

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

UNIDAD DE INGENIERIA DE SISTEMAS

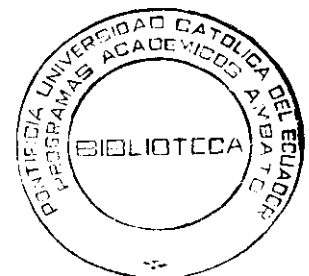
**DISERTACIÓN DE GRADO PREVIA LA OBTENCIÓN DEL
TÍTULO DE INGENIERÍA EN SISTEMAS**

**“ Integración de Sistemas Nativos en diferentes plataformas aplicado a la
Cooperativa de Ahorro y Crédito O.S.C.U.S. Ltda. “**

Néstor Vinicio Vivas Carrillo.

DIRECTOR DE TESIS : Ing. Msc. ROXANA MERIÑO.

Ambato, 1999



PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

UNIDAD DE INGENIERIA DE SISTEMAS

**DISERTACIÓN DE GRADO PREVIA LA OBTENCIÓN DEL
TÍTULO DE INGENIERÍA EN SISTEMAS**

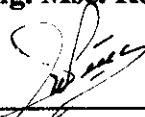
**“ Integración de Sistemas Nativos en diferentes Plataformas aplicado a la
Cooperativa de Ahorro y Crédito O.S.C.U.S. Ltda. “**

Director :



Ing. Msc. Roxana Meriño.

Revisores :



Ing. Msc. Wigberto Sánchez.



Ing. David Guevara.

Néstor Vinicio Vivas Carrillo.

Ambato, 1999

DEDICATORIA

Este trabajo lo dedico a mi esposa e hijo, fuente de inspiración y estímulo para mi superación personal, a mis padres que me han sabido apoyar y guiar por el buen camino para poder llegar a la consecución de mis metas.

AGRADECIMIENTO

Mi profundo agradecimiento a todas las personas e instituciones que de una u otra manera supieron apoyarme desinteresadamente en la realización de este trabajo práctico.

De manera especial a la Cooperativa de Ahorro y Crédito "OSCEUS" Ltda. A la directora de tesis Ing. Msc. Roxana Meriño, a los asesores Ing. Msc. Wigberto Sánchez e Ing. David Guevara por su valiosa ayuda y a nuestra universidad que sembró en mi la semilla del saber.

INDICE

CAPITULO PRIMERO	3
1 GENERALIDADES	3
1.1 Alcance y propósito del proyecto.	3
1.2 Justificación, delimitación, metas.	4
1.3 Temas generales.	5
1.3.1 Plataformas.	5
1.3.1.1 Windows NT.	6
1.3.1.2 UNIX.	7
1.3.1.3 Netware Novell 3.x y 4.x.	7
1.3.2 Estándares Nativos.	8
1.3.3 Protocolos de Comunicación.	10
1.3.3.1 Protocolos de Enlace de Datos.	10
1.3.3.2 Protocolos del Nivel de Red.	11
1.3.3.3 Protocolos del Nivel de Transporte.	13
1.3.4 Sistemas Distribuidos.	14
1.3.4.1 La arquitectura NFS.	15
CAPITULO SEGUNDO	17
2 ESTUDIO DE LA INSTITUCION.	17
2.1 Actividades y Estructura de la Institución.	17
2.2 Estudio de las actividades de Organismo.	18
2.3 Recursos Existentes.	19
2.3.1 Recursos Humanos.	19
2.3.2 Recursos de Hardware.	20
2.3.3 Recursos de Software.	21
2.3.4 Recursos de Comunicaciones.	22
CAPITULO TERCERO	24
3 ANALISIS DE LA RED.	24
3.1 Redes de Ordenadores.	24

1.3.1.2 UNIX.

Es un sistema operativo multiusuario, multitarea y de tiempo compartido que soporta las redes interconectadas y los sistemas de archivos de redes, este sistema operativo es bastante difundido por su versatilidad, tal es el caso de que existen versiones de UNIX por diversos fabricantes así en IBM el AIX, de SUN Microsystem SUN SOLARIES, de Microsoft el AIX y otros. UNIX es el responsable de la adición de los protocolos de conexión de redes: Protocolo de control de transmisión / Protocolo Internet (TCP/IP, Transmission Control Protocol / Internet Protocol). Este Sistema Operativo se popularizo en los entornos de Ingeniería, diseño asistido por computadora (CAD, Computer Asisted Design) y científicos, esta escrito en lenguaje C. La Cooperativa de Ahorro y Crédito El Sagrario Ltda., y la Cooperativa de Ahorro y Crédito Oscus Ltda. trabajan en esta plataforma.

1.3.1.3 Netware Novell 3.x y 4.x.

Novell NetWare es un sistema operativo en red que permite interconectar sus equipos a los usuarios de computadoras y compartir recursos, archivos y programas. Son sistemas operativos de 32 bits que utilizan un único espacio de direcciones sin segmentación, esto permite que los programas trabajen en forma más eficiente, el sistema operativo puede gestionar miles de interrupciones y proceso de miles de peticiones de clientes por segundo, Netware 3.x y 4.x son modulares ampliables que permiten cambios, actualizaciones y adiciones de la red. La Cooperativa de Ahorro y Crédito San Francisco Ltda. y la Cooperativa de Ahorro y Crédito La Merced Ltda. trabajan con Novell.

1.3.2 Estándares Nativos.

Estándares Nativos se refiere a los ficheros de bases de datos propios o nativos que posee un servidor o cualquier estación de trabajo, estos ficheros no pueden ser descifrados o utilizados por otras bases de datos diferentes; para poder utilizar estos ficheros se tendrá que recurrir a la técnica de migración de datos teniendo en cuenta que cada base de datos tiene drivers para conectividad abierta en bases de datos (ODBC, Open Database Connectivity) para cada plataforma nativa.

Si se requiere migrar los datos de un fichero de una plataforma a otra se debe tomar en cuenta varios aspectos como las seguridades de migración, forma de transmisión de datos, datos que se van a migrar, forma de almacenamiento de información entre otros.

Otra de las formas para el acceso a datos es la utilización de Librerías² ODBC, la cual es un desarrollo de la compañía Microsoft que se ha convertido en estándar, para el acceso a través de él a gran cantidad de tipos de datos. Básicamente, por tanto, cualquier aplicación simple que emplee ODBC puede acceder a las Bases de datos soportadas por este estándar.

El funcionamiento es: la aplicación sólo necesita comunicarse con un paquete de archivos (ODBC), e instantáneamente puede trabajar con cualquier tipo de datos soportados por este paquete.

² Librería: Procedimientos compartidos almacenados para múltiples funciones

Existen cientos de sistemas de gestión de bases de datos (DBMS, database management system) disponibles para los ordenadores personales, y miles de aplicaciones que acceden a los datos contenidos en las bases de datos.

Hay dos posibles accesos a la base de datos (directo e indirecto), el directo consiste en que normalmente, una compañía diseña una aplicación que accederá a datos. Esta aplicación se desarrolla hacia un tipo de datos, y por tanto sólo se implementa el acceso para ese tipo de datos. Si se requiere acceso a otro DBMS, es necesario por tanto, implementar un nuevo controlador o driver de acceso. Este sistema hace que el acceso sea directo a la base de datos, pero tiene el inconveniente de que hay que desarrollar el enlace para cada DBMS que se quiera soportar.

Otro posible acceso es el indirecto, es decir, si el DBMS posibilita que con ODBC puedan ser accedidos los datos, funciona con lo que se denomina fuente de datos ODBC (ODBC data source), la aplicación accede a través del paquete de archivos ODBC "indirectamente", y si añadimos que ODBC es soportado por los DBMS más comunes, con una misma aplicación y con un mismo paquete de drivers, podremos acceder a todas las DBMS sin necesidad de hacer un desarrollo para cada tipo.

Quizás, la mayor ventaja es que el acceso a datos a través del ODBC permite el poder gestionar un amplio rango de datos con un solo interface. Desde que las más populares DBMS ofrecen drivers ODBC, muchas son las aplicaciones que lo incluyen como drivers de acceso.

La principal desventaja son las capas a través de las cuales tiene que pasar la consulta, esto hace que el tiempo de respuesta hasta que se obtienen los datos se incremente. El proceso es que la petición ha de "traducirse" a ODBC, para que éste entienda la consulta. ODBC determina que fuente de datos contiene los datos que se piden y transmite la petición a la siguiente capa que es la fuente de datos ODBC (ODBC data source). La fuente de datos analiza la petición y "traduce" de nuevo la consulta a un formato que pueda ser "comprendido" por la DBMS. Este complejo proceso puede verse alterado por cualquier fallo en cualquiera de sus fases y por tanto la consulta no tendría éxito.

1.3.3 Protocolos de Comunicación.

A los protocolos se los define como las reglas y normas diseñadas para establecer, mantener y cerrar la comunicación entre las estaciones que forma la red de datos. Los protocolos están definidos por niveles, tomando como referencia el modelo de Interconectividad de sistemas abiertos (OSI, Open Systems Interconnection), para entornos de redes, por lo tanto en una red se manejan varios protocolos así para el acceso al canal de transmisión encaminamiento de los paquetes en la red o para las aplicaciones.

1.3.3.1 Protocolos de Enlace de Datos.

Se los conoce como protocolos de línea o protocolos de enlace de datos, gestionan el enlace físico y controlan el tráfico de comunicaciones que atraviesa el enlace físico de comunicaciones. Funciona en el primero y segundo nivel correspondiente al modelo OSI. Los protocolos más conocidos en este nivel son:

a) CSMA/CD

(Carrier Sense Multiple Access with Detection Collisions), Método de acceso múltiple con detección de portadora y detección de colisiones. La versión mas extendida de este método es la especificación Ethernet, es independiente del medio, por lo tanto, no le afecta que la red sea de banda ancha o de banda base. Para transmitir envía una señal al canal para averiguar si está libre o si alguna estación está transmitiendo usando el canal principal, esta señal se la conoce como portadora, cuando la portadora detecta la liberación del canal la estación empiezan a transmitir al mismo tiempo, entonces se produce las colisiones, este protocolo las detecta y las combate.

b) Testigo.

Se usa generalmente en redes con topología en anillo lógico y físico, variando un poco de acuerdo a cada fabricante, su funcionamiento esta basado que por anillo circule un testigo, este testigo le ofrece a cada estación la posibilidad de transmitir, es decir, le otorga el dominio de la red. Cuando ninguna estación está transmitiendo el testigo se encuentra libre caso contrario está en un estado ocupado, al terminar la transmisión el testigo sigue su camino.

1.3.3.2 Protocolos del Nivel de Red.

Estos protocolos proporcionan el encaminamiento de los paquetes en la red, trabajan conjuntamente con los protocolos del nivel de transporte para asegurar la fiabilidad en la transmisión de datos. Los protocolos más conocidos en este nivel son:

a) IP (Protocolo Internet, Internet Protocol).

Se introdujo en la época de los 80's y desde entonces ha estado funcionando con gran aceptación en estos tiempos sobre todo porque es el protocolo oficial de la super autopista de la información INTERNET, fue desarrollado como parte del proyecto ARPANET fundado por la agencia de proyectos avanzados para la investigación de los Estados Unidos, con el objetivo de proporcionar conectividad entre sistemas de comunicación independientes. IP es un protocolo **no orientado a conexión**, basado en la idea de los *datagramas*, que son paquetes de datos, los cuales recorren a través de la red hasta llegar a su destino por diferentes caminos, es decir, no hay una ruta fija por lo tanto, no necesariamente llegan en orden a su destino, el receptor ensambla e integra la información, en caso de error o pérdida de paquetes solicita retransmisión de los datagramas correspondientes entonces aquí no se da control de flujo, reconocimiento de recepción, comprobación de error y secuenciamiento, IP maneja la congestión con el descarte de paquetes, para esto trabaja con los protocolos del nivel superior; el protocolo de control de transmisión (TCP, Transmission Control Protocol), trabaja en el nivel de transporte y tiene asignado todas estas tareas, que cuidan el resecuenciamiento y manejo de errores de modo que IP es más rápido, más eficiente y trabaja con redes de área local y extensa.

b) IPX (Intercambio de paquetes entre redes, Internetwork Packet Exchange).

IPX es el protocolo de conexión de red par a par incorporado en Netware de Novell desarrollado a partir del Sistema de red de Xerox. Netware es un S.O cliente servidor que proporciona servicios de compartición de archivos, de servicios de impresoras, seguridad, además servicios de comunicaciones (Fax y Correo Electrónico)

c) NETBEUI (NETBIOS Extended User Interface).

IBM y Microsoft diseñaron los protocolos básicos de entrada y salida en red NETBIOS (Network Basic Input Output System) y NETBEUI este es un protocolo de niveles de transporte y red del modelo de protocolos OSI, diseñado especialmente para redes LAN de pequeño a mediano tamaño, se integra con NETBIOS para ofrecer un sistema de comunicaciones eficiente, de tal manera que NETBEUI proporciona los servicios subyacentes para red y transporte de datos y NETBIOS establece y mantiene las sesiones de comunicaciones entre las redes ya que es un protocolo de nivel de sesión según el modelo de referencia OSI.

1.3.3.3 Protocolos del Nivel de Transporte.

Proporcionan servicios de control en la distribución de los datos, usa el sistema de comunicaciones orientado a conexión. Los protocolos más representativos dentro de este nivel son:

a) TCP (Protocolo de control de transmisión, Transport Control Protocol).

Es una parte del grupo de protocolos TCP/IP de INTERNET y UNIX. Se encarga del control de flujo, secuenciamiento, control de errores y transmisión en caso de pérdida de paquetes, requiere de la fase establecimiento de conexión (Una vez realizada proporciona distribución fiable y eficiente de los datos) en donde el emisor envía un mensaje al receptor, luego el receptor debe devolver el mensaje al emisor confirmando la conexión, entonces, se da la transferencia inicial de datos para establecer los controles de la transferencia.

1.3.4.1 La arquitectura NFS.

El Sistema de archivos en red (NFS, Network File System) es un sistema operativo de red, la cual su idea fundamental es permitir que una colección de clientes y servidores compartan un sistema de archivos común. En la mayoría de los casos, todos los clientes y servidores están en la misma LAN, pero esto es necesario. Es posible ejecutar NFS en una red amplia, el NFS permite que cada máquina sea un cliente y un servidor al mismo tiempo.

La característica básica de la arquitectura NFS es que los servidores exportan directorios y los clientes lo montan de manera remota, si dos o más clientes montan el mismo directorio al mismo tiempo, ellos pueden comunicar o compartir archivos en sus directorios comunes. Un programa en un cliente puede crear un archivo y un programa en otro cliente distinto puede leer dicho archivo. Una vez llevados a cabo los montajes, no hay que hacer nada en especial para lograr compartir los archivos. Los archivos compartidos están ahí, en la jerarquía de directorios de varias máquinas y se pueden leer o escribir en ellos de manera usual. Esta sencillez es uno de los grandes atractivos de NFS.

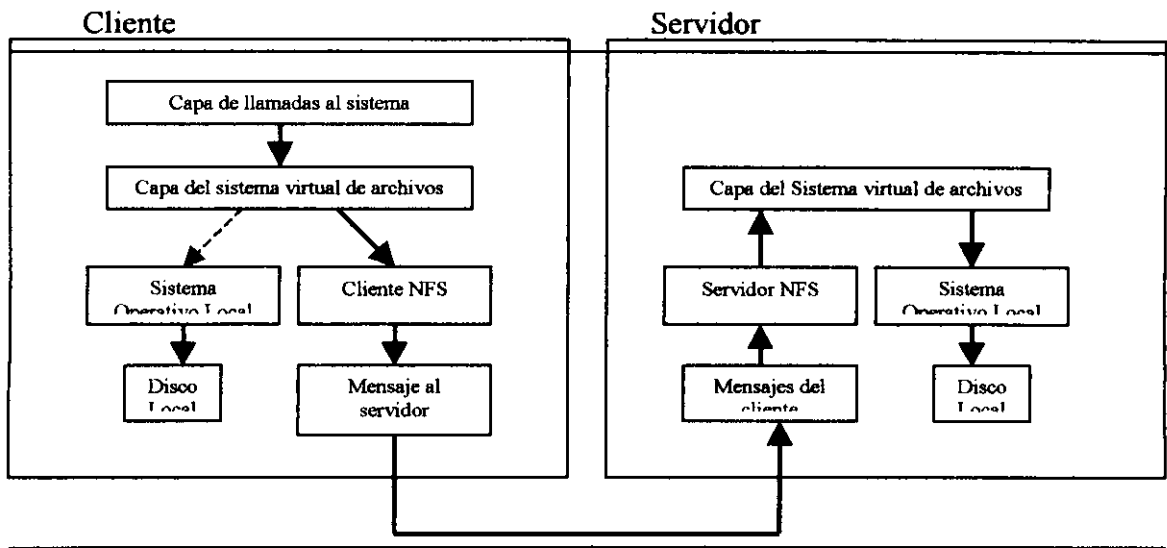


Figura 1.1 Estructura de la capa NFS (Network).

CAPITULO SEGUNDO

2 ESTUDIO DE LA INSTITUCION.

2.1 Actividades y Estructura de la Institución.

"O.S.C.U.S." que su significado es "Obra Social Cultural Sopeña", es una empresa formada por personas naturales y jurídicas para ofrecer créditos y servicios sin fines de lucro; está estructurada de la siguiente manera:

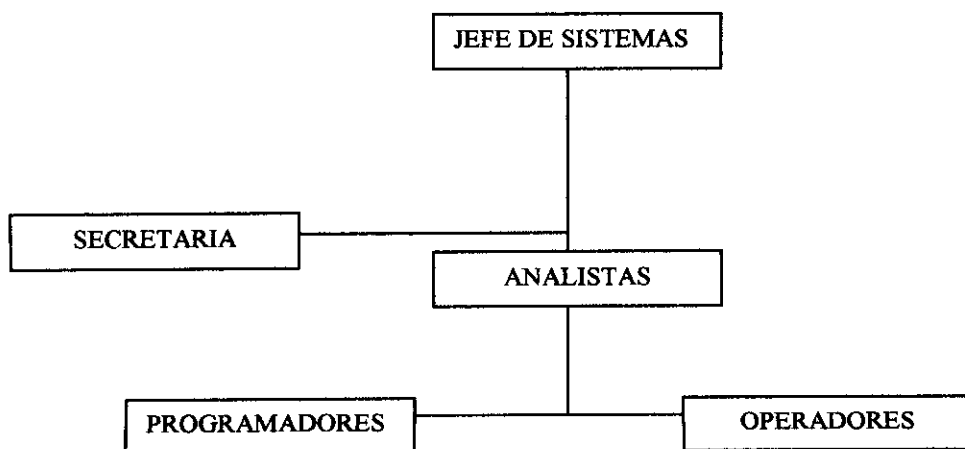
- Asamblea General. Es la máxima autoridad de la Cooperativa, se renova cada 2 años, se reúne anualmente para conocer los avances de la institución y aprobar planes, presupuestos e informes.
- Consejo de Administración. Organismo legislativo elegido por la Asamblea General, toma importantes decisiones para el desarrollo institucional.
- Consejo de Vigilancia. Es el organismo elegido también por la Asamblea General, controla el proceso económico y financiero de la Cooperativa.
- Comisiones. Son organismos que apoyan al cumplimiento de las metas institucionales.
- Gerencia. Desarrolla las decisiones de los organismos directivos.
- Empleados. Trabajadores que llevan a la práctica los programas en beneficio del socio.

2.2 Estudio de las actividades de Organismo.

Existe un organismo que es el encargado de procesar las transacciones de los socios, de este departamento depende la exactitud de los datos en cuanto a cálculos de intereses de ahorros y préstamos tiene que ver; dicho organismo es el responsable de tener informado al día los movimientos de la institución a los directivos para que estos puedan tomar decisiones; este departamento es el de Sistemas.

El área de sistemas en los actuales momentos está sufriendo un cambio profundo en cuanto a su sistema informático, se esta cambiando el viejo sistema por un sistema moderno e integrado en su totalidad.

De acuerdo con la investigación realizada en la Cooperativa de Ahorro y Crédito "O.S.C.U.S." Ltda., el departamento de sistemas esta estructurado de la siguiente manera:



2.3 Recursos Existentes.

2.3.1 Recursos Humanos.

Según el estudio realizado en la Institución, el personal que labora en el Departamento de Sistemas es suficiente para atender sus requerimientos en cuanto al procesamiento de datos tiene que ver.

En la Agencia Matriz en el departamento de Sistemas, se encuentra conformado por el siguiente personal humano:

- Un jefe de sistemas
- Dos analistas programadores
- Un programador
- Un operador de sistemas
- Una secretaria.

En cuanto a las Sucursales de la Institución no registran personal en el área de Sistemas, llevando un control de las mismas en la agencia matriz.

El personal que labora en esta área se encuentra debidamente capacitado y preparado para realizar lo que se les asigne, es decir cada uno de ellos es responsable en sus funciones, además se capacita continuamente al personal.

2.3.2 Recursos de Hardware.

La Institución ha venido funcionando con un equipo antiguo, el cual no pasaría las pruebas para el año 2000; por lo cual la Cooperativa se encuentra en pleno proceso de actualización de equipos.

Según el inventario de los equipos tenemos:

- Servidores Actuales
 - Un Data general con 1 CPU, 6 Mb. Ram, 3 Hd. 350 Mb c/u.
 - Un Data General 1 CPU, 1 Mb Ram, Hd. 170,36,15 Mb respectivamente.
- Servidores Futuros
 - Un SUN Solaris 3350, 1 CPU, 256 Mb Ram, Hd. 1 Tb.
 - Un SUN Solaris 250, 1 CPU, 256 Mb Ram., Hd. 500 Gb.
 - Seis Dell Pentium II, 300 Mhz, 1 CPU, 128 Mb Ram, Hd. 8 Gb.
- Terminales Inteligentes
 - Setenta y siete Dell Pentium, 300 Mhz., 32 Mb Ram, Hd. 2.3 Gb.
- Seis impresoras matriciales.
- Seis impresora a Inyección de tinta.
- Cinco impresoras láser.
- Doce modems externos de distintas velocidades de transmisión.
- Doce hubs.
- Se estima implementar ruteadores.

2.3.3 Recursos de Software.

La Cooperativa trabaja actualmente con un sistema de archivos indexados, el cual fue desarrollado por la misma Institución en lenguaje Cobol versión 3.50; para enfrentar los retos del nuevo milenio se está implantado un nuevo sistema desarrollado en lenguaje Visual Basic versión 4.0, el cual tiene un motor de bases de datos Informix de tipo Universal Server; este sistema fue desarrollado por la empresa Mexicana Aseinform.

A continuación detallamos el software existente:

- S.O. Multiusuarios
 - Windows95 versión 4.0 en 4 terminales.
 - AOS/VS versión 7.6 en 1 servidor.
 - AOS versión 7 en 4 terminales.
- Bases de datos
 - Infos versión 3.1 (Sistema actual)
 - Informix Universal Server versión 7.2 (Sistema futuro)
- Herramientas de desarrollo
 - Cobol versión 3.50.
 - Visual Basic versión 4.0
- Otros paquetes adicionales
 - MS. Office 97
 - Cristal Report versión 3.0

En cuanto a los procesos que se encuentran automatizados tenemos los siguientes:

- Captaciones automatizado 90%; el nuevo 100% automatizado.
- Colocaciones automatizado 80%; el nuevo 100% automatizado.
- Contabilidad automatizado 100%; el nuevo 100% automatizado.
- Tesorería automatizado 50%; el nuevo 100% automatizado.
- Presupuesto automatizado 70%; el nuevo 100% automatizado.

Cabe indicar que el sistema que se encuentra funcionando no se encuentra integrado en sus módulos; en cambio el sistema nuevo que se esta implantado y que se encuentra en proceso de integración y pruebas está totalmente integrado en todos sus módulos; además no se contaba con una comunicación entre agencias y sucursales lo cual se esta integrando.

2.3.4 Recursos de Comunicaciones.

La interconexión entre terminales o computadoras es un factor primordial en el intercambio de información, la Cooperativa OSCUS posee una red de área local (LAN) de mas de 30 equipos conectados entre sí de la siguiente forma:

- Sistema Unix, su topología³ de tipo árbol y cableado UTP5 o par trenzado, conectado a varios hubs⁴ de 16 y 8 puertos, utilizando el protocolo TCP/IP.
- Red Windows NT , su topología de tipo estrella y cableado UTP5 conectado a un hubs de 8 puertos, utilizando como protocolo principal TCP/IP y como alterno Netbeui.

³ Topología: La forma como se conectan los equipos, es decir la arquitectura de la red.

⁴ Hub: Es un conmutador de red que acondiciona y amplifica la intensidad de la señal.

- Red de área extensa (WAN), la forma de comunicación es por medio línea telefónica digital dedicada (LP) y la conexión es permanente, para esto cuenta con modems⁵ externos de varios tipos.
- La Cooperativa cuenta con el servicio de Internet.

⁵ Modems: Convertir señales digitales en analógicas y transmitir sobre la red telefónica.

CAPITULO TERCERO

3 ANALISIS DE LA RED.

3.1 Redes de Ordenadores.

El término *red* significa dos o más computadoras conectadas entre sí. Algunas de las razones para unir las computadoras son permitir comunicarse a las personas y compartir recursos; una red de ordenadores permite conectar a los ordenadores que la forman con la finalidad de compartir información, como documentos o bases de datos, o recursos físicos, como impresoras o unidades de disco. Las redes suelen clasificarse según su extensión en:

a) **LAN** (Local Area Network) Son las redes de área local. La extensión de este tipo de redes suele estar restringida a una sala o edificio, aunque también podría utilizarse para conectar dos o más edificios próximos. Diversas LANs pueden conectarse entre sí mediante un encaminador, de esta forma podemos formar una red de área extensa, (Wide Area Network o WAN).

b) **WAN** (Wide Area Network) Son redes que cubren un espacio muy amplio, conectando a ordenadores de una ciudad o un país completo. Para ello se utilizan las líneas de teléfono y otros medios de transmisión más sofisticados, como pueden ser las microondas. La velocidad de transmisión suele ser inferior que en las redes locales.

Varias redes pueden conectarse entre sí formando una red lógica de área mayor. Para que la transmisión entre todas ellas sea posible se emplean los routers, que son los sistemas que conectando físicamente varias redes se encargan de dirigir la información por el camino

adecuado. Cuando las redes que se conectan son de diferente tipo y con protocolos distintos se hace necesario el uso de los gateways, los cuales además de encaminar la información también son capaces de convertir los datos de un protocolo a otro. Generalmente los términos router y gateway se emplean indistintamente para referirse de forma general a los sistemas encargados del encaminamiento de datos en Internet.

Lo que se conoce como Internet es en realidad una red de redes, la interconexión de otras redes independientes de manera que puedan compartir información entre ellas a lo largo de todo el planeta. Para ello es necesario el uso de un protocolo de comunicaciones común. El protocolo que proporciona la compatibilidad necesaria para la comunicación en Internet es el TCP/IP.

Una de las redes más populares es la Ethernet, que tiene una topología de bus y el acceso al medio de transmisión es muy sencillo: cuando una estación quiere comunicarse con otra mira que no haya tráfico en la red y si no lo hay pone la información que quiere enviar en diversos paquetes. Existe la posibilidad de que dos estaciones envíen a la vez paquetes y es este caso se producen colisiones. Cuando las estaciones detentan una colisión, interrumpen la emisión y después de esperar un tiempo aleatorio repiten el proceso.

3.1.1 Topologías de red.

Cuando hablamos de topología de una red, hablamos de su configuración. Esta configuración recoge tres campos: físico, eléctrico y lógico.

El nivel físico y eléctrico se puede entender como la configuración del cableado entre máquinas o dispositivos de control o conmutación. Cuando hablamos de la configuración lógica tenemos que pensar en como se trata la información dentro de nuestra red, como se dirige de un sitio a otro o como la recoge cada estación.

Así pues, para ver más claro como se pueden configurar las redes vamos a explicar de manera sencilla cada una de las posibles formas que pueden tomar.

3.1.1.1 Mallada.

Este tipo de redes son las más caras, pero a su vez son las más flexibles. Vienen caracterizadas por encontrar caminos entre estaciones muy rápidamente, pero hay que tener en cuenta que para N nodos, necesitamos $N-1$ enlaces, teniendo pues en total $(N(N-1))/2$ enlaces.

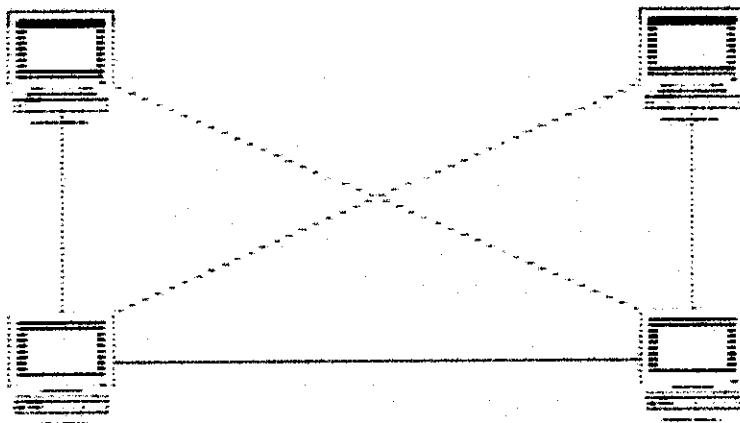


Figura 3.1 Red mallada

3.1.1.2 Bus.

Tenemos un enlace por cada nodo, y estos se conectan a un enlace que une todas las estaciones. Típica configuración que usa Ethernet.

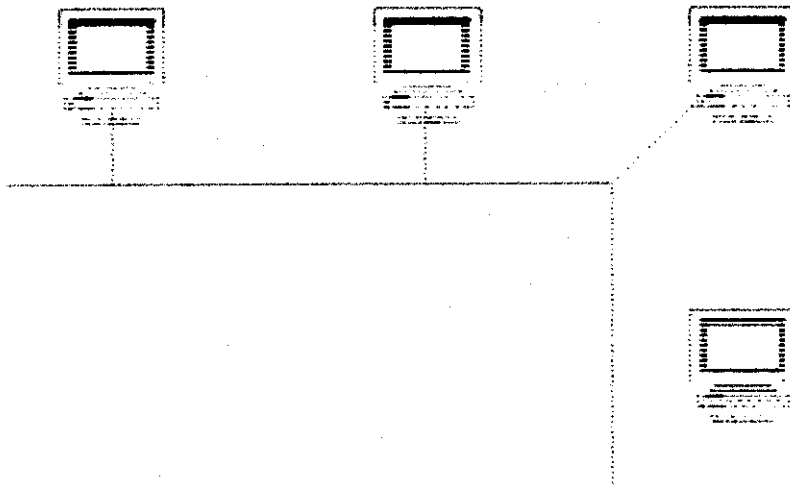


Figura 3.2 Red con topología de bus

3.1.1.3 Anillo.

Cada nodo tiene dos enlaces, puesto que la información siempre le vendrá de un lado y la enviará hacia el otro. Un ejemplo de esta configuración la encontramos en un Token Ring.

3.1.1.5 Árbol.

Todas las estaciones cuelgan de un ordenador central y se conectan entre ellas a través de los hubs que haya instalados.

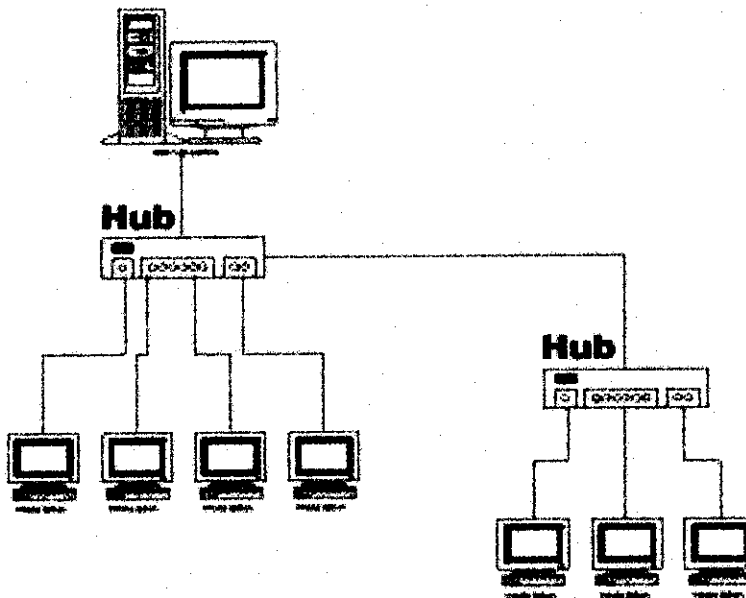


Figura 3.5 Red en árbol

3.1.2 Cableado.

Podemos clasificar los tipos de cables basándonos en estos factores:

- Velocidad de Transmisión.
- Longitud máxima.

- Protección contra interferencias.

Los tres tipos de cable más populares son:

- **Par trenzado.** El cable de par trenzado es exactamente eso: dos hilos conductores de cobre aislados y trenzados entre sí, y en la mayoría de los casos, cubiertos por una malla protectora.
- **Cable coaxial.** Se utiliza generalmente para señales de televisión. Consiste en un núcleo de cobre rodeado por una capa aislante. Existen diversos tipos de cable coaxial, que puede ser grueso o fino. Los tramos largos de una red pueden implementarse con cable grueso; sin embargo es más caro que el fino, que permite cubrir distancias menores.
- **Cable de fibra óptica.** El cable de fibra óptica transmite señales de datos mediante luz. La luz modulada pasa por un conductor de vidrio, rodeada por una capa reflectante. Este conjunto está envuelto en una capa protectora.

3.1.3 Características de una red.

Las características o beneficios de una red de computadoras las detallamos a continuación.

3.1.3.1 Compartición de archivos y programas.

Las versiones de red de muchos paquetes populares de software están disponibles, con un ahorro considerable costos cuando se compara con la compra de copias de licencias

individuales. Los archivos de programas y datos se almacenan en un servidor de archivos y acceden a ellos muchos usuarios de red.

3.1.3.2 Compartición de recursos de red.

Los recursos de red incluyen impresoras, trazadores gráficos y dispositivos de almacenamiento. La red ofrece un enlace de comunicación que permite a los usuarios compartir estos dispositivos.

3.1.3.3 Compartición de bases de datos.

Un sistema gestor de base de datos es una aplicación ideal para una red. Una característica de red llamada bloque de registro permite que múltiples usuarios accedan simultáneamente a un archivo sin corromper los datos. El bloqueo de registro asegura que dos usuarios no editen el mismo registro a la vez.

3.1.3.4 Expansión económica de la base del PC.

Las redes proporcionan una forma económica de aumentar el número de computadoras en una organización. Se pueden enlazar económicas estaciones de trabajo sin disco a una red que utilice la unidad de disco rígido del servidor para el arranque y el almacenamiento.

3.1.3.9 Mejora de la estructura corporativa.

Las redes pueden cambiar la estructura y gestión de una organización. Los usuarios que trabajan en un departamento concreto y para un directorio específico ya no tienen la necesidad de estar en la misma área física. La red los une con sus directores y compañeros de departamento.

3.1.4 Métodos de Transmisión.

3.1.4.1 Señalización analógica.

Son voltajes continuamente variables u ondas que pueden representar un número finito de valores dentro del rango del dispositivo que mide genera o transmite señales; el sistema telefónico usa líneas analógicas conmutadas para proporcionar la conmutación de redes. Una señal analógica es una forma de energía propagada como una onda de sonido que vibra en el aire y viaja a través de él.

3.1.4.2 Señalización digital.

El nivel de tensión en un canal va entre un estado alto y otro bajo. Principalmente se caracteriza por presentar la información codificada como los dígitos binarios de 0 y 1 en la señal transmitida, la señal digital puede ser voz, videos imágenes pueden transmitir eficazmente mediante su codificación como valores binarios y con la transmisión de estos

valores como pulsos eléctricos. El voltaje varía entre un estado alto y otro bajo, es decir un voltaje positivo indica un 1 binario y un voltaje bajo un 0 binario.

3.1.5 Modos de transmisión.

- a) **Circuitos Simple**, es una transmisión en un solo sentido.
- b) **Circuito Semiduplex**, transmisión bidireccional pero en un solo sentido a la vez, es decir cuando el uno envía el otro espera hasta que termine el mensaje y pueda responder.
- c) **Circuito Fullduplex**, transmisión bidireccional en dos sentidos, es decir, dos estaciones pueden estar enviándose información al mismo tiempo.

3.1.6 Componentes de una red.

Una red de computadoras consta tanto de hardware como de software. El hardware incluye tarjetas de interfaz de red, y el cable que las une. Los componentes software incluye sistemas operativos, protocolo de comunicación y controladores de la tarjeta de la interfaz de red del servidor.

3.1.6.1 Sistema operativo de red.

En una red par a par, cada nodo de la red ejecuta un sistema operativo y con el soporte de conexión de red incorporado, el cual permite que los usuarios compartan archivos y periféricos. Normalmente también se incluyen características de seguridad y de gestión. El

sistema operativo de red para una red dedicada, se ejecuta en servidores o estaciones de trabajo autónomos que ejecutan el software del cliente que se comunica con el servidor.

3.1.6.2 Servidores.

Un servidor ofrece los siguientes servicios a los usuarios de la red. Los sistemas operativos de red modulares como NetWare de Novel pueden proporcionar alguno de o todos estos servicios en uno o más servidores, dependen de qué componentes modulares elija instalar el administrador.

- **Servidor de archivos.** Proporciona servicios de almacenamiento y recuperación de archivos, incluidas las utilidades de seguridad que controlan los derechos de acceso a los archivos.
- **Servicio de correo electrónico o pasarela.** Ofrece servicios de correo electrónico de corporación extensa o local y traducción entre distintos sistemas de correo.
- **Servicios de comunicaciones.** Permite los servicios de conexión en sistemas de computadora central o de minicomputadora, o en sistema y redes de computadora remotas por medio de enlaces de área extensa.
- **Servidores de base de datos.** Un servidor dedicado que gestiona las peticiones y respuestas del usuario de la base de datos.
- **Servidor de archivos.** Un sistema dedicado o copias de seguridad y almacenamiento de archivos en la red.

3.1.6.3 Sistemas clientes (Nodos o estaciones de trabajo).

Los sistemas clientes se unen a la red por medio de tarjetas de la interfaz de red. El sistema operativo que le ejecuta en la estación de trabajo puede incluir el software ya incorporado para soportar las tarjetas, o puede ser necesario cargar el software del cliente.

3.1.6.4 Tarjetas de la interfaz de red.

Las estaciones de trabajo de la computadora necesitan la instalación de una tarjeta de la interfaz de red. Hay algunas computadoras que ya incorporan estas interfaces.

3.1.6.5 Sistema de cableado.

El sistema de cableado de red es el medio que conectan juntos servidores y estaciones de trabajo. No es necesario el cable en las redes inalámbricas por radio o infrarrojos.

3.1.6.6 Recursos y periféricos compartidos.

Los recursos y periféricos compartidos incluyen dispositivos de almacenamiento unidos al servidor como unidades de disco óptico, impresoras, trazadores gráficos y otros equipos disponibles que utiliza cualquier usuario autorizado de la red.

3.1.7 Formas de Comunicación.

Existen dos tipos o formas de comunicación: comunicación síncrona y comunicación asíncrona.

3.1.7.1 Comunicación Síncrona.

En esta comunicación el emisor y el receptor están sincronizados con un reloj, así se sabe cada que tiempo recoger un bloque de bits. Entonces se transfiere la información en paquetes de bits que se sincronizan con señales de reloj, se usan caracteres especiales para empezar la sincronización y periódicamente se comprueba su precisión.

3.1.7.2 Comunicación Asíncrona.

La información que transmite como un flujo de bits en donde un carácter se codifica como un conjunto de bits, la información se envía carácter a carácter, que tiene un bit de principio de carácter y otro de parada para separar cada carácter. Se usa un bit de paridad para la detección y corrección de errores. La comunicación asíncrona implica que no un mecanismo, como un reloj, sino que utiliza un modelo de inicio-parada de transmisión, lo cual implica que las transmisiones vuelven a empezar para cada carácter nuevo.

3.2 Estudio y Análisis de la Red Existente.

3.2.1 Sistema Operativo de red existente.

La Cooperativa de Ahorro y Crédito "O.S.C.U.S." Ltda. Se encuentra conectada a una red LAN, en la actualidad esta implantando una amplia red de tipo WAN compuesta por varias LAN en cada agencia de la Institución, en cuanto tiene que ver al S.O. de red, se está implantando Unix de SUN Solaris.

3.2.1.1 Tecnología Ethernet.

Ethernet es el nombre de una de las redes de área local más populares hoy en día inventada por Xerox PARC a principios de los 70's. La versión a continuación descrita fue estandarizada por Xerox Corporation, Intel Corporation y Digital Equipment Corporation en 1978.

La tecnología ethernet consiste fundamentalmente en un cable coaxial llamado ether de aproximadamente <Imagen> pulgada de diámetro y hasta 500 metros de longitud, estos pueden ser extendidos por medio de dispositivos llamados repetidores que duplican señales eléctricas de un cable a otro.

Sólo dos repetidores pueden ser usados entre dos computadoras, por lo que la longitud máxima de un ethernet es bastante moderada (1500 metros)

3.2.1.2 Sistema Operativo de Red Unix

Al igual que otros sistemas operativos, el sistema operativo UNIX es un conjunto de programas de utilidad y un conjunto de instrumentos que permiten al usuario conectar y utilizar esas utilidades para construir sistemas y aplicaciones.

Al conjunto de programas que componen UNIX y que se encargan de proporcionar los recursos del sistema y de coordinar todos los detalles internos de la computadora se les llama en conjunto SISTEMA OPERATIVO o KERNEL.

UNIX se caracteriza por ser un sistema "MULTIUSUARIO" porque permite que dos o más personas utilicen la computadora al mismo tiempo.

Los Usuarios se comunican con el Kernel a través de otro programa conocido como el shell. El shell es un "Intérprete de Línea de Comandos" que traduce los comandos tecleados por el usuario y los convierte en instrucciones que puede entender el Kernel.

a) Características principales del Sistema Operativo UNIX.

Los siguientes conceptos son comunes para todos los sistemas UNIX, por lo cual se puede afirmar que éstos componen las características principales de UNIX.

KERNEL : Este es el componente principal del sistema operativo. Se encarga de asignar tareas y manejar el almacenamiento de datos. El usuario rara vez opera directamente con el kernel, que es la parte residente en memoria del sistema operativo.

SHELL : Esta es la utilidad que procesa las peticiones de los usuarios. Cuando alguien teclea un comando en la terminal, el shell interpreta el comando y llama el programa deseado. También es un lenguaje de programación de alto nivel que puede utilizarse en la combinación de programas de utilidad para crear aplicaciones completas.

El shell puede soportar múltiples usuarios, múltiples tareas, y múltiples interfaces para sí mismo. Los dos shells más populares son el BourneShell (System V) y el Cshell (BSD Unix), debido a que usuarios diferentes pueden usar diferentes shells al mismo tiempo, entonces el sistema puede aparecer diferente para usuarios diferentes. Existe otro shell conocido como KornShell (así llamado en honor de su diseñador), que es muy popular entre los programadores.

PROGRAMAS DE UTILIDAD (UTILERIAS) : El Sistema Operativo UNIX incluye una gran variedad de programas de utilidad que pueden ser fácilmente adaptadas para realizar tareas específicas. Estas utilerías son flexibles, adaptables, portables y modulares, y pueden ser usadas junto con filtros y redireccionamientos para hacerlos más poderosos.

SISTEMA MULTIUSUARIOS : Dependiendo del equipo disponible, un UNIX puede soportar desde uno hasta más de 100 usuarios, ejecutando cada uno de ellos un conjunto diferente de programas.

SISTEMA MULTITAREAS : UNIX permite la realización de más de una tarea a la vez. Pueden ejecutarse varias tareas en su interior, mientras se presta toda la atención al programa desplegado en la terminal.

FTP (File Transfer Protocol) permite la transferencia de ficheros entre maquinas. Otras aplicaciones incluyen sendmail, sistema para enviar y recibir correo electrónico usando el protocolo SMTP; Sistemas de noticias (news) basados en NNTP como C-News e INN; telnet, rlogin y rsh, que permiten iniciar una sesión y ejecutar ordenes en otras maquinas de la red; y finger, que permite obtener información de otros usuarios de Internet. Hay literalmente cientos de aplicaciones basadas en TCP/IP y protocolos disponibles por la red.

Todos los lectores de correo y noticias están disponibles para Unix, como elm, pine, rn, nn y tin. Sea cual sea tus preferencias, se puede configurar un sistema Unix para enviar y recibir correo electrónico y noticias por todo el mundo.

Si se tiene experiencia con aplicaciones TCP/IP en otros sistemas, Unix será muy familiar. El sistema proporciona el interface estándar de programación por 'sockets', lo que virtualmente permite que cualquier programa que use TCP/IP pueda ser llevado a Unix. El servidor Unix de X también soporta TCP/IP, permitiendo ver aplicaciones que están corriendo en otros sistemas sobre tu pantalla.

UUCP (UNIX-to-UNIX Copy) es un viejo mecanismo usado para transferir ficheros, correo electrónico y noticias entre maquinas UNIX. Clásicamente las maquinas UUCP conectan entre ellas mediante líneas telefónicas y módem, pero UUCP es capaz de funcionar también sobre una red TCP/IP. Si no se tiene acceso a una red TCP/IP o a un servidor SLIP, se puede configurar un sistema para enviar y recibir ficheros y correo electrónico usando UUCP.

c) Unix vs. Otros Sistemas Operativos.

Están surgiendo un gran número de sistemas operativos avanzados en el mundo del PC. Concretamente, OS/2 de IBM y Windows NT de Microsoft comienzan a tener popularidad a medida que los usuarios de MS-DOS migran a ellos.

Ambos, OS/2 y Windows NT son sistemas operativos completamente multitarea, muy parecidos a Unix. Técnicamente, OS/2, Windows NT y Unix son bastante similares, soportan aproximadamente las mismas características en términos de interfaz de usuario, redes, seguridad, y demás.

¿Que hace a UNIX tan importante?. No solo es el sistema operativo más popular para máquinas multiusuario, también es la base de la mayoría del mundo del software de libre distribución. Si tiene acceso a Internet, casi todo el software de libre distribución disponible está específicamente escrito para sistemas UNIX. (Internet en si está profundamente basada en UNIX.).

Hay muchas implementaciones de UNIX, de muchos vendedores, y ni una sola organización es responsable de su distribución. Hay un gran pulso en la comunidad UNIX por la estandarización en forma de sistemas abiertos, pero ninguna corporación controla este diseño.

Por eso, ningún vendedor (o como parece, ningún Hacker) puede adoptar estos estándares en una implementación de UNIX.

Por otro lado, OS/2 y Windows NT son sistemas propietarios. El interface y diseño están controlados por una sola corporación, y solo esa corporación puede implementar ese diseño. De alguna forma, este tipo de organización es beneficiosa: establece un estándar

estricto para la programación y la interfaz de usuario distinto al encontrado incluso en la comunidad de sistemas abiertos. OS/2 es OS/2 vaya donde vaya lo mismo ocurre con Windows NT.

Sin embargo, el interface UNIX esta constantemente desarrollándose y cambiando. Varias organizaciones están intentando estandarizar el modelo de programación, pero la tarea es muy difícil. Unix, en particular, es en su mayoría compatible con el estándar POSIX para el interface de programación UNIX, a medida que pase el tiempo, se espera que el sistema se adhiera a otros estándares, pero la estandarización no es la etapa primaria en la comunidad de desarrollo de Unix.

3.2.1.3 Comando Telnet.

El comando telnet se comunica con otro host usando el protocolo TELNET. Si telnet es llamado sin argumentos, este entrará al modo de comando, indicado por su prompt (ejemplo, telnet >). En este modo, este acepta y ejecuta los comandos. Si este es invocado con argumentos, este representa un comando abierto con estos argumentos. Una vez que se ha establecido la conexión, telnet estará en el modo de entrada. En el modo de entrada, el texto escrito es inmediatamente enviado al host remoto para ser procesado.

3.2.1.4 Estándares de comunicación.

a) El modelo OSI

El modelo OSI (Open System Interconnection) es utilizado por prácticamente la totalidad de las redes del mundo. Este modelo fue creado por el ISO (Organización Internacional de Normalización), y consiste en siete niveles o capas donde cada una de ellas define las funciones que deben proporcionar los protocolos con el propósito de intercambiar información entre varios sistemas. Esta clasificación permite que cada protocolo se desarrolle con una finalidad determinada, lo cual simplifica el proceso de desarrollo e implementación. Cada nivel depende de los que están por debajo de él, y a su vez proporciona alguna funcionalidad a los niveles superiores. Los siete niveles del modelo OSI son los siguientes:

a.1 Aplicación:

El nivel de aplicación es el destino final de los datos donde se proporcionan los servicios al usuario.

a.2 Presentación:

Se convierten e interpretan los datos que se utilizarán en el nivel de aplicación.

a.3 Sesión:

Encargado de ciertos aspectos de la comunicación como el control de los tiempos.

a.4 Transporte:

Transporta la información de una manera fiable para que llegue correctamente a su destino.

a.5 Red:

Nivel encargado de encaminar los datos hacia su destino eligiendo la ruta más efectiva.

a.6 Enlace:

Enlace de datos. Controla el flujo de los mismos, la sincronización y los errores que puedan producirse.

a.7 Físico:

Se encarga de los aspectos físicos de la conexión, tales como el medio de transmisión o el hardware.

3.3 Protocolos de Comunicación

Los protocolos de comunicaciones definen las normas que posibilitan que se establezca una comunicación entre varios equipos o dispositivos, ya que estos equipos pueden ser diferentes entre sí. Una interfaz, sin embargo, es el encargado de la conexión física entre los equipos, definiendo las normas para las características eléctricas y mecánicas de la conexión.

Exceptuando a los routers cualquier ordenador conectado a Internet y, por tanto, capaz de compartir información con otro ordenador se conoce con el nombre de host (anfitrión). Un host debe identificarse de alguna manera que lo distinga de los demás para poder recibir o enviar datos. Para ello todos los ordenadores conectados a Internet disponen de una dirección única y exclusiva. Esta dirección, conocida como dirección de Internet o dirección IP, es un número de 32 bit que generalmente se representa en cuatro grupos de 8 bit cada uno separados por puntos y en base decimal.

3.3.1 Protocolo existente TCP/IP.

TCP/IP es el protocolo común utilizado por todos los ordenadores que forman parte de la Cooperativa de Ahorro y Crédito "O.S.C.U.S." Ltda., también existe el protocolo NETBEUI que es un protocolo generalizado del S.O. Windows.

Hay que tener en cuenta que en Internet se encuentran conectados ordenadores de clases muy diferentes y con hardware y software incompatibles en muchos casos, además de todos los medios y formas posibles de conexión. Aquí se encuentra una de las grandes ventajas del TCP/IP, pues este protocolo se encargará de que la comunicación entre todos sea posible. TCP/IP es compatible con cualquier sistema operativo y con cualquier tipo de hardware.

TCP/IP no es un único protocolo, sino que es en realidad lo que se conoce con este nombre es un conjunto de protocolos que cubren los distintos niveles del modelo OSI. Los dos protocolos más importantes son el TCP (Transmission Control Protocol) y el IP (Internet Protocol), que son los que dan nombre al conjunto. En Internet se diferencian cuatro niveles o capas en las que se agrupan los protocolos, y que se relacionan con los niveles OSI de la siguiente manera:

- **Aplicación:** Se corresponde con los niveles OSI de aplicación, presentación y sesión. Aquí se incluyen protocolos destinados a proporcionar servicios, tales como correo electrónico (SMTP), transferencia de ficheros (FTP), conexión remota (TELNET) y otros más recientes como el protocolo HTTP (Hypertext Transfer Protocol).

3.3.1.1 Protocolos alternativos a TCP.

TCP es el protocolo más utilizado para el nivel de transporte en Internet, pero además de éste existen otros protocolos que pueden ser más convenientes en determinadas ocasiones. Tal es el caso de UDP y ICMP.

a) UDP (User Datagram Protocol)

El protocolo de datagramas de usuario (UDP) puede ser la alternativa al TCP en algunos casos en los que no sea necesario el gran nivel de complejidad proporcionado por el TCP. Puesto que UDP no admite numeración de los datagramas, éste protocolo se utiliza principalmente cuando el orden en que se reciben los mismos no es un factor fundamental, o también cuando se quiere enviar información de poco tamaño que cabe en un único datagrama.

Cuando se utiliza UDP la garantía de que un paquete llegue a su destino es mucho menor que con TCP debido a que no se utilizan las señales de confirmación. Por todas estas características la cabecera del UDP es bastante menor en tamaño que la de TCP. Esta simplificación resulta en una mayor eficiencia en determinadas ocasiones.

Un ejemplo típico de una situación en la que se utiliza el UDP es cuando se pretende conectar con un ordenador de la red, utilizando para ello el nombre del sistema. Este nombre tendrá que ser convertido a la dirección IP que le corresponde y, por tanto, tendrá que ser enviado a algún servidor que posea la base de datos necesaria para efectuar la conversión. En este caso es mucho más conveniente el uso de UDP.

un rendimiento en el envío de datos de 1544Mbits/seg, aunque se tratan de implementar velocidades aún mayores.

La sobrecarga del X.25 se suele comparar con frecuencia con Frame Relay, por ejemplo, en X.25, cada nodo del camino del paquete debe recibir la totalidad del paquete y realizar un test de errores sobre el mismo, antes de enviarlo. Los nodos de Frame Relay simplemente consultan con el encabezamiento del paquete cual es su dirección de destino, e inmediatamente lo envía, en algunos casos incluso antes de haberlo recibido por completo. Frame Relay no requiere el uso de las tablas de estado que el X.25 emplea en cada nodo intermedio para el trato de la gestión, el control del flujo y la comprobación de errores. Los nodos finales detectan los fragmentos perdidos y solicitan su retransmisión.

3.4.5.7 Cómo se accede a los servicios de conmutación.

El mayor problema de cualquier servicio de conmutación es cómo ganar acceso a la facilidad de la compañía de telecomunicaciones que proporciona el servicio. Las conexiones entre dos usuarios cualesquiera, uno de una LAN local y otro de una LAN remota, no están previstas en las redes públicas del emplazamiento del cliente, pero sí en el equipo de conmutación de la compañía de telecomunicaciones. Los clientes deben alquilar líneas dedicadas de tipo T1o T3, con suficiente capacidad para comunicar el tráfico de datos locales multiplexados a la facilidad de conmutación de la compañía de telecomunicaciones, o usar líneas de conmutación como Switched-56 o las interfaces ISDN para acceder a la facilidad.

Cuando la demanda crezca, las redes de datos llegarán a ser tan frecuentes y transparentes, como lo son las redes de voz. La Red óptica síncrona (SONET, Synchronous Optical Network) es una red transparente que posibilita esto. Define un estándar de multiplexación para la transmisión a través del cable de fibra óptica con velocidades en el rango entre 51 Megabits/seg. y 2488 Mbits/seg.

El Modo de transferencia asíncrono (ATM) transporta datos en las redes SONET, Multiplexa células de datos (paquetes de tamaño fijo) que proceden de diversas fuentes de red física (SONET). ATM proporciona trayectos de comunicación virtual a través de la red SONET. Un circuito virtual orientado a la conexión entre dos puntos puede alcanzar velocidades de 45 Mbits/s a 622 Mbits/s, aunque la limitación habitual de las compañías de telecomunicaciones es de 45Mbits/s. La conmutación ATM es habitual en el nivel LAN de los concentradores cableados. También es frecuente su uso como una técnica de conmutación de las redes de telecomunicaciones internacionales y globales.

Por encima de ATM está la Red de servicios integrados de banda ancha (BISDN, Broadband-Integrated Services Digital Network). ATM es la base de BISDN. B-ISDN es un sucesor de ISDN que define cómo proporcionar la comunicación entre casas y oficinas a base de conmutación de circuitos, en incrementos de 64Kbits/seg. B-ISDN usa la tecnología ATM y la red física SONET para difundir datos con una velocidad de transferencia entre 155Mbits/seg y 622Mbits/seg, junto con diversos servicios al cliente. Otra oferta de las posibles es el Servicio de conmutación de datos Multimegabits (SMDS, Switched Multimegabit Data Service), que fue desarrollado por Bellcore, una división de las compañías

regionales de operaciones Bell. SMDS se diseñó para que proporcione redes de datos públicas y privadas dentro de las áreas metropolitanas.

3.4.5.9 Entradas Relacionales.

Area local para acceso y transporte (LATA); AT&T; compañías regionales de operaciones Bell (RBOCs); línea alquilada; Red pública de conmutación de datos; redes(PSDN); Redes públicas de datos (PDNs); Servicio de conmutación de circuitos; Servicios T1/T3; y telecomunicaciones.

CAPITULO CUARTO

4 MOTORES DE BASES DE DATOS.

4.1 Bases de datos relacionales.

Las BDMS relacionales operan sobre archivos o tablas de datos y no sobre datos individuales contenidos en los archivos. Las tablas son medios de representar la información de forma compacta y de fácil acceso.

Estas tablas están formadas por filas y columnas. Las filas de una tabla equivalen a los registros - contienen los valores de los objetos o entidades descritas -, y las columnas lo son a los campos – atributos de los objetos o entidades descritas -.

Otros conceptos importantes:

- Dominio: Valores que puede tomar un campo.
- Clave: Referencia que se utiliza para identificar los objetos de forma única. Está formada por uno o varios campos.
- Clave primaria o principal: Es la mínima en cuanto a número de campos que la componen.

4.1.1 Sistema de Gestión de archivos.

Un sistema de gestión de archivos, generalmente proporcionado por el fabricante del computador como parte del sistema operativo, llevaba la cuenta de los nombres y ubicaciones

de los archivos. El sistema de gestión de archivo básicamente no tenía un modelo de datos; no sabía nada acerca de los contenidos internos de los archivos. Para el sistema de gestión de archivos, un archivo que contuviera un documento de procesamiento de textos y un archivo que contuviera datos de nóminas aparecían igual.

El conocimiento acerca del contenido de un archivo qué datos contuviera y como estuvieran organizados estaba incorporado a los programas de aplicación que utilizaban el archivo, tal como se muestra en la Figura 4.1 En esta aplicación de nóminas, cada uno de los programas COBOL que procesaban el archivo maestro de empleados contenía una descripción de archivo (DA) que describía la composición de los datos en el archivo. Si la estructura de los datos cambiaba – por ejemplo, si un ítem adicional de datos fuera a ser almacenado por cada empleado – todos los programas que accedían al archivo tenían que ser modificados. Como el número de programas y archivos crecía con el tiempo, todo el esfuerzo de procesamiento de datos de un departamento se perdía en mantener aplicaciones existentes en lugar de desarrollar otras nuevas.

Los problemas de mantener grandes sistemas basados en archivos condujo a finales de los setenta al desarrollo de los sistemas de gestión de base de datos. La idea detrás de estos sistemas era sencilla: tomar la definición de los contenidos de un archivo y la estructura de los programas individuales, y almacenarla, junto con los datos, en una base de datos. Utilizando la información de la base de datos, el DBMS que la controlaba podría tomar un papel mucho más activo en la gestión de los datos y en los cambios a la estructura de la base de datos.

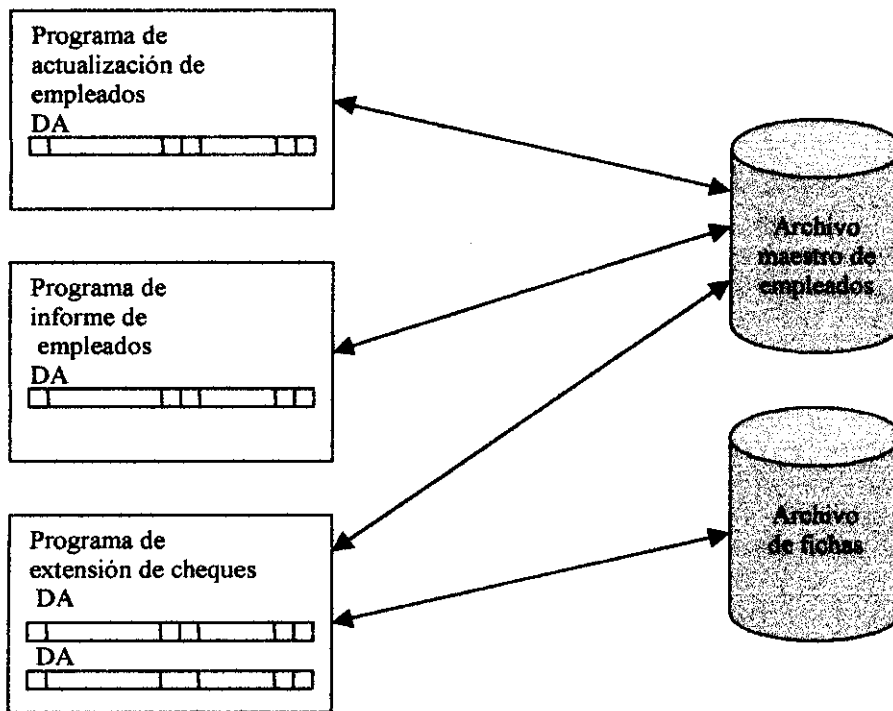


Figura 4.1 Aplicación de nómina utilizando un sistema de gestión de archivos

4.1.2 Diseño de una base de datos.

El paso preliminar es la planificación del tipo de información que se quiere almacenar, atendiendo a dos premisas: la información disponible y la información que necesitamos.

Diseño de la estructura de una tabla: descripción de los campos y valores o datos que contendrán.

- Campos: Tipos de datos que componen una tabla.
- Definición del campo: Debemos especificar lo siguiente:

-
- Nombre del campo.
 - Tipos de datos: caracteres, valores numéricos, fechas, etc.
 - Anchura del campo.
-
- Fases de diseño de una base de datos: Es importante dividir los datos en varias tablas independientes interrelacionadas.
 - Definición de los datos. Lista de campos.
 - Refinamiento de los datos: Refinamos la lista inicial de datos de manera que contengan una descripción precisa de los mismos.
 - Consideración de las relaciones entre tablas:
 - Debemos introducir un campo común en las tablas relacionadas.
 - Los tipos de relaciones son:
 - Uno a uno.
 - Uno a varios.
 - Varios a vario

4.2 Análisis de la DBMS existente.

Universal Server de Informix es el motor de bases de datos que se encuentra utilizando la Cooperativa de Ahorro y Crédito "OSCUS LTDA."

Universal Server de Informix es un motor de bases de datos avanzado, con arquitectura cliente servidor concretada en la arquitectura dinámica escalable DSA, soportada por bases de datos

- Arquitectura fichero-servidor. Se utiliza un servicio de ficheros remotos para leer y escribir en las páginas de la base de datos. El cliente soporta la mayoría de las funciones.
- Hoy por hoy existen entornos de desarrollo estables que aplican la orientación a objetos para la creación de aplicaciones. Sin embargo, los SGBDOO son adecuados para tratar tipos de datos complejos que se estructuran según la metodología POO y son manipulados mediante lenguajes de programación orientados a objetos. Las aplicaciones que se desarrollan para estos sistemas se basan en la utilización de punteros a través de los cuales se accede a los datos y su mantenimiento resulta muy caro.

c) **Bases de datos Relacional-objeto.**

Como resultado de las carencias que las bases de datos orientadas a objetos presentan en cuanto a la gestión de contenidos no estructurados, desde informix software inc. Se nos propone una solución que se basa en la gestión de datos complejos de forma inteligente, permitiendo al servidor la toma de decisiones para optimizar el acceso a dichos datos, no en forma de BLOBs, sino como objetos nativos del sistema.

Las definiciones de funciones en el SGBDRO incluyen instrucciones sobre el coste de E/S y de CPU de la función, lo cual permite escoger un plan de consulta más adecuado. Las funciones se ejecutan en el servidor, cerca de los datos, lo que aumenta su rendimiento porque no hay que sacar los datos del servidor para manipularlos.

El desarrollo de aplicaciones que se basan en la información que contienen se hace posible cuando el sistema de gestión de bases de datos integra las estructuras de los datos (objetos), así como las operaciones que con ellas se pueden realizar, la forma de indexarlos, la forma de optimizar las consultas sobre ellos, etc.

Con universal Server los clientes pueden definir sus aplicaciones estableciendo la estructura de los datos que necesiten y su funcionamiento. Se proporciona al sistema de gestión la capacidad de gestionar los nuevos tipos de datos de forma que sea el contenido de la información el que defina que todas sus partes estén interrelacionadas.

Para que nos aclaremos, los datos se tratan como extensiones, donde se integran texto, imágenes, videos, páginas HTML, datos de las operaciones que con ellos se pueden realizar, entre otras informaciones que puedan estar relacionadas.

d) Datablades: Objetos y Funciones.

Los DataBlades definen la estructura de un objeto de información, además de las funciones asociadas a ese objeto y los métodos de acceso para su recuperación y visualización. Esta es la definición que Informix da a DtaBlade, y también puntualiza sus ventajas:

- El sistema de gestión podrá optimizar cualquier consulta que haga referencia al objeto de información, permitiendo que el rendimiento global del sistema no se vea afectado sea cual sea la complejidad de la estructura de los datos con los que trabajemos.
- A la hora de desarrollar aplicaciones basadas en los objetos, no es necesario de utilizar nuevos estándares ni lenguajes de programación especiales.

- Se pueden integrar perfectamente unos con otros permitiendo así desarrollar aplicaciones multimedia basadas en el contenido de la información, dinámicas e interactivas. En un único Universal Server se pueden cargar simultáneamente muchos módulos DataBlade y los usuarios pueden extender e incluso sustituir las funciones de estos módulos, según sus necesidades.

e) Metodología POO en los DataBlades.

La concepción de DataBlade implica que universal server tenga las siguientes características:

- Universal Server permite la definición de nuevos tipos de datos, totalmente nuevos o basados en otros ya existentes; un tipo puede incluir otros tipos y hacer referencia a los mismos como parte de su definición. Así, los usuarios pueden crear tipos diferentes, conjuntos, multiconjuntos, listas matrices, identificadores de objetos (referencias a otros objetos), que además podrán formar parte de otros tipos, y otros, según sus necesidades y completamente adaptados a la información que requieran gestionar. Esto se traduce en la posibilidad de creación de nuestras propias estructuras de datos, basadas en otros objetos ya definidos de forma que hereden sus propiedades y funciones, o bien empezar de cero, adaptándose a nuevos datos.
- Los tipos basados en otros tipos pueden copiar su definición o heredarla, se soporta la herencia, permitiendo que un tipo recoja partes de sus estructura y funcionamiento de otro tipo predefinido.
- Pueden heredarse la estructura. las reglas (triggers en Universal Server), los operadores, las funciones y las expresiones de agrupación, manteniendo así las ventajas de la OO.

- Los tipos están definidos completamente, es decir tipos diferentes no pueden compararse directamente, debiéndose realizar un cast para poder realizar la comparación, se soporta así la definición de funciones de conversión.
- Las funciones definidas por el usuario en Universal Server se corresponden con los llamados métodos en OO. Las funciones pueden ser implementadas en lenguaje C, en Java o en SQL y están limitadas por las capacidades del lenguaje utilizado, de esta forma, si se utiliza Java, se podrá realizar cualquier tarea que Java permita. Un usuario puede definir cualquier función para ser ejecutada en el servidor o en el cliente. En la definición de la función se instruirá al sistema sobre los costes de E/S y de CPU para que las consultas donde dicha función aparezca puedan ser optimizadas.
- Se admite la sobrecarga, lo que en OO equivale a polimorfismo. Las funciones se componen de un nombre y una lista de parámetros, los cuales, junto con sus tipos, forman el identificativo de la función (signature). La sobrecarga de funciones permite que pueda haber varias funciones con el mismo nombre pero con distintos identificativos. Universal Server determina que función debe ser llamada durante la ejecución examinando su identificativo.
- Se consigue la encapsulación de objetos, permitiendo a sus diseñadores modificar su estructura interna o la de las funciones que operan sobre los objetos sin que se aprecie en las aplicaciones.

f) Seguridad.

- Control sobre los privilegios de conexión, recursos de base de datos y DBA.
- Control sobre el acceso de datos hasta nivel del campo.

- Facilidades para la auditoría de todas las operaciones exitosas/no exitosas de I/O del sistema, de otorgamiento/revocación de permisos, consulta, y actualización.

g) Soporte de Transacciones.

- COMMIT WORK y ROLLBACK WORK.
- Puntos de conservación con nombre.
- Consistencia de lectura a nivel de instrucción y transacción.
- Transacciones solo lectura (read only).

h) Tolerancia a Fallas

- Respaldo en línea por: archivo, espacio de tabla o base de datos.
- Recuperación en línea por: archivo, espacio de tablas.
- Hora de recuperación definible.
- Protección completa hacia adelante (roll-forward) prácticamente sin sobrecarga.
- Exportación, importación: incremental, acumulativa y total.

i) Otras características.

Las características que definen a Universal Server, además de las mencionadas anteriormente, pretenden conseguir la escalabilidad y la flexibilidad; como características añadidas podemos mencionar:

-
- Soporte de objetos de gran tamaño. Los objetos que exceden del tamaño que el servidor soporte, pueden almacenarse como ficheros externos o internos en la BD, pudiéndose recuperar en ambos casos en el supuesto de un posible fallo de la máquina.
 - Como utilidad del sistema, en máquinas multiprocesador, el cargador en paralelo gestiona tipos de datos abstractos y acelera el proceso de la carga de la BD que se añaden.
 - Para facilitar la extensibilidad de las herramientas del sistema de base de datos, se utilizan las siguientes interfaces.
 - Utilización de SQL embebido en C.
 - Librerías de clase de C++, para soportar el desarrollo de aplicaciones en este lenguaje.
 - Interfaz OLE para crear controles personalizados OLE (OCX) a la vez que los módulos DataBlade, utilizando la librería de C del nivel de llamadas para interactuar con Universal Server. Los OCX se distribuyen con módulos de ejecución de aplicaciones cliente y aparecen como iconos u operaciones en la aplicación.
 - Interfaz para Java. Una aplicación cliente de Java podrá descargar applets de java de distintas redes; el applet se ejecutará entonces como la aplicación cliente. Además, se soporta la especificación de funciones de servidor en Java.
 - Se permite la creación de funciones de agregación (suma, media,.....) con lo que se intenta facilitar la búsqueda de los datos. Estas funciones nuevas pueden ejecutarse en paralelo, en máquinas multiprocesador. La versión actual de SQL incluye algunas funciones como sum, average, etc. Sin embargo no permite crear otras funciones nuevas.

b) Portabilidad

INFORMIX es totalmente portable a más de 80 plataformas de hardware y sistemas operativos, incluyendo VMS, MVS, Unix, MS-DOS, OS/2, Macintosh, además de muchas otras plataformas propietarias. Esta portabilidad le permite total libertad de escoger la plataforma de servidores de bases de datos que satisfagan los requerimientos de su sistema.

La arquitectura cliente-servidor de INFORMIX aprovecha al máximo el hardware más reciente, incluyendo los más avanzados multiprocesadores simétricos (SMPs) y también aquellos multiprocesadores LOOsely coupled.

INFORMIX ofrece una capacidad de aumento en SMPs hasta de un 95 por ciento a mayor número de transacciones por segundo.

c) Sistemas Abiertos

INFORMIX ofrece la implementación líder del estándar SQL de la industria. La arquitectura abierta de INFORMIX integra sistemas DBMS INFORMIX y no INFORMIX, con el conjunto más amplio de herramientas, aplicaciones y productos de software de terceros de la industria.

La arquitectura abierta de INFORMIX brinda acceso en forma transparente a datos de otras bases de datos relacionales como DB2 y SQL/DS de IBM, e incluso a bases de datos no relacionales y sistemas de archivo tales como RMS de DEC.

INFORMIX integra este recurso unificado de datos con su amplísimo conjunto de herramientas y un creciente número de aplicaciones, también interactúa con el más amplio conjunto de productos de software de terceros de la industria incluyendo paquetes populares

como Lotus 1-2-3, dBASE, HyperCard, y muchos otros que van desde procesadores de palabras hasta sistemas expertos.

d) Rendimiento

INFORMIX ha registrado marcas de rendimiento auditadas en las plataformas OS/2, Unix, VMS y MVS.

El DBMS de INFORMIX se adapta en forma única tanto al procesamiento en línea (OLTP) como a los sistemas DSS siendo el único DBMS que puede manejar las exigencias de la ejecución simultánea de DSS y OLTP sobre los mismos datos críticos.

e) Control de Concurrencia

INFORMIX usa consultas con bloqueo no restringido libre de contención a nivel de fila, para minimizar y en muchos casos, eliminar por completo los tiempos de espera por contención

f) Bloqueo a Nivel de la Fila.

El bloqueo de INFORMIX a nivel de fila puede reducir en uno o más órdenes de magnitud los tiempos de espera por contención con respecto a los esquemas de bloqueo a nivel de página utilizados por la mayoría de las bases de datos. INFORMIX brinda bloqueo a nivel de fila tanto para datos como para índices, sin crecer proporcionalmente los bloqueos ni limitar el número de filas bloqueadas por transacción, tabla o base de datos.

g) Consultas Libres de Contención

Se puede ejecutar reportes aplicaciones de apoyo a la toma de decisiones y consultas ad-hoc sobre datos críticos, sin afectar el rendimiento del sistema OLTP. El control de concurrencia multiversión de INFORMIX mantiene una total integridad y consistencia de los datos sin bloqueo de lectura, de forma tal que las actualizaciones y consultas no se bloqueen entre ellas. Además elimina la necesidad de los bloqueos de lectura, cerciorándose de que usted vea los datos como existían en la base de datos en el instante en que emitió la consulta. INFORMIX elimina por completo los tiempos de espera por contención de lector/escritor, ya que los lectores no bloquean a los escritores ni los escritores a los lectores.

h) Eliminación de Cuellos de Botella por I/O

Las tecnologías de INFORMIX commit rápido, commit de grupo y commit de escritura diferida reducen en forma dramática las operaciones de I/O a discos y eliminan los cuellos de botella por I/O.

Mientras algunas bases de datos escriben bloques completos de datos al disco en el momento de hacer el COMMIT, INFORMIX confirma las transacciones con un máximo de una escritura secuencial a los archivos de bitácora (log) en disco. En los sistemas de alto rendimiento, una escritura secuencial típicamente hará el commit de múltiples transacciones.

Los datos leídos por una transacción permanecen en memoria compartida en forma tal que otras transacciones pueden tener acceso a esos datos sin tener que leerlos nuevamente del disco.

i) Características Adicionales de Rendimiento

INFORMIX incorpora una multitud de características únicas orientadas a los requerimientos de rendimiento prácticos de los sistemas de hoy. Estos incluyen generadores de números secuenciales que eliminan el cuello de botella producido por la necesidad de generar secuencias únicas de números; una interfaz de arreglos para transferir rápidamente gran cantidad de filas entre bases de datos y aplicaciones, y grupos de tablas múltiples o unitarias para reducir el I/O a disco.

j) Operación Continua

Las aplicaciones operan sin interrupción durante las 24 horas del día. Funciones cruciales del sistema tales como las de respaldo, recuperación y administración de la base de datos, tienen lugar en línea, sin interrumpir las transacciones.

Una falla en una porción de la base de datos no afecta aplicaciones que no están relacionadas. Las aplicaciones no afectadas continuarán procesando transacciones mientras que la porción de la base de datos que presenta fallas se saca de línea y se recupera.

k) Soporte a Grandes Bases de Datos

El tamaño de su base de datos solo está limitado por la capacidad de almacenamiento disponible. Una tabla, índice o base de datos INFORMIX, puede utilizar todos los dispositivos físicos del disco disponibles. Usted tiene control absoluto sobre la ubicación de tablas e índices, la asignación de espacio a una tabla a medida que vaya creciendo el espacio al que puede tener acceso un usuario y sobre el espacio que puede consumir un usuario.

l) Datos Compartidos en Forma Distribuida

Las capacidades de interconexión en redes y bases de datos distribuidas de INFORMIX se pueden usar para tener acceso a datos almacenados en servidores remotos, tan fácilmente como si hubiesen sido almacenados en un único computador local. Con una sola instrucción SQL se puede tener acceso a datos en múltiples instalaciones. Usted puede almacenar los datos en el sitio en el que el rendimiento, la seguridad o la disponibilidad de su sistema lo requiera.

4.3 Recuperación de datos.

ODBC, DAO y RDO2 forman parte principal del conjunto de tecnologías que gran cantidad de aplicaciones utilizan actualmente para interconexión de datos. De hecho, muchas de ellas se comprometen plenamente con éstas utilizándolas extensiva e intensivamente.

ODBC, DAO y RDO2 son tecnologías conocidas, maduras y consolidadas, sin embargo, en muchas áreas no pueden considerarse como la o las soluciones integradas para las aplicaciones actuales que demandan mayor funcionalidad.

Como herramienta de consulta de una base de datos el lenguaje SQL proporciona extensiones significativas a las capacidades especificadas en el estándar SQL ANSI/ISO.

Sentencias

El lenguaje SQL consta de unas treinta sentencias, cada sentencia demanda una acción específica por parte de DBMS, tal como la creación de una nueva tabla, la recuperación de

datos o la inserción de nuevos datos en la base. Todas las sentencias SQL tienen la misma forma básica, como a continuación se describe.

Sentencia	Descripción
<i>Manipulación de Datos</i>	
SELECT	Recupera datos de la base de Datos
INSERT	Añade filas de datos a la base de datos
DELETE	Suprime filas de datos de la base de datos
UPDATE	
<i>Definición de datos</i>	
CREATE TABLE	Añade una nueva tabla a la base de datos
DROP TABLE*	Suprime una tabla de la base de datos
ALTER TABLE*	Modifica la estructura de una tabla existente
CREATE VIEW*	Añade una nueva vista a la base de datos
DROP VIEW*	Suprime una vista de la base de datos
CREATE INDEX*	Construye un índice para una columna
DROP INDEX*	Construye un índice para una columna
CREATE SYNONYM*	Define un alias para un nombre de tabla
DROP SYNONYM*	Suprime un alias para un nombre de tabla
COMMENT*	Defino comentarios para una tabla
LABEL*	Define el título de una columna

<i>Control de Acceso</i>	
GRANT	Concede privilegios de acceso a un usuario
REVOKE	Suprime privilegios de acceso a un usuario
<i>Control de Transacciones</i>	
COMMIT	Finaliza la transacción actual
ROLLBACK	Aborta la transacción actual
<i>SQL Programático</i>	
DECLARE	Define un cursor para una consulta
EXPLAIN*	Describe plan de acceso a datos para consult
OPEN	Abre cursor para recuperar datos de consulta
FETCH	Recupera fila de resultados de una consulta
CLOSE	Cierra un cursor
PREPARE*	Prepara sentencia SQL para ejecución dinámica
EXECUTE*	Ejecuta dinámicamente una sentencia SQL
DESCRIBE*	Describe una sentencia preparada

Las sentencias seguidas de un asterisco no forman parte del estándar SQL ANSI/ISO, pero se encuentran en la mayoría de los productos más populares basados en SQL.

Todas las sentencias SQL comienzan con un verbo, una palabra clave que describe lo que la sentencia hace. CREATE, INSERT, DELETE y COMMIT son verbos típicos. La sentencia continúa con una o más *cláusulas*. Una cláusula puede especificar los datos sobre los que debe actuar la sentencia, o proporcionar más detalles acerca de lo que la sentencia se supone que hace. Todas las cláusulas comienzan con una palabra clave, tal como WHERE, FROM, INTO y HAVING.

Algunas cláusulas son opcionales; otras son necesarias. La estructura y contenido específicos varían de una cláusula a otra. Muchas cláusulas contienen nombres de tablas o columnas; algunas pueden contener palabras clave adicionales, constantes o expresiones.

A continuación se presenta la forma de escribir una sentencia

DELETE FROM *nombre de tabla* WHERE *condición de búsqueda*

Nombres

Los objetos de una base de datos basada en SQL se identifican asignándoles nombres únicos. Los nombres se utilizan para identificar el objeto de la base de datos sobre la que la sentencia debe actuar. El estándar SQL ANSI/ISO especifican nombres de tabla (que identifican tablas), nombres de columna (que identifican columnas) y nombres de usuario (que identifican usuarios de la base de datos). Muchas implementaciones SQL soportan objetos nominados adicionalmente, tales como procedimientos almacenados (Sysbase y SQL Server), relaciones clave primaria/clave foránea (DB2) y formularios de entrada de datos (Ingres).

Nombres de Tabla. Cuando se especifica un nombre de tabla en una sentencia SQL , presupone que se está refiriendo a una de las tablas propias (es decir una tabla ya creada). Con el permiso adecuado, también se puede referir a tablas propiedad de otros usuarios, utilizando un *nombre de la tabla cualificado*. Un nombre de tabla cualificado especifica el nombre del propietario de la tabla junto con el nombre de la tabla, separados por un punto(.). Por ejemplo la tabla CUMPLEAÑOS, propiedad del nombre del usuario SAM, tiene el nombre de la tabla cualificado:

SAM.CUMPLEAÑOS.

Un nombre de tabla cualificado puede ser utilizado generalmente dentro de una sentencia SQL en cualquier lugar que pueda aparecer un nombre de tabla.

Nombres de Columna. Cuando se especifica el nombre de columna en una sentencia SQL, puede determinar normalmente a qué columna se refiere a partir del contexto. Sin embargo, si la sentencia afecta a dos columnas con el mismo nombre correspondiente a dos tablas diferentes, debe utilizarse un *nombre de columna cualificado* para identificar sin ambigüedad la columna designada. Un nombre de columna cualificada especifica tanto el nombre de la tabla que contiene la columna como el nombre de la columna, separados por un punto(.). Por ejemplo la columna de nombre VENTAS en la tabla REPVENTAS tiene el nombre de columna cualificado:

REPVENTAS.VENTAS.

Un nombre de columna cualificado puede ser utilizado generalmente dentro de una sentencia SQL en cualquier lugar que pueda aparecer un nombre de columna simple (no cualificado).

Tipos de Datos

Las bases de datos basadas en SQL pueden almacenar varios tipos de datos entre los que incluyen texto, enteros, números decimales, números de coma flotante.

Funciones Internas

Aunque el estándar ANSI/ISO no las especifica, la mayoría de las implantaciones SQL incluyen una serie de *funciones internas* útiles. Estas utilidades proporcionan con frecuencia facilidades de conversión de tipo de datos. Por ejemplo, las funciones internas DB2 Month() y Year() aceptan un valor Date() o Timestamp como entrada y devuelven un entero que es el mes o el año del valor.

Las funciones internas también suelen utilizarse para reformatear los datos.

La función interna de INFORMIX To_Char(), por ejemplo, acepta un tipo de datos Date y una especificación de formato como argumentos, y devuelve una cadena que contiene una versión formateada de la fecha.

Falta de datos (valores NULL)

Puesto que una base de datos es generalmente un modelo de una situación del mundo real, ciertos datos pueden inevitablemente faltar, ser desconocidos o no ser aplicables.

SQL soporta explícitamente los datos que faltan, son desconocidos o son inaplicables, a través del concepto del *valor nulo*. Un valor nulo es un *indicador* que dice a SQL (y al usuario) que el dato falta o no es aplicable. Por conveniencia, un dato que falta normalmente se dice que

tiene el valor NULL, pero el valor NULL no es el valor de dato real como 0,473,83. En vez de ello, es una señal o un recordatorio de que el valor de datos falta o es desconocido.

4.4 Seguridad e Integridad de Datos.

Los términos **seguridad** e **integridad** se escuchan mucha frecuencia juntos en contextos de bases de datos, aunque en realidad los conceptos son distintos. Seguridad se refiere a la protección de los datos contra una revelación, alteración o destrucción no autorizada; integridad se refiere a la exactitud o validez de los datos. En otras palabras:

La **seguridad** implica asegurar que los usuarios están **autorizados** para llevar a cabo lo que tratan de hacer, la **integridad** implica asegurar que lo que tratan de hacer es **correcto**.

Desde luego, existen similitudes entre ambos conceptos: en los dos casos el sistema necesita estar al tanto de ciertas restricciones que no deben ser violadas por los usuarios; en ambos casos esas restricciones se deben especificar (por lo general se encarga el administrador de la base de datos DBA) en algún lenguaje apropiado y se deben mantener en el catálogo o diccionario del sistema; y en los dos casos el sistema de gestión de bases de datos (SGDB) debe supervisar de algún modo las interacciones de los usuarios con el fin de garantizar que se conservan dichas restricciones.

4.4.1 Seguridad.

4.4.1.1 Consideraciones generales.

El problema de la seguridad tiene muchos aspectos, entre ellos los siguientes:

- Aspectos legales, sociales y éticos (por ejemplo, ¿tiene la persona que solicita, por ejemplo el crédito de un cliente, derecho legal a obtener la información solicitada?)
- Controles físicos (por ejemplo, ¿debe estar cerrada o resguardada de algún modo la habitación de los terminales?)
- Cuestiones de política interna (por ejemplo, ¿cómo decide la empresa propietaria del sistema quiénes pueden tener acceso a qué?)
- Problemas de operación (por ejemplo, si se usa un sistema de contraseñas, ¿cómo se mantienen en secreto las contraseñas?, ¿Con qué frecuencia se cambian?)
- Controles del equipo (por ejemplo, ¿posee la CPU características de seguridad tales como claves para la protección de las áreas de almacenamiento o un modo de operación privilegiado?)
- Seguridad del sistema operativo (por ejemplo, ¿borra el sistema operativo subyacente el contenido de las áreas de almacenamiento y los archivos de datos cuando ya no se necesitan?).
- Materias de relevancia específica para el sistema mismo de bases de datos (por ejemplo, ¿tiene el sistema de bases de datos un concepto de propiedad de los datos?)

- Debe existir alguna forma de verificar una solicitud de acceso dada contra las restricciones de autorización aplicables (con solicitud de acceso nos referimos aquí a la combinación de operación solicitada más objeto solicitado más solicitante). Esta verificación se hace por el **subsistema de seguridad** del SGBD, también llamado **subsistema de autorizaciones**.
- Para poder decir qué restricciones son aplicables a una solicitud dada, el sistema debe ser capaz de reconocer el origen de dicha solicitud; es decir, debe ser capaz de reconocer al usuario específico del cual proviene una determinada solicitud. Por esa razón cuando los usuarios tienen acceso al sistema, casi siempre se les solicita no sólo que proporcionen su **identificador de usuario** (decir quienes son), sino también una **contraseña** (para demostrar que son quienes dicen ser). En teoría, sólo el sistema y los usuarios legítimos del identificador en cuestión conocen la contraseña.

4.4.1.2 Control de acceso por privilegios.

Como ya se ha comentado, la mayoría de los SGBD's soportan bien control por privilegios, bien control por niveles, o ambos. De hecho, sería más exacto decir que la mayoría de los sistemas soportan control por privilegios, y algunos de ellos soportan también control por niveles; el control por privilegios se encuentra con más frecuencia en la práctica.

Como ya se ha apuntado en las consideraciones generales, se necesita un lenguaje que soporte la definición de reglas de seguridad (por privilegios). Describamos un ejemplo hipotético de tal lenguaje.

4.4.2 Integridad.

4.4.2.1 Consideraciones generales.

Como se explicó, el término integridad se usa para referirnos a la exactitud de los datos en una base de datos. Esto es distinto al de seguridad, concepto con el que guarda ciertas similitudes enunciadas al comienzo del tema. Un punto de diferencia entre ambos conceptos, es que las reglas de integridad, a diferencia de las de seguridad, no son específicas de un usuario, si lo fuesen por definición serían reglas de seguridad y no reglas de integridad.

4.4.2.2 Reglas de integridad.

Comencemos expresando una regla de integridad en un lenguaje hipotético.

Supongamos que PX es una variable sobre la relación PIEZA.

CREAR REGLA INTEGRIDAD ejemplo1

PARA-TODO PX (PX.PESO > 0)

SI INTENTO-VIOLACIÓN RECHAZAR;

Esta regla expresa que los pesos de las piezas deben ser positivos. El ejemplo sirve para ilustrar que las reglas de integridad tienen en general tres componentes:

- Un nombre (ejemplo1). La regla se registra en el catálogo del sistema bajo ese nombre. Este nombre aparecerá en cualquier respuesta producida por el sistema a un intento de violación de esta regla.
- La restricción que se debe cumplir, formulada con expresiones que se evalúen a verdadero o falso. La regla de integridad se dice que se satisface si y sólo si la expresión se evalúa a verdadero, y que es violada si y sólo si se evalúa a falso. En el ejemplo se requiere que *todas* las piezas tengan un peso mayor que cero (PARA-TODO es el cuantificador universal). En la práctica, por supuesto, es suficiente con revisar sólo la pieza que acaba de ser insertada o modificada, no todas las piezas (antes de realizar la operación la base de datos sería correcta).
- La respuesta en caso de intento de violación de la regla. En la regla ejemplo1 la respuesta sería simplemente rechazar. Esta es la opción más extendida en la práctica, por lo que se considerará la opción por defecto, pero en general, la respuesta podría ser un procedimiento de cualquier complejidad.

Cuando una sentencia CREAR REGLA INTEGRIDAD se ejecuta, el sistema debe revisar si el estado actual de la base de datos satisface la restricción que se especifica. Si no es así la nueva regla debe ser rechazada y en caso contrario aceptada y se forzará a que se cumpla desde ese momento. Para velar por el cumplimiento de ejemplo1, el SGBD debe controlar todas las operaciones que sean insertar un valor o actualizar un valor del atributo peso de la tabla pieza.

En general las reglas de integridad son clasificadas en 2 grandes categorías:

a) Reglas Estáticas o de Estado

Reglas de dominio.

Una regla de dominio especifica los valores lícitos para un dominio dado.

Reglas de atributo.

Una regla de atributo especifica los valores lícitos para un atributo dado.

Reglas de relación.

Una regla de relación especifica los valores lícitos para una relación dada.

Reglas de base de datos.

Una regla de base de datos especifica los valores lícitos para una base de datos.

b) Reglas Dinámicas o de Transacción

Empezaremos viendo cada uno de los tipos de reglas estáticas para finalizar con las de transición.

Reglas de Dominio.

La regla de integridad de dominio para un dominio dado es, precisamente, la definición del conjunto de valores que formarán ese dominio; en otras palabras, se enumeran los valores del dominio. No hay por tanto necesidad de una regla "CREAR REGLA INTEGRIDAD-

DOMINIO" ya que esto se especifica en la definición de dominios. En el lenguaje hipotético podría ser:

CREAR DOMINIO cantidad DÍGITO(9)

PARA-TODO cantidad (cantidad >0 Y cantidad <5000 Y

MÓDULO(cantidad,50)=0);

El nombre de la regla es justo el nombre del dominio, ya que los dominios, por sí no se actualizan, las reglas de dominio no tiene sentido revisarlas y por tanto el concepto de respuesta en caso de violación no existe en estas reglas.

La restricción en la regla de dominio se da por medio de una fórmula en cálculo relacional (normalmente de dominios), en la que (a) hay exactamente una variable, (b) la variable está bajo el alcance de un cuantificador universal, (c) se recorre el dominio en cuestión. En el ejemplo, la restricción dice que los valores del dominio *cantidad* deben ser positivos, múltiplos de 50 y no superiores a 50000.

Estas reglas sólo desaparecen si se elimina el dominio en sí.

La expresión "VALORES(lista de valores separados por coma)", es útil par expresar de una forma corta algunas restricciones de dominio.

CREAR DOMINIO color CHARACTER(8)

VALORES("rojo", "amarillo", "azul", "verde")

Pero estas expresiones siempre son equivalentes a una fórmula en cálculo relacional:

CREAR DOMINIO color CHARACTER(8)

RATODO color (color="rojo" O color="amarillo" O color="azul" O color="verde")

Reglas de Atributo

La regla de integridad de atributo para un atributo dado es, precisamente, la especificación del dominio desde el que el atributo puede tomar valor. Debido a esto no es necesario una regla específica "CREAR REGLA INTEGRIDAD-ATRIBUTO", ya que la regla de integridad se especifica como parte directa de la definición del atributo.

Ejemplo: color_pieza dominio(color)

En este caso el nombre de la regla es el del dominio.

Las reglas de atributo son revisadas *inmediatamente*, es decir, cualquier intento de insertar o actualizar el valor de un atributo, que lleve a un valor que no pertenezca al dominio sobre el que se ha definido será rechazada inmediatamente. Por tanto, como se acaba de indicar, la respuesta ante un intento de violación en las reglas de atributo es siempre rechazar. Una regla de integridad de atributo se puede destruir sólo destruyendo la definición del atributo en sí.

En definitiva una regla de integridad de atributo, no es más que la especificación del tipo de datos del atributo, donde el tipo de datos puede ser de los que proporciona el sistema o construido por el usuario.

Reglas de Relación

Se llama regla de relación para una relación dada a aquella que se refiere sólo a la relación dada, ni a otras relaciones, ni dominios.

CREAR REGLA INTEGRIDAD-RELACIÓN EJEMPLO2

PARA-TODO PIEX (SI PIEX.PESO>50 ENTONCES PIEX.COLOR<>"ROJO")

[ON ATTEMPTED VIOLATION REJECT]

La sintaxis general es por tanto:

CREAR REGLA INTEGRIDAD-RELACIÓN nombre_regla

restricción

[ON ATTEMPTED VIOLATION acción]

Las reglas de relación son siempre revisadas inmediatamente, es decir, cada operación sobre una relación dada incluye conceptualmente como paso final la revisión de todas las reglas de relación que se aplican a esa relación.

La restricción es normalmente una fórmula cerrada en cálculo relacional de tuplas. Es decir, es una expresión que no contiene variables libres y se evalúa a verdadero o falso. La acción es normalmente un procedimiento de complejidad arbitraria que se invoca cuando la restricción se evalúa a falso. Por defecto, como ya se ha comentado se suele rechazar la operación, pero, en algunas situaciones más complejas puede ser apropiado ejecutar actualizaciones de compensación en algunos sitios de la base de datos para dejarla en un estado que cumpla todas las restricciones. Ya sea la acción rechazar o un determinado procedimiento a ejecutar,

esta acción se considerará parte de la operación de actualización que intenta violar la regla y se volverá a revisar la integridad una vez el procedimiento se haya ejecutado.

Ejemplos de reglas de relación son:

CREAR REGLA INTEGRIDAD-RELACIÓN ejemplo3

PARA-TODO PIEX (PIEX.PRECIO>0 AND PIEX.PRECIO<10000)

Esta regla que hace que los precios de las piezas estén entre 0 y 1000 es una regla de relación siempre que se asuma que el dominio del precio de las piezas no es sólo ese intervalo.

CREAR REGLA INTEGRIDAD-RELACIÓN ejemplo4

EXISTE PIEX (EXISTE PIEY (PIEX.CODIGO<>PIEY.CODIGO Y
PIEX.COLOR="ROJO" Y PIEY.COLR="ROJO"))

Esta regla indica que tienen que existir al menos dos piezas rojas. Esta regla se creará después de que un conjunto apropiado de tuplas se hayan insertado, pues en otro caso daría error. Esta regla difiere, por ejemplo, de la anterior en que, a *ejemplo3* las operaciones de borrado en principio no le afectan, mientras que a *ejemplo4* sí. Si se cumple la regla, las inserciones no afectan a su violación, mientras sí que pueden afectar a *ejemplo3*.

Un claro ejemplo de reglas de relación son la definición de claves candidatas. Supongamos *provx* y *provy* dos variables sobre la relación proveedores:

CREAR REGLA INTEGRIDAD-RELACIÓN candidata

SI *provx.código*=*provy.código*

ENTONCES provx.nombre=provy.nombre Y

provx.nombre=provy.nombre Y

provx.nombre=provy.nombre

Estas reglas que se suelen definir en todas las relaciones con independencia de la información que vaya a contener (a diferencia de las anteriores donde se enuncian aspectos concretos) se suele tener previsto en el lenguaje de definición una forma más abreviada de expresarlas.

CLAVE CANDIDATA (código) PARA proveedores

4.5 Conexiones Cliente/Servidor.

La arquitectura cliente-servidor es una forma específica de diseño de aplicaciones, aunque también se conoce con este nombre a los ordenadores en los que estas aplicaciones son ejecutadas. Por un lado, el cliente es el ordenador que se encarga de efectuar una petición o solicitar un servicio. El cliente no posee control sobre los recursos, sino que es el servidor el encargado de manejarlos. Por otro lado, el ordenador remoto que actúa como servidor evalúa la petición del cliente y decide aceptarla o rechazarla consecuentemente. Una vez que el servidor acepta el pedido la información requerida es suministrada al cliente que efectuó la petición, siendo este último el responsable de proporcionar los datos al usuario con el formato adecuado. Finalmente debemos precisar que cliente y servidor no tienen que estar necesariamente en ordenadores separados, sino que pueden ser programas diferentes que se ejecuten en el mismo ordenador.

La figura 4.2 muestra la emergente arquitectura *cliente/servidor* para gestión de bases de datos. En esta arquitectura, los computadores personales están combinados en una red de área local junto con un *servidor de base de datos* que almacena las bases de datos compartidas. Las funciones de DBMS están divididas en dos partes. Los (*frontales*) de base de datos, tales como herramientas de consulta interactiva, escritores de informe y de programas de aplicación, se ejecutan en el computador personal. La máquina de soporte (*back-end*) de la base de datos que almacena y gestiona los datos se ejecuta en el servidor. SQL se ha convertido en el lenguaje de base de datos estándar para comunicación entre las herramientas frontales y la máquina de soporte en esta arquitectura.

Consideremos una vez más la consulta que solicita el tamaño medio de los pedidos. En la arquitectura cliente/servidor, la consulta viaja a través de la red hasta el servidor de base de datos como una petición SQL. La máquina de base de datos en el servidor procesa la petición y explora la base de datos, que también reside en el servidor. Cuando calcula el resultado, la máquina de base de datos envía de vuelta a través de la red una única contestación a la petición inicial, y la aplicación frontal la muestra en pantalla del PC.

La arquitectura cliente/servidor reduce el tráfico de red y divide la carga de trabajo de la base de datos. Las funciones de intensiva relación con el usuario, tales como el manejo de la entrada y la visualización de los datos, se concentran en el PC. Las funciones intensivas en proceso de datos, tales como la entrada/salida de archivos y el procesamiento de consultas, se concentran en el servidor de la base de datos. Lo que es más importante, el lenguaje SQL proporciona una interfaz bien definido entre los sistemas frontales y de soporte, comunicando las peticiones de acceso a la base de los datos de una manera eficiente.

CAPITULO QUINTO

5 ANALISIS Y DISEÑO DEL SISTEMA

5.1 Cuadros comparativos.

Sistemas operativos.

Cooperativa / S.O.	Novell	Unix	Windows NT	Windows 95	MSDOS
El Sagrario Ltda.		X		X	
San Francisco Ltda.	X			X	X
Oscus Ltda.		X	X	X	
Cámara de Comercio			X	X	
La Merced Ltda.	X		X	X	

Topología de Red.

Cooperativa / Red	Estrella	Estrella-Bus
El Sagrario Ltda.	X	
San Francisco Ltda.	X	X
Oscus Ltda.	X	
Cámara de Comercio	X	
La Merced Ltda.	X	

Tipo de Cable.

Cooperativa	Cable	Coaxial	UTP 5
El Sagrario Ltda.			X
San Francisco Ltda.		X	X
Oscus Ltda.			X
Cámara de Comercio			X
La Merced Ltda.			X

Base de Datos.

Cooperativa	B. D.	Informix	Oracle	C Free	Visual Foxpro	Foxpro
El Sagrado Ltda.		X			X	
San Francisco Ltda.				X		X
Qucus Ltda.	X					
Cámara de Comercio					X	X
La Merced Ltda.						X

Protocolos de Comunicación.

Cooperativa	Com.	ICP/IP	NetBeui	IPX/SPX
El Sagrado Ltda.		X		
San Francisco Ltda.		X		X
Qucus Ltda.		X		
Cámara de Comercio			X	
La Merced Ltda.			X	X

Velocidad Módem.

Cooperativa	Mod.	14.4 Kbps	33.6 Kbps	56 Kbps
El Sagrado Ltda.		X		
San Francisco Ltda.			X	
Qucus Ltda.				X
Cámara de Comercio			X	
La Merced Ltda.	X			

5.2 Métodos de Comunicación.

Las computadoras se comunican sobre medios de comunicación de datos, entre los que se incluyen los cables de cobre o de fibra óptica y ondas de radio, para el intercambio de datos. Algunas de las aplicaciones que se pueden realizar en la comunicación de datos son las siguientes:

- Intercambio de Archivos.
- Intercambio de correo electrónico.
- Transacciones remotas de negocios.

El éxito de la transmisión de datos depende principalmente de dos factores: la calidad de la señal y las características del medio de transmisión. El objetivo de este punto es proporcionar una visión de cuáles son los medios de transmisión que se pueden utilizar en el presente proyecto y qué características presentan dichos medios.

Hay que tener en cuenta que las características del medio de transmisión determinan en gran medida cómo se transmiten los datos, de forma que dependiendo de dicho medio la señal se deteriorará más o menos rápida. Atendiendo a las características físicas de los medios de transmisión se verán los más difundidos.

La comunicación entre computadoras situadas en lugares diferentes requiere el compromiso de un servicio de comunicaciones público, en el cual se incluye las compañías telefónicas locales y de larga distancia, u otros proveedores.

Otros métodos para conexión de sistemas a través de áreas geográficas extensas incluyen los sistemas de comunicación por satélites y microondas.

Radioenlaces.

La comunicación por radio consiste en que los datos que se pretenden enviar se deben insertar en una portadora radioeléctrica (ondas electromagnéticas que se propagan por el espacio sin guía artificial), de manera que se debe superponer la información que se desea transmitir a dicha portadora radioeléctrica. Cuando la información llega a su destino el receptor recoge una parte de la energía radioeléctrica transmitida. Junto a estos elementos básicos, se requieren órganos de acoplamiento entre los equipos y el medio de propagación; estos elementos son las antenas transmisoras y receptoras.

Para ver más claro cómo se envían los datos y cómo actúan el emisor y el receptor supóngase el siguiente ejemplo: la onda portadora es analógica (onda sinusoidal) y los datos son también analógicos (onda sinusoidal). El emisor lo que hará al enviar los datos es sumar ambas señales sinusoidales, de forma que se propagará una nueva onda que será la suma de ambas. Cuando la señal llega al receptor éste restará a la onda que llegue la señal portadora consiguiendo así la señal correspondiente a los datos. Como es lógico, el emisor y el receptor conocen la frecuencia de la señal portadora, ya que si no fuera así el receptor no podría obtener los datos a partir de la onda que llega.

El funcionamiento sería exactamente el mismo si la onda portadora fuera digital o si los datos fueran analógicos. La onda portadora es una señal electromagnética y, por tanto, puede ser

perturbada por los campos electromagnéticos debidos a otras señales. Existe un límite por debajo del cual no puede reducirse el nivel de la señal, ya que en otro caso no podría extraerse la información útil de la onda que llega al receptor (debido a las interferencias). Se definen las interferencias como el efecto de una energía no deseada (debida a una o varias emisiones de radiaciones o a sus combinaciones) sobre la recepción de un sistema de radiocomunicación. Estas interferencias se manifiestan como degradación de la calidad, o pérdida de la información en la señal recibida.

Los sistemas de radioenlace se pueden clasificar en analógicos o digitales, dependiendo de si la onda portadora es analógica o digital. Por lo general, los radioenlaces funcionan en las bandas del espectro por encima de 1 GHz.

Además de la antena emisora y la receptora, también existen elementos repetidores intermedios. La función de estos elementos repetidores es salvar la falta de visibilidad impuesta por la curvatura terrestre y así poder conseguir enlaces superiores al horizonte óptico, y las distancias son 50 y 80 Km.

Los elementos repetidores se pueden clasificar en activos o pasivos. La función de estos últimos es simplemente modificar la trayectoria de la señal que llega. Los activos, además, tienen ganancia, lo cual quiere decir que recuperan o amplifican la señal en cierta forma y la retransmiten para que siga la trayectoria deseada..

Fibra Óptica.

La fibra óptica consiste en un hilo largo, delgado y flexible de vidrio o de otro material transparente (generalmente, materiales plásticos), capaz de conducir en su interior un rayo luminoso (rayo láser). Se consigue con ello un sistema de transmisión unidireccional, que acepta una señal eléctrica, la convierte, la transmite mediante pulsos de luz y la recupera en el destino para volver a convertirla en señal eléctrica.

En un cable de fibra óptica se pueden distinguir dos partes con distinto índice de refracción: el núcleo (es macizo y se encuentra en el interior del cilindro de fibra óptica) y el revestimiento (es la parte exterior de la fibra y tiene un índice ligeramente inferior al del núcleo). Los sistemas de fibra óptica disponibles actualmente pueden transmitir del orden de los 1.000 Mbps.

Sus principales características son: un gran ancho de banda, con una atenuación pequeña, e inmunidad frente a las interferencias electromagnéticas (debido a la naturaleza luminosa de la señal transmitida). Además, el espacio entre repetidores es mucho mayor que en el caso del cable coaxial, lo cual contribuye a la reducción del coste total de la instalación.

Dentro de sus principales desventajas está su alto coste, tanto de su instalación como de las interfaces que requieren.

Microondas Vía Satélite.

La transmisión vía satélite utiliza microondas, de forma que antes de pasar a explicar cómo se realiza dicha transmisión se explicará el fundamento de la transmisión de datos utilizando microondas.

Los sistemas de microondas se basan en la utilización de ondas electromagnéticas de frecuencias altas (entre 1 y 2,5 GHz). Se cumple que cuanto mayor es la frecuencia, mayor es el ancho de banda disponible y, por tanto, se podrá utilizar un mayor número de canales al mismo tiempo.

El inconveniente de este tipo de sistemas es que la propagación de la señal es de tipo óptico (debido a la longitud de onda tan pequeña que tienen estas señales), lo cual quiere decir que para que se pueda realizar la transmisión entre dos antenas, ambas deben ser visibles, ya que cualquier obstáculo físico haría que la transmisión no fuera posible. Además, fenómenos atmosféricos como la lluvia, niebla o nieve dificultan la propagación de las señales microondas.

Debido a esto, las distancias entre la antena emisora y la receptora suelen ser inferiores a 50 Km., incluso utilizando antenas parabólicas dicha distancia no supera los 100 Km. La utilización de los satélites viene a solucionar en cierta forma el problema que se presentaba en el caso de los sistemas microondas terrestres, ya que los obstáculos naturales pueden ser superados utilizando como estación intermedia un satélite.

Debido a esto, la función principal que realizan los satélites es servir como repetidor; en estos casos la frecuencia de la señal puede ser superior a 2,5 GHz, permitiendo por tanto un mayor ancho de banda.

El funcionamiento básico de un satélite es el siguiente:

- Reciben los datos procedentes de la Tierra a una frecuencia determinada.
- Amplifica la señal si la transmisión es analógica o la repite en caso de que sea digital.
- Retransmite los datos con otra frecuencia a la estación destino.

Cables de Cobre Trenzado.

Es el medio de transmisión más antiguo y todavía el más ampliamente utilizado. También es uno de los más simples, ya que consiste en dos alambres de cobre aislados, en general de un milímetro de espesor, que se encuentran entrelazados entre sí.

La forma trenzada del cable se utiliza para reducir las interferencias electromagnéticas con respecto a los pares cercanos que se encuentran a su alrededor (hay que recordar que cuando la corriente eléctrica pasa por dos hilos paralelos genera un campo electromagnético). La principal utilización del par trenzado es el servicio telefónico, donde las distancias son relativamente cortas y las frecuencias de utilización bajas.

Generalmente, casi todos los teléfonos están conectados a la oficina de la compañía telefónica a través de un par trenzado. La distancia que las señales pueden recorrer con estos cables, sin

Interoperabilidad.

Definimos interoperabilidad como la posibilidad de enlazar ordenadores de distintas marcas con la seguridad de que trabajarán conjuntamente, dando la sensación de que funcionan como un único sistema. En este aspecto es en el que existe una mayor similitud entre todos los sistemas operativos, aunque siguen existiendo diferencias en favor de Unix. Los protocolos que podemos considerar estándares de facto (TCP/IP, NFS, X.400, X.25), vienen implementados en Unix por defecto. NT también incluye la mayoría de ellos (TCP/IP, por ejemplo), si bien, por defecto NT trabaja con protocolos propios de Microsoft (NetBEUI). Por último, Netware 4.11 también integra el TCP/IP en el producto básico.

En cualquier caso, aunque tengamos cargado el protocolo TCP/IP el acceso a una máquina NT desde el exterior tiene que ser a través de máquinas W95, WFW o NT e instalando algunas herramientas adicionales. La situación en Netware es semejante. Por contra, en Unix, mediante el uso del comando telnet es posible conectarnos a cualquier máquina Unix.

5.4.2 Estabilidad de los sistemas.

En general, los sistemas operativos de red van a actuar como servidores de redes corporativas. En este tipo de redes, en muchas ocasiones es muy conflictivo que la máquina se bloquee durante la jornada de trabajo, porque no afecta al trabajo de una persona sino a toda la empresa.

Para la mayoría de los entornos, se considera que el nivel de seguridad C2 es suficiente, ya que este nivel garantiza una protección de accesos controlados. Los tres sistemas cumplen este nivel de seguridad en las siguientes versiones: Netware 4.11, Windows NT 3.5 y varios sistemas Unix. Algunas versiones de sistemas Unix (Trusted XENIX 3.0 y 4.0, HP-UX BLS, Trusted IRIX/B, entre otros) alcanzan los niveles B1, B2 y B3.

Del estudio realizado anteriormente se ha llegado a la conclusión que el sistema operativo adecuado para el software a desarrollar en el presente proyecto es Windows NT. El mismo que deberá ser instalado en el servidor que se utilizará como base principal de consulta. Para la comunicación de las Cooperativas con el servidor se requiere terminales con sistema operativo Windows 95/98, del estudio efectuado en el capítulo 2 se determinó que todas las cooperativas poseen terminales con estas características.

En Windows NT están disponibles dos tipos de accesos remotos: acceso a nodo remoto y acceso a control remoto. Ambos solucionan el problema de acceso LAN remoto, aunque tengan capacidades y restricciones diferentes.

El acceso a nodo remoto trata a la computadora del usuario como nodo real de la red, extendiendo la conexión de red más allá de las líneas telefónicas a través de un módem. La red ve a la computadora remota como si fuera una conexión local; todo el procesamiento se hace en la computadora remota. El nodo remoto puede realizar casi las mismas tareas desde su localización remota, que las que puede realizar un nodo local.

La sentencia SELECT y la cláusula FROM.

La sentencia SELECT "selecciona" los campos que conformarán la consulta, es decir, que establece los campos que se visualizarán o compondrán la consulta. El parámetro 'lista_campo' está compuesto por uno o más nombres de campos, separados por comas, pudiéndose especificar también el nombre de la tabla a la cual pertenecen, seguido de un punto y del nombre del campo correspondiente. Si el nombre del campo o de la tabla está compuesto de más de una palabra, este nombre ha de escribirse entre corchetes ([nombre]). Si se desea seleccionar todos los campos de una tabla, se puede utilizar el asterisco (*) para indicarlo.

Una sentencia SELECT no puede escribirse sin la cláusula FROM. Una cláusula es una extensión de un mandato que complementa a una sentencia o instrucción, pudiendo complementar también a otras sentencias. Es, por decirlo así, un accesorio imprescindible en una determinada máquina, que puede también acoplarse a otras máquinas. En este caso, la cláusula FROM permite indicar en qué tablas o en qué consultas (queries) se encuentran los campos especificados en las sentencias SELECT. Estas tablas o consultas se separan por medio de comas (,), y, si sus nombres están compuestos por más de una palabra, éstos se escriben entre corchetes ([nombre]):

He aquí un ejemplo de mandatos SQL en la estructura SELECT...FROM...:

```
SELECT nombre, apellidos FROM clientes;
```

Selecciona los campos 'nombre' y 'apellidos' de la tabla 'clientes'.

5.7 Requerimientos de la institución.

De acuerdo al estudio realizado en cada una de las instituciones que intervienen en nuestro proyecto, los requerimientos propuestos por cada una de ellas se describen a continuación:

- Compartir datos personales de los socios con los siguientes campos: nombre del socio, Cédula del socio. Cooperativa a la que pertenece
- Controlar los datos de las cuentas que ya han sido liquidadas no sean desplegados lo cual hemos controlado con un campo llamado status.
- Que los datos sean actualizados constantemente
- Que exista una opción de reportes para impresión.

Toda esta información les ayudará a tener un mejor control de los socios, y de esa forma cumplir con el reglamento expuesto en la Ley de Cooperativas que indica que una persona no puede ser socio en dos o más cooperativas a la vez.

5.8 Recursos para la elaboración del sistema.

Para la elaboración del sistema se necesitan varios recursos como son: humanos, software, hardware y comunicaciones los mismos que se describen a continuación.

5.8.1 Recursos humanos.

Como recursos humanos que intervienen en este proyecto son las cinco personas que han realizado el estudio de cada una de las Cooperativas los mismos que se han unido para el desarrollo del software piloto y así cumplir con los objetivos planteados.

5.8.2 Recursos del hardware

Para la elaboración del presente proyecto se utilizará el siguiente hardware:

- Un servidor NT 4.0, instalado con diversos tipos de Bases de Datos.
- Dos computadoras 486 o superior con 16 Mb de RAM o superior conectadas en red.
- Una Impresora

5.8.3 Recursos del software.

Las herramientas de desarrollo que utilizaremos en este proyecto son:

- Visual Basic 5.0
- Access 97
- Procesador de textos
- Internet Explorer 4.0
- Drivers de diferentes Bases de Datos.
- Apis de Windows.

-
- 200 Megabytes de espacio libre en disco
 - Un monitor SVGA o superior
 - 16 Megabytes de memoria RAM o superior
 - Una unidad de CD-ROM
 - Una tarjeta de red compatible
 - Un módem de 28800 bits/seg. Como mínimo.

5.9.3 Recursos del software.

Se necesita que una Cooperativa posea una computadora con Windows NT Server 4.0, la misma que será la que contenga la información centralizada. El software que deberán tener las terminales en el resto de cooperativas es Windows 95 o superior. Se necesita para la comunicación el protocolo de comunicación NetBeui y/o TCP/IP.

5.9.4 Recursos de comunicación.

El recurso de comunicación necesario para que este software funcione adecuadamente es una línea telefónica digital.

5.10 Análisis y Diseño orientado a objetos.

El paradigma de la programación orientada a objetos es un modelo de programación de propósito general que simula la forma en que el hombre trabaja. Es una evolución natural de las primeras innovaciones al diseño de los lenguajes de programación: es más estructurada

que todos los intentos previos de estructurar la programación y es más abstracta y modular que los intentos de abstracción de datos y ocultación de detalles.

La filosofía fundamental de la programación orientada a objetos consiste en que toda programación puede verse como una simulación. El programa personifica a los objetos físicos o computacionales del dominio del problema en objetos del dominio del programa.

Para la elaboración del software piloto, nos basamos en el análisis orientado a objetos (AOO) que se basa en conceptos de objetos, clasificación, herencia y comunicación.

5.10.1 Etapas del Análisis.

La etapa del análisis en el enfoque orientado a objetos comprende a la definición de clases con sus atributos, responsabilidades y guiones.

5.10.1.1 Definición de Clases.

Se ha determinado las siguientes clases:

- Usuarios.
- Servidor Base.
- Conexión.
- Comunicación.
- Transmisión.

- Integración.
- Generar.

5.10.1.2 Definición de responsabilidades y atributos.

Nombre : Usuario.

Atributos :

- Identificación del Usuario.
- Código de Identificación.
- Línea Telefónica.
- Plataforma Disponible.
- Hardware Disponible.

Responsabilidades :

- Solicitar.
- Enviar.
- Recibir.

Nombre : Servidor base.

Atributos :

- Línea Telefónica.

- Plataforma Disponible.
- Hardware Disponible.
- Software Disponible.

Responsabilidades :

- Manejar información.
- Enviar.
- Recibir.
- Ordenar.

Nombre : Conexión.

Atributos :

- Módem.
- Línea Telefónica.

Responsabilidades :

- Establecer Conexión punto a punto.

Nombre : Comunicación.

Atributos :

- Velocidad.

- Comunicación RAS.

Responsabilidades :

- Comunicación Datos.

Nombre : Transmisión.

Atributos :

- Medio de Transmisión, método de Transmisión.
- Modos de Transmisión.
- Equipo de Transmisión.

Responsabilidades :

- Transmitir Información.

Nombre : Integración.

Atributos :

- Plataforma Nativa.
- Plataforma Base.
- Campos de los Datos.
- Registros.

- Filtros.
- Ordenación.

Responsabilidades :

- Integrar sistemas nativos de diferentes plataformas.

Nombre : Generar.

Atributos :

- Tipos de Filtros.
- Campos.

Responsabilidades :

- Generar reportes.

5.10.1.3 Definiciones de los guiones.

Los guiones, consisten en conjuntos de mensajes a clases, prueban la habilidad del diseñador de satisfacer la especificación de requisitos del sistema.

5.10.2.3 Refinamiento de las Colaboraciones.

CLASE	SERVICIO SOLICITADO	COLABORADOR
Usuario	Informes	Colaborador, Servidor Base.
Servidor Base	Integración de Sistema Nativo	Conexión, Comunicación. Transmisión, Usuario.
Conexión	Conectarse	Servidor Base, Usuario
Comunicación	Comunicarse	Conexión
Transmisión	Transmitir Información	Comunicación
Integración	Datos de los Usuarios	Usuario, Servidor Base, Transmisión
Generación	Generar Informes	Integración.

5.10.2.4 Diagrama de transición de estados.

ESTADOS	ACCION
Usuario	Solicita informe, envía información, recibe Información.
Servidor base	Envía o recibe la información al usuario.
Conexión	Realiza la conexión entre el usuario y el servidor central.
Comunicación	Es el encargado de realizar la comunicación luego de la transmisión.
Transmisión	Genera la transmisión de datos entre los usuarios y el servidor.
Integración	Es el encargado de integrar la plataforma base a la plataforma nativa y viceversa.

Generar

Se encarga de la generación de los informes solicitados por el usuario.

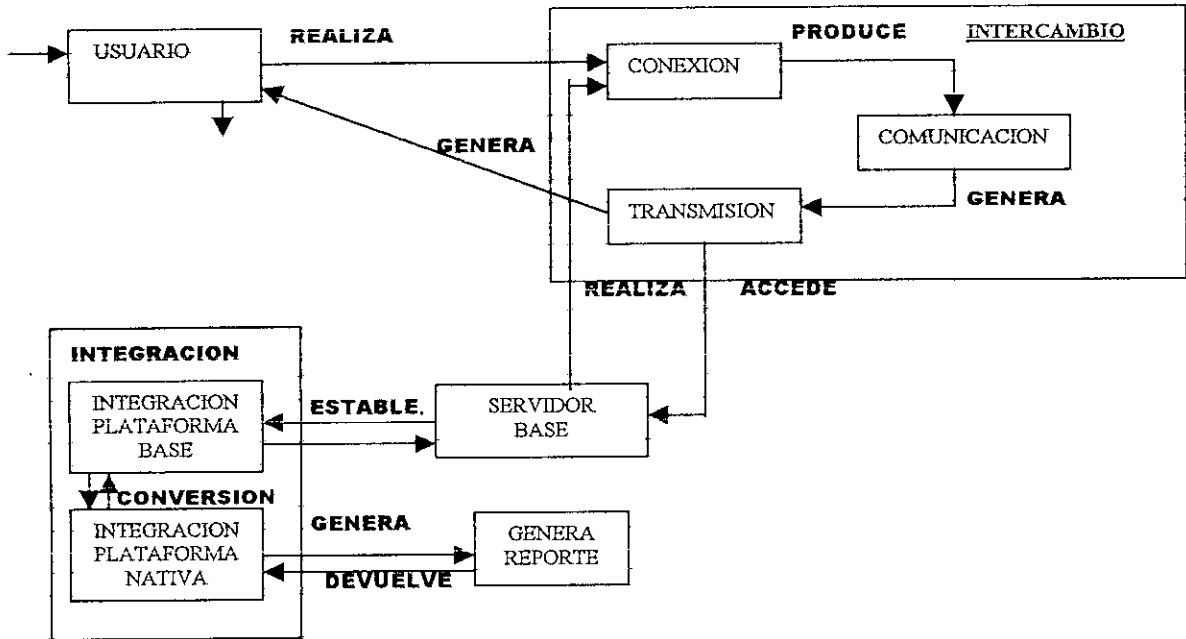


Figura 5.2. Diagrama de transición de estados.

Nodo : Usuario.
Alcanzado desde : Transmisión.
Acción : Solicitud de datos, enviar datos, recibir datos.
Salida : Conexión.

Intercambio

Nodo : Conexión.

Alcanzado desde : Usuario, servidor base.

Acción : Establece conexión.

Salida : Comunicación.

Nodo : Comunicación.

Alcanzado desde : Conexión.

Acción : Realiza comunicación.

Salida : Transmisión.

Nodo : Transmisión.

Alcanzado desde : Comunicación.

Acción : Transmitir datos.

Salida : Usuario, servidor base.

Nodo : Servidor base.

Alcanzado desde : Integración, transmisión.

Acción : Enviar datos, recibir datos.

Salida : Conexión, integración.

Nodo : Genera informe.

Alcanzado desde : Integración.

Acción : Realiza la generación de reportes.

Salida : Integración.

Integración

Nodo : Integración plataforma base.

Alcanzado desde : Servidor base, integración, plataforma nativa.

Acción : Realiza integración de plataforma.

Salida : Servidor base, integración plataforma nativa.

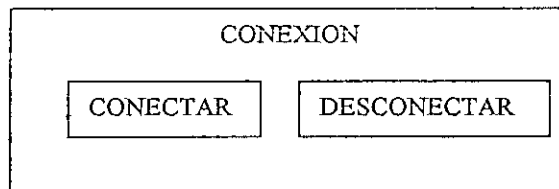
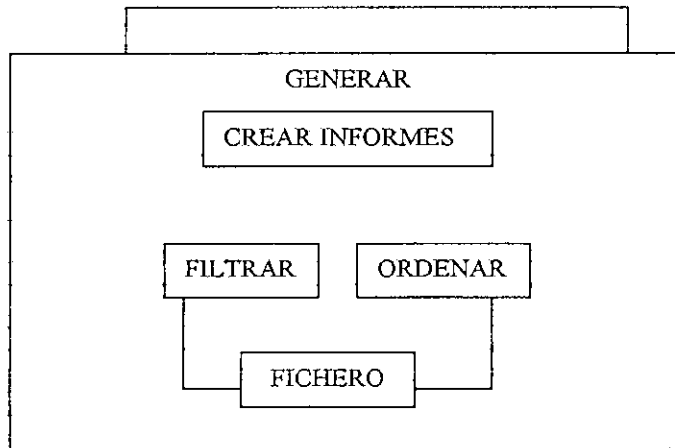


Figura 5.10. Grafo de colaboración de la clase conexión.

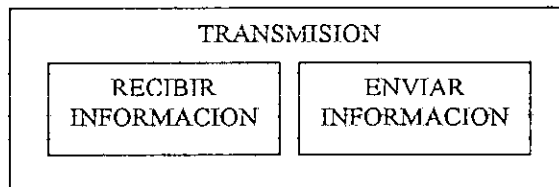


Figura 5.11. Grafo de colaboración de la clase transmisión.

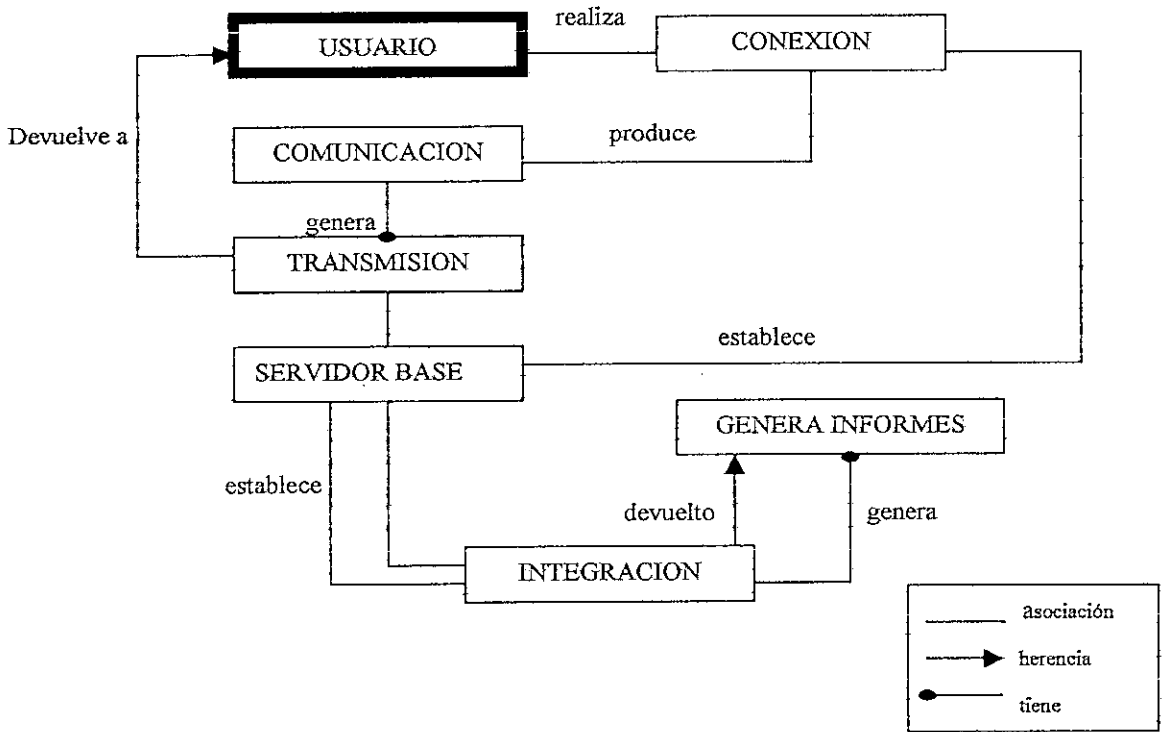
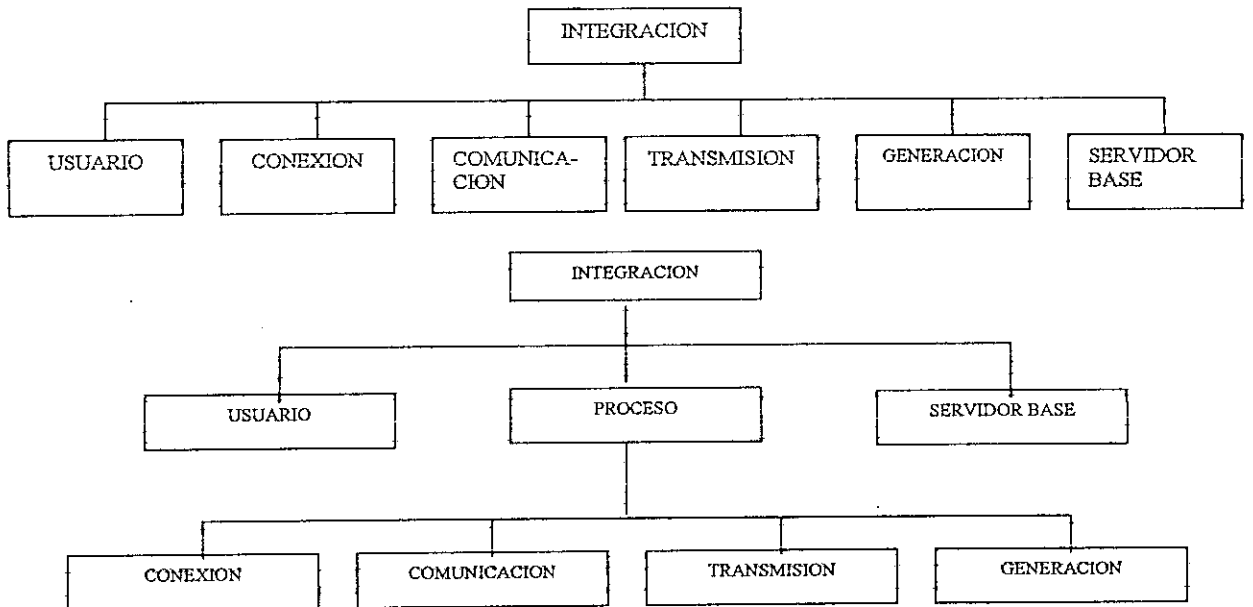


Figura 5.12. Colaboraciones entre las clases.

5.10.2.7 Refinamiento de Jerarquías.



5.10.2.8 Diagrama de Colaboraciones entre Clases.

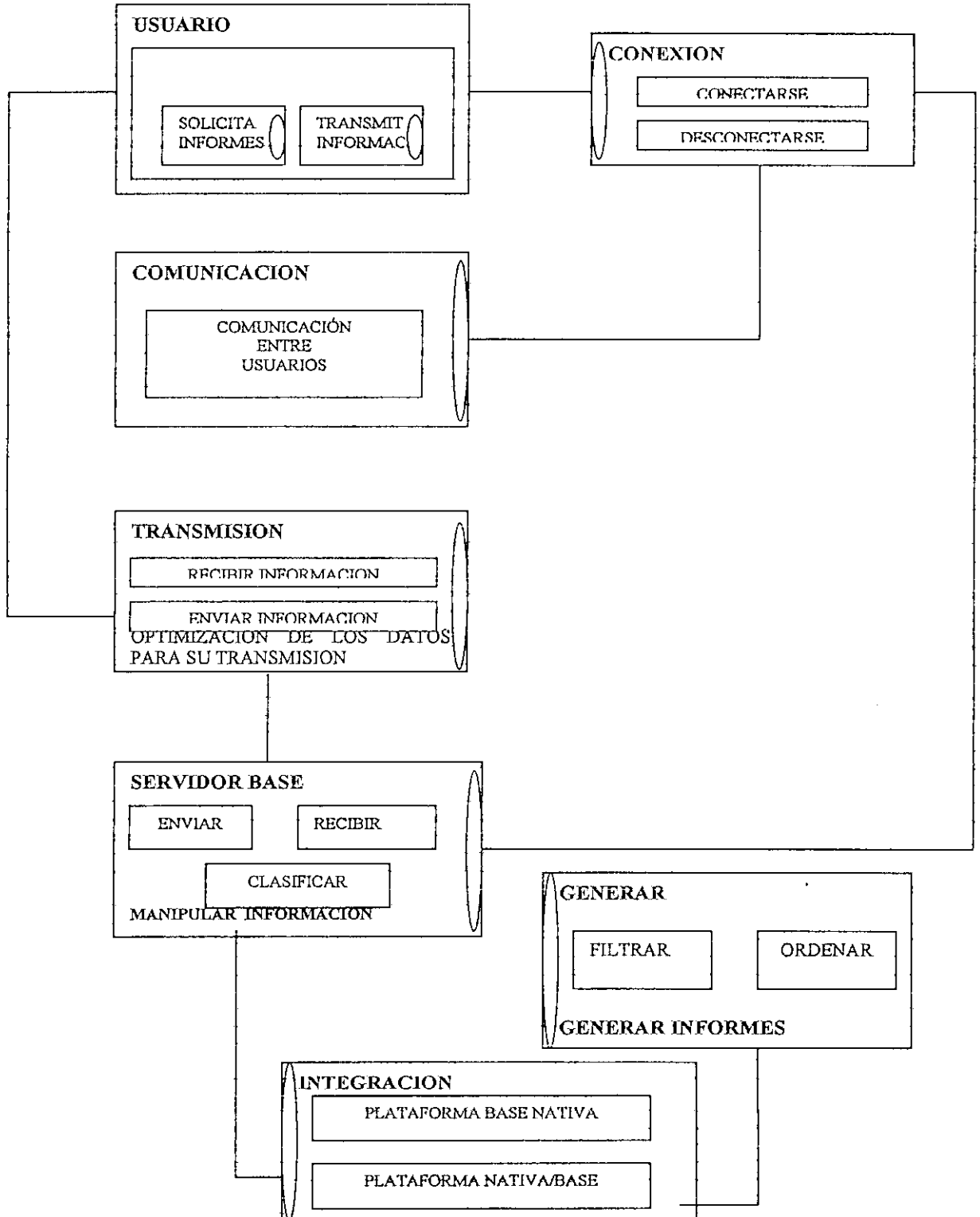


Figura 5.13. Diagrama.

5.10.2.9 Diccionario de clases del diseño.

Clase : *Usuario.*

La clase usuarios se refiere a las diferentes cooperativas, las cuales solicitan información del servidor central de las otras cooperativas; es decir la clase usuarios envía información y recibe información solicitada.

Responsabilidades.

- Solicita información a cooperativas proveedoras.
- Enviar información solicitada usuario proveedor.
- Recibir información solicitada usuario destino.

Atributos.

- Identificar usuario – tipo de usuario.
- Línea telefónica. Tipo de línea y número que utiliza.
- Plataforma que el usuario trabaja.
- Tipo de Hardware que dispone un usuario.

Clase : *Servidor Base.*

Base de Datos Central donde se almacenará la información de los usuarios.

Responsabilidades *Conexión*

- La responsabilidad de conexión se procede a establecer una comunicación entre los usuarios.
- Manipulación de la información de los usuarios.
 - Enviar y recibir la información solicitada.
 - Ordenación de la información para ser enviada.

Atributos. *Conexión*

- Número y tipo de línea telefónica.
- Tipo de plataforma base.
- Tipo de Hardware que dispone el servidor.
- Software disponible del servidor.

Clase : *Conexión.*

Conexión entre el usuario fuente servidor y usuario destino.

Responsabilidades.

- Establecer conexión entre los usuarios.

Atributos.

- Módem que dispone.
- Línea de teléfono que dispone.

SUBSISTEMA : FUSION DE DATOS

CONTRATOS	CLASES A LAS QUE DELEGA
Cambiar los sistemas nativos a una plataforma base y viceversa	Integración
Crear el Informe Solicitado por el Usuario	Comunicación

SUBSISTEMA : MOVIMIENTOS

CONTRATOS	CLASES A LAS QUE DELEGA
Solicitar Informe, Transmitir información	Usuario
Manipular la Información	Servidor Base

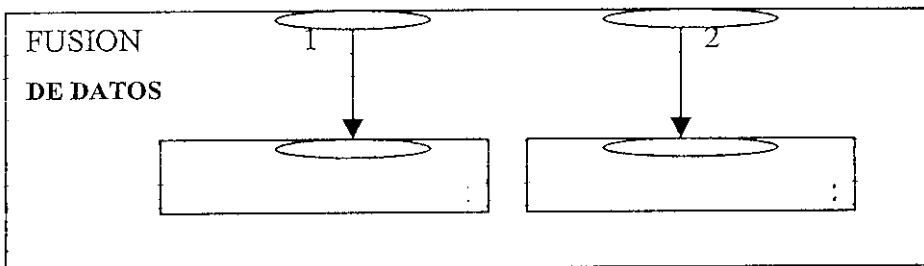


Figura 5.14. Subsistema de fusión de datos

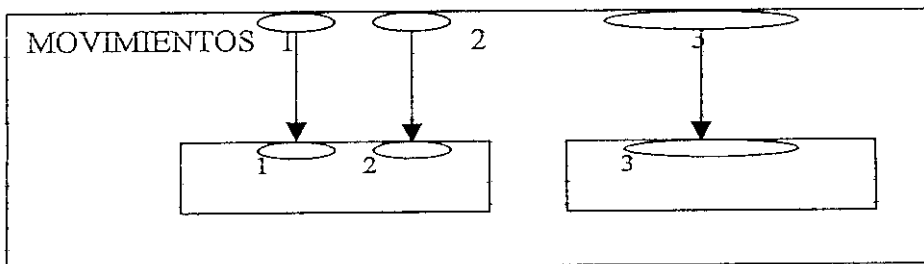


Figura 5.15. Subsistema de movimientos.

Para cada estación de trabajo ubicadas en las cooperativas se debe instalar el software piloto (programa ejecutable) el mismo que manejará una base de datos llamada DBTERMINAL la que contiene una tabla llamada TBLDATOS con los siguientes campos:

Nombre del campo	Tipo de datos	Descripción
camino	Texto	path del servidor y la base de datos
password	Texto	password de la base de datos del servidor
usuariodb	Texto	nombre de usuario de la base de datos
dominio	Texto	nombre del dominio
usuario	Texto	nombre del usuario
telefono	Texto	telefono del servidor al cual se va a conectar
fechact	Fecha/Hora	fecha de ultima actualización del archivo plano
ncedula	Texto	nombre de campo de tabla que almacena cedula
nnombre	Texto	nombre de campo de tabla que almacena nombre del titular
nstatus	Texto	nombre de campo de tabla que almacena el status
cstatus	Texto	caracter que identifica si un socio es activo
nodbc	Texto	nombre del odbc
ntabla	Texto	nombre de la tabla de consulta
cooperativa	Texto	nombre de la cooperativa
ndatabase	Texto	nombre de la base de datos
nfechain	Texto	nombre del campo fecha de ingreso

Figura 5.18. Tabla TBLDATOS de la base DBTERMINAL

Estos campos son utilizados para la configuración de cada terminal con el servidor base.

5.12 Descripción de la aplicación

La aplicación está diseñada para que funcione como servidor o terminal, su estructura se presenta de la siguiente manera:

- Se ha diseñado la aplicación para configurar el camino de la base de datos, del servidor y los datos correspondientes a la conexión ODBC de la base fuente (terminal).

Conexión de Acceso Remoto

Esta opción permite realizar la conexión entre el Usuario y el Servidor mediante la línea telefónica.

Actualizar Bases de Datos

Permite realizar la actualización de las Bases de Datos de cada una de las Cooperativas (fuente) al Servidor principal, la primera vez que se utilice esta opción actualizará todos los datos existentes; después solamente permitirá actualizar los datos de nuevos socios de cada una de las Cooperativas.

Configuración de Terminales

Mediante esta opción podemos configurar los datos que se necesitan en cada terminal para poder conectarse con el servidor central. A continuación se describen cada uno de los campos que van hacer ingresados por los usuarios para la configuración del sistema.

Cooperativa. Nombre de la Cooperativa que esta utilizando el sistema.

Camino Servidor DB. Camino donde se encuentra la Base de Datos del Servidor central.

Password DB Fuente Clave para acceder a la Base de Datos Fuente.

Usuario DB Fuente. Nombre del Usuario de la DB fuente.

Nombre de la conexión ODBC. Nombre ODBC de la Base de Datos del Servidor Fuente (DSN).

Nombre campo fecha ingreso Socio. Nombre del campo fecha de ingreso del socio en la Base Fuente.

Nombre del campo Cédula. Nombre del campo donde se almacena el número de cédula del Socio.

Nombre del campo Titular. Nombre del campo donde se almacena el nombre del Socio.

Nombre del campo Status. Nombre del campo donde se almacena el status de cédula del Socio.

Carácter del Status del Socio. Tipo de carácter con el cual se identifica a un liquidado.

Nombre de la Tabla a Consultar. Nombre de la tabla de la cooperativa (fuente) donde se encuentra la información a migrar.

Nombre de la Base de Datos. Nombre de la Base de Datos fuente que va ha ser abierta mediante ODBC.

Datos Generales.

Dominio nombre. Nombre del dominio del Servidor al que se va ha establecer conexión.

Usuario. Nombre del usuario que va a acceder al Dominio del Servidor.

Teléfono a conectarse. Número de teléfono para hacer el acceso remoto.

A continuación se describen la funcionalidad de los botones presentes en la pantalla de la actualización de datos.

< **Actualizar** >

Este botón sirve para grabar los datos ingresados en la pantalla en la tabla de datos de cada terminal.

< **Cancelar** >

Esta opción permite realizar la cancelación de los Datos que se han ingresado en la pantalla.

< **Eeditar** >

Permite realizar la edición del registro que se encuentra en pantalla.

< **Salir** >

Permite retornar al menú principal del Sistema.

Consulta de Datos

Esta opción permite realizar la consulta de datos, se pueden consultar los datos por Cédula, Nombre ó Cooperativa haciendo click en el cualquiera de las opciones. En el cuadro de texto Dato a Consultar ingresamos el nombre, cédula o cooperativa que se desea visualizar. A continuación se detallan los campos que se presentan en la pantalla.

Consultar Datos por: Cédula Nombres Cooperativa

Seleccionar cualquiera de los botones para escoger el campo de datos a consultar.

Datos a consultar:

Este campo permite ingresar el tipo de dato a consultar ya sea por Número de Cédula, Nombre o Titular de la Cuenta y la Cooperativa.

A continuación se detallan las funciones que realiza cada una de las opciones que se muestra.

< Impresión >

Permite generar un reporte de los datos consultados por impresora.

< Consultar >

Permite desplegar los datos consultados en la ventana que se encuentra en la parte media de la Consulta de datos.

< Salir >

Permite retornar a menú principal de la Integración de Sistemas.

Salir

Sale al ambiente de Windows.

5.14 Cómo crear un ODBC

Los controladores Microsoft ODBC le permiten abrir y consultar una base de datos a través de la interfaz Open Database Connectivity (ODBC). Estos controladores están diseñados para su uso en Microsoft Windows 95 o posterior, o bien en Windows NT 3.51 o posterior. En

Windows 95 o posterior solamente se admiten aplicaciones de 32 bits, mientras que en Windows NT 3.51 o posterior se admiten aplicaciones de 16 bits y de 32 bits.

El controlador ODBC de una Base de Datos se instala automáticamente al instalar el software cliente, en un PC. Si desea más información sobre la instalación del software cliente, consulte los manuales adjuntos al Software. Si necesita reinstalar solamente el controlador ODBC, puede ejecutar el programa setup de ODBC del subdirectorío ODBC o del directorío de instalación apropiado del software cliente.

Para configurar un DSN del ODBC se debe acceder al cuadro de diálogo Instalación de ODBC de la siguiente forma:

- En el Panel de control, haga doble clic en el icono ODBC de 32 bits.
- Aparece la ventana "Administrador de orígenes de datos ODBC". Elija la página "DSN de usuario"
- Elija el botón Agregar.
- Aparecerá la ventana que contiene todos los driver de las bases de datos instaladas en Windows.
- En la lista "Controladores ODBC instalados", seleccione el driver adecuado y a continuación elija "Aceptar".
- Ingresar el nombre del origen de datos, descripción, nombre del servidor de bases de datos, dirección IP y biblioteca de red.

El cuadro de diálogo Instalación de ODBC tiene las siguientes opciones.

Nombre del origen de datos.- El nombre del origen de datos deseado. Por ejemplo, "Datos de personal".

Descripción.- Una descripción de los datos del origen de datos. Por ejemplo, "Fecha de contratación, historial de sueldo y revisión actual de todos los empleados".

Servidor.- El nombre de un Servidor de la red. Puede seleccionar un servidor de la lista o introducir el nombre del servidor; puede indicar "(local)" como servidor en un PC Microsoft Windows NT.

Dirección de red.- La dirección del sistema de administración de bases de datos (DBMS) desde el cual el controlador recupera los datos. Normalmente puede dejar este valor establecido como (Predeterminado).

Biblioteca de red.- El nombre de la DLL Biblioteca de red que el controlador seleccionado anteriormente utiliza para comunicarse con el software de red. Normalmente se utiliza la Biblioteca de red predeterminada del PC cliente.

CONCLUSIONES

Luego de haber realizado el análisis individual de cada cooperativa en lo referente al área informática, y específicamente en lo que respecta al sistema operativo hemos decidido trabajar con las plataformas de Microsoft Windows NT Server 4.0 para el Servidor Base y con Windows95 para las estaciones de trabajo (cooperativas)

- Microsoft Windows NT Server 4.0 ha sido escogido para el Servidor base por ser un sistema operativo más sencillo para las necesidades más exigentes. Es flexible, escalable y compatible, que representa una significativa reducción en los costes de Hardware y Software.
- El interfaz de usuario Windows 95 ha sido integrado en Windows NT Server 4.0, da facilidad de uso y proporcionando consistencia entre sí. El hecho de proporcionar una interfaz de usuario consistente en todas las plataformas Windows y facilita la migración de los usuarios dentro de sus sistemas operativos.
- Configura y detecta los módems conectados a nuestro ordenador.
- En una red Windows NT server, los clientes pueden ejecutar casi cualquier sistema operativo, incluyendo MS-Dos, Microsoft Windows 3.1, Windows95, Novell , Unix, Machintosh OS y OS/2

-
- La seguridad, base de los servicios de directorio de Windows NT Server 4.0, es una importante característica de la solución, ofreciéndonos un control total sobre quién tiene acceso a la información Cooperativa importante.

Como motor de base de datos se ha escogido Microsoft Access ya que puede optimizar las consultas de actualización en gran volumen para orígenes de datos ODBC enviando la consulta al servidor, donde todos los registros apropiados se procesan a la vez en lugar de registro por registro. Además Microsoft Access puede importar o vincular una tabla de datos de otras bases de datos

- El software piloto es un sistema demostrativo que no necesita manejar grandes cantidades de información.

La herramienta de desarrollo escogida para nuestro software piloto es Visual Basic 5.0 ya que:

- Visual Basic permite comunicarse con bases de datos creadas con Microsoft Access, Microsoft Fox Pro, Borland Paradox, Borland dBase, ODBC.
- Visual Basic es un Front End (Herramienta de desarrollo) que maneja bases de datos de manera fácil a través de ODBC.
- Visual Basic permite trabajar con APIS que utilizaremos para la conectividad Cliente/servidor.

- Visual Basic es un lenguaje de programación de propósito general, con una gran potencia en toda su estructura. Su implementación en el sistema operativo Windows y sus herramientas visuales

Para realizar la consulta de los datos en las diferentes cooperativas se ha escogido SQL (Structured Query Language ó Lenguaje Estructurado de Consulta), es un lenguaje bastante sencillo, principalmente orientado a bases de datos y, sobre todo, al manejo de consultas.

Visual Basic incorpora esta extensión junto a nuestras bases de datos, obteniendo potentes resultados. De hecho, las consultas que se realizan en Access, están desarrolladas o basadas en este lenguaje, por lo que su implementación en Visual Basic no es complicada.

Para la conectividad entre las diferentes bases de datos de las cooperativas utilizaremos la conectividad abierta en bases de datos de microsoft (ODBC) ya que forma parte de un conjunto de controladores que hace funcionar productos de los sistemas frontales como Access y Excel de Microsoft, Foxpro, dBase y Paradox, Oracle y otros sistemas de bases de datos que ofrecen soporte a los sistemas posteriores.

RECOMENDACIONES

Para que dos o más cooperativas se puedan comunicar a la vez se recomienda:

- Dos líneas telefónicas de preferencia dedicadas, o a la vez un pool modem que permitirá realizar varias conexiones al servidor base al mismo tiempo.
- Microsoft Access Server, si se requiere manipular mayor cantidad de datos se recomienda cambiar a una base de datos más potente.
- Windows NT Server 4.0 instalado en el servidor de Bases de Datos principal.
- Windows 95/98 instalado en cada una de las terminales (Cooperativas).
- En el caso de trabajar con dos líneas telefónicas se debe configurar al servidor Windows NT para múltiple enlace,
- Crear los DSN (ODBC) correspondientes a las Bases de Datos con las cuales trabaja cada terminal.

APENDICE

ACRONIMOS

ANSI	American National Standards Institute	Instituto Nacional Americano de Standarización
API	application program interface	Interfaz de programación de aplicaciones
CAD	Computer Asisted Design	Diseño asistido por computadora
CSMA/CD	carrier sense multiple access/collision detection	Acceso múltiple con detección de portadora/detección de colisiones
DBMS	database management system	Sistema gestor de bases de datos
IP	Internet Protocol	Protocolo Internet
IPX	Internetwork Packet Exchange	Intercambio de paquetes entre redes
ISO	International Organization for Standardization	Organización internacional de normalización
LAN	local area network	Red de área local
LAT	Local Area Transport	Transporte en área local
LATA	Local Access and Transport Area	Area local para acceso y transporte
LEC	local exchange carrier	Compañía de telecomunicaciones de intercambio local
MAN	metropolitan area network	Red de área metropolitana
NFS	Network File System	Sistema de archivos en red
ODBC	Open Database Connectivity, Microsoft	Conectividad abierta en bases de datos de Microsoft
ODI	Open Data-link Interface	Interfaz abierta de base de datos

OSI	Open Systems Interconnection	Interconexión de sistemas abiertos
PDN	public data network	Red pública de datos
SPX	Sequenced Packet Exchange	Intercambio secuencial de paquetes
SQL	Structured Query Language	Lenguaje de estructuras de consultas
TCP/IP	Transmission Control Protocol/Internet Protocol	Protocolo de control de transmisión/Protocolo Internet
UTP	unshielded twisted pair	Par trenzado no apantallado
VAC	value added carrier	Compañía de telecomunicaciones de valor añadido
WAN	Wide Area Network	Red de área extensa

BIBLIOGRAFIA

SHELDON, Tom, Enciclopedia LAN TIMES de redes (networking), Ed. MCGRAW-HILL
, España 1995

RUSSEL, Charlie, CRAWFORD, Sharon, Microsoft Windows NT Server 4.0, Ed.
MCGRAW-HILL , España 1998

PRESSMAN, Roger, Ingeniería del Software, Ed. MCGRAW-HILL , España 1994

TANENBAUM, Andrew, Sistemas Operativos Modernos, Ed. PRENTICE HALL INC.,
México 1992

GROFF, James, WEINBERG, Paul, Aplique SQL, Ed. CONSULTORES EDITORES,
España 1996

SHELDON, Tom, Novell Netware 386, Ed. MCGRAW-HILL , México 1992

JONES, Edward, Aplique FoxPro 2.5 para windows, Ed. MCGRAW-HILL , España 1993

ANEXOS

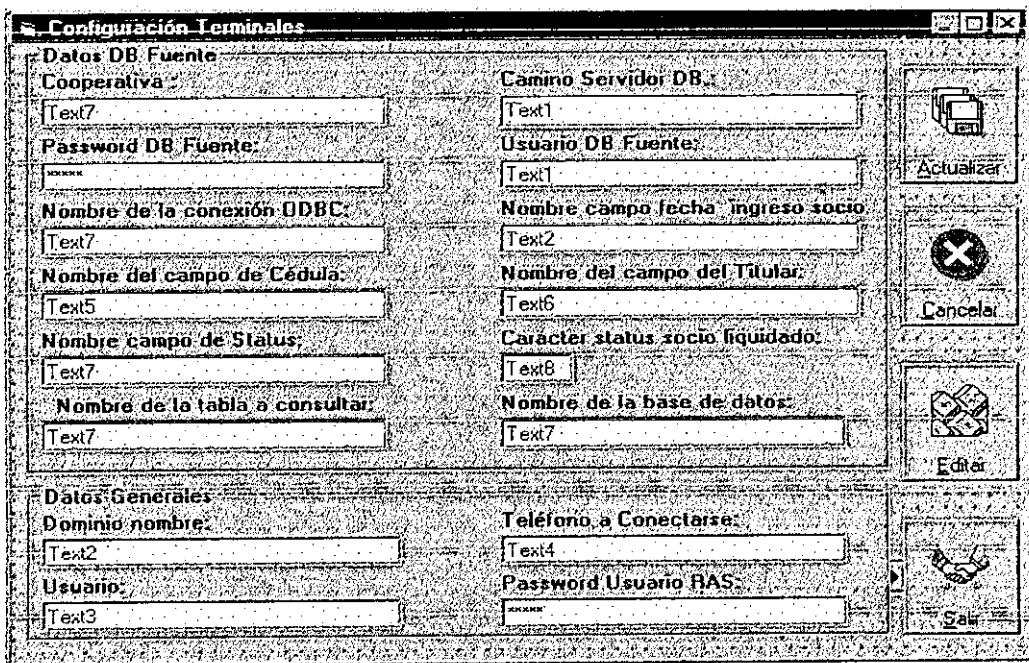
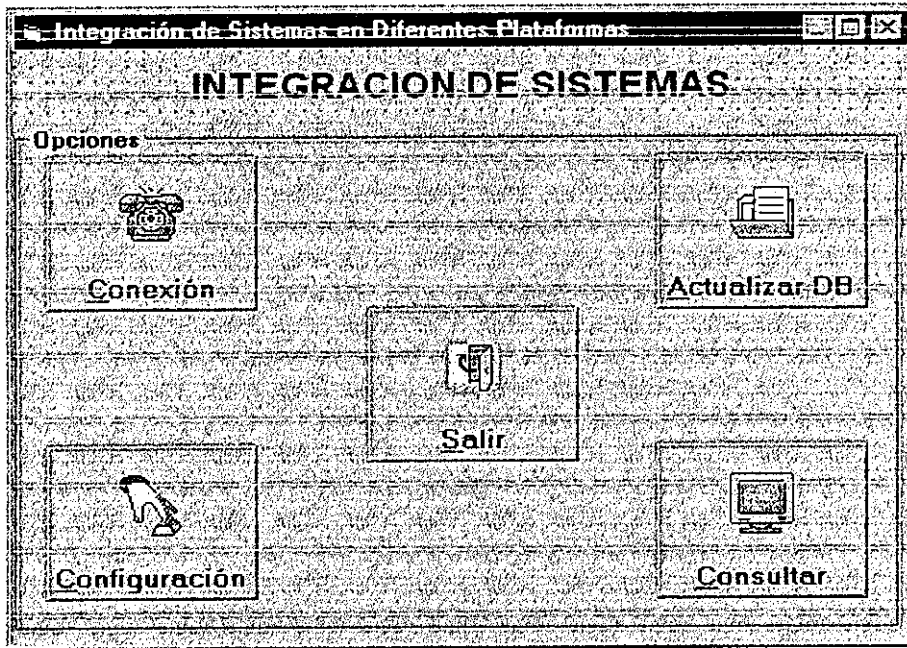
TITULO III

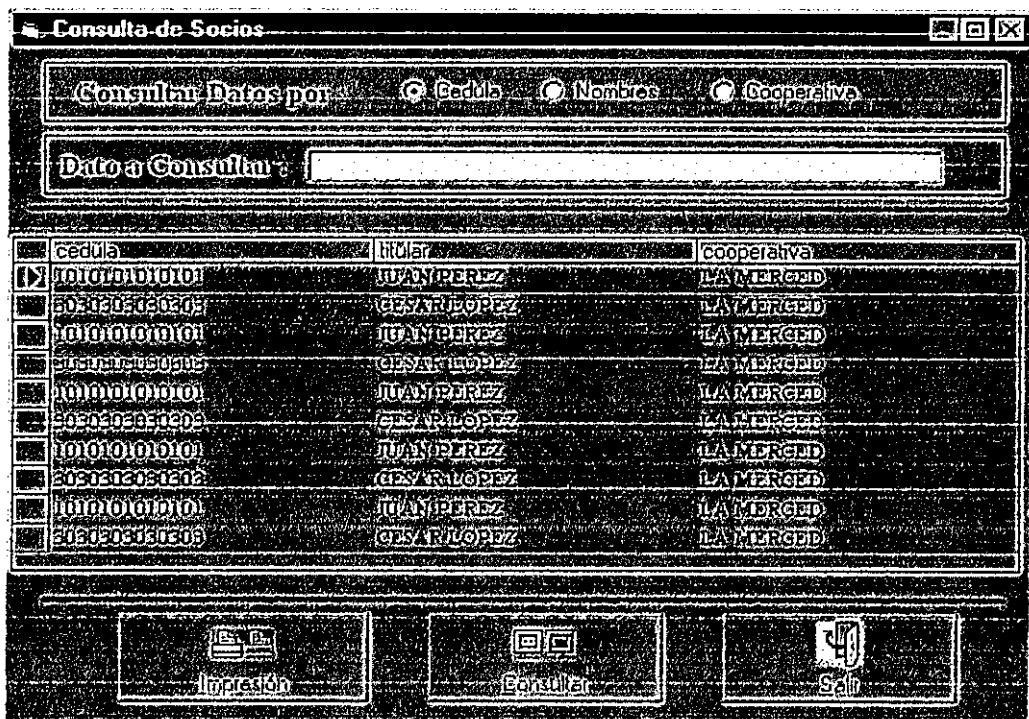
De los Socios

Art. 12.- Ninguna persona, podrá ser miembro de una cooperativa de la misma clase o línea de aquella a la esa persona o su conyuge ya pertenecen; salvo las excepciones contempladas en el Reglamento General.

Art. 13.- Tampoco podrán ser socios de una cooperativa quienes hubieren defraudado en cualquier institución pública o privada, o quienes hayan sido expulsados de otra cooperativa por falta de honestidad o probidad.

Entradas/Salidas del Sistema





INTEGRACION DE SISTEMAS NATIVOS

Listado General de Socios

Fecha de Reporte : Ambato, 07 de Octubre de 1999

Cédula	Nombre o Titular de la Cuenta	Cooperativa
3030303030303	CESAR LOPEZ	LA MERCED
3030303030303	CESAR LOPEZ	LA MERCED
3030303030303	CESAR LOPEZ	LA MERCED
3030303030303	CESAR LOPEZ	LA MERCED
3030303030303	CESAR LOPEZ	LA MERCED

Código Fuente del Sistema.

Formulario FRMMAIN

Option Explicit

```
Private Declare Function RasDial Lib "RasApi32.DLL" Alias "RasDialA" (lpRasDialExtensions As Any, _  
    ByVal lpszPhonebook As String, lprasdialparams As Any, ByVal dwNotifierType As Long, lpvNotifier As Long, lphRasConn As Long) As Long  
Dim path As Variant
```

Private Sub CmdActual_Click()

```
Dim wrkODBC, wrkJET As Workspace  
Dim rsfuente, Rsrecord, rsterminal As Recordset  
Dim dbfuente, dbbase, dbterminal As Database  
Dim fechact1 As Date  
Dim sql1, formato1 As String  
Dim camino1, odbcl, ntabla1, ncedula1, ntitular1, cstatus1, nstatus1, _  
    ndatabase1, password1, usuariodb1, nfechain1 As Variant  
Dim msj, coop1 As String  
Dim fldBucle As Field  
Dim aux, aux1, swtrans As Integer
```

Inicializa el controlador de error.

```
Err.Clear
```

'ABRIR TABLA FUENTE DE DATOS C/COOPERATIVA

' Crea un nuevo Workspace Microsoft Jet.

```
Set wrkJET = CreateWorkspace("NuevoWorkspaceJet", _  
    "admin", "", dbUseJet)
```

```
Workspaces.Append wrkJET
```

' Crea un nuevo Workspace ODBCDirect.

```
Set wrkODBC = CreateWorkspace("NuevoWorkspaceODBC", _  
    "admin", "", dbUseODBC)
```

```
Workspaces.Append wrkODBC
```

```
Set dbterminal = DBEngine.Workspaces("NuevoWorkspaceJET").OpenDatabase(path &  
"dbterminal.mdb")
```

```
Set rsterminal = dbterminal.OpenRecordset("tbldatos")
```

```
swtrans = 0
```

```
odbcl = rsterminal!nodbc
```

```
ntabla1 = rsterminal!ntabla
```

```
cstatus1 = rsterminal!cstatus
```

```
nstatus1 = rsterminal!nstatus
```

```
camino1 = rsterminal!camino
```

```
ncedula1 = rsterminal!ncedula
```

```
ntitular1 = rsterminal!nnombre
```

```
coop1 = rsterminal!cooperativa
```

```
ndatabase1 = rsterminal!ndatabase
```

```
password1 = rsterminal!password
```

```
usuariodb1 = rsterminal!usuariodb
```

```
fechact1 = rsterminal!fechact
```

```
nfechain1 = rsterminal!nfechain
```

On Error Resume Next

```
If Err.Number = 94 Then  
    fechact1 = #1/1/99#  
    Err.Clear  
End If
```

```
If Right$(camino1, 1) <> "\" Then  
    camino1 = camino1 & "\"  
End If
```

```
' Abre el objeto ODBC utilizando la información proporcionada  
' Si esta información no es suficiente,  
' podría interceptar un error en vez de ir al cuadro de diálogo  
Set dbfuente = wrkODBC.OpenDatabase(odbc1, _  
    dbDriverNoPrompt, True, _  
    "ODBC;DATABASE=" & ndatabase1 & ";UID=" & usuariodb1 & ";PWD=" _  
    & password1 & ";DSN=" & odbc1)
```

```
If Err.Number <> 0 Then  
    msj = "Error..Revise su ODBC..." & Chr(13)  
    MsgBox msj, , "Error", Err.HelpFile, Err.HelpContext  
    GoTo cierre  
End If
```

```
'ABRE LA TABLA FUENTE EN FORMA FILTRADA CON UNA CUENTAS NO LIQUIDADAS  
sql1 = "SELECT " & ncedula1 & "," & ntitular1 & "," & nfechain1 & " FROM " & ntabla1 & " where " &  
    nstatus1 & "<>" & "" & UCase(cstatus1) & " and [" & nfechain1 & "]>" & fechact1 & "" _  
Set rsfuente = dbfuente.OpenRecordset(sql1)
```

```
'CONTROLA SI SE HA CONECTADO LA TABLA CON DATOS  
aux = rsfuente.RecordCount  
If aux = 0 Then  
    msj = "No existen datos que añadir...."  
    aux = MsgBox(msj, vbOKOnly)  
    GoTo cierre  
End If
```

```
'Obtiene la posición del campo de la tabla fuente  
'y verifica si el campo cédula es diferente de char  
For Each fldBucle In rsfuente.Fields  
    If UCase(fldBucle.Name) = UCase(ncedula1) Then  
        aux = fldBucle.OrdinalPosition  
        If fldBucle.Type = dbBigInt Or fldBucle.Type = dbDouble Then  
            swtrans = 1  
        End If  
    End If  
    If UCase(fldBucle.Name) = UCase(ntitular1) Then  
        aux1 = fldBucle.OrdinalPosition  
    End If  
Next fldBucle
```

'ABRIMOS TABLA DE SERVIDOR DE DATOS

```
Set dbbase = DBEngine.Workspaces("NuevoWorkspaceJET").OpenDatabase(caminoI & "dbnafiva.mdb")  
Set Rsrecord = dbbase.OpenRecordset("tbldatos")
```

'AÑADE LOS DATOS FILTRADOS AL SERVIDOR

```
Do While Not rsfuente.EOF()
```

```
  If swtrans = 1 Then
```

```
    dbbase.Execute "INSERT INTO tbldatos (str(cedula),titular,cooperativa) "  
      & "values (" & rsfuente.Fields(aux).Value & "," & rsfuente.Fields(aux1).Value & "," & Trim(coop1)  
      & ")"
```

```
  Else
```

```
    dbbase.Execute "INSERT INTO tbldatos (cedula,titular,cooperativa) "  
      & "values (" & rsfuente.Fields(aux).Value & "," & rsfuente.Fields(aux1).Value & "," & Trim(coop1)  
      & ")"
```

```
  End If
```

```
  rsfuente.MoveNext
```

```
Loop
```

'ACTUALIZA FECHA DE ACTUALIZACION EN TERMINAL

```
rsterminal.Edit
```

```
rsterminal!fechact = Date
```

```
rsterminal.Update
```

'CERRADO DE BASES DE DATOS Y AREAS DE TRABAJO

```
GoTo cierre
```

```
msj = "Actualización Finalizada....."
```

```
aux = MsgBox(msj, vbOKOnly)
```

cierre:

```
  dbbase.Close
```

```
  dbfuente.Close
```

```
  dbterminal.Close
```

```
  wrkODBC.Close
```

```
  wrkJET.Close
```

```
  Exit Sub
```

```
End Sub
```

```
Private Sub Cmdconexion_Click()
```

```
  Dim IRetVal As Long
```

```
  IRetVal = tapiRequestMakeCall(Trim$("09479416"), App.Title, _  
    Trim$("marco"), "")
```

```
  IRetVal=RasDial(0&,vbnullstring, lprasdialparams As Any, ByVal dwNotifierType As Long, lpvNotifier As  
  Long, lphRasConn As Long) As Long
```

```
  If IRetVal <> 0 Then 'Couldn't make the call.
```

```
    'Take appropriate action
```

```
  End If
```

```
End Sub
```

```
Private Sub cmdconfig_Click()
    Frmconfig.Show vbModal
End Sub
```

```
Private Sub Cmdconsulta_Click()
    frmconsulta.Show vbModal
End Sub
```

```
Private Sub Cmdsalir_Click()
    Unload Me
End Sub
```

```
Private Sub Command2_Click()
    Dim wrkODBC As Workspace
    Dim conEditores As Connection
    Dim Rsrecord As Recordset
    Dim db As Database
    Dim sqli As String
```

```
' Crea el objeto ODBCdirect Workspace.
```

```
Set wrkODBC = CreateWorkspace("NuevoWorkspaceODBC", _
    "admin", "", dbUseODBC)
```

```
' Abre el objeto Connection utilizando la información proporcionada
```

```
' en la cadena de conexión. Si esta información no es suficiente,
```

```
' podría interceptar un error en vez de ir al cuadro de diálogo
```

```
Set db = wrkODBC.OpenDatabase("conexion", _
    dbDriverNoPrompt, True, _
```

```
"ODBC;DATABASE=conexion;UID=;PWD=;DSN=fpd26")
```

```
Set Rsrecord = db.OpenRecordset("rolpri01.dbf")
```

```
' Administrador del controlador ODBC.
```

```
MsgBox "Abriendo conexiónI..."
```

```
sqli = "SELECT rolpri01.cedula, rolpri01.nombre"
```

```
sqli = sqli & " FROM rolpri01 "
```

```
wrkODBC.Close
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
App.HelpFile = App.path & "\isndp.hlp"
```

```
path = App.path
```

```
If Right$(path, 1) <> "\" Then
```

```
    path = path & "\"
```

```
End If
```

```
End Sub
```

Formulario FRMCONSULTA.

```
Option Explicit
Dim dbbase As Database
Dim camino1 As String
Dim path As Variant
Dim Rsrecord As Recordset
```

```
Private Sub cmdconsultar_Click()
    Dim sql, msj As String
    If Option1(0).Value Then
        sql = "SELECT tbldatos.cedula, tbldatos.titular, tbldatos.cooperativa"
        sql = sql & " FROM tbldatos where tbldatos.cedula=" & """" & txtsql.Text & """"
    End If
    If Option1(1).Value Then
        sql = "SELECT tbldatos.cedula, tbldatos.titular, tbldatos.cooperativa"
        sql = sql & " FROM tbldatos where tbldatos.titular like " & """" & txtsql.Text & ""*"
    End If
    If Option1(2).Value Then
        sql = "SELECT tbldatos.cedula, tbldatos.titular, tbldatos.cooperativa"
        sql = sql & " FROM tbldatos where tbldatos.cooperativa like " & """" & txtsql.Text & ""*"
    End If
    Data1.RecordSource = sql
    If Err.Number < 0 Then
        msj = "Error..de Path del DB Servidor....." & Chr(13)
        MsgBox msj, , "Error", Err.HelpFile, Err.HelpContext
        Exit Sub
    End If
    Data1.Refresh
End Sub
```

```
Private Sub Cmdsalir_Click()
    Unload Me
End Sub
```

```
Private Sub impresion_Click()
    Dim sql, msj As String
    Dim dbbase, dbterminal As Database
    Dim qdfNuevo As QueryDef
    Dim Rsrecord As Recordset
    path = App.path
    If Right$(path, 1) <> "\" Then
        path = path & "\"
    End If
    Set dbterminal = Workspaces(0).OpenDatabase(path & "dbterminal.mdb")
    Set Rsrecord = dbterminal.OpenRecordset("tbldatos")
    camino1 = Rsrecord!camino & "\" & "dbnativa"
    dbterminal.Close
    If Option1(0).Value Then
        sql = "SELECT tbldatos.cedula, tbldatos.titular, tbldatos.cooperativa"
        sql = sql & " FROM tbldatos where tbldatos.cedula=" & """" & txtsql.Text & """"
    End If
    If Option1(1).Value Then
        sql = "SELECT tbldatos.cedula, tbldatos.titular, tbldatos.cooperativa"
```

```

    sql = sql & " FROM tbldatos where tbldatos.titular like " & "" & txtsql.Text & ""
End If
If Option1(2).Value Then
    sql = "SELECT tbldatos.cedula, tbldatos.titular, tbldatos.cooperativa"
    sql = sql & " FROM tbldatos where tbldatos.cooperativa like " & "" & txtsql.Text & ""
End If 'AGREGA UN QUERY A DBTERMINAL
    On Error Resume Next
Set dbbase = Workspaces(0).OpenDatabase(camino1)
If Err.Number <> 0 Then
    msj = "Error..de Path del DB Servidor....." & Chr(13)
    MsgBox msj, "Error", Err.HelpFile, Err.HelpContext
Exit Sub
End If
Set qdfNuevo = dbbase.CreateQueryDef("qryconsulta", sql)
rpttodos.ReportFileName = App.path & "\rptconsulta.rpt"
rpttodos.SQLQuery = sql
rpttodos.PrintReport
' Elimina el QueryDef nuevo porque esto es un ejemplo.
dbbase.QueryDefs.Delete qdfNuevo.Name
dbbase.Close
Data1.RecordSource = sql
If Err.Number <> 0 Then
    msj = "Error..de Path del DB Servidor....." & Chr(13)
    MsgBox msj, "Error", Err.HelpFile, Err.HelpContext
Exit Sub
End If
Data1.Refresh
End Sub

```

```

Private Sub Form_Load()
    Dim msj As String
    Dim aux As Integer
    On Error Resume Next
    Err.Clear
    path = App.path
    If Right$(path, 1) <> "\" Then
        path = path & "\"
    End If
    Set dbbase = Workspaces(0).OpenDatabase(path & "dbterminal.mdb")
    Set Rsrecord = dbbase.OpenRecordset("tbldatos")
    camino1 = Rsrecord!camino & "\" & "dbnativa"
    dbbase.Close
    Data1.DatabaseName = camino1
    Data1.RecordSource = "tbldatos"
End Sub

```

