base de Datos usando los parámetros especificados: Userid, Password, y ConnectionName.
- Si la Base de Datos correspondiente a DatabaseFile no está en funcionamiento, la aplicación envía una petición (requerimiento) al motor o servidor de red para inicializar una Base de Datos usando los parámetros DatabaseFile, DatabaseName, y DatabaseSwitches (el parámetro AutoStop determina si la Base de Datos se detiene automáticamente cuando la última conexión a la Base de Datos es desconectada). La aplicación entonces intenta conectarse a la Base de Datos usando los parámetros: Userid, Password, y ConnectionName.
- Si la conexión no se produce, el intento ha fallado en este punto.

Sentencia CONNECT de ISQL:

La utilidad ISQL tiene una conducta diferente a la incluida por defecto en SQL cuando una sentencia CONNECT es usada mientras ya se ha conectado a una Base de Datos. Si

ninguna Base de Datos o motor es especificado en la sentencia CONNECT, ISQL se conecta a la actual Base de Datos, en lugar de la Base de Datos por defecto. Esta conducta es requerida para operaciones de volver a cargar (recargar) la Base de Datos.

Conectándose desde las utilidades de SQL Anywhere.

Todas las utilidades de base de datos SQL Anywhere que se comunican con el motor de base de datos (actúan directamente sobre los archivos de la base de datos) usan SQL empotrado.

Cómo las herramientas de Base de Datos obtienen los valores de los parámetros de conexión.

Muchas de las herramientas de Base de Datos obtienen los valores de los parámetros de conexión de la Base de Datos de la siguiente manera:

- Si hay valores especificados en la línea de comando, esos valores son usados como parámetros de conexión. Por ejemplo, el siguiente comando inicializa una copia de seguridad de la Base de Datos por default en el motor de Base de Datos por default usando la identificación de usuario DBA y el password SQL:
`DBBACKUP -C "UID=DBA; PWD=SQL" c:\backup`
- Si falta cualquier valor en una línea de comando, la aplicación muestra la variable de entorno SQLCONNECT. Esta variable no es configurada automáticamente por SQL Anywhere. Si usa una configuración simple de parámetros de conexión, frecuentemente puede querer usar una variable de entorno SQLCONNECT en su archivo AUTOEXEC.BAT (bajo DOS Y Windows 3.x), CONFIG.SYS (bajo OS/2), panel de control (bajo NT) o el login file (bajo QNX).
- Si los parámetros no son configurados en la línea de comando, o la variable de entorno SQLCONNECT, entonces durante el proceso de conexión, la aplicación permite el ingreso de la identificación del usuario y el password para conectarse a la Base de Datos por default en el motor de Base de Datos por default.

Conectándose desde una aplicación ODBC-enabled (activado).

La interface **Open Database Connectivity (ODBC)** es definida por Microsoft Corporation, y es una interface standard para conectarse las aplicaciones cliente a un sistema manejador de Base de Datos en los entornos Windows y Windows NT. Muchas aplicaciones cliente, incluyendo sistemas de desarrollo de aplicaciones, usan la interface ODBC para acceder completamente al sistema de Base de Datos. Hay aplicaciones ODBC-enabled. SQL Anywhere permite la interface ODBC.

Las aplicaciones ODBC-enabled usan un grupo de parámetros de conexión. Sin embargo, ellas obtienen los valores para esos parámetros de formas diferentes en las aplicaciones SQL. ODBC define el origen de los datos. Cada descripción del origen de datos contienen cada uno de los parámetros de conexión requeridos. Los demás parámetros de conexión serán obtenidos del usuario (por instancia, se puede ingresar en una caja de diálogo su identificación de usuario y su password), o internamente por la aplicación cliente.

Los ODBCs permitidos por SQL Anywhere son principalmente Windows NT, DOS y QNX, y OS/2.

CAPITULO III

INTRODUCCIÓN A WINDOWS NT SERVER, POWER DESIGNER Y POWER BUILDER

3.1. Windows Nt Server.

Windows NT es un sistema operativo que ayuda a organizar la forma de trabajar a diario con la PC. Las letras NT significan Nueva Tecnología. Fue diseñado para uso de compañías grandes, por lo tanto realiza muy bien algunas tareas tales como la protección por contraseñas. Windows actúa como su ejecutivo personal, personal de archivo, mensajeros, guardias de seguridad, asistentes administrativos y mantenimiento de tiempo completo.

Microsoft Windows NT Server es un sistema operativo diseñado para su uso en servidores de red de área local (LAN). Ofrece la potencia, la manejabilidad y la capacidad de ampliación de Windows NT en una plataforma de servidor e incluye características, como la administración centralizada de la seguridad y tolerancia a fallos más avanzada, que hacen de él un sistema operativo idóneo para servidores de red.

Windows NT Server es a la vez un sistema operativo para computadoras (ordenadores) personales y un sistema operativo para red. Puesto que incorpora funciones de red, las redes de Windows NT Server se integran de forma óptima con el sistema operativo básico, facilitando el uso y la administración de las funciones.

3.1.1. Descripción general de Windows Nt Server.

Windows NT Server es un sistema operativo para servidores, ampliable e independiente de la plataforma. Puede ejecutarse en sistemas basados en procesadores Intel x86, RISC y DEC Alpha, ofreciendo al usuario mayor libertad a la hora de elegir sus sistemas informáticos. Es ampliable a sistemas de multiproceso

simétrico, lo que permite incorporar procesadores adicionales cuando se desee aumentar el rendimiento.

Posee múltiples threads (subprocesos) de ejecución, lo que permite utilizar aplicaciones más potentes. La protección de la memoria garantiza la estabilidad mediante la asignación de áreas de memoria independientes para el sistema operativo y para las aplicaciones, con el fin de impedir la alteración de los datos. La capacidad de multitarea de asignación prioritaria permite al sistema operativo asignar tiempo de proceso a cada aplicación de forma eficaz. Windows NT Server incluye, asimismo, diversas funciones de red.

Seguridad.

Windows NT ofrece gran seguridad por medio del acceso por cuentas y contraseñas. Es decir un usuario debe tener su cuenta asignada y una contraseña para poder tener acceso al sistema.

Contiene protecciones para directorios, archivos, y periféricos, es decir que todo esto se encuentra con una contraseña para poder ser utilizados.

Concepto de Derechos: Permite a un grupo de usuarios efectuar determinadas operaciones.

Cuenta Administrador: Controla todos los permisos y con ellas se puede: dar de alta; asignar cuentas; cancelar derechos

Comunicación: Permite acceder y compartir discos en red. Permite compartir archivos, directorios y periféricos.

Administración Centralizada

Las herramientas de administración de Windows NT Server hace posible trabajar con toda la red desde computadoras corriendo:

- Microsoft Windows 3.x
- Microsoft Windows for Workgroups 3.1x
- Microsoft Windows 95
- Microsoft Windows NT Workstation
- Microsoft Windows NT Server

3.2. Power Designer.

3.2.1. Términos usados en el diseño de Bases de Datos.

Un modelo conceptual de datos (CDM) representa la estructura lógica de un sistema de información, independientemente de cualquier software o sistema de almacenamiento.

Entidad.- Es un objeto distinguible con propiedades descriptivas. Se definen como entidades solo a los objetos cuya información es requerida para el trabajo planeado. Una entidad puede ser tangible o intangible, concreta o abstracta, animada o inanimada.

Atributo.- Es una parte atómica o no susceptible de descomponer, que se necesita conocer a cerca de la entidad. Un atributo es un detalle asociado con una entidad.

Item.- Es una pequeña unidad significativa de información en un modelo de datos. Un ítem corresponde a un atributo cuando está asociado con una entidad.

Dominio.- Es un conjunto de valores desde donde uno o más ítems extraen sus valores actuales.

Relación.- Es una conexión o asociación entre entidades.

Enlace de herencia.- Es una relación especial que define una entidad como un caso especial de una entidad más general.

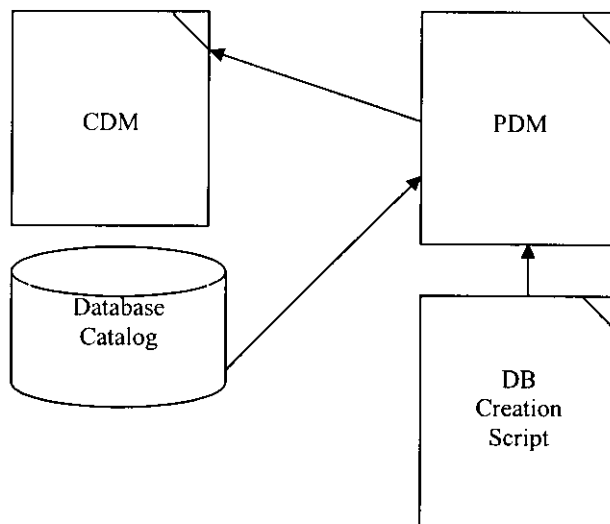
Items y Atributos.- En el entorno de la arquitectura de datos, los ítems y atributos no son objetos diferentes. Un atributo es simplemente un ítem que está asociado con una entidad.

Cuando se define un atributo, realmente se define una asociación entre un ítem y una entidad. Si el ítem no existe, automáticamente la arquitectura de datos lo crea.

ODBC.- La interface ODBC (Open DataBase Connectivity), definido por Microsoft Corporation, es una interface standard para acoplar los sistemas de bases de datos en el entorno de Windows.

Ingeniería reversa.

En arquitectura de datos, la ingeniería reversa es el proceso de leer un catálogo de base de datos o la creación de un script de base de datos para construir un modelo físico de datos (PDM). Adicionalmente, la arquitectura de datos puede construir un modelo conceptual de datos (CDM) desde un PDM.



La ingeniería reversa es usada en algunas situaciones:

- ❑ Una base de datos existente fue designada e implementada sin utilizar el modelamiento de datos.
- ❑ El modelo de datos no fue creado para permitir modificaciones en la base de datos existente.
- ❑ El modelo de datos necesita ser convertido a un diferente RDBMS que requiere de un modelo físico diferente.

- Qué clase de información necesita almacenar.
- Qué clase de información necesita recopilar, cómo obtenerla y en qué forma.
- Cómo será accesada la base de datos.

Cuando las necesidades de implementación son claras, se puede empezar desarrollando el PDM.

Los índices aceleran el acceso a los datos cuando se busque información en las tablas. Se pueden crear índices para una columna en particular de la tabla en el PDM. Cuando ejecuta una búsqueda en una tabla, por ejemplo, al ejecutar una consulta SELECT en SQL, el proceso de búsqueda puede encontrar en un par de filas el criterio buscado más pronto cuando el modelo está indexado. Es mucho más eficiente buscar un índice que buscar a través de los datos contenidos en cada fila.

Índices de clave primaria y clave foránea.

Se pueden crear índices para clave primaria y clave foránea, pero se puede crear también un índice propio para otras columnas, dependiendo del tipo de información que se requiera ingresar a la base de datos.

Por ejemplo, si se necesita buscar un autor en la base de datos, necesita realizar una búsqueda por nombres (o apellidos) de autores. En este caso debe indexar la columna que contenga esta información.

3.2.4. Definiendo una referencia y una integridad referencial.

Una **referencia** simboliza el enlace entre dos tablas. Se puede crear una referencia entre dos tablas en el PDM. Se hace esto cuando se desea una columna en referencia a otra. Cuando se crea una referencia, la clave primaria en la tabla padre migra como una clave foránea en la tabla hija.

La **integridad referencial** determina qué sucede en una columna de clave foránea en una tabla hija cuando se actualiza o elimina el valor correspondiente en la columna de clave primaria de la tabla padre.

3.2.5. Crear vistas.

Una **vista** es un camino alternativo para observar los datos en una o más tablas. Una vista contiene un subgrupo de columnas de una o más tablas. Se puede crear una vista para permitir al usuario ver los subgrupos de las tablas sin necesidad de acceder a las tablas mismas.

Cuando se crea una vista, esto equivale a una definición SQL SELECT para seleccionar objetos en la base de datos. Cuando se crea una vista se selecciona las tablas y columnas que se desea incluir en ella. Esta vista genera una consulta SQL, la cual automáticamente selecciona todas las columnas en estas tablas y despliega un símbolo gráfico que representa la vista.

3.2.6. Usar los Triggers y Procedures.

Un **trigger** es un procedimiento que hace efecto cuando se inserta, elimina o actualiza una tabla específica o una columna. Los Triggers hacen cumplir la integridad referencial, por ejemplo, se despliega un mensaje de error si se intenta actualizar una columna de clave primaria la cual tiene una restricción para actualizarla.

Un PDM puede generar automáticamente ciertos triggers de integridad referencial dependiendo del tipo de base de datos de origen. Se puede también definir triggers adicionales definidos por el usuario.

Se puede generar un Script SQL que contenga triggers, o puede generar los triggers directamente en la base de datos.

Las plantillas de triggers.

Un trigger script se basa en una plantilla. Hay una plantilla diferente para cada tipo de trigger: insertar, eliminar, y actualizar. Se puede crear su propia plantilla de trigger o modificar una plantilla existente, por ejemplo cambiando el tipo de mensajes de error que se van a desplegar.

Una plantilla de trigger contiene variables para tablas, columnas y otras. Antes de generar el script, se puede visualizar previamente el trigger viendo el script con las variables reemplazadas con los nombres actuales.

3.2.7. Generar un Database Script.

Puede generar una base de datos directamente desde el PDM, o puede generar un database script el cual puede correr en su entorno DBMS. Se crea un script para una base de datos en particular.

Los parámetros de generación los cuales están disponibles dependen de la base de datos de origen que haya seleccionado. Por default, la base de datos de origen es seleccionada cuando se abre el PDM, pero se puede seleccionar otra antes de generar el script.

3.3. POWER BUILDER.

3.3.1. Características de Power Builder

Es un entorno de programación que está compuesto por diferentes herramientas, para el desarrollo rápido de una aplicación en el ambiente cliente - servidor.

Permite usar ventanas, botones y todas las herramientas que presenta windows, facilitando su manejo.

Se maneja básicamente con el ratón, y el lenguaje de programación llamado PowerScript.

Esta herramienta de desarrollo está completamente orientada a objetos, lo que permite a equipos de programadores crear aplicaciones gráficas sofisticadas con acceso a información de base de datos locales o en servidores de red.

Características Básicas

PowerBuilder es un desarrollador de aplicaciones, cuyas características básicas son:

- ❑ Soporta una gran variedad de sistemas de gestión de base de datos, tales como: Sybase, Informix, Oracle, Watcom, entre otras.
- ❑ Tiene capacidad de acceder a información de múltiples bases de datos y mostrar esa información en una única ventana.
- ❑ Se trabaja en ambiente cliente - servidor.
- ❑ Posee un objeto inteligente llamado Datawindow que realiza directamente la interface con la base de datos, sin requerir que el programador conozca SQL.
- ❑ Capacidad de utilizar sentencias SQL combinadas en el código.
- ❑ Se puede trabajar en múltiples plataformas, ya que soporta diferentes sistemas operativos y posee drivers nativos para las bases de datos más comerciales.
- ❑ La creación de aplicaciones es sencilla.
- ❑ Se realiza poca programación.
- ❑ Permite usar ventanas, botones y todas las herramientas que presenta el Windows facilitando su manejo.
- ❑ Para construir la aplicación se utilizan painters, allí se definen las propiedades de los objetos y se agregan los controles.
- ❑ Se puede trabajar con múltiples ventanas.

PowerBuilder, ofrece a los programadores:

- ❑ Un entorno de desarrollo profesional.
- ❑ La Orientación a Objetos.
- ❑ Rapidez de aprendizaje y de desarrollo.
- ❑ Generación automática de código SQL.
- ❑ Acceso a las aplicaciones de Windows.
- ❑ Una herramienta de desarrollo asequible y rentable en explotación.
- ❑ Un entorno abierto de desarrollo que dispone de interfaces inteligentes a otras tecnologías Cliente/Servidor.
- ❑ Abierto a la mayoría de Base de Datos del mercado.

- ❑ Abierto a diferentes herramientas CASE.
- ❑ Abierto a Librerías de objetos y control de versiones.
- ❑ Abierto a aplicaciones Windows: DLL, DDE, OLE.

Eventos y Scripts

Un evento se ejecuta por ejemplo cuando se hace **Click** en un botón o al salir de cualquier control.

En el Script se escriben los comandos y funciones que se realizan cuando se ejecuta algún evento.

Funciones

Permite al programador hacer más fácil la programación. PowerBuilder posee una gran cantidad de funciones que se pueden usar en los objetos, por ejemplo se pueden usar para abrir o cerrar una ventana o para activar algún botón.

Librerías

Las librerías o PBL es el contenedor básico de objetos PowerBuilder (ventanas, menús, datawindows). Cualquier cosa que se cree deberá estar contenida en una PBL o librería. Una aplicación puede tener una o más librerías.

De una PBL se puede generar una PBD, que es lo mismo que una PBL pero sin el código fuente, o una DLL típica de windows.

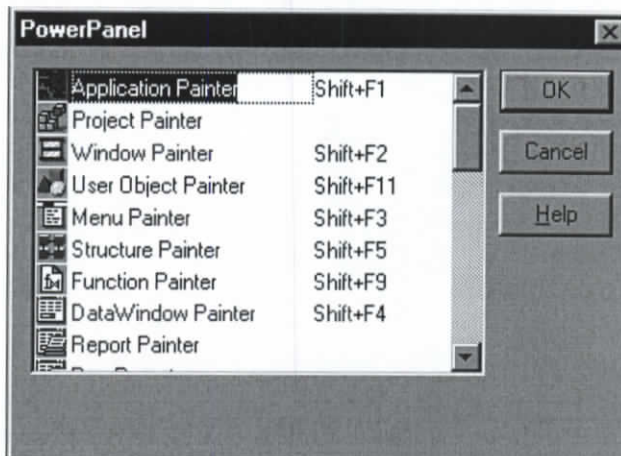
3.3.2. Ambiente de Powerbuilder

Una vez que se ha iniciado el PowerBuilder, lo primero que se aprecia es una ventana con un menú y una barra de herramientas (PowerBar), similar a la que se presenta a continuación:



En PowerBuilder abre por defecto la aplicación activa. Si es la primera vez que se carga el PowerBuilder, aparecerá por defecto la aplicación de ejemplo del PowerBuilder.

En el ambiente de desarrollo, para crear los objetos se usan painters o pintores. Cada pintor tiene un icono asociado. Para los diferentes iconos, seleccionar la opción **PowerPanel** desde el menú File.



Power Panel

La caja de diálogo mostrada arriba se llama 'PowerPanel'. Para invocar un pintor desde el PowerPanel, se necesita seleccionar el pintor haciendo "click" sobre el icono/texto y seleccionar el botón 'OK'.

Antes de hacer cualquier otra cosa se describirá cada uno de los painters o pintores.

PAINTERS O PINTORES DE OBJETOS:

Son pintores para crear objetos que se almacenan en las librerías Power Builder.



Pintor de Aplicación

El objeto de aplicación es el punto de entrada a cualquier aplicación Power Builder. Si se hace una comparación, sería algo así como `main()` en un programa de 'C'.



Pintor de Ventana

En cualquier aplicación, la pantalla es la interface principal entre el usuario y la aplicación. Una pantalla permite hacer entrada de datos, ver informes, etc.

En PowerBuilder, las pantallas se llaman '**Windows**'.

El Pintor de Ventana, como el nombre lo dice, permite pintar y salvar objetos de ventana. Sobre una ventana se puede pintar diversos controles, tales como CommandButtons, ListBoxes, Pictures, etc.



Pintor de Menú

Un menú es simplemente una lista de opciones o comandos. Un ejemplo de menú, el menú del PowerBuilder.

En PowerBuilder se puede crear objetos menú en el pintor de menú y asignarlos a las pantallas (ventanas). También se pueden pintar menús popup (menús sensibles de contexto) en el pintor de menú.



Pintor de Datawindow

El objeto Datawindow permite recobrar datos desde la base de datos u otras fuentes. Mediante un Datawindow también se puede hacer entrada de datos.

Después de pintar el objeto de Datawindow, no puede usarse el objeto solo. Se necesita asociar el objeto Datawindow con el control Datawindow y poner el Datawindow control sobre la pantalla (ventana).



Pintor de Tubería de Datos

El objeto de Tubería de Datos permite copiar tablas y sus datos desde una base de datos a otra, con pocas líneas de código.



Pintor de Objetos de Usuario

Es una colección de uno o más objetos de PowerBuilder/ no-PowerBuilder. Los objetos no-PowerBuilder incluyen objetos de VB, DLLS, OCX y otros. El Objeto de Usuario permite re-usar el código.



Pintor de Consultas

El Pintor de Consultas permite construir declaraciones SQL SELECT que pueden salvarse como objetos de Consulta en la biblioteca de PowerBuilder.



Pintor de Funciones

El pintor de funciones permite construir objetos de función que pueden usarse para definir una serie de comandos, frecuentemente usados. En PowerBuilder, una función es una colección de comandos PowerScript o declaraciones de SQL dinámicos. Se puede referir también a estas funciones en su PowerScript y también en el pintor de Datawindow.



Pintor de Estructura

El Pintor de Estructura permite crear objetos de estructura. Una estructura es un conjunto de variables conexas (que puede tener los tipos diferentes de datos) agrupadas bajo un nombre único.

Otros Pintores



Pintor de Biblioteca

Los objetos que crean en los Pintores de PowerBuilder (aplicaciones, ventanas, menús, funciones, consultas, estructuras, objetos de usuario y DataWindows) se almacenan en un archivo de sistema con extensión ".pbl". En la terminología de PowerBuilder, este archivo se llama "la Biblioteca". El Pintor de Biblioteca permite crear y mantener bibliotecas de PowerBuilder y objetos en esas bibliotecas.



Pintor de base de datos

El Pintor de base de datos permite crear objetos de base de datos como tablas, vistas, índices, etc., sin saber la verdadera sintaxis. Todos los objetos de base de datos que se crean en este pintor se almacenan en la base de datos conectada, no en la biblioteca de PowerBuilder. Los objetos que se crean en los otros painters, como ventanas o el menú, se almacenan en la biblioteca de PowerBuilder.

Se puede hacer también administración de base de datos desde el pintor de base de datos.



Pintor de Proyecto

El Pintor de Proyecto, permite crear Bibliotecas Vinculadas Dinámicas (PBDS PowerBuilder), ejecutables.



Corrida

El Icono de Corrida permite ejecutar la aplicación.



Debug

El pintor de Debug permite inspeccionar la ejecución de la aplicación actual gradualmente, simplificando el descubrimiento de los errores en la aplicación.

Usando los PAINTERS o PINTORES

Al usar los painter o pintores se visualizan 4 componentes básicos:

- ❑ The title bar: Identifica la ventana y muestra el nombre de la aplicación actual.
- ❑ The menu bar: Muestra las opciones disponibles del menú.
- ❑ Workspace: Espacio en el cual se pueden colocar los objetos a utilizar.
- ❑ Toolbars: Provee varias barras de herramientas para hacer el trabajo rápido.

Usando la BARRA DE HERRAMIENTAS

PowerBuilder usa 4 barras de herramientas (Toolbars):

- ❑ PowerBar : contiene iconos para abrir pintores y otras herramientas.
- ❑ PainterBar: manipula componentes en el painter activo.
- ❑ StyleBar: cambia atributos de los textos (tipo, tamaño y fondo).
- ❑ ColorBar: cambia los colores de los componentes en el pintor actual.

El programador puede controlar qué barra de herramientas puede usar y dónde visualizarlas. Además se puede agregar, mover y eliminar iconos en el Power Bar y en el PainterBar.

Usando el PowerBar

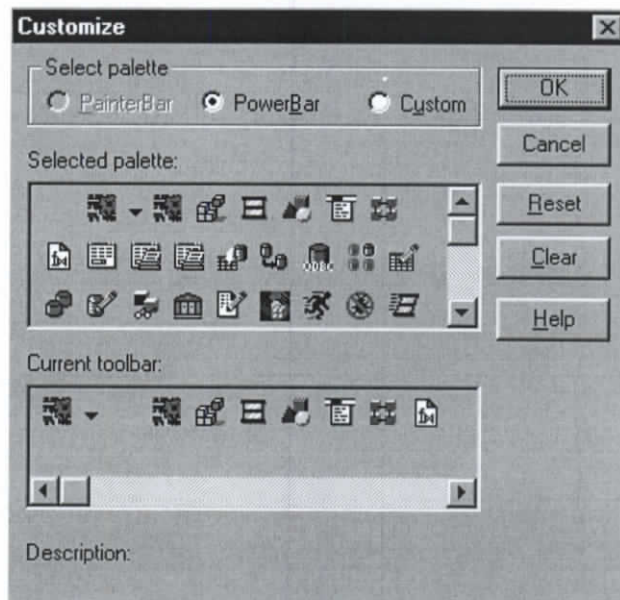
El Power Bar se muestra cuando se inicia una sesión de Power Builder.

El PowerBar es el control principal para construir aplicaciones PowerBuilder. Desde el PowerBar se puede abrir el Painter de PowerBuilder, depurar o correr la aplicación actual, pedir ayuda o personalizar PowerBuilder para cualquier necesidad.

Se puede personalizar PowerBuilder. Por ejemplo, se puede agregar iconos para operaciones que se necesiten con mayor frecuencia.

Usando Iconos

Los iconos en el Power Bar (y en el Power Panel) representan cada una de las herramientas y pintores frecuentemente usados en PowerBuilder.



Usando el Menú POPUP

Power Builder provee un contexto sensible de menú popup que lista los ítems apropiados para el objeto seleccionado. El menú popup está disponible en cualquier lugar del PowerBuilder.

Para usar el Menú PopUp:

- ❑ Seleccionar uno o más objetos
- ❑ Dar click con el botón derecho del mouse, el menú apropiado se desplegará.
- ❑ Seleccionar el ítem deseado.

Trabajando con Ventanas en Power Builder

Cada Painter de PowerBuilder muestra su propia ventana. Se puede abrir varios painters a la vez y también varias instancias de un painter.

Ejemplo : Si ha abierto varias ventanas en PowerBuilder, puede moverse de una a otra, pulsando los botones: **Ctrl + Tab**.

Registro de la Compra de Energía

Sistema Control de Fideicomiso EEASA USUARIO: patty Conexión: SERVER

Registro de Facturas

Periodo: 2 FEBRERO DEL 2007

Id: 2 FEBRERO DEL 2007

Total: 24.435.635,40 Estatus: Nuevo Borrar

II ^a Reg	GENERADOR	II ^a Doc.	Tipo Documento	FECHA EMISION	CANTIDAD ENERGIA (k.h)	VALOR FACTURA	% RETEIL.	Valor Retencion	TOTAL
1	ELECAUSTRO	5452	FACTURA	31/01/2007	15.500.000,00	955.403,00	1 %	9.564,03	946.838,97
2	ELECTROGUAYAS	6942	FACTURA	31/01/2007	15.412.000,00	11.961.569,00	1 %	119.615,69	11.841.953,31
3	EMAAP-Q	6471	FACTURA	31/01/2007	45.121.200,00	63.890,00	1 %	638,90	63.251,10
4	HIDROAGOYAN	9111	FACTURA	31/01/2007	448.745.400,00	1.218.000,00	1 %	12.180,00	1.205.820,00
5	HIDRONACION	4068	FACTURA	31/01/2007	3.961,00	3.921,00	1 %	39,21	3.881,79
6	HIDROPAUTE	1033	FACTURA	31/01/2007	4.206.164,00	4.206.164,00	1 %	42.061,64	4.164.102,36
7	TERMOESMERALDAS	5361	FACTURA	31/01/2007	547.000.000,00	1.850.300,00	1 %	18.503,00	1.831.797,00
8	TERMOPICHINCHA	6305	FACTURA	31/01/2007	48.172.500,00	941.725,00	1 %	9.417,25	932.307,75
9	ELECTROQUIL	4519	FACTURA	31/01/2007	948.048.800,00	3.480.488,00	1 %	34.804,88	3.445.683,12
					TOTALES:	24.682.460,00		248.824,60	24.435.635,40

GENERADOR	TOTAL INICIAL	SALDO
ELECAUSTRO	946.838,97	946.838,97
ELECTROGUAYAS	11.841.953,31	11.841.953,31
EMAAP-Q	63.251,10	63.251,10
HIDROAGOYAN	1.205.820,00	1.205.820,00

Registro de Fondos

Sistema Control de Fideicomiso EEASA USUARIO: patty Conexión: SERVER

Registro de Fondos

Periodo: 2 FEBRERO DEL 2007 Fecha Reg: 13/07/2007

Periodo: 2 FEBRERO DEL 2007

Fecha Reg: 13/07/2007

Total Fondo: 371823927,00 Estatus: DISPONIBLE

Item	RUBROS	FECHA DEPOSITO	VALOR	Estatus
1	CONSUMO ENERGIA	01/03/2007	275.598.483,00	DISPONIBLE
2	ALUMBARDO PUBLICO	01/03/2007	37.859.521,00	DISPONIBLE
3	BOMBEROS	01/03/2007	20.543.901,00	DISPONIBLE
4	FERUM	01/03/2007	11.121.115,00	DISPONIBLE
5	SEGURO CONTRA INCENDIOS	01/03/2007	136.100,00	DISPONIBLE
6	TASA BASURA	01/03/2007	16.642.238,00	DISPONIBLE
7	TASAS Y TERCEROS	01/03/2007	10.830.969,00	DISPONIBLE
8	RETENCION	01/03/2007	-514.083,00	DISPONIBLE
9	RETENCION BANCOS	01/03/2007	-227.693,00	DISPONIBLE
10	VENATANILLA 16	01/03/2007	-166.624,00	DISPONIBLE
TOTAL:			371.823.927,00	

Ready | lw_registro_fondos | 13/Jul/2007 12:19

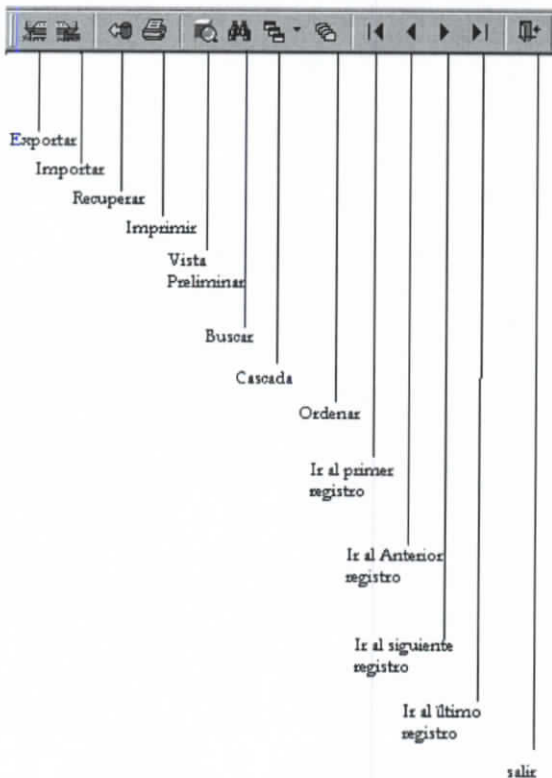
4.8. Diseño de salidas

Al hablar del diseño de salida, estamos refiriéndonos a las pantallas de Reportes, información que se imprimen o que se guardan en archivos magnéticos, estas pantallas también son estándares, es así que mantienen una misma barra de herramientas y los menús se ayudan también de la combinación de teclas rápidas, las que facilitan el manejo del menú una vez que el usuario se familiarice con el sistema.

Una de las ventajas de este sistema es que puede exportar datos a otras aplicaciones tal como Excel, Word, Dbase III, archivos TXT. Con lo cual facilita que los datos puedan ser manipulados en otros tipos de formatos.

Las opciones de la barra de herramientas y las pantallas de salidas las podemos ver en los siguientes gráficos:

4.9. Barra de Herramientas de Pantallas de salida



4.10. Pantallas de Salida

Reporte de Fondos del Fdieicomiso

Reporte General de Fondos

Recuperar Por: I Criterio: II Criterio: III Criterio: Ordenar Por: I: II: III: Aso Des Vista Previa Reglas 1 Regs

EMPRESA ELECTRICA AMBATO RCN. Dirección: Av. 12 de Noviembre y Espejo Telfs: 927111 Faxi: 422091 Ambato - Ecuador

REPORTE GENERAL DE FONDOS Pág 1 de 1 Fecha: 13-Jul-2007 12:22

Nº	ID. DOC	FECHA	TOTAL	Estatus
1	2	FEBRERO DEL 2007	371.823.927,00	DISPONIBLE
RUBRO		F. DEPOSITO	VALOR	
CONSUMO ENERGIA		01/03/2007	275598483,00	
ALUMBARDO PUBLICO		01/03/2007	37859521,00	
BOMBEROS		01/03/2007	20543901,00	
FERUM		01/03/2007	11121115,00	
SEGURO CONTRA INCENDIOS		01/03/2007	136100,00	
TASA BASURA		01/03/2007	16642238,00	
TASAS Y TERCEROS		01/03/2007	10830969,00	
RETENCION		01/03/2007	-514083,00	
RETENCION BANCOS		01/03/2007	-227693,00	
VENATANILLA 16		01/03/2007	-166624,00	
SUMAN:			371.823.927,00	
TOTAL:			371.823.927,00	

Reporte de Fondos Disponible

Reporte General de Disponible

Recuperar Por: I Criterio: II Criterio: III Criterio: Ordenar Por: I: II: III: Aso Des Vista Previa Reglas 1 Regs

EMPRESA ELECTRICA AMBATO RCN.

REPORTE GENERAL DISPONIBLE Pág 1 de 1 Fecha: 13-Jul-2007 12:23

Nº	ID. DOC	FECHA	DISP. ANTERIOR	TOTAL	SALDO
1	1	13/07/2007	0	-126649,00	-126649,00
TIPO CARDO					VALOR
INTERESES					85390,00
COMISIONES					-212038,00
SUMAN:					-126.649,00
TOTAL:					-126649

Reporte de la Distribución del VAD

Reporte General de VAD

Recuperar Por: I Criterio: II Criterio: III Criterio: Ordenar Por: I: II: III: Aso Des Vista Previa Reglas 1 Regs

EMPRESA ELECTRICA AMBATO RCN. DIRECCIÓN : Av. 12 de Noviembre y Espejo Telfs: 827111 Fax: 422091 Ambato - Ecuador

REPORTE GENERAL VAD Pág 1 de 1 Fecha: 13-Jul-2007 12:22

Nº	ID. DOC	FECHA	VALORES 100%	VALORES Otros%	TOTAL
1	1 ENERO DEL 2007	13/07/2007	58365923,00	154879599,78	213.245.522,78
RUBRO			VALOR RUBRO	%	VALOR
CONSUMO ENERGIA			275598483,00	49,41	136.173210,45
ALUMBARDO PUBLICO			37859521,00	49,41	18706389,33
			SubTotal:		154.879.599,78
TASAS Y TERCEROS			10830969,00	100,00	10830969,00
RETENCION			-514083,00	100,00	-514083,00
TASA BASURA			16642238,00	100,00	16642238,00
SEGURO CONTRA INCENDIOS			136100,00	100,00	136100,00
FERUM			11121115,00	100,00	11121115,00
RETENCION BANCOS			-227693,00	100,00	-227693,00
VENATANILLA 16			-166624,00	100,00	-166624,00
BOMBEROS			20543901,00	100,00	20543901,00
			SubTotal:		58.365.923,00
			SUMAN:		213.245.522,78

Reporte

Reporte General de Facturas

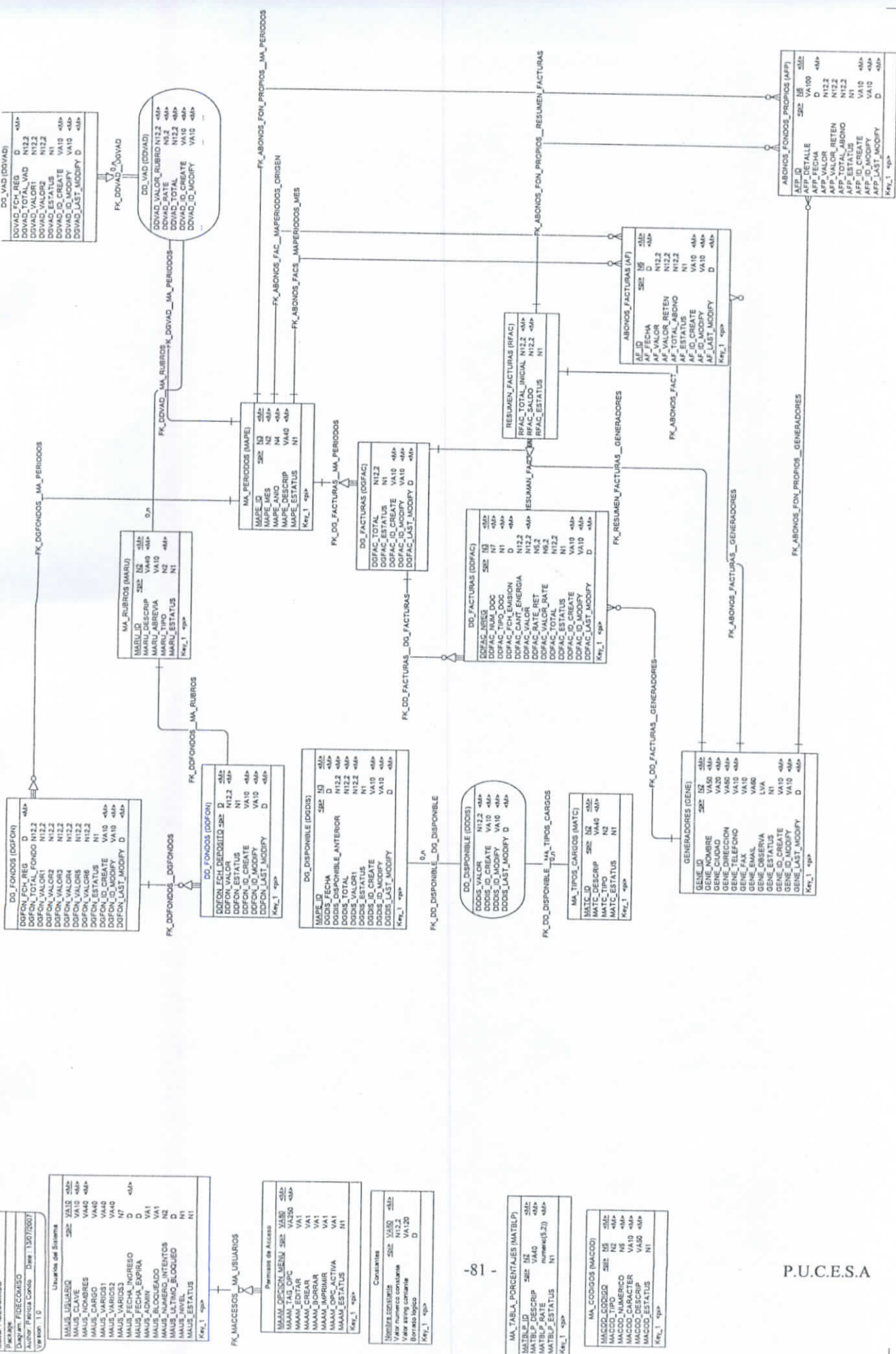
Recuperar Por: I Criterio: II Criterio: III Criterio: Ordenar Por: I: II: III: Aso Des Vista Previa Reglas 1 Regs

EMPRESA ELECTRICA AMBATO RCN. DIRECCIÓN : Av. 12 de Noviembre y Espejo Telfs: 827111 Fax: 422091 Ambato - Ecuador

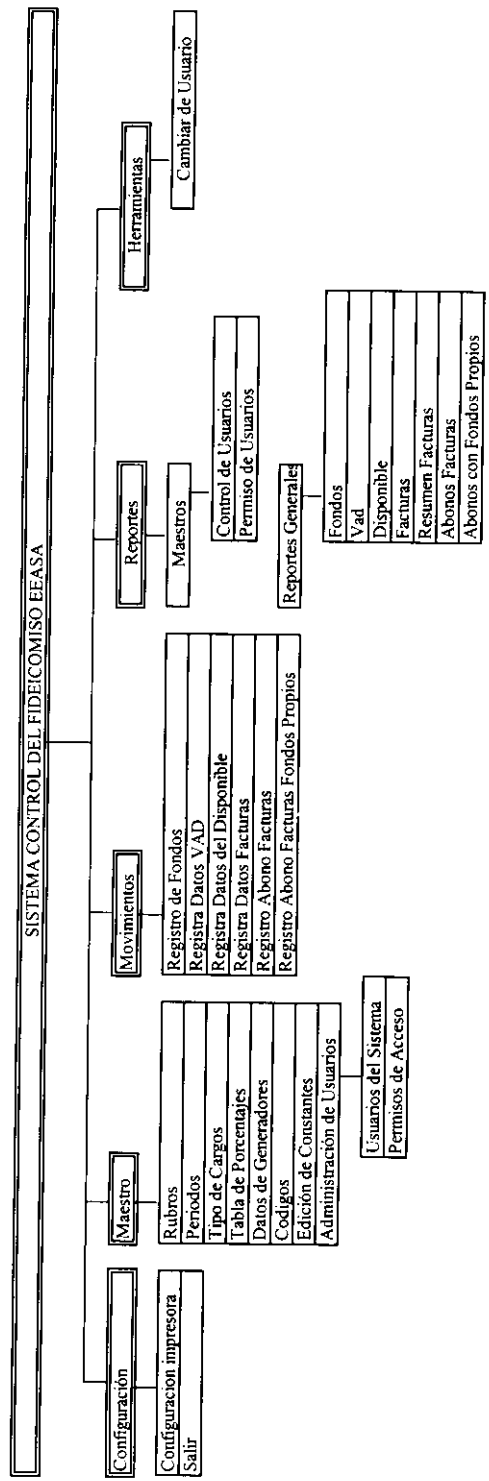
REPORTE GENERAL FACTURAS Pág 1 de 1 Fecha: 13-Jul-2007 12:23

Nº	ID. DOC	TOTAL							
1	2	24.435.635,40							
Reg.	GENERADOR	TIPO DOC.	Nº DOC.	FCH. EMI	CANT. ENER.	VALOR	%	V.RETENCION	TOTAL
1	ELECAUSTRO	FACTURA	5452	31/01/2007	15.500.000,00	956.403,00	1,00	9.564,03	946.838,97
2	ELECTROGUAYAS	FACTURA	6942	31/01/2007	15.412.000,00	11.961.569,00	1,00	119.615,69	11.841.953,31
3	EMAAP-Q	FACTURA	6471	31/01/2007	45.121.200,00	63.890,00	1,00	638,90	63.251,10
4	HIDROAGOYAN	FACTURA	9111	31/01/2007	448.745.400,00	1.218.000,00	1,00	12.180,00	1.205.820,00
5	HIDRONACION	FACTURA	4068	31/01/2007	3.961,00	3.921,00	1,00	39,21	3.881,79
6	HIDROPALUTE	FACTURA	1033	31/01/2007	4.206.164,00	4.206.164,00	1,00	42.061,64	4.164.102,36
7	TERMOESMERALDAS	FACTURA	5361	31/01/2007	547.000.000,00	1.850.300,00	1,00	18.503,00	1.831.797,00
8	TERMOPICHINCHA	FACTURA	6305	31/01/2007	48.172.500,00	941.725,00	1,00	9.417,25	932.307,75
9	ELECTROQUIL	FACTURA	4519	31/01/2007	948.048.800,00	3.480.488,00	1,00	34.804,88	3.445.683,12
									SUMAN: 24.435.635,40
TOTAL:									24.435.635,40

Modelo Entidad - Relación

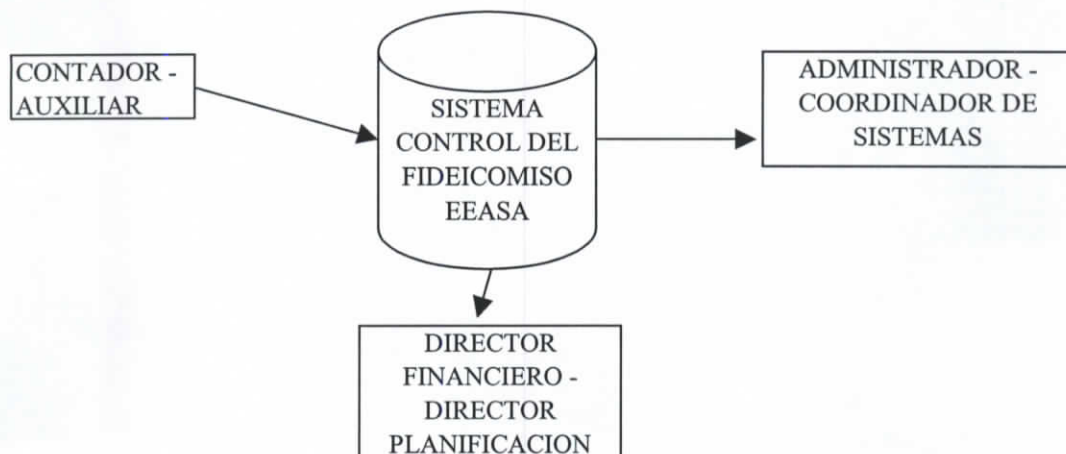


4.12. Diagrama del Sistema



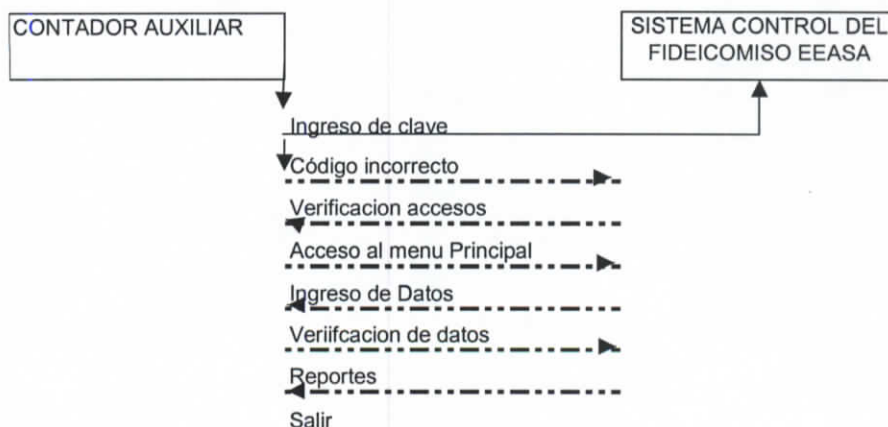
4.13. Diagrama de Iteración

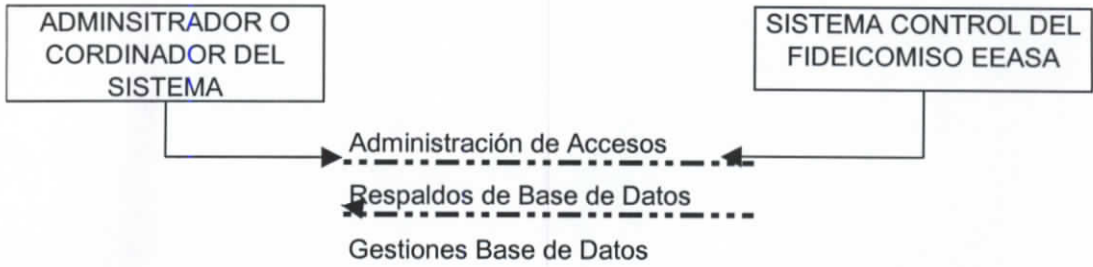
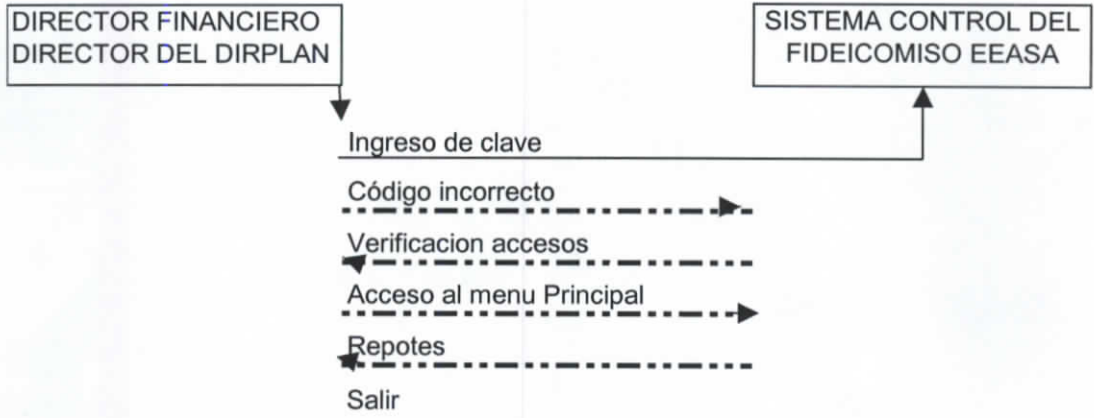
Se presenta en el diagrama como está conectado la Base de datos con cada uno de los actores: Director Financiero, Director del Departamento de Planificación, Contador Auxiliar, Administrador.



4.14. Diagrama de Secuencia

Presenta la interacción entre los diferentes objetos en el Sistema; los diferentes objetos del diagrama se Secuencia interactúan entre ellos a través del paso de mensajes:





4.15. Diccionario de Datos

TABLA ABONOS FACTURAS

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
AF_ID	numeric(6)	Clave Principal
MAPE_ID_MES	numeric(3)	Mes de compra de energía
GENE_ID	numeric(2)	Identificador del Generador
MAPE_ID_ORIGEN	numeric(3)	Identificador de Fondos para pagos
MAPE_ID_RFACTURAS	numeric(3)	Numero de la factura de compra
GENE_ID_RFACTURAS	numeric(2)	Identificador de factura a recibir el pago
AF_FECHA	date	Fecha del Abono
AF_VALOR	numeric(12,2)	Valor del Abono
AF_VALOR_RETEN	numeric(12,2)	Retención en abono
AF_TOTAL_ABONO	numeric(12,2)	Total de Abono
AF_ESTATUS	numeric(1)	Estado
AF_ID_CREATE	varchar(10)	Identificador de quien ingresa el abono
AF_ID_MODIFY	varchar(10)	Identificador de quien modifica el abono
AF_LAST_MODIFY	date	Fecha de ultima modificación

TABLA ABONOS FONDOS PROPIOS

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
AFP_ID	numeric(6)	Clave Principal
MAPE_ID_MES	numeric(3)	
GENE_ID	numeric(2)	Identificador del Generador
MAPE_ID_RFACTURAS	numeric(3)	Numero de la factura de compra
GENE_ID_RFACTURAS	numeric(2)	Identificador de factura a recibir el pago
AFP_DETALLE	varchar(100)	Detalle del abono con Fondos Propios
AFP_FECHA	date	Fecha del Abono
AFP_VALOR	numeric(12,2)	Valor del abono
AFP_VALOR_RETEN	numeric(12,2)	Retención en abono con fondos propios
AFP_TOTAL_ABONO	numeric(12,2)	Total de Abono
AFP_ESTATUS	numeric(1)	Estado
AFP_ID_CREATE	varchar(10)	Identificador de quien ingresa el abono
AFP_ID_MODIFY	varchar(10)	Identificador de quien modifica el abono
AFP_LAST_MODIFY	date	Fecha de ultima modificación

TABLA DD DISPONIBLE

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
MAPE_ID	numeric(3)	Clave periodo
MATC_ID	numeric(2)	Clave tipo de cargos
DDDIS_VALOR	numeric(12,2)	Valor del detalle del valor disponible para pago
DDDIS_ID_CREATE	varchar(10)	Identificador del usuario que procesa el valor disponible
DDDIS_ID_MODIFY	varchar(10)	Identificador del usuario que modifica
DDDIS_LAST_MODIFY	date	Fecha de ultima modificación

TABLA DD FACTURAS

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
MAPE_ID	numeric(3)	Clave periodo
DDFAC_NREG	numeric(3)	Clave Factura
GENE_ID	numeric(2)	Identificador del generador
DDFAC_NUM_DOC	numeric(7)	Numero de factura
DDFAC_TIPO_DOC	numeric(1)	Tipo de Documento
DDFAC_FCH_EMISION	date	Fecha de emisión
DDFAC_CANT_ENERGI	numeric(12,2)	Cantidad de energía en KWH
DDFAC_VALOR	numeric(12,2)	Valor de factura
DDFAC_RATE_RET	numeric(5,2)	Valor de retención
DDFAC_VALOR_RATE	numeric(9,2)	Porcentaje de retención
DDFAC_TOTAL	numeric(12,2)	Total de factura
DDFAC_ESTATUS	numeric(1)	Estado
DDFAC_ID_CREATE	varchar(10)	Identificador del usuario que procesa el valor disponible
DDFAC_ID_MODIFY	varchar(10)	Identificador del usuario que modifica
DDFAC_LAST_MODIFY	date	Fecha de ultima modificación

TABLA DD FONDOS

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
MAPE_ID	numeric(3)	Clave periodo
MARU_ID	numeric(2)	Clave de Rubro
DDFON_FCH_DEPOSITO	date	Clave principal del detalle de fondos
DDFON_VALOR	numeric(12,2)	Valor del fondo
DDFON_ESTATUS	numeric(1)	Estado
DDFON_ID_CREATE	varchar(10)	Identificador del usuario que procesa el valor disponible
DDFON_ID_MODIFY	varchar(10)	Identificador del usuario que modifica
DDFON_LAST_MODIFY	date	Fecha de ultima modificación

TABLA DD VAD

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
MAPE_ID	numeric(3)	Clave periodo
MARU_ID	numeric(2)	Clave de Rubro
DDVAD_VALOR_RUBRO	numeric(12,2)	Rubros asignados para el calculo del VAD
DDVAD_RATE	numeric(5,2)	Porcentaje de asignacion del Vad
DDVAD_TOTAL	numeric(12,2)	Total Vad a recibir
DDVAD_ID_CREATE	varchar(10)	Identificador del usuario que procesa el valor disponible
DDVAD_ID_MODIFY	varchar(10)	Identificador del usuario que modifica
DDVAD_LAST_MODIFY	date	Fecha de ultima modificación

TABLA DG DISPONIBLE

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
MAPE_ID	numeric(3)	Clave periodo
DGDIS_FECHA	date	Fecha de los mfondos disponibles
DGDIS_DISPONIBLE_ANTERIOR	numeric(12,2)	Valor del disponible anterior
DGDIS_TOTAL	numeric(12,2)	Total disponible
DGDIS_VALOR1	numeric(12,2)	Valor que recibe los fondos disponibles para el pago
DGDIS_ESTATUS	numeric(1)	Estado del disponible
DGDIS_ID_CREATE	varchar(10)	Identificador del usuario que procesa
DGDIS_ID_MODIFY	varchar(10)	Identificador del usuario que modifica
DGDIS_LAST_MODIFY	date	Fecha de ultima modificación

TABLA DG FACTURAS

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
MAPE_ID	numeric(3)	Clave principal
DGFAC_TOTAL	numeric(12,2)	Total de factura
DGFAC_ESTATUS	numeric(1)	Estado
DGFAC_ID_CREATE	varchar(10)	Identificador del usuario que procesa el valor disponible
DGFAC_ID_MODIFY	varchar(10)	Identificador del usuario que modifica
DGFAC_LAST_MODIFY	date	Fecha de ultima modificación

TABLA DG FONDOS

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
MAPE_ID	numeric(3)	Clave periodo
DGFON_FCH_REG	date	Fecha del fondo
DGFON_TOTAL_FON DO	numeric(12,2)	Valor total del Fondo
DGFON_VALOR1	numeric(12,2)	Valores a utilizarse del fondo por vad
DGFON_VALOR2	numeric(12,2)	Valores a utilizarse del fondo por prealciones
DGFON_VALOR3	numeric(12,2)	Valores a utilizarse del fondo pagos
DGFON_VALOR4	numeric(12,2)	Valores a utilizarse del fondo por pagos fondos propios
DGFON_VALOR5	numeric(12,2)	Valores a utilizarse del fondo comisiones
DGFON_VALOR6	numeric(12,2)	Valores a utilizarse del fondo rendimiento
DGFON_ESTATUS	numeric(1)	Estado
DGFON_ID_CREATE	varchar(10)	Identificador del usuario que procesa el valor disponible
DGFON_ID_MODIFY	varchar(10)	Identificador del usuario que modifica
DGFON_LAST_MODIF Y	date	Fecha de ultima modificación

TABLA DG VAD

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
MAPE_ID	numeric(3)	Clave periodo
DGVAD_FCH_REG	date	Fecha de registro del vad
DGVAD_TOTAL_VAD	numeric(12,2)	Total del calculo del Vad
DGVAD_VALOR1	numeric(12,2)	Valor vad tasas
DGVAD_VALOR2	numeric(12,2)	Valor vad otros ingresos
DGVAD_ESTATUS	numeric(1)	Estado
DGVAD_ID_CREATE	varchar(10)	Identificador del usuario que procesa el valor disponible
DGVAD_ID_MODIFY	varchar(10)	Identificador del usuario que modifica
DGVAD_LAST_MODIF Y	date	Fecha de ultima modificación

TABLA GENERADORES

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
GENE_ID	numeric(2)	Clave generador
GENE_CAMPO	varchar(50)	Nombre del Generador
GENE_CIUADAD	varchar(20)	Ciudad
GENE_DIRECCION	varchar(60)	Dirección
GENE_TELEFONO	varchar(10)	Telefono
GENE_FAX	varchar(10)	Fax
GENE_EMAIL	varchar(60)	E mail
GENE_OBSERVA	long varchar	Observacion
GENE_ESTATUS	numeric(1)	Estado
GENE_ID_CREATE	varchar(10)	Identificador del usuario que procesa el valor disponible
GENE_ID_MODIFY	varchar(10)	Identificador del usuario que modifica
GENE_LAST_MODIFY	date	Fecha de ultima modificación

TABLA MA ACCESOS MENU

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
MAUS_USUARIO	varchar(10)	Identificador del Usuario
MAAM_OPCION_MENU	varchar(80)	Clave principal
MAAM_TAG_OPC	varchar(250)	Opciones de Accesos
MAAM_EDITAR	varchar(1)	Acceso de edicion
MAAM_CREAR	varchar(1)	Acceso para crear
MAAM_BORRAR	varchar(1)	Acceso para Borrar
MAAM_IMPRIMIR	varchar(1)	Acceso para imprimir
MAAM_OPC_ACTIVIA	varchar(1)	Opcion Activo
MAAM_ESTATUS	numeric(1)	Estado

TABLA MA CODIGOS

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
MACOD_CODIGO	numeric(5)	Calve principal
MACOD_TIPO	numeric(2)	Tipo de Codigo
MACOD_NUMERICO	numeric(5)	Asignacion del Codigo
MACOD_CHARACTER	varchar(10)	Nombre del codigo
MACOD_DESCRIP	varchar(50)	Descripción del codigo
MACOD_ESTATUS	numeric(1)	Estado

TABLA MA CONSTANTES

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
MACONS_NOMBRE	varchar(60)	Campo constante
MACONS_VALOR	numeric(12,2)	Valor numerco constante
MACONS_DETALLES	varchar(120)	Valor string contante
MACONS_ESTADO	date	Borrado lógico

TABLA MA PERIODOS

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
MAPE_ID	numeric(3)	Clave periodo
MAPE_MES	numeric(2)	Mes del periodo
MAPE_ANIO	numeric(4)	Años del periodo
MAPE_DESCRIP	varchar(40)	Descripción del periodo
MAPE_ESTATUS	numeric(1)	Estado del periodo

TABLA MA RUBROS

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
MARU_ID	numeric(2)	Clave de Rubro
MARU_DESCRIP	varchar(40)	Descripción del rubro
MARU_ABREVIAS	varchar(10)	Abreviación del Rubro
MARU_TIPO	numeric(2)	Tipo del rubro
MARU_ESTATUS	numeric(1)	Estado

TABLA MA TABLA PORCENTAJES

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
MATBLP_ID	numeric(2)	Clave principal
MATBLP_DESCRIP	varchar(40)	Descripción del porcentaje
MATBLP_RATE	numeric(5,2)	Detalle del porcentaje
MATBLP_ESTATUS	numeric(1)	Estado

TABLA MA TIPOS CARGOS

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
MATC_ID	numeric(2)	Clave tipo de cargos
MATC_DESCRIP	varchar(40)	Descripción del Tipo de Cargo
MATC_TIPO	numeric(2)	Tipo de cargo
MATC_ESTATUS	numeric(1)	Estado del cargo

TABLA MA USUARIOS

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
MAUS_USUARIO	varchar(10)	Identificador del Usuario
MAUS_CLAVE	varchar(10)	Clave de usuario
MAUS_CAMPOS	varchar(40)	Nombre del usuario
MAUS_CARGO	varchar(40)	Cargo del Usuario
MAUS_VARIOS1	varchar(40)	Campo varios datos
MAUS_VARIOS2	varchar(40)	Campo varios datos
MAUS_VARIOS3	numeric(7)	Campo varios datos
MAUS_FECHA_INGRESO	date	Fecha de ingreso
MAUS_FECHA_EXPIRACION	date	Fecha de Expiración del Usuario

MAUS_ADMIN	varchar(1)	Columna para Identificar si se trata de un Usuario con perfil de administrador
MAUS_BLOQUEADO	varchar(1)	Control de Bloqueo
MAUS_NUMERO_INTENTOS	numeric(2)	Control de Numero de intento para acceso
MAUS_ULTIMO_BLOQUEO	date	Fecha de ultimo bloqueo
MAUS_NIVEL	numeric(1)	Nivel del usuario
MAUS_ESTATUS	numeric(1)	Estado del Usuario

TABLA RESUMEN FACTURAS

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>
MAPE_ID	numeric(3)	Clave periodo
GENE_ID	numeric(2)	Clave generador
RFAC_TOTAL_INICIAL	numeric(12,2)	Valor inicial del la factura
RFAC_SALDO	numeric(12,2)	Saldo de factura
RFAC_ESTATUS	numeric(1)	Estado

CAPITULO V

IMPLEMENTACION

5.1. Estándares de codificación

Los programas que forman el sistema fueron codificados de manera que la programación no sea confusa, facilitando la comprensión y utilización de las diferentes estructuras que provee Power Builder como son: *for*, *while* además la utilización del *if* y muchas otras sentencias que nos proporciona esta herramienta de programación.

En gran parte de los programas se ha documentado el código, especialmente al iniciar una rutina o subrutina ó en procesos complejo para guiar a los programadores sobre la lógica del mismo.

En el desarrollo del programa se ha utilizado algunas funciones de mucha utilidad cumpliendo los objetivos para los cuales fueron creadas, entre ellas podemos mencionar a las siguientes: *f_datos_fondos*, *f_existe_disponible*, *F_existe_vad*

Las siguientes instrucciones forman parte del código de la función para validación de los datos de los Fondos Mensuales del Fideicomiso.

f datos fondos:

```
// Función para validación de datos

String ls_Ret
Long ll_periodo
Date ld_fch_reg

SELECT a.mape_id, a.dgfon_fch_reg
INTO :ll_periodo,
      :ld_fch_reg
FROM dg_fondos a
WHERE a.mape_id = :al_periodo ;
CHOOSE CASE sqlca.sqlcode
CASE 0
CASE 100
      RETURN (")
```

CASE -1

```
gs_mensaje= ' DG_FONDOS '  
f_mensajes(4)  
RETURN ("
```

END CHOOSE

```
IF f_IsEmpty_n(ll_periodo) THEN ll_periodo = 0  
IF f_IsEmpty_d(ld_fch_reg) THEN ld_fch_reg = Date(0,0,0)
```

```
ls_Ret= String(ll_periodo)+'-'+String(ld_fch_reg)  
RETURN(ls_Ret)
```

El siguiente código es de la función para validar si existe fondos disponible para el proceso del Fideicomiso:

f existe disponible

Long ll_Cuenta

```
SELECT COUNT(*)  
INTO :ll_Cuenta  
FROM dg_disponible a  
WHERE a.mape_ID = :al_periodo ;
```

```
CHOOSE CASE sqlca.sqlcode  
CASE 0  
    IF IsNull(ll_Cuenta) THEN  
        ll_Cuenta = 0  
    END IF  
CASE 100  
    ll_Cuenta = 0  
CASE -1  
    ll_Cuenta = -1  
END CHOOSE
```

Return(ll_Cuenta)

F existe vad

Long ll_Cuenta

```
SELECT COUNT(*)  
INTO :ll_Cuenta  
FROM dg_vad a  
WHERE a.mape_ID = :al_periodo ;
```

```
CHOOSE CASE sqlca.sqlcode  
CASE 0
```

```
IF IsNull(Il_Cuenta) THEN
    Il_Cuenta = 0
END IF
CASE 100
    Il_Cuenta = 0
CASE -1
    Il_Cuenta = -1
END CHOOSE

Return(Il_Cuenta)
```

En el sistema también constan las funciones para el manejo- de registros; añadir, eliminar, buscar, filtrar registros, actualizar base de datos

5.2. Convenciones sobre los nombres de Objetos y Controles

Existen algunas convenciones para nombres de objetos y controles y por eso tomamos la que Power Builder sugiere, esta es una nomenclatura de nombre propio del programa, que en la práctica resulta beneficioso y útil para los usuarios de esta herramienta de programación.

Deben emplearse las convenciones que mejor se adapten a las necesidades de cada uno. Lo más importante es utilizar en el código la convención relativa a los nombres. Estas facilitan en gran medida las tareas de mantenimiento y depuración.

Las convenciones de nombres sugieren que se utilicen prefijos en los nombres para indicar el ámbito y el tipo de dato de las variables e identificadores.

Las convenciones sobre nombres que se ha adoptado es la siguiente:

Tipo de Control	Prefijo	Ejemplo
Application		FIDECO
CommandButton	cb_	Cb_aceptar
DataWindow	dw_	Dw_fondos
DropDown DataWindow	d_dddw_	D_dddw_tipo_rubro

Objeto DataWindow	d_	D_lista_generadores
Tab	tab_	Tab_valor
TabPage	tabpage_	TabPage_1 (2,3,4...n)
unction(global)	f_	f_new_reg
Function(Window)	wf_	wf_valida
Window	w_	w_tipo_fondo
Menu	m_	m_principal

Ambito de la Variable	Prefijo	Nombre	Descripción
Global	g	Gs_usuario	Variable Global tipo string
Local	l	Ll_filas	Variable Local tipo long
Argumento	a	As_tabla	Argumento tipo string

// Programa Principal

SetPointer(Hourglass!)

f_comun()

// Asignar valores

gs_logo_ingreso = "..\fideco\bmp\logo.bmp"

gs_logo_frame = "..\fideco\bmp\logo_1.bmp"

gs_archivo_ini = "FileString.ini"

Open(w_acceso)

gl_policia = f_constante_n('POLICIA')

IF gl_policia = -1 THEN

 RETURN

END IF

Idle(gl_policia)

5.3. Pruebas

5.3.1. Pruebas de integridad

Primero se ha procesado datos de prueba típicos que se maneja diariamente en el Control y Manejo de los Fondos del Fideicomiso de la Empresa Eléctrica Ambato, aquellas que cubren el mayor volumen de trabajo, si el sistema trabaja con transcripciones normales, luego se agregaron datos con variaciones los cuales se incluyeron datos inválidos para asegurarnos que el sistema puede detectar errores de manera adecuada.

También se ha verificado las salidas de datos mediante la obtención de reportes, para constatar que los datos se hayan procesado de una forma adecuada para satisfacer las necesidades del usuario. Estos reportes sirvieron para comparar las variables que se relacionaron con la Operación del Sistema.

5.3.2. Pruebas de funcionamiento

A medida que se fueron desarrollando y codificando los distintos programas y opciones que conforman el Sistema se fueron realizando pruebas a fin de comprobar el buen funcionamiento de estos y verificar si están dando los resultados esperados, estas pruebas se las realizo con la aprobación y visto bueno del Director Financiero, quien es el mejor juez que se podría tener.

Para realizar estas pruebas utilizamos datos reales que nos fueron facilitados por personal del Departamento Financiero, gracias a los cuales se pudo apreciar como funcionará el Sistema en el momento de la explotación misma.

Al momento que se han ingresado datos reales que maneja el Departamento Financiero de la Empresa Eléctrica, se ha podido comprobar las validaciones con las que cuenta el sistema por ejemplo en el ingreso de fechas, y campos principales como son los rubros de recaudación , estos datos han sido correctamente procesados

y de esta, forma se han obtenido los resultados esperados mediante la emisión de reportes.

5.4. Implementación

La Implantación del sistema realiza de una manera progresiva:

- Se instalará el nuevo sistema y tendrá un funcionamiento paralelo con el sistema antiguo, esto significa que los datos se ingresarán en los dos sistemas y se validarán los datos obtenidos.
- Además la instalación también se la realizará en los computadores de Planificación, para que ellos puedan realizar las consultas necesarias.
- Identificar las dificultades especiales que se hubieran encontrado durante la utilización del sistema actual.
- Verificar hasta que punto se han cumplido las bases de referencia
- Examinar el uso que se hace de los datos de salida, asegurándose si se han obtenido los resultados óptimos al mejorar el servicio y la información.
- Para que los usuarios trabajen con el nuevo sistema es necesario capacitarles por lo cual se considera que a los usuarios se les instruirá por un tiempo de dieciséis horas.
- Esta fase es muy importante para la explotación del sistema, terminando solo, cuando entendamos que los usuarios estén en condiciones de obtener el máximo rendimiento de software.

5.5. Beneficios que se obtienen del Sistema del Fideicomiso

Disponibilidad de información rápida y confiable, Una de las ventajas de trabajar con una Base de Datos como Sybase Sql Anywhere es la seguridad que esta mantiene sobre su información, pues permite tener integridades de información asegurando que la información sea confiable. Con el desarrollo de aplicaciones Cliente-Servidor se mejora el rendimiento de las aplicaciones pues el trabajo ya no esta centralizado

en el servidor de base de datos sino que es distribuido a los usuarios, quienes procesan sus propios datos para finalmente ser enviados al servidor para su almacenamiento.

Además de lo antes mencionado las aplicaciones Cliente-Servidor mejoran el rendimiento de la red pues el tráfico de información disminuye notablemente.

Mejorar la integración organizacional, al disponer de una aplicación que permita mantener organizada la información, esta podrá estar a disposición de toda la Empresa lo que permitirá integrar los distintos departamentos y secciones.

Además la información que genera esta aplicación puede servir de entrada para otros Sistemas Informáticos que mantiene la empresa en los diferentes Departamentos.

Toma de Decisiones, una de las finalidades al desarrollar esta aplicación fue dotar de una herramienta que sea un soporte al momento de la toma de decisiones especialmente en los niveles superiores, a quienes les interesa principalmente los resultados finales de una gestión o trabajo, y es aquí en donde los diferentes reportes cumplen con su objetivo.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Todos los objetivos propuestos para la elaboración de este trabajo han sido cumplidos.

- Al finalizar este trabajo podemos afirmar con certeza que el motor de base de datos SQL Anywhere es una herramienta que sin tener costos muy altos puede ser la solución para pequeñas, medianas y grandes empresas.

- Una vez que se desarrolla una aplicación del cliente y funciona en la configuración de SQL Standalone en monousuario, no se requiere ningún cambio en la aplicación para que trabaje como una aplicación del cliente en un ambiente de la red con un servidor SQL Anywhere.

- La Empresa no está utilizando todos los recursos que posee, por falta de una Unidad de sistemas informáticos a nivel de toda la empresa, pues los que existen trabajan en forma aislada en cada uno de sus respectivos departamentos.

- El Departamento esta en capacidad de informar a las autoridades el registro de los movimientos financieros del Fideicomiso EEASA.

- Entrega reportes en base de los cuales se puede proyectar la información y obtener una visión de la liquidez financiera de la Empresa Eléctrica Ambato S.A., en forma mensual en lo que se refiere a los fondos del Fideicomiso.

RECOMENDACIONES

- ❑ Es necesario el trabajo conjunto de los profesionales en el área de sistemas de las distintas áreas de la empresa, para que los recursos existentes puedan ser explotados en su real capacidad.
- ❑ Recomendamos la creación de un área, Unidad o Departamento de Sistemas que sea el encargado de administrar y distribuir y coordinar los recursos de hardware y software a todos los departamentos y áreas de la empresa, previo un estudio detallado.
- ❑ Crear políticas de establecimiento de respaldo de la información para mejorar los datos del sistema.
- ❑ Establecer un plan periódico de encuestas a los usuarios para realizar un sondeo del grado de satisfacción del mismo y de su reacción ante el sistema

MANUAL DEL USUARIO

MANUAL DE USUARIO

INSTALACIÓN DEL PROGRAMA DEL CONTROL DE FIDEICOMISO EN LA EEASA. RCN SA.

Para instalar el programa de Control de Fideicomiso, el usuario deberá seguir los siguientes pasos:

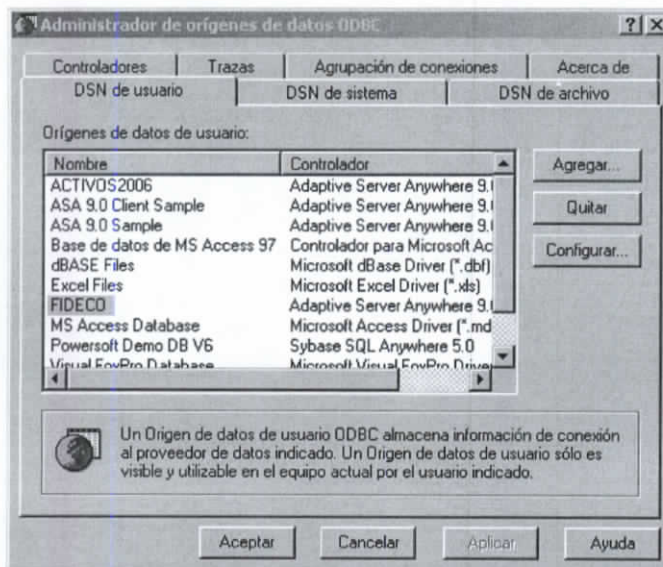
- 1.- Insertar el Cd en la unidad de CD_ROM.
- 2.- Copiar la carpeta FIDECO al disco duro desde el CD.
- 3.- Obtener un acceso directo desde el archivo **FIDECO.EXE** en el escritorio de Windows
- 4.- Crear el ODBC (ver apartado CREACIÓN DE ODBC PARA EL SISTEMA CONTROL DE FIDEICOMISO)

Al terminar estos cuatro pasos nos encontramos listos para ingresar al sistema de Control de Fideicomiso.

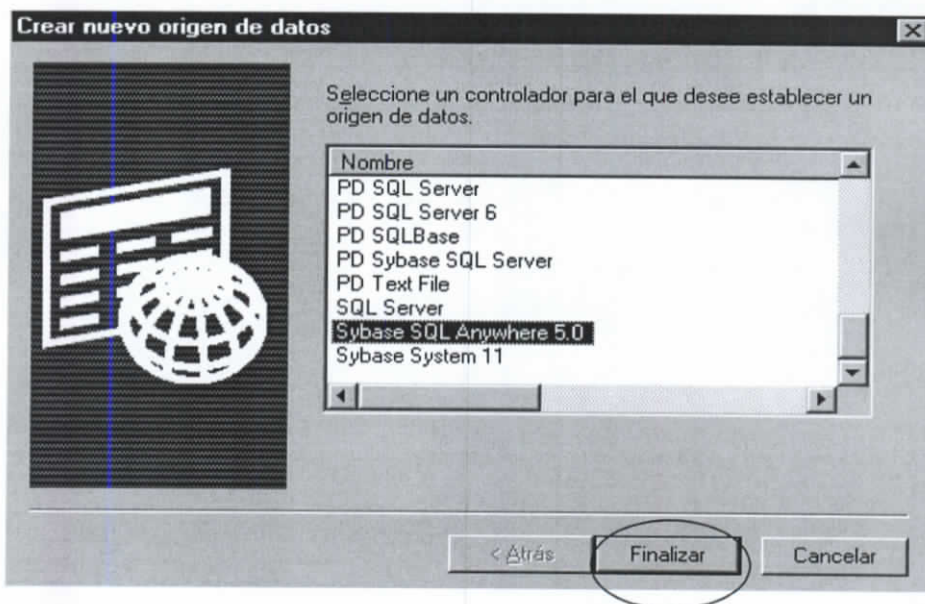
CREACIÓN DE ODBC PARA EL SISTEMA DE CONTROL DE FIDEICOMISO

Como parte de la configuración del sistema el usuario deberá crear un ODBC para la conexión a la Base de datos Sysbase Sql Anywhere, por lo tanto es necesario indicar los siguientes pasos para obtener este ODBC.

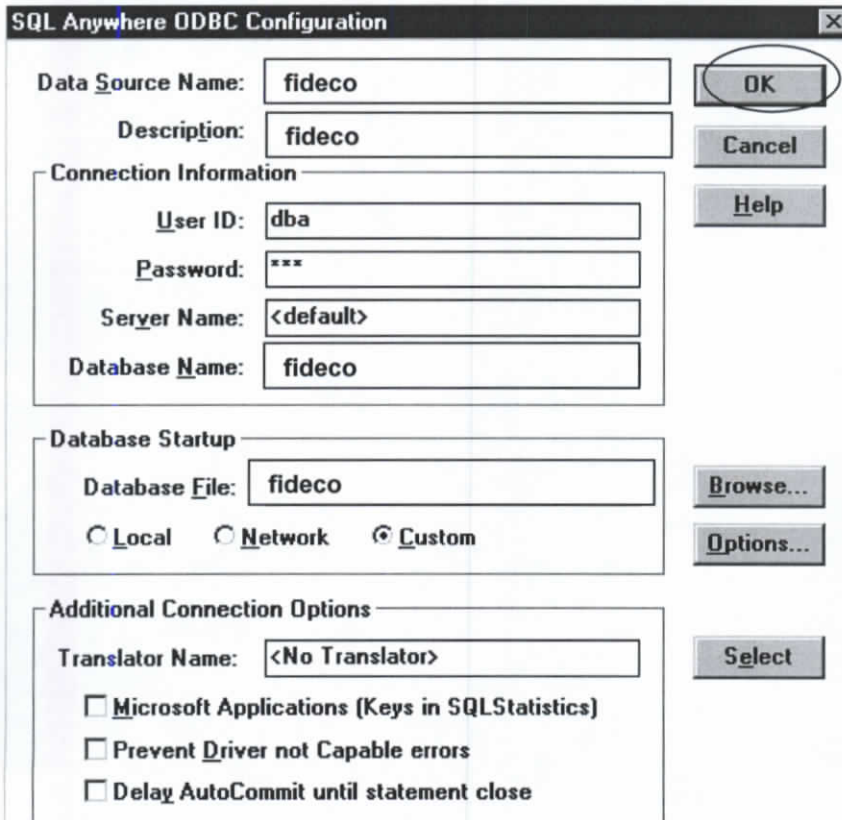
Desde el escritorio, entrar por Mi Pc, Panel de Control, Fuentes de datos ODBC, el sistema mostrará una pantalla como la siguiente:



Seleccionar Agregar e inmediatamente aparece la siguiente pantalla, en la cual deberá el usuario buscar con la barra deslizante de la derecha la opción Sybase SQL Anywhere .



Una vez seleccionado hacer click en el botón Finalizar entonces el sistema presenta la pantalla para configurar el ODBC, en ella se deben ingresar los datos siguientes:



Luego de configurar la pantalla anterior hacer click en OK, y luego en Aceptar, con lo que la configuración del ODBC, quedará listo.

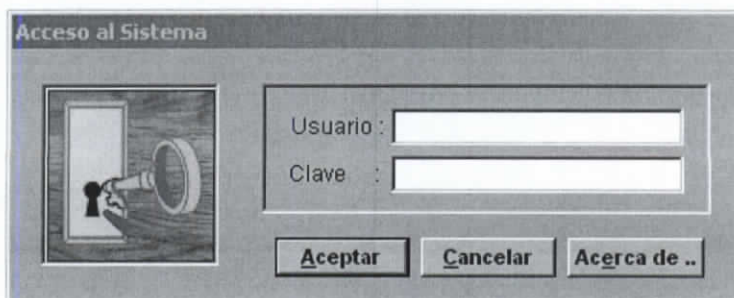
COMO INGRESAR AL SISTEMA

En usuario encontrándose en el escritorio de Windows deberá realizar lo siguiente

- Doble Click en el Icono de acceso directo a FIDEICOMISO

Después de lo cual aparecerá un cuadro de diálogo como el que se muestra en la pantalla siguiente:

Pantalla para el ingreso de usuarios al sistema



El sistema cuenta con una pantalla de ingreso que restringe el acceso de usuarios al sistema, para lo cual se solicita ingresar el nombre del usuario y la contraseña; las mismas que son proporcionadas por el administrador del sistema que es la persona autorizada para crear usuarios.

Los botones del cuadro de dialogo de ingreso realiza lo siguiente:

ACEPTAR.- Permite el ingreso al sistema

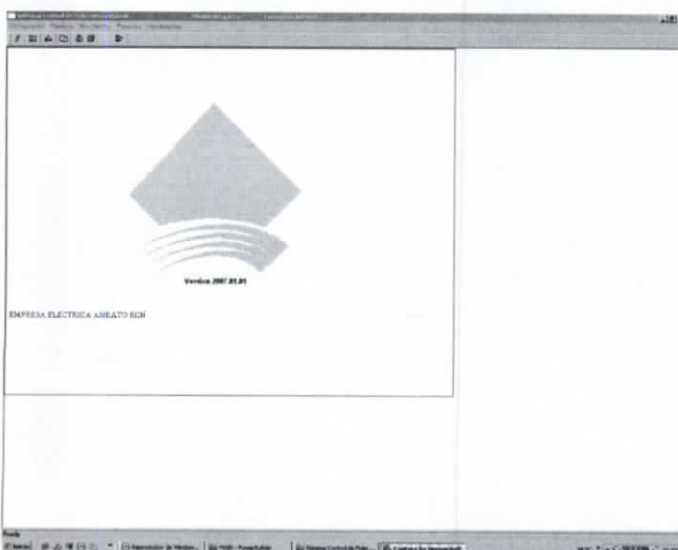
CANCELAR.- Termina la aplicación

ACERCA DE .- Muestra una pantalla sobre información general del sistema.

Después de tres fallidos al intentar ingresar, el sistema termina su ejecución automáticamente.

EJECUCIÓN DEL PROGRAMA

Una vez ingresada la clave correcta con su respectivo password se despliega la pantalla principal del sistema Figura siguiente



Para manejar todas las opciones se encuentra un menú que consta de seis botones que permiten acceder a submenús como :

CONFIGURACION.- En este submenú tenemos dos opciones: Configura Impresoras y Salir.

MAESTROS.- En este menú tenemos el manejo de datos de todas las tablas que intervienen en el sistema, existe una opción para cada una de ellas así,

RUBROS

PERIODOS

TIPOS DE CARGOS

TABLA DE PORCENTAJE

DATOS GENERADORES

MOVIMIENTOS.- Al seleccionar esta opción se despliega opciones para el ingreso y procesamiento de datos:

REGISTRO DE FONDOS

REGITRO DATOS DE VAD

REGISTRO DE DATOS DEL DISPONIBLE

REGISTRO DE FACTURAS

REGISTRO DE ABONOS

REGISTRO DE PAGOS FONDOS PROPIOS

REPORTES.- Esta es la opción del Menú que llama a los reportes habilitados en este sistema.

HERRAMIENTAS.- Proporciona al usuario la posibilidad de cambiar a otro usuario

DESCRIPCIÓN DE LOS ICONOS DE LA BARRA DE MENU



Como podemos observar inicialmente los Iconos de Grabar y Cancelar se encuentran inhabilitados.

Exportar Registros.- Para guardar los registros que se encuentran en la pantalla se utiliza esta opción, con la cual se puede grabar en un archivo de diferente formato.



Importar Registros.- Con esta opción podemos importar registros de archivos con diferente formato.



Editar.- Este Icono permite realizar correcciones en los registros que ya se encuentran ingresados, para lo cual el puntero deberá encontrarse en el registro que se desea cambiar y luego hacer click en el Icono, el programa dejará habilitado solamente los iconos de Grabar, Cancelar y Salir.



Nuevo: Inmediatamente se inserta en la parte final de la pantalla de ingreso un registro en blanco y en la columna de código se inserta el numero secuencial correspondiente a este código y en la columna de descripción, se debe digitar la descripción correspondiente, cuando se selecciona esta opción en el menú se encuentran habilitados solamente los Iconos de Grabar, Cancelar y Salir.



Grabar.- Al hacer click en este icono se graba el registro último ingresado.



Cancelar.- Si se hace click en este icono todos los cambios e ingresos realizados en los datos se cancelan y vuelven a lo anteriormente existente.



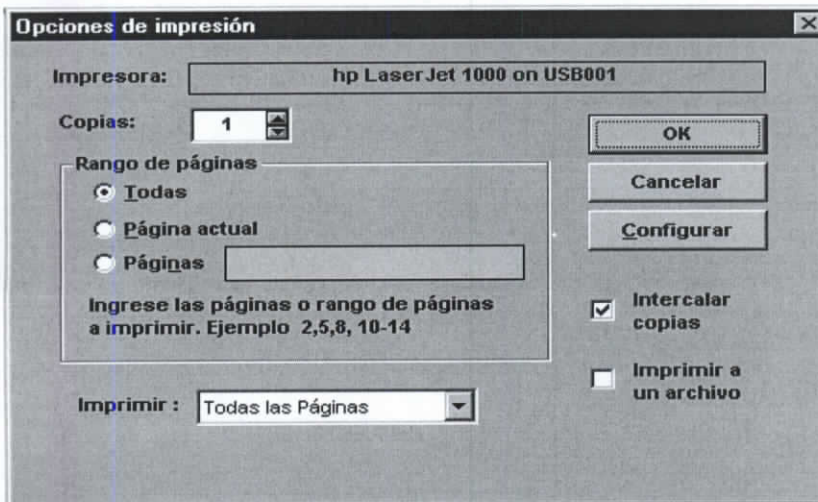
Borrar.- Esta opción borra el registro en el que se encuentra actualmente el cursor.



Buscar.- Con la selección de esta opción se puede localizar inmediatamente un registro que se busca, y el cursor se ubica en el registro encontrado.



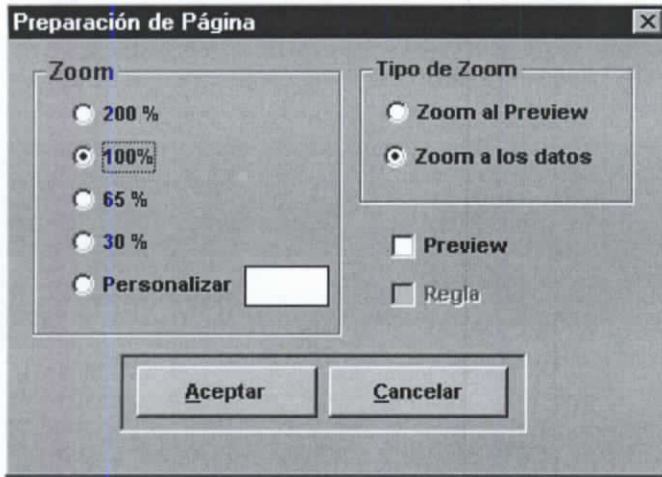
Imprimir.- Al hacer click en este icono aparece la siguiente pantalla:



Aquí podemos seleccionar y configurar algunas opciones de la impresora antes de aceptar la opción de imprimir.

Mirar.- Podemos visualizar la pantalla de acuerdo a los siguientes parámetros:

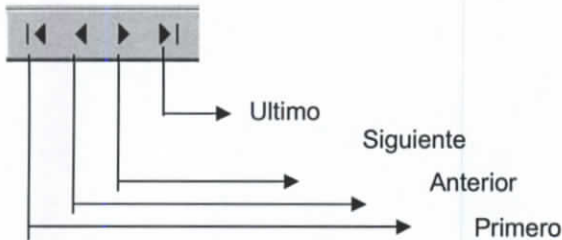




Ventana.- Para cambiar el estilo de pantalla tenemos este Icono.



Botones para posiciones de registros.- Los siguientes son los botones que nos ayudan a cambiarse de registro a registro, posiciona el cursor en el registro de acuerdo al gráfico siguiente:

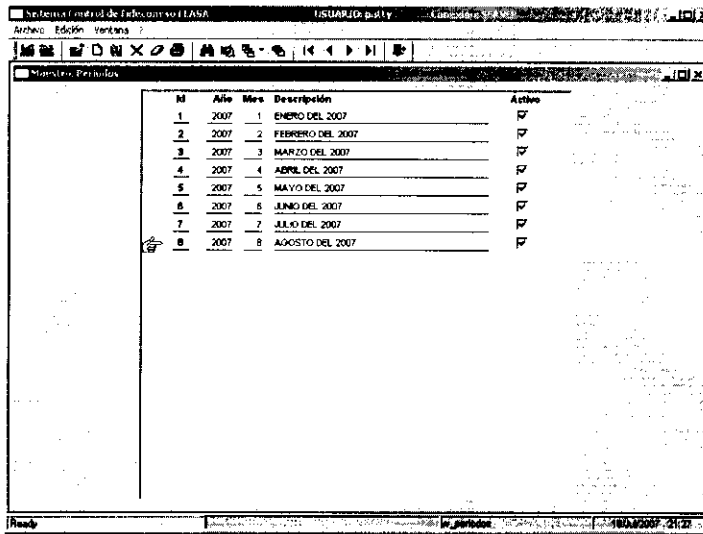


Salir.- Para salir de un determinado proceso utilizamos esta opción de Salir, con lo cual regresamos a la pantalla principal.



MAESTROS: PERIODOS

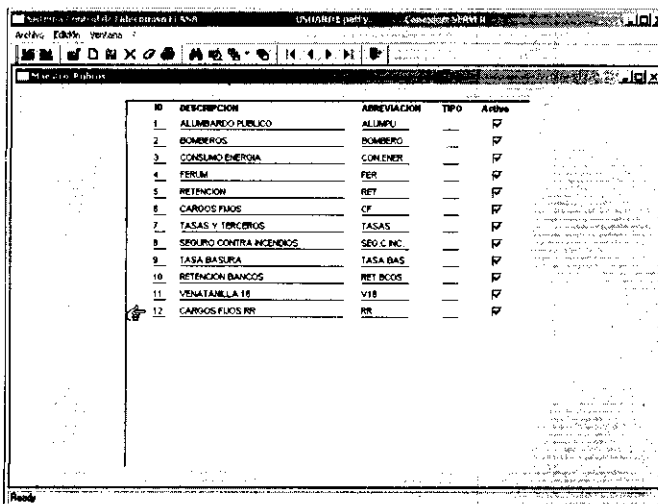
El usuario deberá seleccionar desde el menú principal la opción Maestros, seguido de la opción Maestro de Períodos, e inmediatamente aparecerá la pantalla siguiente:



Mediante esta pantalla el usuario puede ingresar datos de Año y mes, para ello en la parte superior se encuentra la barra de opciones del menú que controla esta pantalla, se puede añadir, borrar, imprimir, buscar registros de periodos, para seleccionar cualquiera de estas opciones con el puntero del mouse debe ubicarse en el icono deseado y hacer click.

MAESTROS: RUBROS

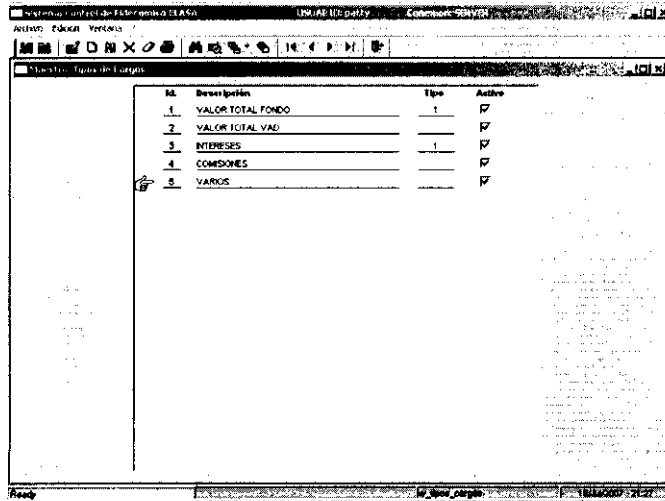
El usuario deberá seleccionar desde el menú principal la opción Maestros, seguido de la opción Maestro de Rubros, e inmediatamente aparecerá la pantalla siguiente:



Esta pantalla permite ingresar, modificar, borrar y buscar un tipo de rubro, utilizando las diferentes opciones de la barra de menú, que ya se describió anteriormente.

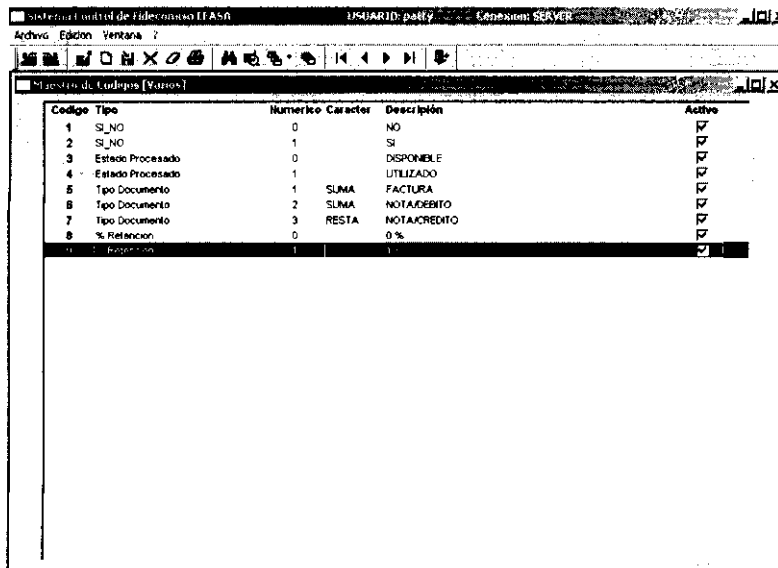
MAESTRO: TIPO DE CARGOS

El usuario deberá seleccionar desde el menú principal la opción Maestros, seguido de la opción Maestro de Tipos de Cargos, e inmediatamente aparecerá la pantalla siguiente:



MAESTRO: DE CODIGOS

El usuario deberá seleccionar desde el menú principal la opción Maestros, seguido de la opción Maestro de Códigos, e inmediatamente aparecerá la pantalla siguiente:



MAESTRO: Usuarios

El usuario deberá seleccionar desde el menú principal la opción Maestros, seguido de la opción Maestro de Usuarios, e inmediatamente aparecerá la pantalla siguiente:

Usuario	Nombres	Cargo	Admin	Bloqueado	Nivel	Estatus
PATTY	ADMIN	ADMIN	S	N	Administrador	<input checked="" type="checkbox"/>

MOVIMIENTOS

INGRESO DE DATOS DE GENERADORES

Elegir nuevo registro, ingresar datos generales de los Generadores que realizan la venta de energía a la Empresa.

NOMBRE	CIUDAD	DIRECCION	TELEFONO	FAX
1. ELECAUSTRO	Cuenca	Av. 12 de Abril y José Peralta	843 035	843 035
2. ELECTROOJAYAS	Quayquil	Km. 7 1/2 vía a la Costa	870 771	870 771
3. EMAP-O	Guilo	Av. Mariana de Jesús entre Alemania e Italo	501 225	501 225
4. HIDROOJAYAN	Ambato	Sucre No. 458 y Castillo, Edif. Hidroojayan 2do. F	719-123	719-123
5. HIDROOJAYAN	Quayquil	CAROL 704 - AV 8 DE OCTUBRE	304 930	304 930
6. HIDROPALITE	Cuenca	Panamericana Norte Km 7	875 558	875 558
7. TERMOCISMERALDAS	Esmeraldas	KM. 7 1/2 VIA A ATACAMES	700 220	700 220
8. TERMOCISMERALDAS	Guilo		520 482	520 482
9. ELECTROJIL	Quayquil	Km. 19 vía a la Costa	871 005	871 005
10. INTERVEJA TRADE	Quayquil		630 548	630 548
11. EL ORO	Nachala	AV ARIZAGA 503 Y SANTA ROSA - MACHALA	962 808	962 808
12. MANABI	Manta	EDIFICIO D TRAVEL (FRENTE PLAZA CIVICA)	622 881	622 881
13. EMBUNORTE	Ibarré	BORRERO 6-73 CHICA NARVAEZ - IBARRA	955 860	955 860
14. QUITO	Quito	AV 10 DE AOSTO LAS CASAS - QUITO	564 281	564 281
15. REGIONAL SUR	Loja	OLMEDO 884 Y ROCAFUERTE - LOJA	570 243	570 243
16. RIOBAMBA	Riobamba	LARRA - PRADERA CONSTITUYENTE	981 986	981 986
17. SANTA ELENA		CALLE 22 - AV SEGUNDA - PENINSULA DE SANTI	784 757	784 757

REGISTRO DE FONDOS

Se presenta la siguiente pantalla, en la que se debe seleccionar nuevo registro, se

selecciona el periodo al que corresponde la recaudación por distribución de energía, con la opción nuevo ítem se ingresa de rubro, fecha del depósito, valor recaudado.

Sistema Control de Fideicomiso EEASA USUARIO: patty Conexión: SERVER

Archivo Edición Ventana ?

Registro de Fondos

Periodo: 2 FEBRERO DEL 2007 Fecha Reg.: 13/07/2007

Periodo: 2 FEBRERO DEL 2007

Fecha Reg: 13/07/2007

Total Fondo: 3718238.77 Estatus UTILIZADO

Item	RUBROS	FECHA DEPOSITO	VALOR	Estatus
1	CONSUMO ENERGIA	01/03/2007	2,755,984.83	UTILIZADO
2	ALUMBARDO PUBLICO	01/03/2007	378,595.21	UTILIZADO
3	BOMBEROS	01/03/2007	205,439.01	UTILIZADO
4	FERUM	01/03/2007	111,211.15	UTILIZADO
5	SEGURO CONTRA INCENDIOS	01/03/2007	1,361.00	UTILIZADO
6	TASA BASURA	01/03/2007	166,422.38	UTILIZADO
7	TASAS Y TERCEROS	01/03/2007	108,309.69	UTILIZADO
8	RETENCION	01/03/2007	-5,140.83	UTILIZADO
9	RETENCION BANCOS	01/03/2007	-2,276.93	UTILIZADO
10	VENATANILLA 16	01/03/2007	-1,666.24	UTILIZADO
TOTAL:			3,718,238.77	

REGISTRO VAD

En la siguiente pantalla se seleccionará el Mes del Fondo a procesar el cálculo del V.A.D, presionado el Boton OBTENER RUBRO recuperará el detalle del rubro y el valor, después se escoge el porcentaje asignado para cada uno de los rubros, este porcentaje se basa en el Asignado por el Fondo de Solidaridad para el cálculo del Retorno de fondos al la Empresa.

Sistema Control de Fideicomiso EEASA USUARIO: patty Conexión: SERV...

Archivo Edición Ventana ?

Datos del VAD

Período: Fecha Reg: 13/07/2007

Período: 1 ENERO DEL 2007 Total VAD: 2,132,454.98

Fecha Reg: 13/07/2007 Valor1: 583,659.23 100 %
 Valor2: 1,548,795.75 Otros % Estatus: 0

Item	RUBROS	VALOR RUBRO	%	TOTAL
1	ALUMBRADO PUBLICO	137,063.89	VAD	137,063.89
2	BOMBEROS	205,439.01	100 %	205,439.01
3	CONSUMO ENERGIA	2,755,984.33	VAD	1,361,731.86
4	FERUM	111,211.15	100 %	111,211.15
5	RETENCION	-5,140.83	100 %	-5,140.83
6	TASAS Y TERCEROS	108,309.69	100 %	108,309.69
7	SEGURO CONTRA INCENDIOS	1,361.00	100 %	1,361.00
8	TASA BASURA	166,422.38	100 %	166,422.38
9	RETENCION BANCOS	-2,276.93	100 %	-2,276.93
10	VENATANILLA 16	-1,666.24	100 %	-1,666.24
TOTAL:				2,132,454.98

REGISTRO DE COMPRA

En la siguiente pantalla se registran las facturas por compra de Energía que realiza mensualmente la empresa a los Generadores del Mercado Eléctrico Mayorista, ingresando: Número de Factura, fecha de emisión de la factura, cantidad en KWH, valor total de la factura, se selecciona el porcentaje de retención que se efectuar a la factura ingresada, este porcentaje afectará el valor de la factura para ser restado y establecer el valor a pagar.

Sistema Control de Fideicomiso EEASA USUARIO: patty Conexión: SERV2

Archivo Edición Ventana ?

Registro de Facturas

Periodo: 2 FEBRERO DEL 2007

Id: 2 FEBRERO DEL 2007

Total: 4,251,586.38 Estatus: Nuevo Borrar

Nº Reg	GENERADOR	Nº Doc.	Tipo Documento	FECHA EMISION	CANTIDAD ENERGIA (kWh)	VALOR FACTURA	% RETEN.	Valor Retencion	TOTAL
1	ELECTROQUIL	6472	FACTURA	31/01/2007	15,500,000.00	69,419.00	1%	694.19	48,119.00
2	ELECTROGUAYAS	6942	FACTURA	31/01/2007	15,412,000.00	119,615.69	1%	1,196.16	118,419.53
3	EMAAP-Q	6471	FACTURA	31/01/2007	45,121,200.00	63,890.00	1%	638.90	63,251.10
4	HIDROAGOYAN	9111	FACTURA	31/01/2007	448,745,400.00	1,218,000.00	1%	12,180.00	1,205,820.00
5	HIDRONACION	4068	FACTURA	31/01/2007	3,961.00	39.21	1%	0.39	38.82
6	HIDROPALUTE	1033	FACTURA	31/01/2007	4,206,164.00	42,061.64	1%	420.62	41,641.02
7	TERMOCESMERALDAS	5361	FACTURA	31/01/2007	547,000,000.00	1,850,300.00	1%	18,503.00	1,831,797.00
8	TERMOPICHINCHA	6305	FACTURA	31/01/2007	48,172,500.00	9,417.25	1%	94.17	9,323.08
9	ELECTROQUIL	4519	FACTURA	31/01/2007	948,048,800.00	34,804.88	1%	348.05	34,456.83
TOTALES:						4,294,531.70		42,945.32	4,251,586.38

GENERADOR	TOTAL INICIAL	SALDO
ELECTROQUIL	48,119.00	48,119.00
ELECTROGUAYAS	118,419.53	118,419.53
EMAAP-Q	63,251.10	63,251.10
HIDROAGOYAN	1,205,820.00	1,205,820.00

Esta pantalla también se manipula con las opciones de la barra de menú.

USUARIOS DEL SISTEMA

El sistema es multiusuario por lo tanto existe la opción para controlar a los usuarios que tendrán acceso al sistema, para ello se ha creado la pantalla para manejar esta información, en ella se presenta la clave, usuario, nombres, Fecha de Ingreso, La fecha de Expiración, el campo <admin.> que indica si este usuario es administrador o no, La fecha del Ultimo Bloqueo y si es que se encuentra bloqueado o no.

Es importante indicar que esta pantalla solamente pueden revisar o manejarla los usuarios que estén como administradores de este sistema.

Sistema Control de Fideicomiso EEASA USUARIO: patty Conexión: SERVER

Archivo Edición Ventana ?

Maestro de Usuarios

Recuperar Por: Ordenar y Buscar Por: Regs 2

Usuario	Nombres	Cargo	Admin	Bloqueado	Nivel	Estatus
PATTY	ADMIN	ADMIN	S	N	Administrador	<input checked="" type="checkbox"/>
UEEP	DESBANDO	NONE	N	N	Usuario Normal	<input type="checkbox"/>

Usuario: Fecha Ingreso: Ultimo Bloqueo:
 Clave: Fecha Expiración:
 Nombres: Es Administrador: Nivel:
 Cargo: Bloqueado: Activo:

PERMISOS DE ACCESO

Sistema Control de Fideicomiso EEASA USUARIO: patty Conexión: SERVER

Archivo Edición Ventana ?

Control de Accesos

Usuario	Cargo	Admin	Lock	Expira	Nivel	Estatus
PATTY	ADMIN	Si	No	01/01/2008	1	<input checked="" type="checkbox"/>
UEEP	NONE	No	No	01/01/2008	2	<input type="checkbox"/>

Opciones del Menu

Opción Menu	Active	Editar	Crear	Borrar	Imprimir
MAESTROS:CODIGOS	Si	No	No	No	No
MAESTROS:PERIODOS	Si	No	No	No	No
MAESTROS:RUBROS	Si	Si	Si	Si	Si

Maestros
 Rubros
 Periodos
 Tipos de Cargos
 Tabla de Porcentajes
 Datos de Generadores
 Codigos
 Edición de Constantes
 Administración de Usuarios
 Usuarios del Sistema
 Permisos de acceso
Movimientos
 Registro de Fondos
 Registra datos VAD
 Registra datos del Disponible
 Registra Datos Facturas
 Registro Abonos Facturas
 Registro Abonos Facturas con Fondos Propios
Reportes
 Maestros
 Control de Usuarios
 Usuarios del Sistema
 Permisos de Acceso
 Reportes Generales
 Fondos
 Vad
 Disponible
 Facturas
 Resumen Facturas
 Abonos Facturas
 Abonos con Fondos Propios
 Recaudaciones por Mes
 Liquidación Compra Energía
 Abonos Realizados con RIFacturas
 Zonas Realizados con Disponible

Elegir Parámetros, desplazarse hasta Usuarios y seleccionar Permisos de accesos, inmediatamente aparecerá la pantalla siguiente en la que se permite al usuario administrador asignar las opciones a cada usuario según las opciones del menú.

Para asignar las opciones del menú marcar la opción que se desee luego hacer click en nuevo y la opción de menú pasa a la parte izquierda de la pantalla , aquí puede seleccionar con SI y NO en Activar, Editar, Crear, Borrar, e Imprimir, de acuerdo a estas opciones en el menú del programa se habilitan o no los iconos correspondientes a estas opciones.

REPORTES

En base a los datos ingresados se tienen diversos tipos de reportes, desde el menú principal seleccionamos la opción reportes.

Todas las pantallas de reportes tienen asociados un menú de reportes el cual consta de los siguientes Iconos:

A continuación se explican los Iconos que no se han explicado en el apartado anterior sobre DESCRIPCIÓN DE LOS ICONOS DE LA BARRA DE MENU.



Recuperar.- En todas las pantallas de reportes aparecerán estas solamente con los encabezados, entonces para recuperar la información se debe hacer click en el Icono de recuperar que es el que sigue, e inmediatamente se visualizará la información.



Imprimir.- Para enviar el reporte hacia la impresora una vez que este se encuentre visualizado en la pantalla se debe seleccionar este Icono con lo cual se obtendrá el mismo reporte en la impresora.



En el menú de reportes de Maestros, se encuentran todas las opciones de parámetros, es decir se pueden obtener los reportes de toda la base de datos tales como: Rubros, Datos, Periodos, Código y Tipo de Datos

En las siguientes pantallas se muestran algunos de los reportes que se obtienen:

REPORTE GENERAL DE FONDOS

Sistema Control de Fideicomiso EEASA USUARIO: paty Computadora: SERVER

Archivo Edición Ventana ?

Reporte General de Fondos

Empresario Por: Ordenar Por: Asa Des Viste Privar
 I: Criterio I: Asa Des I: Reglas
 II: Criterio II: Asa Des II: Reglas
 III: Criterio III: Asa Des III: Reglas

EMPRESA ELECTRICA AMBIATO RCN REPORTE GENERAL DE FONDOS Pág 1 de 1
 Dirección: Av. 12 de Noviembre y Espejo Fecha: 18-Jul-2007 21:37
 Teléfono: 027111 Fax: 02091 Ambato - Ecuador

Nº	ID. DOC	FECHA	TOTAL	Estatus
1	2	FEBRERO DEL 2007	3,718,238.77	UTILIZADO

RUBRO	F. DEPOSITO	V A L O R
CONSUMO ENERGIA	01/03/2007	2755.984.33
ALLIEMBARDO PUBLICO	01/03/2007	378.595.21
BOMBREOS	01/03/2007	205.439.01
FERUM	01/03/2007	111211.15
SEGURO CONTRA INCENDIOS	01/03/2007	1361.00
TASA BASURA	01/03/2007	164.422.38
TASAS Y TERCEROS	01/03/2007	108.309.63
RETENCION	01/03/2007	5.140.83
RETENCION BANCOS	01/03/2007	-2276.93
VENANTANIA IS	01/03/2007	1466.24
SUMAR		3,718,238.77
TOTAL		3,718,238.77

REPORTE DE RECAUDACION MENSUAL

Sistema Control de Fideicomiso EEASA USUARIO: paty Computadora: SERVER

Archivo Edición Ventana ?

Reporte Recaudaciones Mensuales

Fondo: FEBRERO DEL 2007 Periodo: 01/04/2007 - 30/04/2007 Pág(9) Procesar

EMPRESA ELECTRICA AMBIATO RCN RECAUDACION MENSUAL Pág 1 de 2
 Dirección: Av. 12 de Noviembre y Espejo Fecha: 18-Jul-2007 21:30
 Teléfono: 027111 Fax: 02091 Ambato - Ecuador

RUBRO	VALOR
ALLIEMBARDO PUBLICO	378.595.21
BOMBREOS	205.439.01
CONSUMO ENERGIA	2,755.984.33
FERUM	111211.15
RETENCION	5.140.83
TASAS Y TERCEROS	108.309.63
SEGURO CONTRA INCENDIOS	1,361.00
TASA BASURA	164.422.38
RETENCION BANCOS	-2,276.93
VENANTANIA IS	1,466.24
Sub Total	3,718,238.77

ALLIEMBARDO PUBLICO	378.595.21
BOMBREOS	205.439.01
CONSUMO ENERGIA	2,755.984.33
FERUM	111211.15
RETENCION	5.140.83
TASAS Y TERCEROS	108.309.63
SEGURO CONTRA INCENDIOS	1,361.00

ELABORADO POR: PATRICIA CONDO AUXILIAR CONTABLE
 REVISADO POR: ING. ROBERTO SALAZAR DIRECTOR FINANCIERO

REPORTE GENERAL DEL VAD

Sistema Control de Fideicomiso EEASA USUARIO: patty Conexión: SERVER

Archivo Edición Ventana ?

Reporte General de VAD

Recuperar Por: I Criterio: II Criterio: III Criterio: Ordenar Por: I: II: III: Aso Des Vista Previa Reglas 1 Regs

EMPRESA ELECTRICA AMBATO RCN. REPORTE GENERAL VAD Pág 1 de 1
 Dirección: Av. 12 de Noviembre y Espejo Fecha: 18-Jul-2007 21:37
 Telfs: 827111 Fax: 422091 Ambato - Ecuador

Nº	ID. DOC	FECHA	VALORES 100%	VALORES Otros%	TOTAL	
1	1	ENERO DEL 2007	13/07/2007	583659.23	1540795.75	2,132,454.98
	RUBRO		VALOR RUBRO	%	VALOR	
	CONSUMO ENERGIA		2755984.33	49.41	1361731.86	
	ALUMBRADO PUBLICO		378595.21	49.41	187063.89	
			Sub Total:		1,540,795.75	
	TASAS Y TERCEROS		108309.69	100.00	108309.69	
	RETENCION		-5140.83	100.00	-5140.83	
	TASA BASURA		166422.38	100.00	166422.38	
	SEGURO CONTRA INCENDIOS		1361.00	100.00	1361.00	
	FERLUM		111211.15	100.00	111211.15	
	RETENCION BANCOS		-2276.93	100.00	-2276.93	
	VENANTANILLA 16		-1666.24	100.00	-1666.24	
	BOMBEROS		205439.01	100.00	205439.01	
			Sub Total:		583,659.23	
			SUMAN:		2,132,454.98	

REPORTE GENERAL DE FACTURAS

Sistema Control de Fideicomiso EEASA USUARIO: patty Conexión: SERVER

Archivo Edición Ventana ?

Reporte General de Facturas

Recuperar Por: I Criterio: II Criterio: III Criterio: Ordenar Por: I: II: III: Aso Des Vista Previa Reglas 1 Regs

EMPRESA ELECTRICA AMBATO RCN. REPORTE GENERAL FACTURAS Pág 1 de 1
 Dirección: Av. 12 de Noviembre y Espejo Fecha: 18-Jul-2007 21:38
 Telfs: 827111 Fax: 422091 Ambato - Ecuador

Nº	ID. DOC	TOTAL							
1	2	4,251,586.38							
Reg.	GENERADOR	TIPO DOC.	Nº DOC.	FCH. EMI	CANT. ENER.	VALOR	%	V.RETENCION	TOTAL
1	ELECAUSTRO	FACTURA	5452	31/01/2007	15,500,000.00	956,403.03	1.00	9,564.03	946,839.00
2	ELECTROGLAYAS	FACTURA	6942	31/01/2007	15,412,000.00	119,615.69	1.00	1,196.16	118,419.53
3	EMAAP-Q	FACTURA	6471	31/01/2007	45,121,200.00	63,890.00	1.00	638.90	63,251.10
4	HIDROAGOYAN	FACTURA	9111	31/01/2007	446,745,400.00	1,218,000.00	1.00	12,180.00	1,205,820.00
5	HIDRONACION	FACTURA	4068	31/01/2007	3,961.00	39.21	1.00	0.39	38.82
6	HIDROPAUTE	FACTURA	1033	31/01/2007	4,206,164.00	42,061.64	1.00	420.62	41,641.02
7	TERMOESMERALDAS	FACTURA	5361	31/01/2007	547,000,000.00	1,850,300.00	1.00	18,503.00	1,831,797.00
8	TERMOPICHINCHA	FACTURA	6305	31/01/2007	46,172,500.00	9,417.25	1.00	94.17	9,323.08
9	ELECTROQUIL	FACTURA	4519	31/01/2007	946,046,800.00	34,804.88	1.00	348.05	34,456.83
			SUMAN:						4,251,586.38
			TOTAL:						4,251,586.38

REPORTE DE LIQUIDACION DE COMPRA

Sistema Control de Fideicomiso EEASA USUARIO: patty

Reporte General de Liquidación por Compra de Energía

Empresas: EMPRESA ELECTRICA AMBATO RCN. Dirección: No. 1 Av. 12 de Noviembre y Espejo. No: 827111 Fax: 422051 Ambato - Ecuador.

LIQUIDACION POR COMPRA DE ENERGIA

Página: 1 de 1 Fecha: 18-Jul-2007 21:40

PERIODO - MES: 2 FEBRERO DEL 2007

NO.	GENERADOR	TIPO DOC.	Nº DOC.	FCH. EMI	CANT. ENER.	VALOR	%	V.RETENCION	TOTAL
	ELECAUSTRO	FACTURA	5462	31/01/2007	15,500,000.00	956,403.03	1.00	9,564.03	946,839.00
	ELECTROGUAYAS	FACTURA	6942	31/01/2007	15,412,000.00	119,615.59	1.00	1,196.16	118,419.53
	EMAAP-Q	FACTURA	6471	31/01/2007	45,121,200.00	63,890.00	1.00	638.90	63,251.10
	HIDROAGOYAN	FACTURA	9111	31/01/2007	448,745,460.00	1,218,000.00	1.00	12,180.00	1,205,820.00
	HIDRONACION	FACTURA	4068	31/01/2007	3,961.00	39.21	1.00	0.39	38.82
	HIDROPALITE	FACTURA	1033	31/01/2007	4,206,184.00	42,061.54	1.00	420.62	41,641.02
	TERMOESMERALDAS	FACTURA	5361	31/01/2007	547,000,000.00	1,850,300.00	1.00	18,503.00	1,831,797.00
	TERMOPICHINCHA	FACTURA	6305	31/01/2007	48,172,500.00	9,417.25	1.00	94.17	9,323.08
	ELECTROQUIL	FACTURA	4519	31/01/2007	948,048,800.00	34,804.88	1.00	348.05	34,456.83
	SUMAR:								4,261,566.38

REPORTE DE RESUMEN DE FACTURAS

Sistema Control de Fideicomiso EEASA USUARIO: patty Conexión: SERVER

Reporte General Resumen Facturas

Recuperar Por: I Criterio: II Criterio: III Criterio: Ordenar Por: I: II: III: Aso Des Vista Previa Reglas 9 Regs

EMPRESA ELECTRICA AMBATO RCN. Dirección: Av. 12 de Noviembre y Espejo Telfs: 827111 Fax: 422091 Ambato - Ecuador

REPORTE RESUMEN FACTURAS Pág 1 de 1 Fecha: 18-Jul-2007 21:38

Nº	PERIODO - MES	GENERADOR	TOTAL INICIAL	SALDO
1	FEBRERO DEL 2007	ELECAUSTRO	946839.00	946839.00
2	FEBRERO DEL 2007	ELECTROGUAYAS	118419.53	118419.53
3	FEBRERO DEL 2007	EMAAP-Q	63251.10	63251.10
4	FEBRERO DEL 2007	HIDROAGOYAN	1205820.00	1205820.00
5	FEBRERO DEL 2007	HIDRONACION	38.82	38.82
6	FEBRERO DEL 2007	HIDROPALITE	41641.02	41641.02
7	FEBRERO DEL 2007	TERMOESMERALDAS	1831797.00	1831797.00
8	FEBRERO DEL 2007	TERMOPICHINCHA	9323.08	9323.08
9	FEBRERO DEL 2007	ELECTROQUIL	34456.83	34456.83
TOTALES:			4,251,586.38	4,251,586.38

REPORTE DE DETALLE DE ABONOS

Sistema Control de Fideicomiso EEASA USUARIO: patty Conexión: SERVER

Reporte General Resumen Facturas Detalle Abonos

Recuperar Por: I Criterio: II Criterio: III Criterio: Ordenar Por: I: II: III: Aso Des Vista Previa Reglas 9 Regs

EMPRESA ELECTRICA AMBATO RCN. Dirección: Av. 12 de Noviembre y Espejo Telfs: 827111 Fax: 422091 Ambato - Ecuador

ABONOS REALIZADOS CON RFACTURAS Pág 1 de 1 Fecha: 18-Jul-2007 21:41

Nº	PERIODO - MES	GENERADOR	TOTAL INICIAL	SALDO
1	FEBRERO DEL 2007	ELECAUSTRO	946839.00	946839.00
2	FEBRERO DEL 2007	ELECTROGUAYAS	118419.53	118419.53
3	FEBRERO DEL 2007	EMAAP-Q	63251.10	63251.10
4	FEBRERO DEL 2007	HIDROAGOYAN	1205820.00	1205820.00

ANEXOS



EMPRESA ELECTRICA AMBATO REGIONAL CENTRO NORTE S.A.

Ambato, septiembre 21 del 2007

Ing. Santiago Acurio
Director de la Escuela de Sistemas
PUCESA
Presente

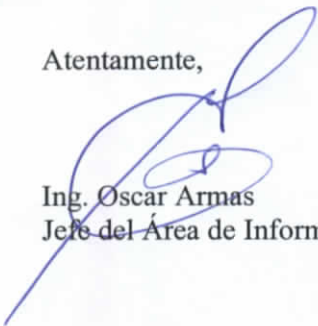
De mi consideración:

El Suscrito Ing. Oscar Armas, Jefe del Área de Informática del Departamento Financiero de la Empresa Eléctrica Ambato Regional Centro Norte, en honor a la verdad certifica que:

El proyecto desarrollado por la Sra. Patricia Condo Carranza bajo el tema "Estudio sobre el Administrador de Base de Datos Sybase SQL aplicado a la Automatización de Información Financiera del Fideicomiso en la EEASA. RCN SA. utilizando Power Builder", en cuanto a la aplicación está funcionando correctamente en el Departamento Financiero, demostrando que este presenta beneficios para el control de la Información acerca del Fideicomiso, siendo eficiente en los reportes que presenta.

Es todo cuanto puedo certificar en honor a la verdad, la interesada puede hacer uso del presente documento para los fines que estime conveniente.

Atentamente,



Ing. Oscar Armas
Jefe del Área de Informática

Ambato, septiembre 21 del 2007

Ing. Santiago Acurio
Director de la Escuela de Sistemas
PUCESA
Presente


De mi consideración:

El Suscrito Ing. Wigberto Sánchez, Profesor Validador de la Pontificia Universidad Católica del Ecuador Sede Ambato , en honor a la verdad certifica que:

El proyecto desarrollado por la Sra. Patricia Condo Carranza bajo el tema "Estudio sobre el Administrador de Base de Datos Sybase SQL aplicado a la Automatización de Información Financiera del Fideicomiso en la EEASA. RCN SA.utilizando Power Builder", en cuanto a la aplicación está funcionando correctamente en el Departamento Financiero, demostrando que este presenta beneficios para el control de la Información acerca del Fideicomiso, siendo eficiente en los reportes que presenta.

Es todo cuanto puedo certificar en honor a la verdad, la interesada puede hacer uso del presente documento para los fines que estime conveniente.

Atentamente,


Ing. Wigberto Sánchez
PROFESOR - VALIDADOR
PUCESA

