

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

UNIDAD DE INGENIERIA DE SISTEMAS

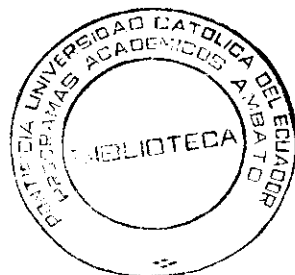
**DISERTACIÓN DE GRADO PREVIA LA OBTENCIÓN DEL
TÍTULO DE INGENIERÍA EN SISTEMAS**

**“ Integración de Sistemas Nativos en diferentes plataformas aplicado a la
Cooperativa de Ahorro y Crédito El Sagrario Ltda. “**

Jhon Wilfrido Remache Segovia.

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 El Sagrario Ltda. “**

Director :

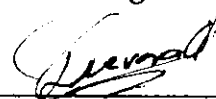


Ing. Msc. Roxana Meriño.

Revisores :



Ing. Msc. Wigberto Sánchez.



Ing. David Guevara.

Jhon Wilfrido Remache Segovia.

Ambato, 1999

Dedicatoria

*Dedicado a mi familia,
a mi esposa y a mi hija,
de manera especial a mis padres
que con su apoyo y consejos
han guiado mi trayectoria.*

Agradecimiento

*De manera especial a la
Cooperativa de Ahorro y Crédito
"El Sagrario" Ltda.,
por las facilidades brindadas*

INDICE

INTRODUCCION	1
CAPITULO I	
1. GENERALIDADES	2
1.1. Alcance y propósito del proyecto.	2
1.2. Justificación, delimitación, metas.	3
1.3. Temas generales.	4
1.3.1. Plataformas.	4
1.3.1.1. Windows NT.	5
1.3.1.2. UNIX.	6
1.3.1.3. Netware Novell 3.x y 4.x.	6
1.3.2. Estándares Nativos.	7
1.3.3. Protocolos de Comunicación.	9
1.3.3.1. Protocolos de Enlace de Datos.	9
1.3.3.2. Protocolos del Nivel de Red.	10
1.3.3.3. Protocolos del Nivel de Transporte.	12
1.3.4. Sistemas Distribuidos	13
1.3.4.1. La arquitectura NFS.	14
CAPITULO II	
2. SITUACION ACTUAL	17
2.1. Actividades y Estructura de la Institución	17
2.2. Estudio de las Actividades del Centro de Cómputo	22
2.3. Recursos Existentes	23
2.3.1. Recursos Humanos	23
2.3.2. Recursos del Hardware	27
2.3.3. Recursos del Software	29
2.3.4. Recursos de Comunicaciones	31
CAPITULO III	
3. ANALISIS DE LA RED	32
3.1. Redes de Computadoras	32
3.1.1. Tipos de Redes.	33
3.1.1.1. Red de área local (LAN, Local Area Network).	33
3.1.1.2. Redes Interconectadas (Internetwork).	33
3.1.1.3. Redes de Area Metropolitana (MAN)	34
3.1.1.4. Redes de área extensa (WAN)	34
3.2. Estudio y Análisis de la Red Existente	35
3.2.1. Redes Ethernet	35
3.2.2. Redes de área local (LAN)	36
3.2.3. Red en estrella	38
3.2.4. Sistema Operativo de Red.	38
3.2.5. Sistema Operativo UNIX	39
3.2.5.1. Historia	39
3.2.5.2. Generalidades	41
3.2.5.3. Núcleo	41
3.2.5.4. Shell o ambiente	42
3.2.5.5. Demonios	43
3.2.5.6. Procesos	43

3.2.6.	Características de Unix	44
3.2.6.1.	Multiusuario	44
3.2.6.2.	Independencia de plataforma	45
3.2.6.3.	Adaptable	45
3.2.6.4.	Para programadores	45
3.2.7.	El Ambiente Operativo Solaris	48
3.2.7.1.	Aspectos de Solaris	49
3.2.7.2.	Ventajas que provee el Ambiente Operativo Solaris	54
3.2.7.3.	Portabilidad	54
3.2.7.4.	Escalabilidad	55
3.2.7.5.	Interoperabilidad	55
3.2.7.6.	Compatibilidad	56
3.2.7.7.	Facilidad de acceso a servicios	56
3.3.	Estándares de Comunicación.	56
3.3.1.	El modelo de referencia OSI	57
3.3.1.1.	La Capa Física	58
3.3.1.2.	La capa de enlace (data link)	59
3.3.1.3.	La capa de red	60
3.3.1.4.	La capa de transporte	62
3.3.1.5.	La capa de sesión	64
3.3.1.6.	La capa de presentación	64
3.3.1.7.	La capa de aplicación	64
3.4.	Protocolos de Comunicación.	65
3.4.1.	El modelo de referencia TCP/IP	66
3.4.1.1.	La Capa Host-Red	67
3.4.1.2.	La Capa Internet	67
3.4.1.3.	La Capa de Transporte	68
3.4.1.4.	La Capa de Aplicación	70
3.4.2.	Open Network Computing ONC.	70
3.4.2.1.	Software NFS	71
3.5.	Redes de área extensa (WANs)	72
3.5.1.	Red privada	74
3.5.2.	Facilidades públicas	76
3.5.3.	Servicios de conmutación de circuitos	76
3.5.4.	Servicios de conmutación de paquetes	77
3.5.5.	Servicios de las compañías de telecomunicación para conexiones WAN	80
3.5.6.	Servicios analógicos de enlace telefónico	81
3.5.7.	Líneas privadas	82
3.5.8.	Servicios de conmutación de circuitos	84
3.5.9.	Red Digital de servicios Integrados (ISDN)	86
3.5.10.	Servicios de comunicación de paquetes	87
3.5.11.	Servicio Frame Relay	88
3.5.12.	Cómo se accede a los servicios de conmutación	88
3.5.13.	Otras tendencias	90
3.5.14.	Entradas Relacionales	91

CAPITULO IV

4.	MOTORES DE BASES DE DATOS	92
4.1	Bases de Datos Relacionales	92
4.2.	Modelos de Datos Primitivos	92
4.2.1.	Sistema de Gestión de archivos	92
4.2.2.	Bases de Datos Jerárquicas.	95
4.2.3.	Bases de Datos en Red	97
4.3.	El Modelo de Datos Relacional	100
4.3.1.	Tablas	102
4.3.2.	Claves Primarias	105
4.3.3.	Claves Foráneas	107
4.4.	Análisis de La Base de Datos existente	109
4.4.1.	Características Principales	109
4.4.1.1.	Control de concurrencia	109
4.4.1.2.	Algoritmos de I/O	109
4.4.1.3.	Bases de Datos Distribuidas	110
4.4.1.4.	Protocolos de Red	110
4.4.1.5.	Gateways	111
4.4.1.6.	Implementación del SQL	111
4.4.1.7.	Configuración de Bases de Datos	111
4.4.1.8.	Seguridad	112
4.4.1.9.	Soporte de Transacciones	112
4.4.1.10.	Tolerancia a Fallas	112
4.4.1.11.	Monitor de Rendimiento Dinámico	113
4.4.1.12.	Utilitarios Adicionales	113
4.4.2.	Beneficios	113
4.4.2.1.	Portabilidad	114
4.4.2.2.	Datos Compartidos en Forma Distribuida	114
4.4.2.3.	Sistemas Abiertos	115
4.4.2.4.	Rendimiento	115
4.4.2.5.	Control de Concurrencia	116
4.4.2.6.	Bloqueo a Nivel de la Fila.	116
4.4.2.7.	Consultas Libres de Contención	116
4.4.2.8.	Eliminación de Cuellos de Botella por I/O	117
4.4.2.9.	Características Adicionales de Rendimiento	117
4.4.2.10.	Operación Continua	118
4.4.2.11.	Soporte a Grandes Bases de Datos	118
4.5.	Recuperación de Datos	119
4.5.1.	Sentencias	119
4.5.2.	Tipos de Datos	122
4.5.3.	Funciones Internas	122
4.5.4.	Falta de datos (valores NULL)	123
4.6	Integridad de Datos	124
4.6.1.	Qué es la Integridad de Datos?	124
4.6.2.	Datos Requeridos	125
4.6.3.	Comprobación de Validez	126
4.6.4.	Integridad de Entidad	127

4.6.5.	Otras restricciones de Unicidad	128
4.6.6.	Unicidad y Valores Null	129
4.6.7.	Integridad Referencial	129
4.7.	Arquitectura cliente/servidor	132
CAPITULO V		
5.	ANALISIS Y DISEÑO DEL SISTEMA	134
5.1.	Cuadros comparativos.	134
5.2.	Métodos de Comunicación.	136
5.3.	Método de comunicación a utilizar.	142
5.3.1.	Módem.	144
5.4.	Sistema Operativo a utilizar .	146
5.4.1.	El punto de vista de los sistemas abiertos.	148
5.4.2.	Estabilidad de los sistemas.	151
5.4.3.	Seguridad.	152
5.5.	Lenguaje de Programación a utilizar	154
5.5.1.	Lenguaje estructurado SQL	155
5.5.1.1.	Consejos para escribir mandatos en SQL.	155
5.5.1.2.	Mandato SQL estándar.	156
5.6.	Base de Datos a utilizar.	159
5.7.	Requerimientos de la institución.	160
5.8.	Recursos para la elaboración del sistema.	161
5.8.1.	Recursos humanos.	161
5.8.2.	Recursos del hardware	161
5.8.3.	Recursos del software.	161
5.8.4.	Recursos de comunicaciones.	162
5.9.	Recursos que necesitan las Instituciones.	162
5.9.1.	Recursos humanos.	162
5.9.2.	Recursos del hardware.	162
5.9.3.	Recursos del software.	163
5.9.4.	Recursos de comunicación.	163
5.10.	Análisis y Diseño orientado a objetos.	164
5.10.1.	Etapas del Análisis.	164
5.10.1.1.	Definición de Clases.	164
5.10.1.2.	Definición de responsabilidades y atributos.	165
5.10.1.3.	Definiciones de los guiones.	168
5.10.2.	Fases del Diseño.	169
5.10.2.1.	Refinamiento de las Clases.	169
5.10.2.2.	Refinamiento de los Atributos y Responsabilidades.	170
5.10.2.3.	Refinamiento de las Colaboraciones.	172
5.10.2.4.	Diagrama de transición de estados.	172
5.10.2.5.	Identificación de los contratos.	175
5.10.2.6.	Grafos de colaboraciones.	177
5.10.2.7.	Refinamiento de Jerarquías.	179
5.10.2.8.	Diagrama de Colaboraciones entre Clases.	181
5.10.2.9.	Diccionario de clases del diseño.	182
5.10.2.10.	Definición de Subsistemas.	186
5.11	Descripción de los campos de la base de datos.	188
5.12.	Descripción de la aplicación	189
5.13.	Características de la aplicación	190
CONCLUSIONES		197
RECOMENDACIONES		200

INTRODUCCION



El impacto de las tecnologías de información en la productividad de las cooperativas es un tema que no deja de causar interés en nuestro medio. Todo indica que el mismo se encamina en una dirección en la cual la tecnología juega un papel preponderante, es así que las cooperativas buscan acogerse al desarrollo tecnológico y para ello deberán contar con una infraestructura adecuada, integrada y de alta seguridad.

Ante estas exigencias las cooperativas necesitan fortalecerse de uno u otro modo lo que les permitirán ubicarse de mejor manera en el medio financiero siendo una forma de fortalecimiento el integrarse entre sí, es así que las personas que formamos parte en el desarrollo de este proyecto hemos querido contribuir de alguna manera con este fortalecimiento por lo que desarrollaremos el siguiente proyecto “ ESTUDIO DE LA INTEGRACION DE SISTEMAS NATIVOS EN DIFERENTES PLATAFORMAS “.

El proyecto tiene como objetivo desarrollar un software piloto que permita compartir información entre las principales cooperativas de la ciudad de Ambato siendo estas: El Sagrario Ltda., Oscus Ltda., San Francisco Ltda., Cámara de Comercio Ambato y la Meced Ltda.

El presente estudio se realizará en la Cooperativa de Ahorro y Crédito "El Sagrario" Ltda., en la misma que analizaremos su sistema operativo, protocolos de comunicación, motor de bases de datos, redes entre otros

CAPITULO I

1. GENERALIDADES

1.1. Alcance y propósito del proyecto.

El principal problema que se presenta en las diferentes Cooperativas de Ahorro y Crédito de la Provincia, es el de no poder establecer una comunicación electrónica de sus equipos de computación para el intercambio de datos.

Se conoce que aunque se desenvuelvan en el mismo campo, cada Cooperativa de Ahorro y Crédito es un mundo diferente con relación a las otras, teniendo cada una de ellas su equipo tanto humano, hardware, software y por supuesto; de organización.

El proyecto a desarrollarse estudiará los siguientes puntos:

- Análisis de los recursos informáticos.
- Redes instaladas.
- Bases de Datos existente.
- Formas de Comunicación entre computadoras
- Principales Protocolos existentes.
- Análisis y diseño de un software piloto que ejemplifique el estudio propuesto.

1.2. Justificación, delimitación, metas.

En la actualidad se debe sacar el máximo provecho tecnológico de los recursos que posee una Institución, en la mayoría de Cooperativas de nuestra ciudad existen los medios técnicos necesarios para la integración electrónica entre ellas. Se debe indicar que cada una de las instituciones trabaja en un sistema operativo diferente, con distintos motores de bases de datos, siendo importante encontrar el medio más adecuado, económico y eficaz para la comunicación entre ellas, así como la forma de migrar los datos.

Se busca migrar datos entre Cooperativas, para lograr consultar los datos de los socios y hacia cumplir con lo dispuesto en el artículo 12 y 13 del reglamento de Ley de Cooperativas¹, el cual prohíbe que los socios pertenezcan a dos Cooperativas a la vez.

Para la elaboración del presente proyecto, por la magnitud y complejidad del mismo se ha conformado un equipo compuesto por cinco personas, en el cual cada integrante se dedicará por completo al estudio y diagnóstico individual en cada cooperativa.

En cada Institución, se estudiará su organización y estructura, así como todos los recursos con que cuentan tanto humanos, hardware, software y comunicaciones. Se examinará sus necesidades y limitaciones y se entregará un diagnóstico de las situaciones actuales de cada una de ellas.

¹ Ver anexo 1

1.3. Temas generales.

1.3.1. Plataformas.

Plataforma tiene varios significados en el mundo de las computadoras y redes:

- Un sistema de computadora basado en un procesador específico, como una plataforma basada en Intel.
- Un entorno de sistema operativo, como una plataforma NetWare.
- Una red corporativa, donde la red en sí misma aparece como una clase de plataforma de conexión y trabajo sobre la cual se pueden conectar diversos dispositivos.

En este punto se tratará acerca de aquellas plataformas relacionadas con los sistemas operativos de redes.

Sistema Operativo de red.

El sistema operativo de red constituye un componente primario de la red y determina su flexibilidad a la hora de soportar aplicaciones que permitan aumentar la eficiencia y mejorar el soporte del cliente. Su selección, además, marca el camino a seguir por los usuarios en el proceso de toma de decisiones tan vitales como la instalación del hardware, cliente servidor y la elección de productos software adicionales.

Los sistemas operativos constituyen una parte importante dentro del área del software para el trabajo de redes, básicamente una red dependerá del alcance funcional que permita el sistema operativo de trabajo. Aunque hay muchos sistemas operativos, sólo alguno de ellos ofrecen el tipo de funcionalidad completa que desean los usuarios. Entre dichos productos los más implantados son NetWare, UNIX, Windows NT. Las cooperativas de ahorro y crédito objeto de nuestro estudio trabajan con los sistemas operativos más populares en nuestro medio, los mismos que se detallan a continuación:

1.3.1.1. Windows NT.

Desde su lanzamiento en 1993 como Windows NT 3.1, el sistema operativo Windows NT se ha mostrado como una promesa para el establecimiento de un estándar en sistemas operativos de red. Windows NT es un sistema operativo de 32 bits con multitareas priorizadas y protección de memoria, además de soporte para multiproceso simétrico y trabajo en red, todo con una inmejorable interfaz gráfica de usuario. La capacidad de Windows NT para acceder a los procesadores de 32 bits le permite trabajar con grandes números y direcciones de memoria, así como grandes cantidades de instrucciones. Por encima de todo consigue rendimiento que es el resultado de haber planeado la combinación de la capacidad de ejecución del procesador, la transferencia de datos y acceso a memoria. La Cooperativa de Ahorro y Crédito de la Cámara de Comercio de Ambato trabaja en esta plataforma.

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:

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.

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:

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.

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)

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:

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.

SPX (Intercambio secuencial de paquetes, Sequenced Packet Exchange)

Protocolo de nivel de transporte orientado a conexión establece una sesión entre dos sistemas de comunicación para el control de flujo secuenciamiento y control de errores de los datos que están siendo transmitidos usando el protocolo IPX (Capa de Red). Su funcionamiento es similar al protocolo TCP. SPX se utiliza para establecer una conexión relativamente permanente entre las estaciones de trabajo de su gestor y un servidor, con el objetivo de proporcionar información continua en tiempo real.

1.3.4. Sistemas Distribuidos

Dentro de la evolución del mundo informático sobresalen dos avances tecnológicos, el primero fue la el desarrollo de los poderosos microprocesadores y el segundo, fue la invención de las redes de área local de alta velocidad (LAN, local area network). El cual

permitió conectar docenas e incluso cientos de máquinas de forma que pudiesen transferir pequeñas cantidades de información entre ellas durante un milisegundo o un tiempo parecido, como resultado neto de estas dos tecnologías es que hoy en día no solo es posible reunir sistemas de cómputo compuestos por un gran número de CPU conectados mediante una red de alta velocidad, los cuales reciben el nombre genérico de **sistemas distribuidos**, en contraste con los sistemas centralizados.

El sistema distribuido es aquel que se ejecuta en una colección de máquinas sin memoria compartida, pero que aparece ante sus usuarios como una sola computadora, además debe controlar las estaciones de trabajo en lo individual, servidores de archivo y también de la comunicación entre ellos.

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.

seguridad débil. Sin embargo, existe un considerable interés mundial por su construcción e instalación.

- Servicio Odontológico

Ubicación Geográfica

Matriz:	Ambato
Ubicación:	Provincia del Tungurahua Ciudad Ambato Sector céntrico de la ciudad calle Sucre y Quito esquina
Sucursal 1:	Riobamba
Ubicación:	Provincia del Chimborazo Ciudad Riobamba Calle García Moreno entre Guayaquil y 10 de Agosto
Sucursal 2:	Latacunga
Ubicación:	Provincia del Cotopaxi Ciudad Latacunga

Finalidad que persigue la empresa

La Cooperativa es una asociación autónoma de personas que se han unido con la finalidad de satisfacer sus necesidades y aspiraciones económicas, sociales y culturales en común mediante una empresa de propiedad conjunta y de gestión democrática.

Nuestra obligación es responder eficientemente a nuevos desafíos, expectativas y necesidades de los socios.

Objetivos y políticas de la empresa

Objetivos:

- Servir a la comunidad es el objetivo de la Cooperativa
- Planificar y realizar actividades de beneficio social, a través de una empresa manejada en común y formada por la aportación económica, pero también intelectual y moral de sus miembros.
- Promover la cooperación económica entre los asociados.
- Otorgar préstamos a los socios que requieran de este servicio.
- Proporcionar a sus asociados mayor capacitación en lo económico y social.
- Brindar servicios de calidad, velando por la salud de sus asociados.
- Hacer contacto con otras entidades similares dentro y fuera del país.
- Obtener fuentes de financiamiento interno y externo.
- Contribuir al desarrollo socio-económico del país, a través de una efectiva gestión de representatividad y entrega de servicios de calidad a los usuarios, asegurando el crecimiento sostenido de la Cooperativa.

Políticas:

- Cumple con una acción de servicio y sin fin de lucro.
- El pago de mejores intereses en ahorros y en depósitos a plazo fijo.
- La Cooperativa de Ahorro y Crédito "El Sagrario" Ltda., sustenta su accionar institucional en la unidad e integridad de sus organismos internos, en la idoneidad profesional de sus funcionarios y empleados y en la activa participación de todos sus socios.
- Adhesión Libre: El ingreso y la salida de los socios depende de su voluntad, sin que exista imposición legal o de hecho que obligue a cualquier persona a asociarse a las cooperativas o permanecer contra su voluntad dentro de ellas. Por lo tanto a una Cooperativa se ingresa libre y espontáneamente así como de igual manera se retira.

Existen desde luego requisitos para afiliarse como para dejar de ser socio.

- Control Democrático: En las cooperativas todos los asociados tienen iguales obligaciones e iguales derechos y esa igualdad se manifiesta en el ejercicio del voto para la elección a los cargos directivos.
- Interés Limitado al Capital: Es evidente que las cooperativas requieren de capitales para organizar, incrementar y desarrollar la empresa económica, para que pueda cumplir los propósitos y servicios sociales del grupo y la comunidad.
- Mientras mayores son los saldos de esos capitales, es evidente que la empresa es más sólida y solvente y además será más fácil el cumplimiento de sus aspiraciones.
- De los Excedentes: Los excedentes totales que se obtienen en la Cooperativa de Ahorro y Crédito "El Sagrario" Ltda. una parte se acredita como bonificación especial a los

Certificados de Aportación y la otra va al patrimonio de la institución, siendo la Asamblea General la que decide al respecto.

- *Educación Cooperativa:* La educación cooperativa es la transformación del individuo para hacerlo capaz de una acción social mas consciente.
- *Integración Cooperativa:* Con el objeto de servir mejor a los intereses de sus miembros y la comunidad, todas las organizaciones cooperativas deben relacionarse con otras cooperativas a nivel local, nacional e internacional. Esta tendencia hacia la integración responde a la naturaleza cooperativa, pues se basa en la unidad de esfuerzos y la colaboración recíproca.

preparados en las funciones a ellos encomendadas permitiendo de esta manera que esta institución ofrezca un mejor servicio a sus asociados.

De igual manera posee actualmente un manual de funciones y procedimientos que está orientado a establecer las acciones que cada uno de los miembros del Departamento de Sistemas debe cumplir tanto en los puestos que en la actualidad existen como aquellos que son necesarios crearlos para el normal funcionamiento de este Departamento.

Cabe destacar que el personal asignado a esta área es evaluado en forma semestral.

El grupo que forma parte de este Departamento se encuentra distribuido en sus tres oficinas de la siguiente manera:

En la Oficina Matriz Ambato

Un Jefe de Sistemas,

Un Analista Programador,

Una digitadora

En la Sucursal Riobamba

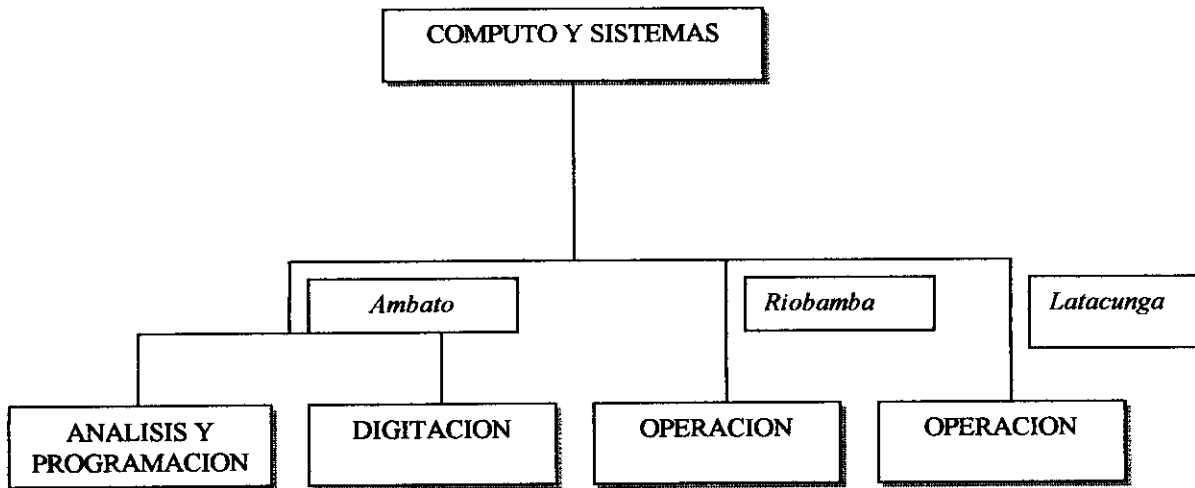
Un Operador del Sistema

En la Sucursal Latacunga

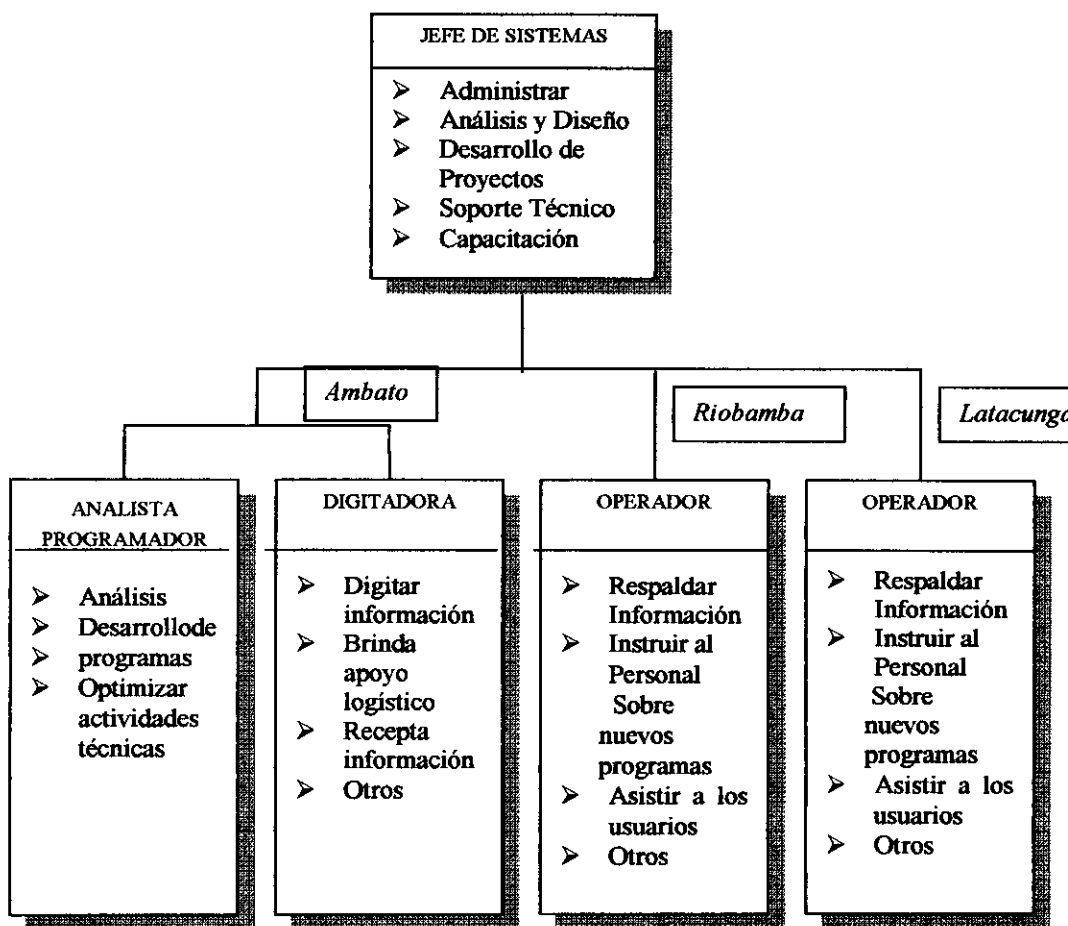
Un Operador del Sistema

ESTRUCTURA ORGANIZACIONAL DEL AREA INFORMATICA

ORGANIGRAMA ESTRUCTURAL



ORGANIGRAMA FUNCIONAL



2.3.2. Recursos del Hardware

La Cooperativa cuenta con la siguiente configuración de equipos:

Matriz Ambato:

- 1 Servidor SUN Microsystems SPARC Station 5 modelo 85 con:
- 64 MB en memoria RAM
- 1 Unidad de CD ROM interno
- 1.05 GB en almacenamiento de disco interno
- 1.05 GB en almacenamiento de disco externo
- 4.02 GB en almacenamiento de disco externo
- 1 Unidad de tape de 525 MB
- 1 Monitor a color de 17"

- 1 Terminal Server para 16 puertos

- 8 Terminales WYSE-150 con teclado

- 1 UPS Fortress LI2KF 2000 VA
- 3 Microcomputadores DTK
- 3 Microcomputadores HP Vectra VE
- 7 Microcomputadores Compaq

- 11 Impresoras matriciales
- 5 Impresoras de inyección a tinta
- 1 Notebook Acer

Sucursal Riobamba

- 1 Servidor SUN Microsystems SPARC Station 5 modelo 70 con:
- 32 MB en memoria RAM
- 1 Unidad de CD ROM interno
- 1.05 GB en almacenamiento de disco interno
- 4.02 GB en almacenamiento de disco externo
- 1 Unidad de tape de 525 MB
- 1 Monitor a color de 17"

- 1 Terminal Server para 16 puertos

- 7 Terminales WYSE-150 con teclado

Sucursal Latacunga:

- 1 Servidor SUN Microsystems SPARC Station Classic con:
- 24 MB en memoria RAM
- 1 Unidad de CD ROM interno

- 1.05 GB en almacenamiento de disco interno
- 1 Unidad de tape de 525 MB
- 1 Monitor a color de 15"

- 1 Terminal Server para 16 puertos

- 4 Terminales WYSE-150 con teclado

2.3.3. Recursos del Software

Los recursos a nivel de Software con los que cuenta la Cooperativa son:

Matriz Ambato:

- Sistema Operativo Unix SOLARIS para el Sevidor
- Sistema Operativo Windows para los Microcomputadores
- Procesador de Palabras WORD
- Hoja Electrónica Excel

- Base de Datos ORACLE

- Sistema Aplicacional para Cooperativas

Sucursal Riobamba:

- Sistema Operativo Unix SOLARIS para el Sevidor
- Sistema Operativo Windows para los Microcomputadores
- Procesador de Palabras WORD
- Hoja Electrónica Excel

- Base de Datos ORACLE

- Sistema Aplicacional para Cooperativas

Sucursal Latacunga:

- Sistema Operativo Unix SOLARIS para el Sevidor
- Sistema Operativo Windows para los Microcomputadores
- Procesador de Palabras WORD
- Hoja Electrónica Excel

- Base de Datos ORACLE

- Sistema Aplicacional para Cooperativas

2.3.4. Recursos de Comunicaciones

Hardware

- 1 Modem Telenetics TCI 442B de 14400 bps
- 1 Hub 3COM OFFICE CONNECT 8TP/BNC
- 1 Conexión telefónica digital

Software

- Administrador de redes NFS, TCP/IP, ONC

Otros

- 1 Conexión a Internet

ejemplo). Un ejemplo de protocolo de sistemas distribuidos podría ser el NFS (Network File System) que permite acceso a ficheros remotos de forma transparente.

3.1.1. Tipos de Redes.

Existen diversos tipos de redes, los cuales vamos a citar algunos de ellos:

3.1.1.1. Red de área local (LAN, Local Area Network).

Es una pequeña red (de 3 a 5 nodos) normalmente localizados en un solo edificio o grupo de edificios pertenecientes a una organización.

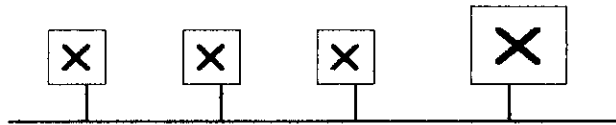


Figura 3.1. Red de área Local

3.1.1.2. Redes Interconectadas (Internetwork).

Se pueden conectar dos o más redes para formar un sistema en red que cubra toda una empresa, También puede dividirse una red extensa en varias redes más pequeñas para optimizar el rendimiento y la gestión.

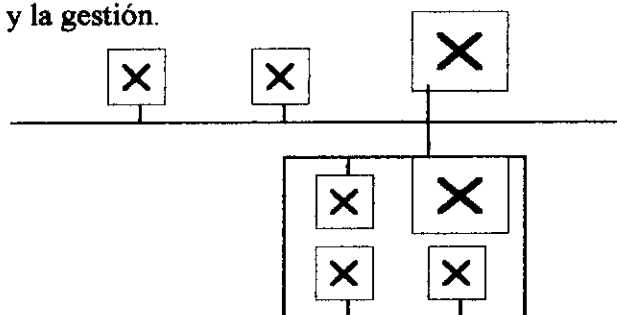


Figura 3.2. Redes Interconectadas.

3.1.1.3. Redes de Area Metropolitana (MAN)

Se trata de un conjunto de redes de área local interconectadas dentro de un área específica, como un campus, un polígono industrial o una ciudad. Se debe utilizar una base de cableado o sistemas de conexión especiales a alta velocidad, como una compañía telefónica, para conectar las redes al sistema interconectado

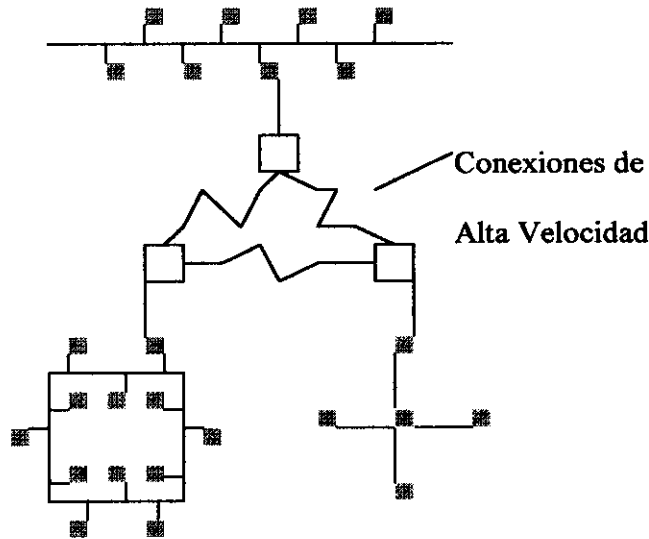


Figura. 3.3. Redes de área Metropolitana.

3.1.1.4. Redes de área extensa (WAN)

Se trata de una red que cubre varios países o incluso el mundo, por ejemplo los sistemas de reservas de las líneas aéreas.

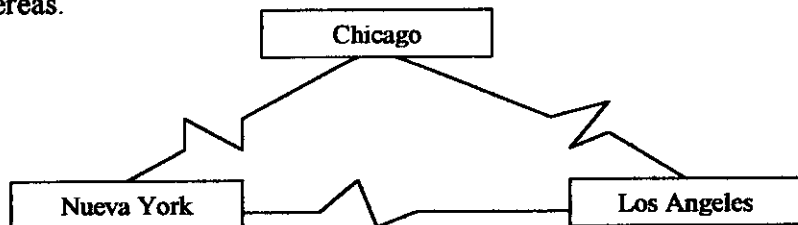


Figura.3.4. Redes de área Extensa.

3.2. Estudio y Análisis de la Red Existente

La Cooperativa de Ahorro y Crédito "El Sagrario" Ltda., trabaja con dos tipos de redes que son:

- **ETHERNET**
- **LAN**

3.2.1. Redes Ethernet

Ethernet, al que también se conoce como IEEE 802.3, es el estándar más popular para las LAN que se usa actualmente. El estándar 802.3 emplea una topología lógica de bus y una topología física de estrella o de bus. Ethernet permite datos a través de la red a una velocidad de 10 Mbps. Ethernet usa un método de transmisión de datos conocido como Acceso Múltiple con Detección de Portadora y Detección de Colisiones (CSMA/CD). Antes de que un nodo envíe algún dato a través de una red Ethernet, primero escucha y se da cuenta si algún otro nodo está transfiriendo información. De no ser así, el nodo transferirá la información a través de la red. Todos los otros nodos escucharán y el nodo seleccionado recibirá la información. En caso de que dos nodos traten de enviar datos por la red al mismo tiempo, cada nodo se dará cuenta de la colisión y esperará una cantidad de tiempo aleatoria antes de volver a hacer el envío. La topología lógica de bus de Ethernet permite que cada nodo tome su turno en la transmisión de información a través de la red. Así, la falla de un solo nodo no hace que falle la red completa.

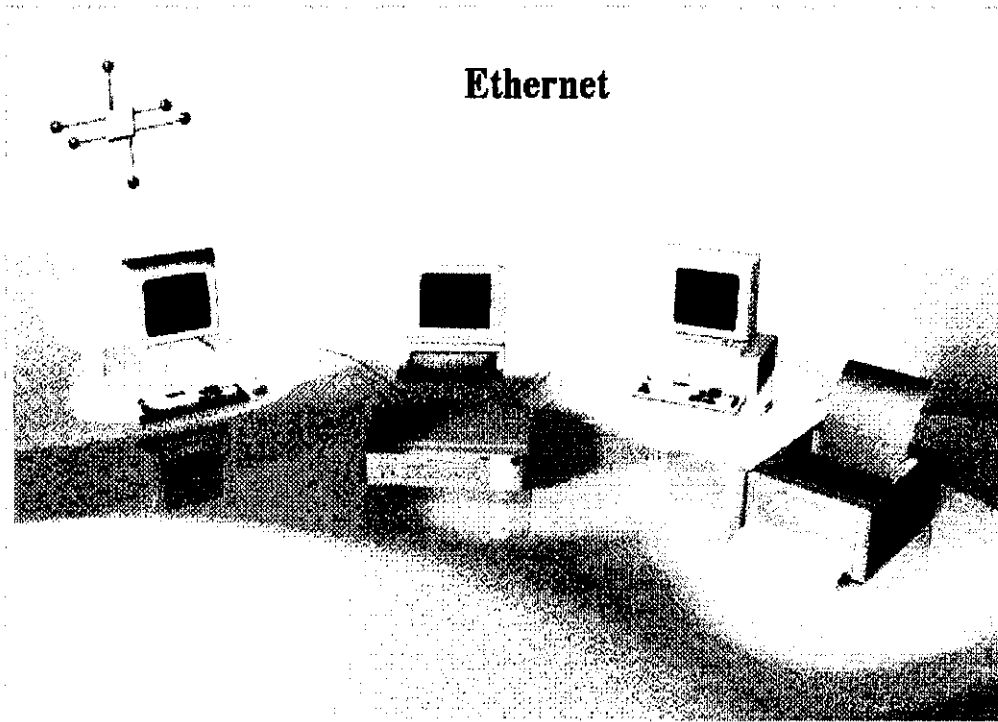


Figura 1. Imagen de una red Ethernet.

Aunque CSMA/CD es una forma rápida y eficiente para transmitir datos, una red muy cargada podría llegar al punto de saturación. Sin embargo, con una red diseñada adecuadamente, la saturación rara vez es preocupante. Existen tres estándares de Ethernet, 10BASE5, 10BASE2, y 10BASE-T, que definen el tipo de cable de red, las especificaciones de longitud y la topología física que debe utilizarse para conectar nodos en la red.

3.2.2. Redes de área local (LAN)

Las características de las redes LAN son las siguientes:

- Tecnología broadcast: medio compartido
- Cableado específico, instalado normalmente a propósito
- Velocidad de 1 a 100 Mbps
- Extensión máxima de 3 KM (FDDI llega a 200 Km)

El alcance limitado de las LANs permite saber el tiempo máximo que un paquete tardará en llegar de un extremo a otro de la red, lo cual permite aplicar diseños que de otro modo no serían posibles, y simplifica la gestión de la red.

Como consecuencia del alcance limitado y del control en su cableado, las redes locales suelen tener un retardo muy bajo en las transmisiones (decenas de microsegundos) y una tasa de errores muy baja.

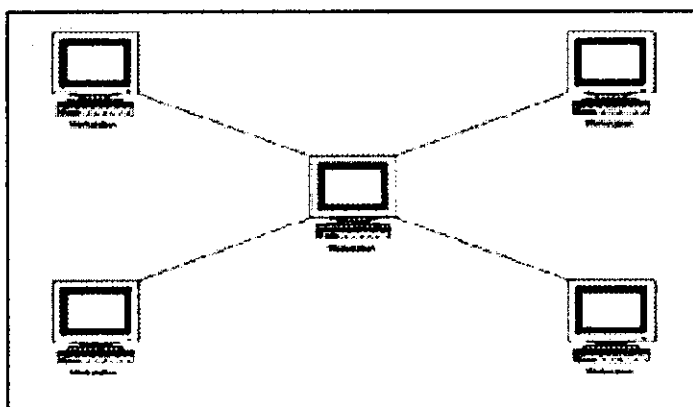
La topología básica de las redes locales suele ser de bus (p. ej. Ethernet) o de anillo (Token Ring o FDDI). Sin embargo, la utilizada en esta institución es la de estrella.

En épocas recientes se ha popularizado una técnica para aumentar el rendimiento de las redes locales, que consiste en dividir una LAN en varias más pequeñas, con lo que el ancho de banda disponible para cada uno es mayor; las diversas LANs así formadas se interconectan en un equipo especial denominado conmutador LAN (o LAN switch); en casos extremos se puede llegar a dedicar una red por equipo, disponiendo así de todo el ancho de banda para él.

La topología usada en la Cooperativa de Ahorros y Crédito el "Sagrario" es de tipo estrella.

3.2.3. Red en estrella

Uno de los tipos más antiguos de topologías de redes es la estrella, la cual usa el mismo método de envío y recepción de mensajes que un sistema telefónico, ya que todos los mensajes de una topología LAN en estrella deben pasar a través de un dispositivo central de conexiones conocido como concentrador de cableado, el cual controla el flujo de datos.



Red Tipo Estrella

3.2.4. Sistema Operativo de Red.

El objetivo del S.O. es ofrecer una abstracción de los recursos hardware para que el software de aplicación no necesite conocer las particularidades de cada máquina.

Disco Conjunto de ficheros con nombres y operaciones de abrir, cerrar, leer, escribir, etc.

Memoria RAM: Espacio de direcciones virtuales.

Teclado + pantalla: Ficheros especiales (terminal)

Puertos: Ficheros especiales (Serie, paralelo, altavoz,...)

Otra función de un S.O. multitarea es el reparto de recursos:

procesador, memoria física, disco, comunicaciones, etc.

La Cooperativa de Ahorro y Crédito "El Sagrario" Ltda., trabaja con dos Sistemas Operativos que son:

- Unix System V Release 4
- Solaris 2.5

3.2.5. Sistema Operativo UNIX

3.2.5.1. Historia

Hacia 1962, se emprendió el desarrollo de un sistema operativo Multiusuario llamado MULTICS. La idea era tener una "planta de computación" enorme, como una planta eléctrica, para todo el mundo en Boston.

MULTICS llegó a funcionar, pero fue demasiado ambicioso para la época. Contribuyo muchas ideas al mundo de sistemas operativos. De hecho, un computista de Bell Labs, Ken Thompson, encontró una DEC PDP-7 (una de las primeras "minicomputadoras", del tamaño de un escritorio) y sobre esta plataforma desarrollo un sistema, que un poco en chiste se llamo "UNICS" por su contraste con el enorme MULTICS.

Cuando surgen las estaciones de trabajo, debido a la iniciativa de SUN y Apollo en 1982 (y adelante), Unix estaba disponible y fue rápidamente portado a estas arquitecturas, de modo que es el sistema operativo de rigor para las estaciones de trabajo.

3.2.5.2. Generalidades

El S.O. Unix se encarga de controlar asignar los recursos físicos del ordenador (hardware) y de planificar tareas . Podemos establecer tres elementos principales dentro de éste S.O.:

- El núcleo del sistema operativo (kernel), el escalón más bajo que realiza tareas tales como el acceso a los dispositivos (terminales, discos, cintas...).

- El intérprete de comandos (shell) es el interfase básico que ofrece UNIX de cara al usuario. Además de ejecutar otros programas, posee un lenguaje propio así como numerosas características adicionales.

- Utilidades "de fabrica"; normalmente se trata de programas ejecutables que vienen junto con el Sistema; algunas de ellas son:
 - Compiladores: C, assembler y en algunos casos Fortran 77 y C++.
 - Herramientas de edición: Editores (viex), formateadores (troff), filtros...
 - Soporte de comunicaciones: Herramientas basadas en TCP/IP (telnet,ftp ..)
 - Programas de Administración del Sistema (sysadm, sa, va....)
 - Utilidades diversas y juegos (éste último se suele instalar aparte).

3.2.5.3. Núcleo

Los inventores de Unix tenían en mente un modelo atómico de un sistema operativo. Alrededor de un "núcleo" giraba, como una capa electrónica, un programa que interactuaba con el usuario, llamado "shell" (por eso mismo de la capa electrónica). Al núcleo se le llama "kernel", completando la analogía.

El modelo atómico vale, pero requiere de capas electrónicas adicionales. Tendríamos, pues, los siguientes elementos:

El núcleo (Kernel). Esto hace las funciones esenciales:

- Manejo de las interrupciones
- Control de procesos
- Control de dispositivos (devices).
- Manejo de los sistemas de archivos

La capa "burocrática" de procesos administrativos

- La capa de los "demonios"
- La capa de los "shells" que interactúan con el usuario.
- La capa de procesos iniciados por los usuarios a través del shell .

3.2.5.4. Shell o ambiente

Un shell (hay muchos) es un programa que interactúa con el usuario (a diferencia de un demonio) y "ayuda" al usuario a correr programas y en general, a sacarle provecho al sistema. No dudo que hay usuarios que opinan que la función del shell es hacerle la vida lo más difícil posible, pero esa probablemente no sea la intención del autor del shell.

No todos los shells son iguales y es casi más fácil cambiar de sistema operativo que cambiar de shell. Los shells más famosos son:

Bourne Shell (sh), el original de ATT.

C-shell (csh), invento de Berkeley. Favorito del IVIC.

K-shell o Korn Shell (ksh), mejora de sh. Favorito del autor.

Bourne-again shell (bash), mejora de sh, de Gnu. Excelente.

Tcsh, mejora de csh. o así dicen.

3.2.5.5. Demonios

Un demonio es un programa, cuyo pintoresco nombre se debe al "demonio de Maxwell" en la termodinámica. Recordamos que este último era una figura que acechaba las moléculas de un gas, esperando a una que cumpliera ciertas condiciones, que luego era apartada a algún rincón del gas. De la misma manera, un demonio en Unix espera un acontecimiento especial. Puede ser un mensaje por la red, una interrupción de un dispositivo, o lo que sea. Cuando este acontecimiento ocurre, el demonio se despierta y toma acción.

3.2.5.6. Procesos

Un proceso es un programa que corre independientemente; es decir, el programa compete con otros procesos por los recursos de la maquina (memoria, CPU, disco, etc.). Unix

mantiene una tabla de procesos donde mantiene la información necesaria para cada proceso.

Uno se puede formar una idea del contenido de la tabla de procesos dando el comando ps. Cada proceso tiene un numero (el PID); ese numero identifica al proceso en todas sus interacciones con el sistema. En cualquier momento, solo un proceso puede usar la maquina; sin embargo Unix divide al tiempo en "tajaditas" (de unos milisegundos cada una) y a cada proceso le regala su tajadita para ejecutar. Cuando el proceso tiene que escribir o leer del disco, u otro dispositivo, pierde su tajadita y otro proceso pasa a ejecutar. Así, cada proceso parece utilizar la maquina simultáneamente (multitasking o multiprocesamiento).

3.2.6. Características de Unix

3.2.6.1. Multiusuario

Un usuario en Unix es (1) un numero o UID y (2) una residencia, o directorio, en una maquina. En general, la idea es que un usuario es una persona (pero puede ser una organización).

Unix garantiza la separación total de los archivos de los usuarios. Cada archivo puede ser individualmente protegido, o desprotegido, por el dueño. Conversamente, cada archivo en Unix tiene un dueño (un usuario), quien controla el destino de ese archivo.

Un usuario muy importante es root, o el superusuario. El superusuario tiene todos los poderes sobre todos los archivos para poder administrar el sistema.

3.2.6.2. Independencia de plataforma

La independencia de plataformas de Unix se debe a dos factores:

- Unix esta escrito casi todo en lenguaje C
- Los conceptos de Unix son totalmente independientes de una arquitectura cualquiera.

Por supuesto, nada es perfecto: a veces Unix muestra su "sabor a PDP". Pero en general, Unix es una abstracción muy poderosa, que se presta bien a todas las arquitecturas.

3.2.6.3. Adaptable

Unix se ha adaptado tanto a los nuevos CPUs (los RISC) como los nuevos dispositivos. La interfase ha pasado de terminales ASCII a pantallas gráficas y ratones con relativamente poco choque cultural.

Las razones de esto son en parte técnicas (la capacidad de incorporar cualquier dispositivo por medio de un device driver) y parte porque Unix en el fondo es conceptualmente simple, con conceptos muy claros.

3.2.6.4. Para programadores

Es muy importante recordar que Unix fue escrito por programadores para programadores. Esto tiene tanto ventajas como desventajas.

Una ventaja del parentesco de Unix es que para desarrollar software, Unix es una plataforma ideal. Tiene muchas herramientas de desarrollo (make, RCS, SCCS, xdb, rdb,...).

La desventaja del "enfoque Unix" es que es difícil de aprender. Los comandos tienen nombres infernalmente cortos y/o oscuros

Breve Historia de Unix

- Ken Thompson, fines de 196x en Bell Labs
 - PDP-7 inutilizada sin SW
 - Assembler
 - Basado en conceptos MULTICS
 - Interactivo
 - Llamado UNICS
- Dennis Ritchie, misma época: BCPL->B->C
- Ritchie y Thompson, 1973: Unix en C
- Portes a otras maquinas 1974-presente.
- Licencias a las universidades, con fuente
- UC Berkeley y las versiones de BSD hasta 4.3
- ATT y versiones I-VII, sistemas III-V.

- Detalles.

Características de Unix

- **Multiusuario.**
 - Separación de los archivos de cada usuario
 - Disco, tiempo de CPU, impresoras compartidas
 - No-interferencia con otros usuarios
- **Multiproceso o multitarea o multitasking.**
- (Casi) independiente de plataforma.
- Adaptable.
 - Redes (TCP/IP y otras)
 - Nuevo hardware (CD, DAT, pantallas,...)
 - Multimedia
- Para programadores
 - Difícil de aprender
 - Fácil de usar
- **Sistema Operativo SOLARIS**
 - El Ambiente Operativo Solaris

- Aspectos de Solaris
- Ventajas que provee el Ambiente Operativo Solaris
- Facilidad de Acceso a Servicios

3.2.7. El Ambiente Operativo Solaris

Solaris 2.5.1, solución de computación distribuida de SunSoft's (TM), es un entorno operativo empresarial o para Internet, que se caracteriza por una integración total con servicios de Internet y Networking; viene con herramientas de desarrollo para construir aplicaciones de comunicación e información avanzadas; tiene una librería de más de 10,000 aplicaciones para todos los líderes del mercado de software.

El centro del Solaris es un UNIX System V Release 4 (el estándar líder en la industria de los sistemas abiertos). Se le menciona como Entorno Operativo y no como Sistema Operativo por que se trata de un gran conjunto de herramientas (entre las que se encuentra el sistema operativo en sí), las cuales fueron desarrolladas con la finalidad de aprovechar al máximo el rendimiento del hardware.

Solaris es el primer entorno operativo UNIX multiplataforma, que ofrece un entorno de trabajo y desarrollo de 32 bits homogéneos. La compatibilidad funcional cruzada permite a SunOS versión 5.5 correr sobre plataformas SPARC, UltraSparc, Intel x86, Pentium, Pentium Pro y Power PC; así como para servidores de red o servidores Intranet/Internet y

sistemas de escritorio. Es decir, abarca desde una sencilla PC hasta el más sofisticado sistema multiprocesador.

Este entorno operativo le brinda todas las ventajas de multitarea, manejo de redes, multiprocesamiento simétrico, librerías multihilo, compatibilidad ABI y la ya conocida capacidad multithreading (la capacidad de realizar operaciones simultáneas dentro de una misma aplicación).

El Solaris para x86 (plataforma Intel), es exactamente igual en todas sus características al Solaris para estación de trabajo. Con esto se esta abarcando sistemas que ejecutan SPARC así como también Intel 486 y Pentium. La simetría entre ambos sistemas garantiza que las aplicaciones desarrolladas para uno de ellos funcionen en el otro con solo recompilar el código fuente.

3.2.7.1. Aspectos de Solaris

El ambiente operativo Solaris (TM) 2.5, esta comprendido de SunOS (TM) versión 5.5 con ONC (TM), OpenWindows (TM), ToolTalk (TM), DeskSet (TM), y OPEN LOOK® así como también otras utilidades.

El Ambiente Operativo Solaris ofrece los aspectos siguientes:

SunOS versión 5.5 es un Sistema Operativo Unix completamente de 32 bits basado en Sistema V versión 4 (SVR4) de Unix y es además el fundamento del Solaris. Como características sobresalientes podemos citar las siguientes:

Portabilidad, Multiprocesamiento con multilectura y Seguridad Normas Industriales incluyendo SVR4 y la familia de protocolos de red ONC. El entorno de computación de Solaris incluye una familia de protocolos y servicios distribuidos, los cuales pueden ser usados para construir un ambiente de computación distribuida heterogéneo. El módulo ONC incluye un sistema distribuido de archivos en red NFS, que es estándar en sistemas abiertos para compartir directorios y archivos a través de la red. Con este sistema, los computadores pueden ofrecer sus archivos a cualquier computador de la red. ONC sigue la filosofía Cliente/Servidor para compartir recursos. Actualmente más de 100 implementaciones de ONC están disponibles para ambientes como DOS, MS-Windows 3.X/95, Appel Macintosh, Ultrix, Novell Netware, OS/2, AIX, MVS, VMS, HP/UX, etc.

Common Desktop Environment (CDE). El Solaris 2.5.1 (2.5 en general), incluye ahora el Common Desktop Environment, el cual le da a los usuarios un potente, abierto y flexible entorno de usuario que provee acceso transparente de todos los desktops soportados (las aplicaciones OpenWindows trabajan sin modificación alguna sobre CDE). Cabe mencionar que el CDE también puede ser instalado sobre Solaris 2.4.

OpenWindows, es una versión ampliada del sistema de ventanas distribuidas en red X/Windows. OpenWindows ofrece actualmente la posibilidad de utilizar un gestor de ventanas OpenLook o Motif. Motif es el estándar de la asociación COSE (Common Open

Software Environment), y es ya (mediante CDE) la interfase del usuario Solaris, pero en la actualidad se mantiene el soporte para la versión OpenLook. El OpenWindows "enfrenta" las necesidades actuales de los desarrolladores mientras provee una tecnología progresiva para las avanzadas aplicaciones del mañana. El servidor de ventanas X11/NeWS (Netware extensible Windows System) es un residente (engine) de alto rendimiento para construir aplicaciones X11.

Interfase gráfica de usuario (GUI) en el Administrador Window OPEN LOOK® .

DeskSet es una suite de herramientas de productividad personal y grupal fácil de usar incluyendo un total de 15 herramientas de administración basadas en la interface gráfica OpenLook y que hacen uso de los servicios ToolTalk. Las herramientas traen Correo, Multimedia, Administrador de Archivo, Herramienta de Impresión, Herramienta de Imagen, Administrador de Calendario, y otros.

Multiprocesamiento Simétrico (SMP), con un kernel de Multilectura (MT), SMP/MT. La Arquitectura avanzada incluye multiprocesamiento totalmente simétrico (SMP) y sofisticado multithreading (MT). El SMP/MT acelera el rendimiento del sistema al permitir que diferentes aplicaciones puedan ejecutarse en múltiples procesadores concurrentemente. Así mismo, SUN implementa un kernel de multithreading sin quebrar las interfaces del SVR4. El Kernel de multithreading le asigna "multithreading" al hardware; esto es, muchos procesos pueden ejecutarse paralelamente en diferentes CPUs incrementando el rendimiento del sistema; muchas aplicaciones pueden beneficiarse con esto, incluyendo manejadores de bases de datos.

La multitarea significa que el sistema operativo puede realizar varias tareas al mismo tiempo. El multiprocesamiento simétrico permite sacar toda la ventaja de los procesadores múltiples. El multiprocesamiento asimétrico (en donde un microprocesador se dedica exclusivamente a una tarea específica), da lugar a que un microprocesador permanezca inactivo en cuanto finaliza su tarea. En el multiprocesamiento simétrico, el sistema operativo puede asignar diferentes tareas a un mismo procesador; así, si uno de ellos termina su trabajo antes que otro, el sistema operativo podrá ocuparlo en otra actividad. El multiprocesamiento simétrico es bastante difícil de implementar, pero ofrece un rendimiento muy superior.

Cabe mencionar que la característica de multithreading sólo se presenta en equipos de arquitectura Sun4m, Sun4d y Sun4u; ¿qué beneficio trae esto?, que se aprovecha al máximo la capacidad de los procesadores.

La herramienta de administración de sistema, Admintool OPEN LOOK, del producto Solaris 2.5, puede usarse únicamente para la administración local de sistema. Las herramientas de administración de sistema usadas para administrar una red de sistemas se proveen con la familia de productos Solstice.

El Administrador de Calendario es una aplicación de gestión de tiempo que muestra citas e items de un vistazo y ofrece un aspecto multibrowse que lo hace fácil de programar entre un grupo.

El Administrador de Archivo provee una manera gráfica e intuitiva para navegar en sistemas de archivo remoto y local.

La Herramienta de Imagen permite a usted cargar, ver y grabar imágenes de mas de 40 formatos diferentes.

Instalación GUI para facilitar la instalación y actualización

Archivos de sistema basados en LOG sobre servidores

Programación con prioridad en tiempo real proveyendo los beneficios de sistemas abiertos mientras encuentran los requerimientos de aplicaciones de control

Servicio Plus de Información de la Red (NIS+) es una versión compatible ascendente del servicio de nombre NIS con administración jerárquica más simple, seguridad mejorada , y actualizaciones más rápidas.

La conformidad de normas para los desarrolladores de aplicación interesadas en los beneficios de portabilidad de aplicación.

Los desktops comerciales usando bases de datos mixtas entre clientes y servidores, pueden ahora usar la interface de programación Open Database Connectivity (ODBC) para una gran flexibilidad, la cual viene incluida en el paquete Solaris 2.5.1.

El Correo Multimedia simplifica mensajes remitentes que incorporan audios, gráficas, y archivos.

Registros de herramientas de audio y grabar mensajes dichos; ellos pueden igual ser adjuntados al enviar correo.

3.2.7.2. Ventajas que provee el Ambiente Operativo Solaris

La norma UNIX SVR4, acomoda las variantes principales DE UNIX (Sistema V, BSD, SunOS, y Xenix (TM)), uniendo la mayoría de la base instalada de usuarios de UNIX. El Ambiente Operativo Solaris, basado en SVR4, da a los desarrolladores de software, administradores de sistema, y usuarios finales los beneficios de una sistema operativo estándar incluyendo amplia compatibilidad, una trayectoria de crecimiento, y reduciendo el tiempo para comercializar. También entrega un producto funcional y poderoso reflejo de los años de refinamiento.

Aunque la fundación del Ambiente Operativo Solaris esta basada en SVR4, SunSoft ha agregado extensiva funcionalidad en áreas tal como multiprocesamiento simétrico con multithreads, funcionalidad en tiempo real, seguridad aumentada, y administración mejorada del sistema.

Entre las muchas ventajas que el Ambiente Operativo Solaris provee están la portabilidad, escalabilidad, interoperabilidad, y compatibilidad.

3.2.7.3. Portabilidad

El SunOS de SunSoft es portátil a través de múltiples plataformas de vendedores. Mediante la portabilidad ABI (Interface de Aplicación Binaria), se provee una interfase que le permite desarrollar aplicaciones UNIX compatibles en cualquier sistema Solaris.

El software ABI corre como software shrink - wrapped sobre todos los sistemas de vendedores con la misma arquitectura de microprocesador. Esto permite a los

desarrolladores de aplicación reducir los costos de desarrollo de software y traer productos para comercializar rápidamente, y permite a los usuarios mejorar el hardware mientras retienen sus aplicaciones de software y minimizando costos de conversión.

3.2.7.4. Escalabilidad

A través del tiempo, las aplicaciones llegan a ser más ampliamente usadas, y requieren que sistemas más poderosos los apoyen.

Para operar en un Ambiente creciente, el software debe ser capaz de correr en una amplia rango de poder y debe ser capaz de tomar ventaja del poder adicional de procesamiento. El sistema Operativo de SunSoft corre sobre máquinas de todos los tamaños, desde laptops a supercomputadoras.

3.2.7.5. Interoperabilidad

Los Ambientes de computación heterogéneos son una realidad hoy. Los usuarios compran sistemas de muchos vendedores para implementar la soluciones que ellos necesitan. La estandarización y las interfaces claras son críticos para un ambiente heterogéneo, permitiendo a los usuarios desarrollar estrategias para comunicar a lo largo de su red.

Los sistemas Solaris pueden interoperar con cada sistema popular sobre el mercado hoy, y las aplicaciones que corren sobre UNIX pueden comunicarse fácilmente.

3.2.7.6. Compatibilidad

La tecnología de Computación continúa avanzando rápidamente, pero la necesidad de permanecer competitivo requiere que los vendedores minimicen sus costos y maximicen sus inversiones. SunSoft asegurará que tanto como la nueva tecnología es introducida, la inversión existente de software se conserve. Los usuarios pueden aprovechar soluciones de hoy y todavía ser compatibles con tecnologías de mañana.

3.2.7.7. Facilidad de acceso a servicios

La Facilidad de Acceso a Servicio (SAF) es la herramienta usada para administrar terminales, modems, y otros dispositivos de red.

La SAF es una solución para sistemas abiertos que controla accesos al sistema y a recursos de red mediante dispositivos TTY y redes de área local (LANS). El SAF ofrece interfaces bien definidas que hacen fácil agregar nuevas características y configurar las existentes.

La SAF no es un programa. Es una jerarquía de procesos y comandos administrativos en background. El programa de mayor nivel SAF es SAC. SAC controla monitores de puerto que se administra mediante el comando sacadm. Cada monitor de puerto puede administrar uno o más puertos.

3.3. Estándares de Comunicación.

Dentro de los Estándares de comunicación utilizados por la Cooperativa de Ahorro y Crédito "El Sagrario" Ltda. tenemos:

- El Modelo OSI

3.3.1. El modelo de referencia OSI

La aportación más importante de la iniciativa OSI ha sido precisamente su arquitectura. Esta ha servido como marco de referencia para describir multitud de redes correspondientes a diversas arquitecturas, ya que la arquitectura OSI es bien conocida en entornos de redes, y su generalidad y no dependencia de ningún fabricante en particular le hacen especialmente adecuada para estos fines. Por este motivo generalmente a la arquitectura OSI se la denomina Modelo de Referencia OSI, o también OSIRM (OSI Reference Model).

Por extensión hoy en día se utiliza a menudo el término modelo de referencia para referirse a una arquitectura de red; así oímos hablar del Modelo de Referencia TCP/IP, el Modelo de Referencia ATM, etc.

El modelo OSI define siete capas, curiosamente como en la arquitectura SNA si bien la funcionalidad es diferente. Las capas son las siguientes:

1. Física
2. Enlace
3. Red
4. Transporte
5. Sesión

Muchas de las normas que existen en la capa física se refieren a la interfaz utilizada para conectar un ordenador con un módem o dispositivo equivalente, que a través de una línea telefónica conecta con otro módem y ordenador en el extremo opuesto.

Este es el caso por ejemplo de las normas EIA RS-232-C, EIA-RS-449, CCITT X.21/X.21bis y CCITT V.35 antes mencionadas. En estos el conector del ordenador y el módem son de diferente "sexo" (macho o hembra). En este contexto se suele utilizar la denominación DTE (Data Terminal Equipment) para referirse al ordenador y DCE (Data Circuit-Terminating Equipment) para referirse al módem. El "módem" en ocasiones no es más que un adaptador, ya que por ejemplo la norma

X.21 se utiliza para líneas digitales. En sentido general al equipo que actúa como adaptador entre el ordenador y el medio de transmisión se le denomina CSU/DSU (Channel Service Unit/ Data Service Unit).

3.3.1.2. La capa de enlace (data link)

La principal función de la capa de enlace es ofrecer un servicio de comunicación fiable a partir de los servicios que recibe de la capa física, también entre dos entidades contiguas de la red. Esto supone que se realice detección y posiblemente corrección de errores. A diferencia de la capa física, que transmitía los bits de manera continua, la capa de enlace transmite los bits en grupos denominados tramas (frames en inglés) cuyo tamaño es típicamente de unos pocos cientos a unos pocos miles de bytes. Si el paquete recibido de la capa superior es mayor que el tamaño máximo de trama la capa física debe encargarse de fragmentarlo, enviarlo y recomponerlo en el lado opuesto. En caso de que una trama no haya sido transmitida correctamente se deberá enviar de nuevo; también debe haber

mecanismos para reconocer cuando una trama se recibe duplicada. Generalmente se utiliza algún mecanismo de control de flujo, para evitar que un transmisor rápido pueda "abrumar" a un receptor lento.

Las redes broadcast utilizan funciones especiales de la capa de enlace para controlar el acceso al medio de transmisión, ya que éste es compartido por todos los nodos de la red. Esto añade una complejidad a la capa de enlace que no está presente en las redes basadas en líneas punto a punto, razón por la cual en las redes broadcast la capa de enlace se subdivide en dos subcapas: la inferior, denominada subcapa MAC (Media Access Control) se ocupa de resolver el problema de acceso al medio, y la superior, subcapa LLC (Logical Link Control) cumple una función equivalente a la capa de enlace en las líneas punto a punto.

Ejemplos de protocolos de la capa de enlace incluyen ISO 7776, la capa de enlace de CCITT X.25, RDSI, LAP-D, ISO HDLC. Como ejemplos de protocolos de la subcapa MAC podemos citar los de IEEE 802.3 (Ethernet), IEEE 802.5 (Token Ring), ISO 9314 (FDDI). El protocolo de subcapa LLC de todas las redes broadcast es el IEEE 802.2.

3.3.1.3. La capa de red

La capa de red se ocupa del control de la subred. Esta es la capa que tiene "conciencia" de la topología de la red, y se ocupa de decidir por que ruta va a ser enviada la información; la decisión de la ruta a seguir puede hacerse de forma estática, o de forma dinámica en base a información obtenida de otros nodos sobre el estado de la red.

En las redes de tipo broadcast el nivel de red es casi inexistente, ya que desde un punto de vista topológico podemos considerar que en una red broadcast los nodos están interconectados todos con todos, por lo que no se toman decisiones de encaminamiento. Sin embargo veremos que la unión de redes broadcast mediante puentes suscita en algunos casos la necesidad de efectuar tareas propias del nivel de red en el nivel de enlace.

3.3.1.4. La capa de transporte

La capa de transporte es la primera que se ocupa de comunicar directamente nodos terminales, utilizando la subred como un medio de transporte transparente gracias a los servicios obtenidos de la capa de red. Por esta razón se la ha llamado históricamente la capa host-host. También se suele decir que es la primera capa extremo a extremo.

La principal función de la capa de transporte es fragmentar de forma adecuada los datos recibidos de la capa superior (sesión) para transferirlos a la capa de red, y asegurar que los fragmentos lleguen y son recompuestos correctamente en su destino.

En condiciones normales la capa de transporte solicita a la capa de red una conexión diferente por cada solicitud recibida de la capa de sesión, pero puede haber razones de costo que aconsejen multiplexar diferentes conexiones en la capa de sesión sobre una sola conexión en la capa de red o, inversamente, razones de rendimiento pueden requerir que una conexión solicitada por la capa de sesión sea atendida por varias conexiones en la capa de red; en ambos casos la capa de transporte se ocupará de hacer la multiplexación más adecuada de forma transparente a la capa de sesión.

La capa de transporte establece el tipo de servicio que recibe la capa de sesión, y en último extremo los usuarios. Éste podría ser por ejemplo un servicio libre de errores que entrega los mensajes en el mismo orden en que se envían; también podría ser un servicio de datagramas, es decir, mensajes independientes sin garantía en cuanto al orden de entrega ni confirmación de la misma, o un servicio broadcast o multicast en que los paquetes se distribuyen a múltiples destinos simultáneamente.

El control de flujo, que ha aparecido en capas anteriores, es necesario también en la capa de transporte para asegurar que un host rápido no satura a uno lento. La capa de transporte realiza también su propio control de errores, que resulta ahora esencial pues algunos protocolos modernos como Frame Relay o ATM han reducido o suprimido totalmente el control de errores de las capas inferiores, ya que con las mejoras en la tecnología de transmisión de datos éstos son menos frecuentes y se considera mas adecuado realizar esta tarea en el nivel de transporte.

Salvo el caso de transmisiones multicast o broadcast el nivel de transporte se ocupa siempre de una comunicación entre dos entidades, lo cual le asemeja en cierto sentido al nivel de enlace. Por esto existen grandes similitudes entre ambas capas en cuestiones tales como el control de errores o control de flujo.

Ejemplos de protocolos de transporte incluyen el CCITT X.224, también llamado protocolo de transporte OSI TP4 (Transport Protocol 4). En Internet existen dos protocolos de transporte: TCP y UDP.

3.3.1.5. La capa de sesión

La capa de sesión es la primera que es accesible al usuario, y es su interfaz más básica con la red. Por ejemplo, mediante los servicios de la capa de sesión un usuario podría establecer una conexión como terminal remoto de otro ordenador. En un sistema multiusuario la capa de sesión se ocupa de ofrecer un SAP a cada usuario para acceder al nivel de transporte.

3.3.1.6. La capa de presentación

Hasta aquí nos hemos preocupado únicamente de intercambiar bits (o bytes) entre dos usuarios ubicados en dos ordenadores diferentes. Lo hemos hecho de manera fiable y entregando los datos a la sesión, es decir al usuario, pero sin tomar en cuenta el significado de los bits transportados. La capa de presentación se ocupa de realizar las conversiones necesarias para asegurar que dichos bits se presentan al usuario de la forma esperada. Por ejemplo, si se envía información alfanumérica de un ordenador ASCII a uno EBCDIC será preciso efectuar una conversión, o de lo contrario los datos no serán interpretados correctamente. Lo mismo podríamos decir de la transferencia de datos enteros, flotantes, etc. cuando la representación de los datos difiere en los ordenadores utilizados.

3.3.1.7. La capa de aplicación

La capa de aplicación comprende los servicios que el usuario final está acostumbrado a utilizar en una red telemática, por lo que a menudo los protocolos de la capa de aplicación se denominan servicios. Dado que se crean continuamente nuevos servicios, existen muchos protocolos para la capa de aplicación, uno o más por cada tipo de servicio.

Ejemplos de protocolos estándar de la capa de aplicación son el CITT X.400, X.420, X.500, FTAM. SMTP, FTP, HTTP, etc.

Resumen de las Capas en el Modelo OSI.

Aplicación	El nivel de aplicación es el destino final de los datos donde se proporcionan los servicios al usuario.
Presentación	Se convierten e interpretan los datos que se utilizarán en el nivel de aplicación.
Sesión	Encargado de ciertos aspectos de la comunicación como el control de los tiempos.
Transporte	Transporta la información de una manera fiable para que llegue correctamente a su destino.
Red	Nivel encargado de encaminar los datos hacia su destino eligiendo la ruta más efectiva.
Enlace	Enlace de datos. Controla el flujo de los mismos, la sincronización y los errores que puedan producirse.
Físico	Se encarga de los aspectos físicos de la conexión, tales como el medio de transmisión o el hardware.

3.4. 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. Un interfaz, sin embargo, es el encargado de la conexión física entre los

use este tipo de líneas, en el caso de que exista mucho tráfico de voz y de datos entre dos lugares.

Una red privada que conectase cuatro posiciones separadas, podría tener un aspecto como el de la Figura R.1. Los cuatro lugares puede que se localicen dentro de la misma área geográfica local para acceso y transporte (LATA, Local Access and Transport Area), con lo que las líneas alquiladas las aportaría la compañía de telecomunicaciones para el intercambio local (LEC, Local Exchage Carrier). Si se sobrepasa los límites de LATA, se involucraría a las LECs de los puestos local y remoto, a través de una compañía de telecomunicaciones para el intercambio a larga distancia que los conectará (IXC).

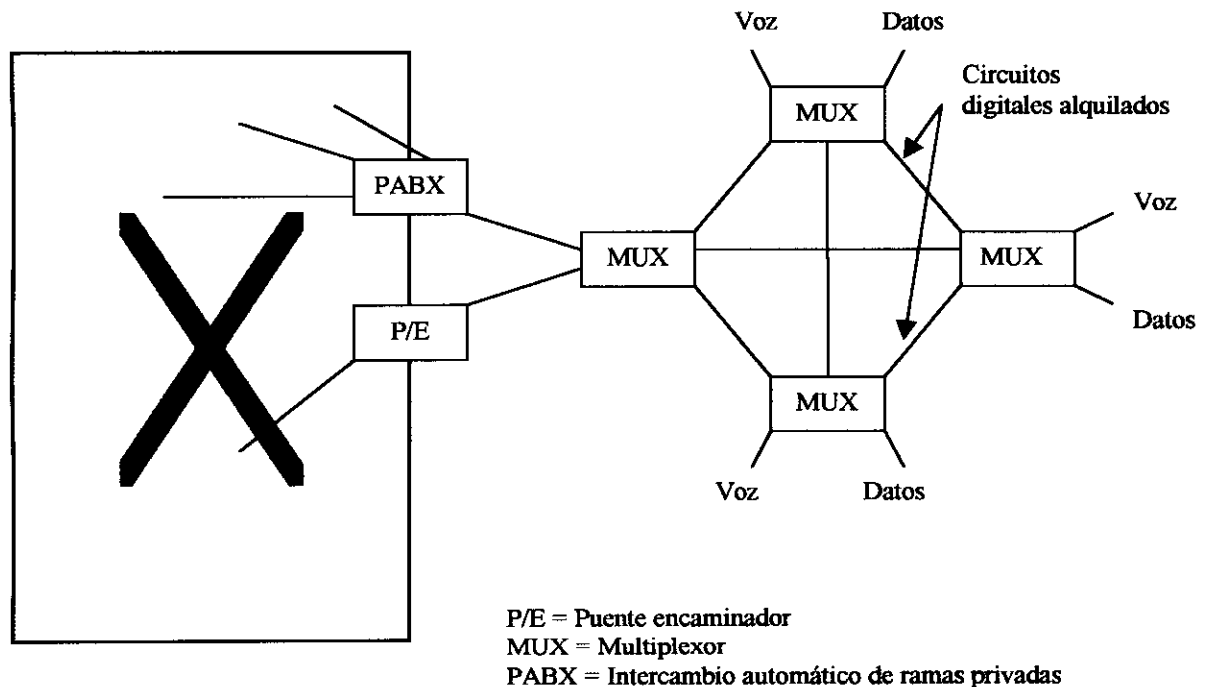


Figura 3.1. Red privada que conecta cuatro lugares remotos

3.5.2. Facilidades públicas

De las *redes públicas de datos* (PDN, Public Data Network) se ocupan las compañías de telecomunicaciones, sean o no de valor añadido (VAC, Value-Added Carrier).

La comunicación se hace en la red de la compañía. Las redes públicas son las mejores para el enlace de muchos lugares remotos y para el enlace de sitios que no existe bastante tráfico entre ningún par de ellos como para que se justifique una línea alquilada dedicada. Además, si la distancia entre las posiciones de dos clientes es tan grande como para que el costo de una línea alquilada se haga prohibitivo, la mejor opción será también una red pública de conmutación de paquetes. Una PDN proporciona servicios tanto de conmutación de paquetes como de circuitos, y es lo que se describe a continuación:

3.5.3. Servicios de conmutación de circuitos

- Estos servicios proporcionan un camino fijo entre dos puntos que se configura previamente al intercambio de información.
- También aportan circuitos dedicados con una anchura de banda conocida y garantizada.
- Cuando se selecciona el circuito, se producen pequeños retrasos.
- La transmisión de datos comienza sólo cuando el circuito se ha establecido.
- Ambos sistemas finales deben estar preparados para efectuar el enlace, al igual que sucede en una línea telefónica.
- Las líneas caídas pueden detener toda la transmisión, o requerir la intervención del usuario para rodear el problema.

3.5.4. Servicios de conmutación de paquetes

- Los paquetes de información se encaminan a través de la malla de redes, de acuerdo con la dirección de destino que figura en su cabecera.
- Los paquetes viajan a través de puertos compartidos, por lo que se pueden producir ligeros retardos, especialmente si son de longitud variable.
- No hay ningún retraso asociado a la elección de un circuito, sino que los circuitos lógicos orientados a la conexión se pueden definir a priori.
- Si se tienen capacidades de ancho de banda variable, se puede soportar el tráfico de ráfagas.
- La fuente puede transmitir en cualquier momento. No es necesario haber prefijado una sesión.
- La red evita de forma automática las líneas caídas o los nodos. En las redes públicas hay muchas rutas <<alternativas>> que pueden ocuparse del tráfico de los caminos que fallen.

La decisión acerca de que método usar, la conmutación de circuitos o la de paquetes, se basa en la cantidad de retraso que es tolerable que se produzca entre el emisor y el receptor. En un sistema orientado a mensajes que se ocupa del correo electrónico y archivos de datos, se hace aceptable un retardo de segundos, minutos e incluso horas, y los sistemas de conmutación de paquetes son más baratos. Para las aplicaciones en tiempo real del tipo de procesamiento de transacciones en línea y otros requisitos interactivos, se hacen esenciales los sistemas de conmutación de circuitos, y el tipo de circuito dependerá de la

esenciales los sistemas de conmutación de circuitos, y el tipo de circuito dependerá de la cantidad de tráfico y del número de posiciones que deban conectarse. Si sólo está involucrado un usuario, una línea de enlace telefónico a alta velocidad puede ser aceptable, pero si hay múltiples usuarios que requieren acceso a una posición central al mismo tiempo y de modo continuo, podría ser necesario construir una red privada o usar facilidades de conmutación de circuitos que proporcionen conexiones casi instantáneas entre dos emplazamientos.

Algunas VACs como Compu Serve Information Service, GE Information Services Corp. Y Tymenet Global Network proporcionan servicios WAN de una parada y facturación de sus clientes. Normalmente, una VAC posee una red nacional establecida para su propio uso mediante la construcción de sus propias facilidades de comunicación o de líneas que haya alquilado, con un descuento considerable, a otras compañías de telecomunicación. La VAC también podría realizar para sus clientes, algún procedimiento de los datos o la conversión de protocolos. Si se usa una PDN, se evitan los problemas inherentes al control de las líneas y la configuración del propio equipo de conmutación. Una PDN también se ocupa de cualquier problema que haya con la red y puede garantizar la distribución de los datos a través de una vasta malla de líneas conmutadas.

En la figura R.2 se ha dibujado la topología de una conexión de área extensa.

Un área local para el acceso y transporte (LATA) se corresponde habitualmente con la geografía de un código de área telefónico. Una compañía de telecomunicaciones para el intercambio local (LEC) que fuera una de las compañías regionales de operaciones Bell (RBOCs, Regional Bell Operating Companies) trabajaría dentro de un área (LATA)

aunque dentro de la misma LATA podrían operar también compañías que no fuesen de operaciones Bell. Se requiere que haya una LEC para que proporcione una facilidad de punto de presencia (POP, Point-of-Presence) a las compañías de telecomunicaciones para el intercambio a larga distancia (IXC), como AT&T, MCI y US Sprint, y así los clientes podrían elegir la IXC que quieran. Algunas compañías de larga distancia ofrecen facilidades de desvío, de forma que sus clientes pueden conectarse directamente a sus servicios de larga distancia en lugar de hacerlo a través de una LEC. Por ejemplo, MCI anunció recientemente que planea la construcción de facilidades alternativas para el acceso a largas distancias que eviten las LECs en las 20 mayores ciudades de los Estados Unidos. Esto supondría un ahorro considerable a los clientes.

En la figura R.16 está representado el equipo físico para las conexiones de una WAN. Un encaminador o un puente hace converger el tráfico desde la red local sobre la conexión de área extensa. Conectado al puente o al encaminador hay un modem para las líneas analógicas o una unidad de servicio de canal/unidad de servicio de datos (CSU/DSU, Channel Service Unit/Data Service Unit) para las líneas digitales. En (A), los modems conectan dos LANs a través de líneas remotas telefónicas analógicas. En (B) una CSU/DSU conecta dos líneas remotas a través de líneas digitales T1 separadas. En (D), una única conexión a una red pública Frame Relay proporciona conexiones a múltiples posiciones remotas, mediante las facilidades de conmutación de las compañías de telecomunicaciones y los servicios de paquetes de la red de retransmisión de tramas. Nótese que en (C) y (D), también pueden emplearse un multiplexor de voz/datos para transmitir tanto voz como datos a través de las mismas líneas.

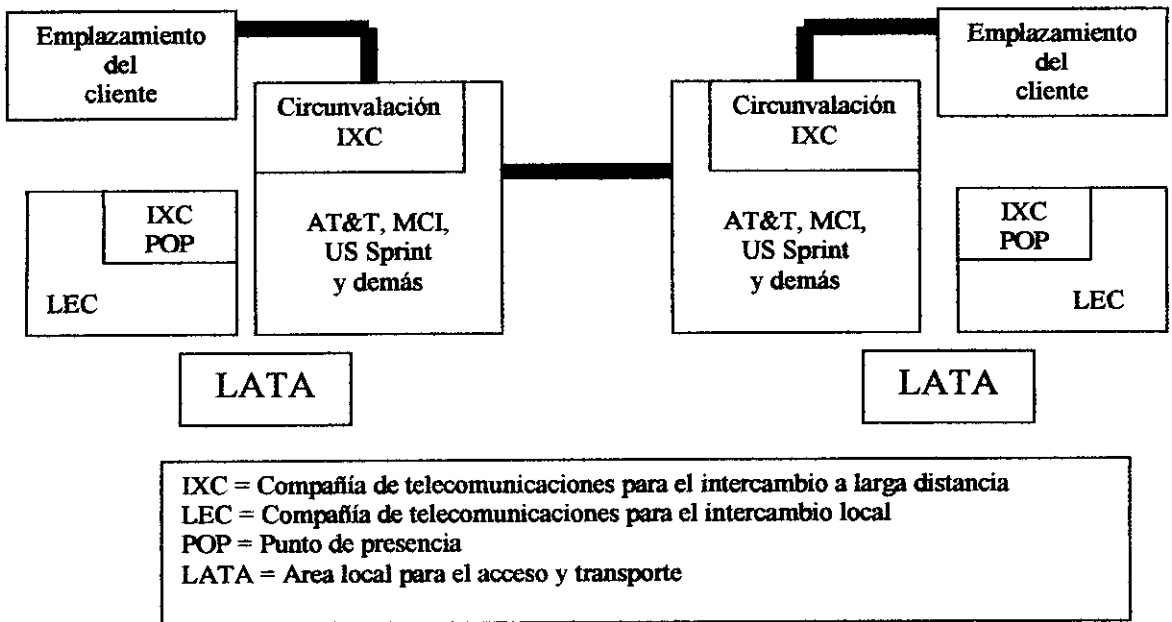


Figura 3.2. Facilidades para distancias locales y grandes para conexiones WAN.

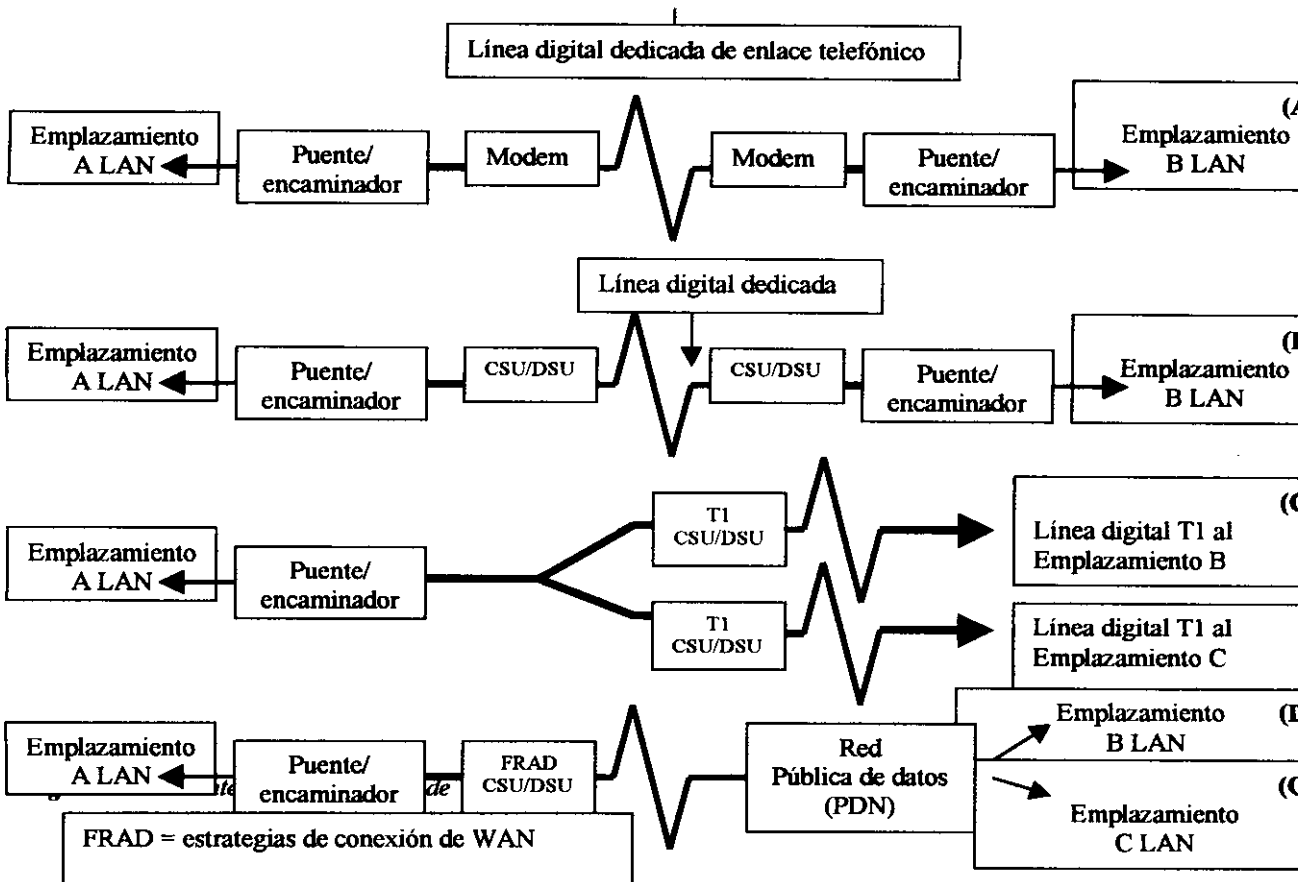
3.5.5. Servicios de las compañías de telecomunicación para conexiones WAN

Estos servicios consisten en líneas de enlace telefónico, líneas dedicadas y servicios de conmutación de circuitos o paquetes. La velocidad de transmisión de las líneas de enlace telefónico analógicas está en el rango entre 1200 y 28800 bits/seg (dentro de los estándares existentes). Los servicios dedicados digitales de conmutación están en el rango entre 56 Kbits/seg y 45 Mbits/seg, con servicios de transmisión de celdas como el Modo de transferencia asíncrono (ATM, Asynchronous Transfer Mode) que actualmente presta servicios a las velocidades de T3 de 45 Mbits/seg. A continuación se dan las aplicaciones típicas:

- Las transferencias de correo electrónico que requieren velocidades de transmisión de 2400 a 9600 bits/seg.

- Los programas de control remoto necesitan velocidades entre 9600 y 56000 bits/seg.
- Las interacciones remotas con aplicaciones basadas en LAN precisan velocidades entre 9600 y 384000 bits/seg.

Para hacerse una idea del tipo de rendimiento que se necesita, hay que multiplicar estas velocidades por el número de usuarios que tengan que acceder al enlace simultáneamente.



3.5.6. Servicios analógicos de enlace telefónico

Las líneas de enlace telefónico usan modems que pueden proporcionar velocidades de transmisión de hasta 28000 bits/seg, a través de líneas de conmutación (usando los nuevos

estándares para modem ITU/CCITT V.34). Como las líneas son de conmutación, sólo se incurre en cargas durante el tiempo de conexión.

Los servicios de enlace telefónico son ideales para los enlaces ocasionales. Pueden ofrecer enlaces para transferencias de enlaces para transferencias de correo electrónico entre dos emplazamientos de la compañía o permitir a usuarios, del lugar o móviles, que se conecten con la base de datos de la compañía. En el caso de la conexión de control remoto, el usuario enlaza con la red de la compañía y toma el control de la computadora perteneciente a la LAN. La computadora conectada a la LAN, realiza el procesamiento, pero envía las pantallas obtenidas al usuario remoto y acepta entradas del teclado del mismo. Esta técnica minimiza la cantidad de tráfico que se transfiere a través del enlace remoto.

3.5.7. Líneas privadas

Las líneas digitales se pueden obtener de las compañías de telecomunicaciones de varios modos y, típicamente, se alquilan por meses. Hay un cargo por la configuración inicial, y el resto de los pagos dependen de la distancia. Si la línea conecta posiciones de dos LATAS separadas, se involucra a la compañía de telecomunicaciones para el intercambio local (LEC) de cada una de las áreas locales para el acceso y transporte (LATA), junto con la compañía de telecomunicaciones de larga distancia. Como se mencionó previamente, esta situación podría cambiar, ya que compañías de larga distancia como MCI trabajan para proporcionar a sus clientes enlaces directos para las conexiones de larga distancia, en lugar de que los enlaces se realicen a través de las LECs.

No todas las líneas dedicadas no son dedicadas. Una organización podría alquilar temporalmente una línea de alta velocidad para uso ocasional, como la transferencia de archivos a fin de mes, o de vídeo conferencias semanales. Este tipo de servicios se contratan de antemano con las compañías de telecomunicaciones. Sin embargo, los servicios digitales de comunicación que se discuten a continuación, han reducido la necesidad de este tipo de líneas.

3.5.8. Servicios de conmutación de circuitos

Básicamente, un servicio de conmutación de circuitos no es muy diferente de una llamada telefónica. Se configura un canal dedicado entre los dos puntos, durante el intervalo de la llamada. Durante ese tipo se proporciona una cantidad específica de ancho de banda. Los usuarios pagan sólo por el tiempo en el que están conectados, y pueden llamar a diferentes sitios. Estos servicios se suelen denominar servicios de *marcar-bajo-demanda*. Habitualmente las llamadas sufren un cierto retardo de configuración asociado a este tipo de servicios; sin embargo, los nuevos equipos de conmutación han reducido este tiempo de configuración a microsegundos. En los siguientes casos, se suelen usar las líneas de conmutación de circuitos como líneas de seguridad para las líneas dedicadas a esta velocidad:

- En el caso de que falle una alquilada dedicada, se conectan una o más líneas de conmutación de circuitos para que se ocupe del tráfico entre los dos emplazamientos, hasta que se reconecte la línea dedicada.

- En caso de que una línea llegue a saturarse, se conectan líneas de conmutación de circuitos para que se ocupen de la excesiva carga de tráfico.
- Esta estrategia proporciona a los administradores de la red un camino económico para enlazar posiciones, sin necesidad de adquirir una cantidad excesiva de servicio dedicado.

Una técnica que convine líneas de conmutación de circuitos con multiplexores inversos, proporcionaría ancho de banda bajo demanda. Con esta técnica, se añaden enlaces adicionales cuando se incrementa el ancho de banda. Los multiplexores inversos de cada extremo dividen y recombinan las señales de múltiples líneas de baja velocidad.

Un servicio de conmutación común *Switched-56*, que opera a 56 Kbits/sec y requieren un *Switched-56* CCU/DECU especial en cada uno de los emplazamientos. Los servicios *Switched-56* tenían originalmente la finalidad de proporcionar una línea de seguridad alternativa a las líneas alquiladas de alta velocidad de tipo T1.

Si una línea fallaba, una línea *Switched-56* establecería rápidamente una conexión alternativa. Todavía pueden usarse para esto, pero además de emplean para manejar picos de tráfico, transmisiones fax, copias de seguridad, transferencias de correo electrónico voluminosas y conexiones de LAN a LAN. Las tarifas se suelen calcular por minuto.

Para velocidades de transmisión de datos más alta, las compañías del tipo AT&T ofrecen servicios como *Switched 384K* y *Switched T1*. Se requiere que el cliente instale

previamente el servicio de Red digital de servicios integrados (ISDN, Integrated Service Network), como se comenta en la siguiente sección.

3.5.9. Red Digital de servicios Integrados (ISDN)

ISDN es un servicio que proporciona todos los servicios digitales en el *bucle local*, que es el cable que corre entre la casa o el negocio de usuario, y la oficina de conmutación de la compañía de telecomunicaciones de intercambio local. Este bucle es en gran medida aún, un cable de par trenzado que soporta transmisiones analógicas . ISDN se ofrece en áreas selectas, y se usa del mismo modo que los servicios telefónicos. Al mismo tiempo que se proporcionan servicios digitales de conmutación de circuitos de alta velocidad, entre un cliente y la compañía telefónica, también ofrece un rango de servicios de integración de voz y datos a partir del cuál se construyen muchas ofertas.

La ISDN básica, conocida como *interfaz de velocidad básica(BRI Basic Rate Interface)* tiene tres canales: dos proporcionan 64 Kbits/sec y un tercero de señalización proporciona 16Kbits/sec. La *Interfaz de velocidad principal* proporciona servicios a aquellos que lo necesiten. Consta de 23 canales de voz o datos a 64Kbits/sec. En ambos interfaces el canal de señalización se ocupa de la importante función de configurar las llamadas. Con esto se consiguen velocidades de configuración de microsegundos y señalización fuera de banda, con lo que se mantienen los canales de datos libres para las transmisiones.

3.5.10. Servicios de comunicación de paquetes

Una red de comunicaciones proporciona una malla de conexiones a través de la cual viajan los paquetes de datos, hasta alcanzar un destino. Los datos del sistema fuente se separan y se dividen en paquetes de tamaño predefinido. Hay dos tipos de servicios para la distribución de paquetes:

- Un *servicio no orientado a la conexión* en el cual cada paquete es una entidad independiente que sigue su propio curso a través de la red, dependiendo del mejor camino disponible o de las decisiones que tomen los conmutadores a lo largo del camino. Como los paquetes pueden seguir distintas rutas, pueden llegar desordenados a su destino, y el sistema destinatario tiene que resecuenciarlos.
- Un *servicio orientado a circuito*, en el cual se establece a través de la red una conexión lógica o camino. El camino está predefinido, con lo que se mejora la eficiencia y sobrecarga debido a los paquetes. Dado que el camino se fija de antemano, los paquetes llegan en orden, con lo que no hay que resecuenciarlos.

X.25 es un protocolo que define conexiones a una red de datos públicos con conmutación de paquetes. (PSPDN, Packet-Switched Public_Data Network). El protocolo define las acciones físicas de la red y las de enlace de datos. Las redes X25 trabajan a velocidad de hasta 64 Kbytes/seg. Realizan una amplia comprobación de errores en cada nodo para asegurarse de que los datos se difundan correctamente a través de las poco fiable líneas telefónicas. En los países desarrollados las líneas son extremadamente fiables, así que esta

aumentando el uso de servicios rápidos que suprimen la comprobación de errores en los nodos. La red Frame Relay es un servicio de este tipo.

3.5.11. Servicio Frame Relay

La red Frame Relay es una innovación que emergió a partir de las especificaciones ISDN. Es un servicio orientado a circuitos, aunque también se la trata como un servicio rápido de paquetes. Perfecciona la técnica de conmutación de paquetes mediante la eliminación del proceso a nivel de red social al X.25. típicamente las redes de retransmisión de tramas tiene 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.

3.5.12. 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 si 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.

En principio, podría parecer poco coherente alquilar una línea dedicada para acceder a la facilidad de conmutación de la compañía de telecomunicaciones, si se tiene en cuenta cuál es el propósito de estas facilidades, pero hay que tener presente que la distancia hasta la red de la compañía de telecomunicaciones es relativamente corta. Las facilidades de conmutación proporcionan conexiones de cualquiera-a-cualquiera a través de grandes distancias, con tarifas que son mucho menores que el costo de construcción de una red privada que salvase estas distancias grandes.

El problema, y lo que encárese a los servicios WAN actuales, es el requisito de que los clientes encaminen el tráfico a través de las facilidades de la compañía de telecomunicaciones para el intercambio local (LEC, Local Exchange Carrier) de su área, donde es recogido por las compañías de telecomunicaciones de larga distancia, a través de la facilidad de punto de presencia (POP, Point of Presence). Como se mencionó anteriormente. MCI y otras compañías que cubren grandes distancias planean aliviar esta situación construyendo alternativas en las mayores áreas metropolitanas.

3.5.13. Otras tendencias

Los actuales sistemas nacionales e internacionales de teléfono don incapaces de manejar los cada vez mayores requerimientos de transferencias de multimegabits de datos para la informática de gran ancho de banda y los sistemas multimedia. Por tanto, merece la pena echar un vistazo a algunas de las tecnologías y servicios que soportarán en el futuro las redes globales de datos, voz e imágenes. La cantidad del tráfico de datos en las redes se incrementará a medida que lo hagan la potencia y la velocidad de los sistemas informáticos. 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 Nedwork). 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 Multimegabit (SMDS, Switchet 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 .

3.5.14. 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 IV

4. MOTORES DE BASES DE DATOS

4.1 Bases de Datos Relacionales

Los sistemas de gestión de bases de datos organizan y estructuran los datos de tal modo que puedan ser recuperados y manipulados por usuarios y programas de aplicación. Las estructuras de los datos y las técnicas de acceso proporcionadas por un DBMS, y las aplicaciones para las cuales esta particularmente bien conformado.

4.2. Modelos de Datos Primitivos

Cuando la gestión de base de datos se popularizó durante los setenta y los ochenta emergieron un puñado de modelos de datos populares. Cada uno de estos primeros modelos de datos tenía ventajas y desventajas que jugaron papeles importantes en el desarrollo de modelo de datos relacional. En muchos sentidos el modelo de datos relacional, representó un intento de simplificar los modelos de datos anteriores para comprender el papel y la contribución de SQL y el modelo relacional, sería útil examinar brevemente algunos modelos de datos que precedieron al desarrollo de SQL.

4.2.1. Sistema de Gestión de archivos

Antes de la introducción de los sistemas de gestión de la base de datos, todos los datos permanentemente almacenados en un sistema informático, tales como la nómina y los registros de contabilidad, se almacenaban en archivo individuales.

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.

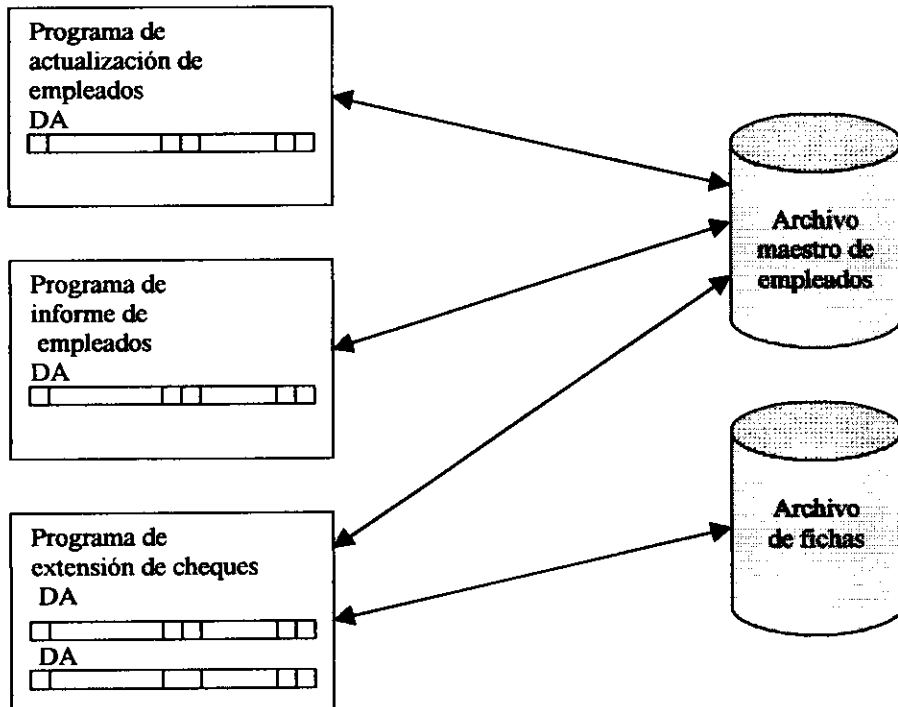


Figura 4.1. Aplicación de nómina utilizando un sistema de gestión de archivos

4.2.2. Bases de Datos Jerárquicas.

Una de las aplicaciones más importantes de los sistemas de gestión de base de datos primitivos era el planteamiento de la producción para empresas de facturación. Si un fabricante de automóviles decidía producir 10.000 unidades de un modelo de coche y 5.000 unidades de otro modelo, necesitaba saber cuántas piezas pedir a sus suministradores. Para responder a la cuestión, el producto (un coche) tenía que descomponerse en ensamblajes (motor, cuerpo, chasis), que a su vez se descomponían en subensamblajes (válvulas, cilindros, bujías) y luego en sub-subensamblajes, etc. El manejo de estas listas de piezas, conocido como una cuenta de materiales, era un trabajo a la medida para los computadores.

La cuenta de materiales para un producto tenía una estructura jerárquica natural . Para almacenar estos datos, se desarrollo el modelo de datos *jerárquico*, ilustrado en la Figura 4.2. En este modelo cada registro de la base de datos representa una pieza específica . Los registros tenían *relaciones padre/hijo*, que ligaba cada pieza a su subpieza, y así sucesivamente.

Para acceder a los datos en la base de datos, un programa podría:

- Hallar una pieza particular mediante su número (como por ejemplo la puerta izquierda)
- Descender al primer hijo (el tirador de la puerta)
- Ascender hasta su padre (el cuerpo)

- Moverse de lado hasta el siguiente hijo (la puerta derecha)

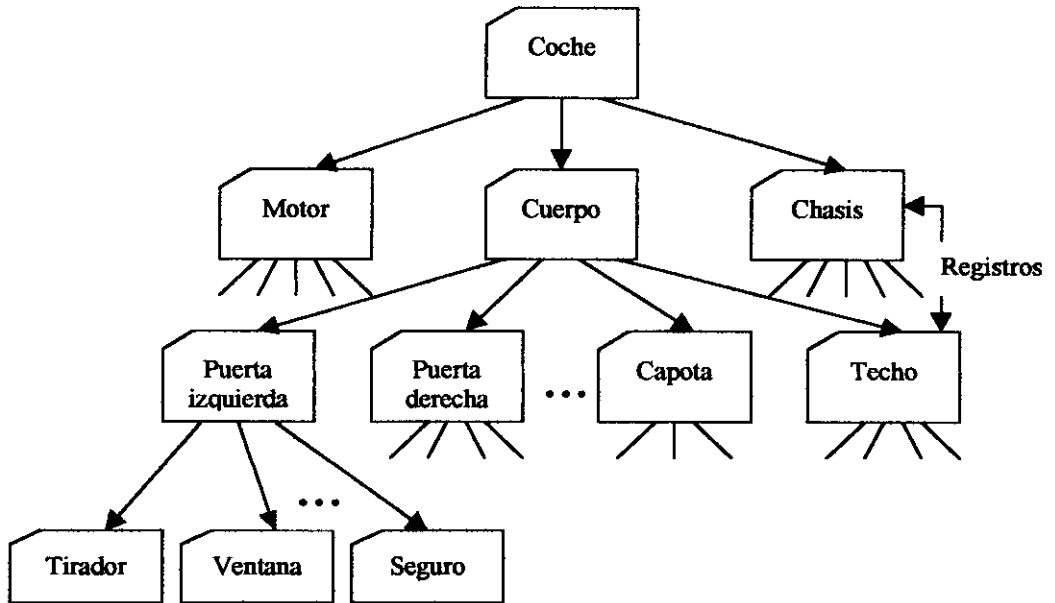


Figura 4.2. Una base de datos de lista de materiales jerárquica.

La recuperación de los datos en una base de datos jerárquica requería, por tanto, *navegar* a través de los registros, moviéndose hacia arriba, hacia abajo y hacia los lados un registro cada vez.

Uno de los sistemas de gestión de base de datos jerárquica mas populares fue el Information Management System (IMS) de IBM, introducido primeramente en 1968. Las ventajas del IMS y su modelo jerárquico son las siguientes:

Estructura simple. La organización de una base de datos IMS era fácil de entender. La jerarquía de una base de datos se asemejaba al diagrama de organización de una empresa o de un árbol familiar.

Organización padre/hijo. Una base de datos IMS era excelente para representar relaciones padre/hijo, tales como (A es una pieza de B) o (A es propiedad de B)

Rendimiento. IMS almacenaba las relaciones padre/hijo como puntero físico de un registro de datos a otro, de modo que el movimiento a través de la base de datos era rápido. Puesto que la estructura era sencilla, IMS podía colocar los registros padre e hijo cercanos unos a otros en los registros, minimizando la entrada/salida de datos.

IMS sigue siendo el DBMS más ampliamente instalado en los maxicomputadores IBM. Se utiliza en más del 25 por 100 de las instalaciones de maxicomputadores IBM.

4.2.3. Bases de Datos en Red

La estructura sencilla de una base de datos jerárquico se convertía en una desventaja cuando los datos tenían una estructura más compleja. En una base de datos de procesamiento de pedidos, por ejemplo, un simple pedido podría participar en tres relaciones padre/hijo *diferentes*, ligando el pedido al cliente que lo remitió, al vendedor que lo aceptó y al producto ordenado, tal como se muestra en la Figura 4.3. La estructura de este tipo de datos simplemente no se ajustaría a la jerarquía estricta de IMS.

Para manejar aplicaciones tales como el procesamiento de pedidos, se desarrolló un nuevo modelo de datos en red. El modelo de datos en red extendía el modelo jerárquico permitiendo que un registro participara en múltiples relaciones padre/hijo, como se muestra en la figura 4.4. Estas relaciones eran conocidas como *conjuntos* en el modelo en red. En 1971 la Conferencia sobre Lenguajes de Sistemas de Datos publicó un estándar oficial de

bases de datos en red, que se hizo conocido como el modelo CODASYL. IBM nunca desarrollo un DBMS en red por sí mismo, eligiendo en su lugar extender el IMS a lo largo de los años. Pero durante los años setenta, compañías de software independientes se apresuraron a adoptar el modelo en red, creando productos tales como el IDMS de Cullinet, el Total de Cincom y el DBMS.

Para un programador, acceder a una base de datos en red era muy similar a acceder a una base de datos jerárquicos. Un programa de aplicación podía:

- Hallar un registro padre específico mediante clave (como por ejemplo un número de cliente),
- Descender al primer hijo en un conjunto particular (el primer pedido remitido por este cliente),
- Moverse lateralmente de un hijo al siguiente dentro del conjunto (la orden siguiente remitida por el mismo cliente),
- Ascender desde un hijo a su padre en otro conjunto (el vendedor que aceptó el pedido).

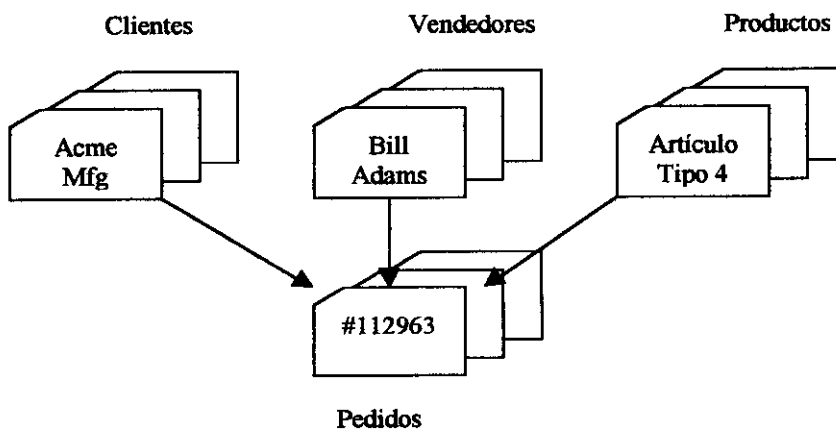


Figura 4.3. Múltiples relaciones padre/hijo.

Una vez más el programador tenía que recorrer la base de datos registro a registro, especificando esta vez qué relación recorrer además de indicar la dirección.

Las bases de datos en red tenían varias ventajas:

Flexibilidad. Las múltiples relaciones padre/hijo permitían a una base de datos en red representar datos que no tuvieran una estructura jerárquica sencilla.

Normalización. El estándar CODASYL reforzó la popularidad del modelo de red, y los vendedores de minicomputadores tales como Digital Equipment Corporation y Data General implementaron bases de datos en red.

Rendimiento. A pesar de su superior complejidad, las bases de datos en red reforzaron el rendimiento aproximándolo al de las bases de datos jerárquicos.

Los conjuntos se representaron mediante punteros a registros de datos físicos, y en algunos sistemas, el administrador de la base de datos podía especificar la agrupación de datos basada en una relación de conjunto.

Las bases de datos en red tenían sus desventajas también. Igual que las bases de datos jerárquicos, resultaban muy rígidas. Las relaciones de conjunto y la estructura de los registros tenían que ser especificadas de antemano. Modificar la estructura de la base de datos requería típicamente la reconstrucción de las bases de datos completa.

Tanto las bases de jerárquicas como las bases en red eran herramientas para programadores.

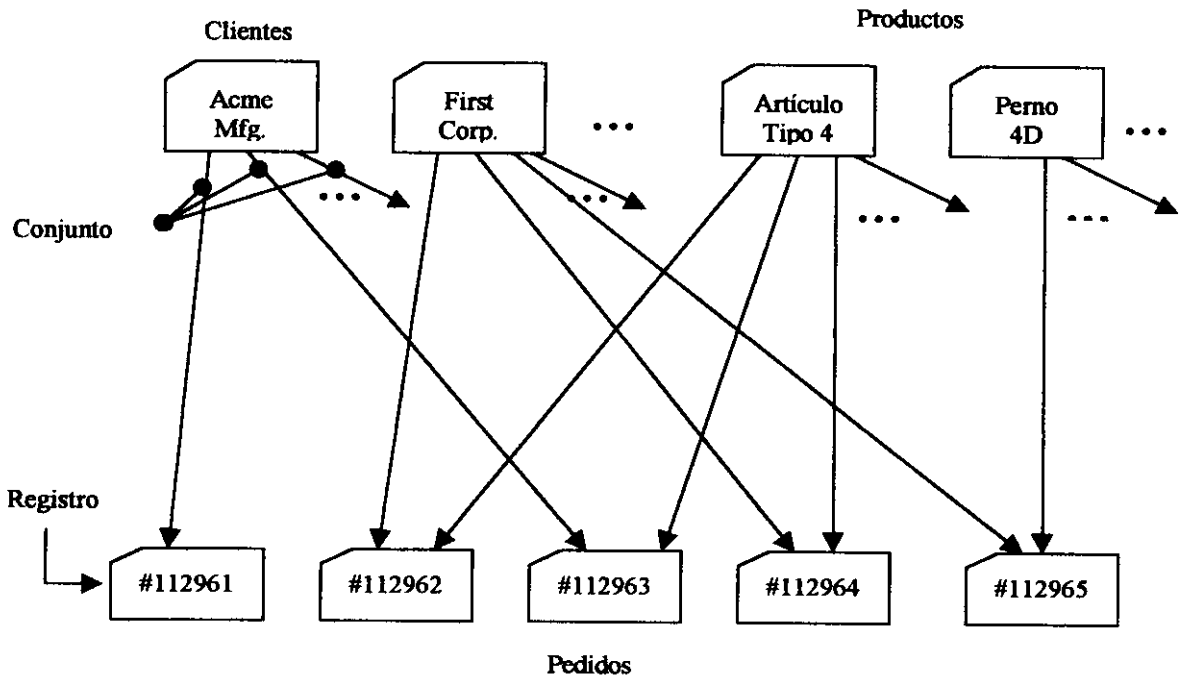


Figura 4.4. Una base de datos en red (CODASYL) para procesamiento de pedidos

4.3. El Modelo de Datos Relacional

El modelo relacional era un intento de simplificar la estructura de las bases de datos. Eliminaba las estructuras explícitas padre/hijo de la base de datos, y en su lugar representaba todos los datos en la base de datos como sencillas tablas fila/columna de valores de datos. La Figura 4.5 muestra una versión relacional de la base de datos en red para procesamiento de pedidos de la figura 4.4.

Una base de datos relacional es una base de datos en donde todos los datos visibles al usuario están organizados estrictamente como tablas de valores, y en donde todas las operaciones de la base de datos operan sobre estas tablas.

La definición está destinada específicamente a eliminar estructuras tales como los punteros estructurados de una base de datos jerárquica o en red. Un DBMS relacional puede representar relaciones padre/hijo, pero éstas se representan estrictamente por los valores contenidos en las tablas de la base de datos.

Tabla PEDIDOS

NUMERO_PEDIDO	EMPRESA	PRODUCTO	CANT
112963	Acme Mfg.	41004	28
112975	JCP Inc.	2A44G	6
112983	Acme Mfg.	41004	6
113012	JCP Inc.	41003	35
.			
.			
.			

Tabla PRODUCTOS

DESCRIPCION	PRECIO	EXISTENCIAS
Artículo Tipo 3	\$107.00	207
Artículo Tipo 4	\$117.00	139
Pasador Bisagra	\$350.00	14

Tabla CLIENTES

EMPRESA	REP_CLI	LIMITE_CREDITO
Acme Mfg.	105	\$50,000.00
JCP Inc.	103	\$50,000.00

Figura 4.5. Una base de datos relacional para procesamiento de pedidos

4.3.1. Tablas

El principio de organización de una base de datos relacional es la *tabla*, una disposición rectangular fila/columna de los valores de datos. Cada tabla de una base de datos tiene un *nombre de tabla* único que identifica sus contenidos. (En realidad cada usuario puede

El dominio de la columna CIUDAD es el conjunto de todos los nombres de ciudades. El dominio de la columna VENTAS es cualquier cantidad de dinero. El dominio de la columna REGION es uno de dos valores de datos, <<Este>> y <<Oeste>> ya que la compañía sólo tiene estas dos regiones de ventas.

Cada columna de una tabla tiene un *nombre de columna*, que se escribe generalmente como encabezamiento en la parte superior de la columna. Las columnas de una tabla deben tener todas nombres diferentes, pero no está prohibido que columnas de tablas diferentes tengan nombres idénticos. De echo, nombres frecuentemente utilizados, tales como NOMBRE, DIRECCION, CNT, PRECIO Y VENTAS, se encuentran a menudo en muchas tablas diferentes de una base de datos de producción.

Las columnas de una tabla tiene un orden de izquierda a derecha, que se define cuando la tabla se crea por primera vez. Una tabla siempre tiene almenos una columna . El estándar SQL ANSI/ISO no especifica un número máximo de columnas en una tabla, pero casi todos los productos comerciales SQL imponen un límite. Generalmente el limite es de 255 columnas por tabla o más.

A diferencia de las columnas, las filas de una tabla no tienen orden particular. De echo, si se utilizan dos consultas de bases de datos consecutivos para visualizar los contenidos de una tabla no hay garantía de que las filas sean listadas en el mismo orden dos veces. Naturalmente se puede solicitar a SQL que ordene las filas antes de visualizarlas, pero el orden de clasificación no tiene nada que ver con la disposición efectiva de las filas dentro de la tabla.

Una tabla puede tener cualquier número de filas. Una tabla de cero filas es perfectamente legal y se denomina una tabla *vacía* (por razones obvias). Una tabla vacía sigue teniendo una estructura, impuesta por sus columnas; simplemente no tiene datos. El estándar ANSI/ISO no limita el número de filas de una tabla, y muchos productos SQL permiten que una tabla crezca hasta que agote el espacio de disco disponible en el computador. Otros productos SQL imponen un límite máximo, pero este es siempre muy generoso –dos millones de filas o más es común.

Tabla OFICINAS

OFICINAS	CIUDAD	REGION	DIR	OBJETIVO	VENTAS
22	DEMVER	OESTE	108	\$300,000.00	\$186,042.00
11	NEW YORK	ESTE	106	\$575,000.00	\$692,637.00
12	CHICAGO	ESTE	104	\$800,000.00	\$735,042.00
13	ATLANTA	ESTE	NULL	\$350,000.00	\$367,911.00
21	LOS ANGELES	OESTE	108	\$725,000.00	\$835,915.00

Campo CIUDAD. Ciudad donde cada oficina está localizada

Campo DIR. Número de empleado del director de la Oficina

Campo VENTAS. Ventas anuales hasta la fecha para la oficina

Figura 4.7. La estructura fila/columna de una tabla relacional

4.3.2. Claves Primarias

En una base de datos relacional bien diseñada cada tabla tiene una columna o combinación de columnas cuyos valores identifican unívocamente cada fila en la tabla. Esta columna (o columnas) se denomina *clave primaria* de la tabla.

Miremos una vez más la tabla OFICINAS en la figura 4.7. A primera vista, tanto la columna OFICINA como la columna CIUDAD podrían servir como clave primaria para la tabla, pero si la empresa se amplía y abre dos oficinas de ventas en la misma ciudad, la columna CIUDAD ya no podría servir como clave primaria.

En la práctica, <<números de ID>>, tales como el número de oficina (OFICINA en la tabla OFICINAS), el número de empleado (NUM_EMP en la tabla REPVENTAS) y los números de clientes (NUM_CLIE en la tabla CLIENTES) se eligen con frecuencia como claves primarias. En el caso de la tabla PEDIDOS no hay elección –lo único que identifica unívocamente un pedido es el número de predio (NUM_PREDIO).

La clave primaria tiene un valor único diferente para cada fila de una tabla de modo que no hay dos filas de una tabla con clave primaria que sean duplicados exactos la una de la otra. Una tabla donde cada fila es diferente de todas las demás se llama una *relación* en términos matemáticos. El nombre << base de datos relacional >> proviene de este término, ya que las relaciones (las tablas con filas distintas) son el corazón de una base de datos relacional.

Tabla PRODUCTOS

ID FAB	ID PRODUCTO	DESCRIPCION	PRECIO	EXISTENCIA
.				
.				
.				
ACI	41003	Articulo Tipo 3	\$107.00	207
ACI	41004	Articulo Tipo 4	\$117.00	139
BIC	41003	Manivela	\$652.00	3
.				
.				
.				

Clave Primaria

Figura 4.8. Una tabla con una clave primaria compuesta

Aunque las claves primarias son parte esencial del modelo de datos relacional, los primeros sistemas de gestión de base de datos relacionales (System/R, DB2, Oracle y Otros), no proporcionaban soporte explícito para claves primarias. Los diseñadores de bases de datos aseguraban generalmente que todas las tablas de sus bases de datos tuvieran una clave primaria, pero el propio DBMS no proporcionaba un modo de identificar la clave primaria de la tabla. DB2 Versión 2, introducido en abril de 1988, añadió finalmente soporte de clave primaria a los productos SQL comerciales de IBM. El estándar ANSI/ISO fue posteriormente ampliado para incluir soporte para claves primarias. Sin embargo el soporte DBMS explícito de claves primarias sigue siendo raro en los productos SQL comerciales.

4.3.3. Claves Foráneas

Una columna de una tabla cuyo valor coincide con la clave primaria de alguna otra tabla se denomina *clave foránea*.

Lo mismo que una combinación de columnas puede servir como clave primaria de una tabla, una clave foránea puede ser también una combinación de columnas. De echo, la clave foránea será *siempre* una clave compuesta(multicolumna) cuando referencia a una tabla con una clave primaria compuesta.

Obviamente el número de columnas y los tipos de datos de las columnas en la clave foránea y en la clave primaria deben ser idénticos unos a otros.

Una tabla puede contener más de una clave foránea si está relacionada con más de una tabla adicional.

Las claves foráneas son parte fundamental del modelo relacional ya que crean relaciones entre tablas en la base de datos. Desgraciadamente, como con las claves primarias, el soporte de claves foráneas falta en los sistemas de gestión de base de datos relacional primitivos. Fueron añadidos a DB2 Versión 2 y desde entonces se han añadido al estándar ANSI/ISO.

Tabla CLIENTES

Tabla REPVENTAS

Tabla PRODUCTOS

NUM. CLIE	EMPRESA	NUM. EMPL	NOMBRE	ID. FAB	ID. PRODUCTO	DESCRIPCIÓN
2103	Acme Mfg.	105	Bill Adams	ACI	41004	Artículo Tipo 4

Tabla PEDIDOS

NUM. PEDIDO	FECHA PEDIDO	CLIE	REP	FAB	PRODUCTO
112963	12/17/1989	2103	105	ACI	41004

Figura 4.9. Múltiples relaciones padre/hijo en una base de datos relacional

4.4. Análisis de La Base de Datos existente

4.4.1. Características Principales

4.4.1.1. Control de concurrencia

- Bloqueo completo de fila sin restricciones
- Bloqueos sin límite por transacción, tabla, o base de datos; sin crecimiento proporcional de los bloqueos
- Consultas libres de contención
- Generación de números secuenciales sin bloqueo-sin I/O (Input/Output) y ni tiempos de espera
- Detección y recuperación automática de deadlocks

4.4.1.2. Algoritmos de I/O

- Commits rápidos máximo un I/O a disco por commit
- Commit en grupo de múltiples transacciones
- Escrituras diferidas a la base de datos
- lectura y escritura de múltiples bloques
- Lecturas "look-aside" - pool de buffers compartido
- Optimizador inteligente de consultas
- Grupos - almacenar los datos provenientes de múltiples tablas juntos en el mismo disco

4.4.1.3. Bases de Datos Distribuidas

- Consulta distribuida en múltiples instalaciones
- Actualización remota en múltiples instalaciones remotas vía Precompiladores
ORACLE
- Actualización remota en una sola instalación vía herramientas Oracle
- Transparencia de la localización
- Autonomía de instalación - no hay dependencia con la instalación central
- Independencia de la red
- Interfaz de arreglos

4.4.1.4. Protocolos de Red

- DECner
- TCP/IP
- APPC (LU6.2)
- 3270 (LU0 y LU2)
- Asíncrona (LU1)
- Novell SPX
- Named Pipes
- Netbios
- DG Xodiac
- Banyan VINES

4.4.1.5. Gateways

- SQL*Connect a DB2
- SQL*Connect a SQL/DS
- SQL*Connect a RMS
- SQL*Connect a TurboIMAGE

4.4.1.6. Implementación del SQL

- Compatibilidad total con ANSI, ISO, DB2
- Extenso agregado de funciones numéricas, de cadenas y de fechas
- UNION, INTERSECT, MINUS
- Auto-uniión interna, unión externa
- Consulta en estructura de árbol para lista de materiales (CONNECT BY)
- Subconsultas, subconsultas correlacionadas
- Sinónimos, vistas, valores nulos
- PL/SQL-extensión de lenguaje procedimental a SQL-interacción, lógica condicional, manejo de cursor, manejo de errores y más

4.4.1.7. Configuración de Bases de Datos

- Diccionario de datos activo
- Eliminar/crear en línea: tablas, columnas, índices, usuarios, espacios de la tabla, vistas

4.4.1.8. 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

4.4.1.9. 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)

4.4.1.10. 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

4.4.1.11. Monitor de Rendimiento Dinámico

- I/O por base de datos, usuario, archivo
- Acceso a tablas por: base de datos, usuario
- Tiempo de uso del procesador y de sesión por usuario, instrucción SQL
- Interpretación, ejecución y conteo de filas por usuario, instrucción SQL

4.4.1.12. Utilitarios Adicionales

- Cargador de datos por volumen
- definición CRT
- Exportación/Importación
- SQL*DBA

4.4.2. Beneficios

Datos Compartidos por toda la Empresa

La insuperable portabilidad y conectividad de DBMS relacional ORACLE permite a todos los sistemas de su organización estar enlazados en un recurso computacional único e integrado.

4.4.2.1. Portabilidad

ORACLE 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 las plataforma de servidores de bases de datos que satisfagan los requerimientos de su sistema.

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

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

4.4.2.2. Datos Compartidos en Forma Distribuida

Las capacidades de interconexión en redes y bases de datos distribuidas de ORACLE 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.

4.4.2.3. Sistemas Abiertos

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

La arquitectura abierta de ORACLE 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.

ORACLE 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.

4.4.2.4. Rendimiento

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

El DBMS de ORACLE 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.

4.4.2.5. Control de Concurrencia

ORACLE 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

4.4.2.6. Bloqueo a Nivel de la Fila.

El bloqueo de ORACLE 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. ORACLE 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.

4.4.2.7. 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 ORACLE 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. ORACLE 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.

4.4.2.8. Eliminación de Cuellos de Botella por I/O

Las tecnologías de ORACLE 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, ORACLE 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.

4.4.2.9. Características Adicionales de Rendimiento

ORACLE 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.

4.4.2.10. 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.

4.4.2.11. 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 ORACLE, 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.

4.5. Recuperación de Datos

Mucha gente utiliza como herramienta de consulta de una base de datos el lenguaje SQL mismo que proporciona extensiones significativas a las capacidades especificadas en el estándar SQL ANSI/ISO.

4.5.1. Sentencias

El lenguaje SQL consta de unas treinta sentencias, que a continuación se resumen en la tabla 4.10. 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, ilustrada en la Figura 4.10.

<i>Sentencia</i>	<i>Descripción</i>
Manipulación de Datos	
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	
Definición de datos	
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

Control de Acceso	
GRANT	Concede privilegios de acceso a un usuario
REVOKE	Suprime privilegios de acceso a un usuario
Control de Transacciones	
COMMIT	Finaliza la transacción actual
ROLLBACK	Aborta la transacción actual
SQL Programático	
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

Figura 4.10. Principales sentencias SQL

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).

4.5.2. 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.

4.5.3. 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 ORACLE 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.

4.5.4. 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.

4.6 Integridad de Datos

El término *integridad de datos* se refiere a la corrección y completitud de los datos en una base de datos. Cuando los contenidos de una base de datos se modifican con sentencias INSERT, DELETE o UPDATE, la integridad de los datos almacenados pueden perderse de muchas maneras diferentes. Por ejemplo:

- Pueden añadirse datos no válidos a la base de datos, tales como un pedido que especifica un producto no existente.
- Pueden modificarse datos existentes tomando un valor incorrecto, como por ejemplo si se reasigna un vendedor a una oficina no existente.
- Los cambios a la base de datos pueden perderse debido a un error del sistema o a un fallo en el suministro de potencia.
- Los cambios pueden ser aplicados parcialmente, como por ejemplo si se añade un pedido de un producto sin ajustar la cantidad disponible para vender.

Una de las funciones importantes de un DBMS relacional es preservar la integridad de sus datos almacenados en la mayor medida posible.

4.6.1. Qué es la Integridad de Datos?

Para preservar la consistencia y corrección de los datos almacenados, un DBMS relacional impone típicamente una o más *restricciones de integridad de datos*.

Estas restricciones restringen los valores que pueden ser insertados en la base de datos o creados mediante una actualización de la base de datos. Varios tipos diferentes de restricciones de integridad de datos suelen encontrarse en la base de datos relacionales, incluyendo:

- Datos Requeridos
- Chequeo de Validez
- Integridad de Entidad
- Integridad Referencial
- Reglas Comerciales

4.6.2. Datos Requeridos

La restricción de integridad de datos más simple requiere que una columna tenga un valor no NULL. El estándar ANSI/ISO y la mayoría de los productos SQL comerciales soportan estas restricciones permitiendo la declaración de que una columna es NOT NULL cuando la tabla que contiene la columna se crea por primera vez, La restricción NOT NULL se especifica como parte de la sentencia CREATE TABLE

Cuando una columna se declara NOT NULL, el DBMS fuerza la restricción asegurando lo siguiente:

- La sentencia INSERT que añade una nueva fila a la tabla debe especificar un valor de datos no NULL para la columna. Un intento de insertar una fila que contenga un valor NULL (explícita o implícitamente) da lugar a un error.
- La sentencia UPDATE que actualiza la columna debe asignarle un valor de datos no NULL. Igualmente, un intento de actualizar la columna con un valor NULL da lugar a un error.

Una desventaja de la restricción no NULL es que generalmente debe ser especificada cuando la tabla se crea por primera vez. Típicamente, no se puede volver atrás una tabla previamente creada y desaprobar los valores NULL para una columna. Generalmente esta desventaja no es seria, puesto que es obvio cuando la tabla se crea por primera vez qué columnas deberían tener permitidos valores NULL y cuáles no.

4.6.3. Comprobación de Validez

El estándar SQL ANSI/ISO proporciona soporte limitado para restringir los valores legales que pueden aparecer en una columna. Cuando se crea una tabla cada columna de la tabla tiene asignado un tipo de datos, y el DBMS asegura que únicamente datos del tipo especificado sean introducidos en la columna.

Varios productos DBMS proporcionan comprobación de validez como parte de sus entradas de datos o paquetes formularios en lugar de soportarla dentro del lenguaje SQL. Por ejemplo el paquete de entrada de datos puede permitir especificar un rango de valores

legales que puedan ser introducidos en un campo de un formulario de entrada de datos. Algunos productos permiten especificar que un valor de dato introducido tiene que ser chequeado con los valores de la base de datos, permitiendo chequeos de validez que no pueden ser implementados con el planteamiento de DB2 o de SQL Server.

4.6.4. Integridad de Entidad

La clave primaria de una tabla debe tener un valor único para cada fila de la tabla o si no la base de datos perderá su integridad como modelo del mundo exterior. Por ejemplo, si dos filas de la tabla REPVENTAS tienen ambas el valor 106 en la columna NUM_EMPL, sería imposible decir qué fila representa realmente a la entidad del mundo real asociada con ese valor clave—Bill Adams, cuyo número de empleado es el 106. Por esta razón la exigencia de que las claves primarias tengan valores únicos se denomina restricción de *integridad de entidad*.

El soporte de claves primarias es aún raro en las bases de datos comerciales SQL. Fue añadido a DB2 en 1988 y ha sido añadido al estándar SQL ANSI/ISO original. Tanto en DB2 como en el estándar ANSI/ISO la clave primaria se especifica como parte de la sentencia CREATE TABLE. El DBMS comprueba automáticamente la unicidad del valor de clave primaria ya existente o de actualizar una fila de modo que la clave primaria pudiera ser un duplicado fallaría con un mensaje de error.

4.6.5. Otras restricciones de Unicidad

A veces es apropiado exigir que una columna que no es clave primaria de una tabla contenga un valor único en cada fila. Por ejemplo, supongamos que se desea restringir los datos en la tabla REPVENTAS de modo que no pueda haber dos vendedores de modo que no pueda haber dos vendedores con exactamente el mismo nombre en la tabla. Se podría conseguir este objetivo imponiendo una restricción de *unicidad* sobre la columna NOMBRE. El DBMS fuerza una restricción de unicidad del mismo modo que fuerza la restricción de clave primaria. Cualquier intento de insertar o actualizar una fila en la tabla que viole la restricción de unicidad fallaría.

El estándar SQL ANSI/ISO utiliza la sentencia CREATE TABLE para especificar restricciones de unicidad en columnas o combinaciones de columnas. Sin embargo, las restricciones de unicidad fueron implementadas en DB2 mucho antes de la publicación del estándar ANSI/ISO, y DB2 las hace parte de su sentencia CREATE INDEX. Esta sentencia es una de las sentencias de administración SQL que tiene que ver con el almacenamiento físico de la base de datos en el disco.

Normalmente el usuario de SQL no tiene que preocuparse de estas sentencias en absoluto; estas sentencias son utilizadas únicamente por el administrador de la base de datos.

La mayoría de los productos SQL comerciales siguen hoy en día la práctica del DB2 en lugar del estándar ANSI/ISO en cuanto a las restricciones de unicidad.

Específicamente SQL Server, Oracle, Ingres, SQLBase y VAX SQL utilizan todos la sentencia CREATE INDEX para implementar restricciones de unidad.

4.6.6. Unicidad y Valores Null

Los valores NULL presentan un problema cuando aparecen en la clave primaria de una tabla o en una columna que está especificada en una restricción de unidad. Supongamos que se intentase insertar una fila con una clave primaria que fuera NULL (o parcialmente NULL, si la clave primaria está compuesta por más de una columna). Debido al valor NULL, el DBMS no puede decidir concluyentemente si la clave primaria está o no duplicada con respecto a otra ya existente en la tabla. La respuesta debe ser <<quizás>>, dependiendo del valor <<real>> del dato que falta (NULL).

Por esta razón SQL requiere que toda columna que forma parte de una clave primaria y toda columna designada en una restricción de unicidad debe ser declarada NOT NULL.

4.6.7. Integridad Referencial

La figura 4.11 muestra las tablas REPVENTAS y OFICINAS, e ilustra una vez más cómo operan las claves foráneas y las claves primarias. La columna OFICINA es la clave primaria para la tabla OFICINAS, e identifica unívocamente a cada fila. La columna OFICINA_REP, en la tabla REPVENTAS, es una clave foránea para la tabla OFICINAS. Identifica la oficina a la cual está asignado cada vendedor.

Tabla OFICINAS

OFICINAS	CIUDAD	REGION	DIR	OBJETIVO	VENTAS
22	DEMVER	OESTE	108	\$300,000.00	\$186,042.00
11	NEW YORK	ESTE	106	\$575,000.00	\$692,637.00
12	CHICAGO	ESTE	104	\$800,000.00	\$735,042.00
13	ATLANTA	ESTE	NULL	\$350,000.00	\$367,911.00
21	LOS ANGELES	OESTE	108	\$725,000.00	\$835,915.00

Clave Primaria

Referencia

Tabla REPVENTAS

Clave Foránea

NUM_EMPL	NOMBRE	EDAD	OFICINA REP	TITULO
105	Bill Adams	37	13	Rep Ventas
109	Mary Jones	31	11	Rep Ventas
102	Sue Smith	48	21	Rep Ventas
106	Sam Clark	52	11	VP Sales
104	Bob Smith	33	12	Dir Ventas
101	Dan Roberts	45	12	Rep Ventas
110	Tom Snider	41	NULL	Rep Ventas
108	Larry Fitch	62	21	Dir Ventas
103	Paul Cruz	29	12	Rep Ventas
107	Nancy Angelle	49	22	Rep Ventas

Figura 4.11. Referencia Clave Primaria Clave Foránea

Las columnas OFICINA_REP y OFICINA crean una relación padre/hijo entre las filas OFICINAS y REPVENTAS, Cada fila OFICINAS (padre) tiene cero o más filas REPVENTAS (hijo) con números de oficina coincidentes. Análogamente, cada fila REPVENTAS (hijo) tiene exactamente una OFICINAS (padre) con un número de oficinas coincidente.

Supongamos que se intenta insertar una nueva fila en la tabla REPVENTAS que contenga un número de oficina inválido, como en este ejemplo:

INSERT INTO REPVENTAS

(NUM_EMPL, NOMBRE, OFICINA_REP, EDAD, CONTRATO, VENTAS)

VALUES (115, 'George 'Smith', 31, 37, '01-ABR-90', 0.00)

Aparentemente, no hay nada erróneo en esta sentencia INSERT. De hecho, la mayoría de las implementaciones SQL actuales añadirían sin problema la fila.

La base de datos mostrará que George Smith trabaja en la oficina número 31, aún cuando no existe la oficina número 31 listada en la tablas OFICNAS. La fila recién insertada <<quiebra>> claramente la relación padre/hijo entre las tablas OFICINAS y REPVENTAS. De hecho, el número de oficina en la sentencia INSERT es probablemente un error – el usuario puede haber pretendido escribir el número de oficina 11,21 o 13.

Parece claro que todo valor legal de la columna OFICINA_REP debería ser forzado para que corresponda algún valor de los que aparezca en la columna OFICINA. Esta regla se conoce como restricción de *Integridad Referencial*. Asegura las relaciones de la integridad de las relaciones padre/hijo creadas mediante claves foráneas y claves primarias.

La integridad referencia ha sido una parte esencial del modelo relacional desde que fue propuesto por primera vez por Codd. Sin embargo, las restricciones de integridad referencial no fueron incluidas en le prototipo System/R de IBM, ni en las primeras ediciones de DB2 o SQL/DS, y también faltan de la mayoría de los productos SQL comerciales hoy día. IBM añadió el soporte de integridad referencia a DB2 en 1989, y la integridad referencial ha sido añadido recientemente al estándar SQL ANSI/ISO. La mayoría de los vendedores de DBMS han hecho planes para incluir el soporte de integridad referencial en las revisiones futuras de sus productos.

lenguaje SQL proporciona un 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.

La arquitectura cliente/servidor ha recibido gran atención con la introducción de las redes PC basadas en OS/2. SQL Server, el Servidor Oracle para OS/2 y SQL Base de Gupta Technologies utilizan esta estrategia.

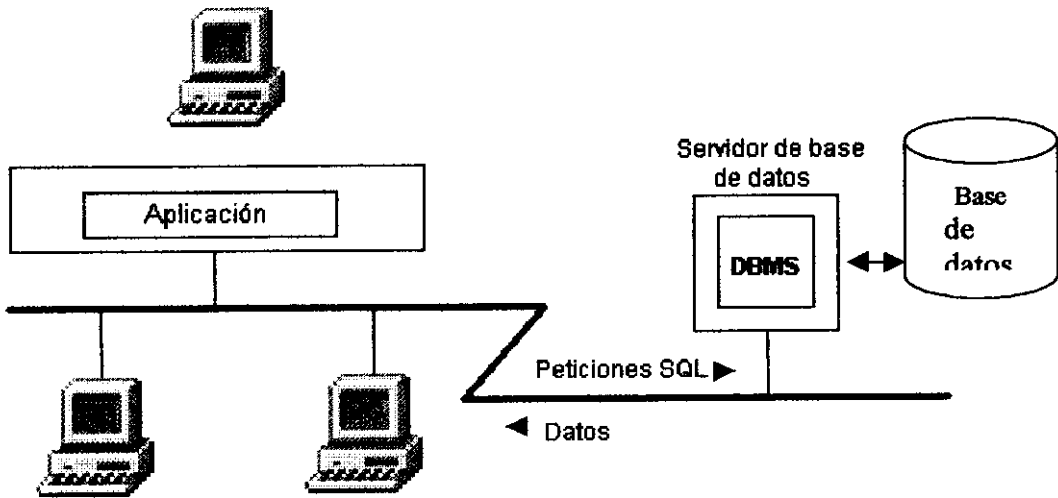


Figura 4.12. Gestión de base de datos en una arquitectura cliente/servidor

CAPITULO V

5. ANALISIS Y DISEÑO DEL SISTEMA

5.1.1. 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 Sagrario Ltda.		X		X	
San Francisco Ltda.			X		X
Oscus Ltda.	X				
Cámara de Comercio				X	X
La Merced Ltda.					X

Protocolos de Comunicación.

Cooperativa / Com.	TCP/IP	NetBeui	IPX/SPX
El Sagrario Ltda.	X		
San Francisco Ltda.	X		X
Oscus Ltda.	X		
Cámara de Comercio		X	
La Merced Ltda.		X	X

Velocidad Módem.

Cooperativa / Mod.	14.4 Kbps	33.6 Kbps	56 Kbps
El Sagrario Ltda.	X		
San Francisco Ltda.		X	
Oscus 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 necesidad de amplificarlas, es de varios kilómetros; en el caso de la transmisión analógica hay que añadir amplificadores cada 5 ó 6 Km, y en el caso de las señales digitales hay que añadir repetidores cada 2 ó 3 Km.

Los cables de pares se han venido utilizando tanto en la transmisión analógica como digital, principalmente en aplicaciones de baja velocidad. En la actualidad, su uso en transmisión de datos se ha visto en aumento debido al desarrollo de nuevas técnicas de codificación que permiten la transmisión digital a velocidades medianamente altas (10 Mbps).

Aparte de las características ya mencionadas, su adecuado comportamiento y su bajo coste han hecho que sean utilizados ampliamente y es probable que su presencia permanezca por muchos años.

5.3. Método de comunicación a utilizar.

Para la elección de método de comunicación que se puede utilizar en nuestro proyecto, se han tomado en cuenta varios aspectos básicos que habrán de considerarse para una comunicación óptima.

Compatibilidad.- Definir los niveles de compatibilidad que el servicio ha de cumplir para las plataformas de sistemas operativos que se utilizan en las Cooperativas de Ahorro y

Crédito de nuestra ciudad, es necesario saber si la utilización de un medio de comunicación específico exigirá la carga de drivers adicionales sobre su sistema operativo, así como en qué medida dichos drivers pueden llegar a afectar a otros subsistemas del sistema operativo.

Arquitectura.- Una de las cosas que han de ser tenidas especialmente en consideración es la manera en que las soluciones o servicios realizan las comunicaciones entre las diferentes instituciones. Cada uno de los entornos de trabajo tiene sus características peculiares, es fundamental analizar cuidadosamente los diferentes modelos de precios existentes y las restricciones de utilización asociadas a las distintas ofertas que satisfacen necesidades de arquitectura específicas.

Actualización.- Antes de optar entre una u otra alternativa, debe determinar el alcance de las actualizaciones que habrá de realizar en sus dispositivos de red instalados para dotarlos de capacidad para soportar el servicio. Puede ser que la migración suponga introducir actualizaciones de hardware en routers, conmutadores, cortafuegos, servidores, etc. Las actualizaciones de software resultan más baratas que las modificaciones en el hardware.

Seguridad.- En que medida los productos y servicios se basan en los estándares de la industria. Los estándares de seguridad contribuyen además a mantener un alto nivel de seguridad interna puesto que cuanto más se utilicen menos agujeros dejaremos sin cubrir.

Integración.- Cómo se integrarán los dispositivos de comunicaciones y red para que la comunicación resulte un éxito, es importante que se asegure que todos los recursos de cada

institución puedan realmente soportar la solución planteada. También deberá plantearse si las cooperativas tendrán que migrar recursos y aplicaciones, y conocer las distintas vías existentes para que los productos de comunicación se implanten en cada institución.

Se tomarán en cuenta los aspectos descritos anteriormente, además de los recursos existentes en cada cooperativa que se describen a continuación:

Cooperativa	Recurso	T. Windows	Módem	Línea Telefónica
El Sagrario Ltda.	X	X	X	X
San Francisco Ltda.	X	X	X	X
Oscus Ltda.	X	X	X	X
Cámara de Comercio	X	X	X	X
La Merced Ltda.	X	X	X	X

Se ha optado por la comunicación telefónica ya que es la más común entre las instituciones involucradas en nuestro estudio, además cumple con los requisitos en compatibilidad, arquitectura, actualización, seguridad e integración estudiados anteriormente. El módem, permite hacer uso de las líneas telefónicas para la comunicación entre computadoras separadas por grandes distancias, por lo que lo estudiaremos a continuación.

5.3.1. Módem.

Al igual que el teléfono permite la comunicación de dos personas situadas a distancia, es posible comunicar dos computadoras, haciendo uso de la red de comunicaciones más

difundida en el mundo: la red telefónica. Recordando la naturaleza digital de las computadoras y la analógica de la red telefónica, es posible darse cuenta de la necesidad de algún dispositivo capaz de realizar la adecuada transformación. Estos dispositivos son los módem.

Los módem son elementos del canal de comunicación que permiten utilizar la red telefónica e integrarla en las comunicaciones entre computadoras. Un módem es un dispositivo que convierte señales digitales dispuestas en serie en señales analógicas, y viceversa. Las principales funciones del módem son:

- Convierte la señal digital de datos en una señal analógica de un ancho de banda determinado, por ejemplo entre 300 y 3.400 Hz para canales telefónicos, adaptando de esta manera la señal a la línea.
- Corrige los defectos que la línea de transmisión pueda presentar mediante circuitos compensadores.
- Detecta errores de transmisión.
- Extrae información digital de la señal analógica que recibe.

En general, el módem se comportará de forma que las computadoras tengan la sensación de estar la una al lado de la otra, sin sufrir las posibles complicaciones que la red telefónica puede implicar.

Como ya se ha dicho, el módem realiza la codificación de información digital, que es la que manejan las computadoras, utilizando señales analógicas, que son utilizadas en las líneas telefónicas.

5.4. Sistema Operativo a utilizar .

En los últimos años, ha existido un fuerte crecimiento del sistema operativo Windows NT. Este crecimiento es muy significativo al haberse producido en uno de los campos más difíciles en el mundo de la informática, el de los sistemas operativos de los servidores de las empresas.

Por contra, en el área de los sistemas operativos de red nunca estamos totalmente seguros de que todo funciona correctamente hasta que no hemos puesto en marcha el sistema en una instalación real. Esta es la principal razón por la que nos ha resultado complicado decidir con que S.O. trabajará este software piloto.

La llegada de NT ha revolucionado a las pequeñas, medianas y grandes empresas. Efectivamente, Netware era más fácil de configurar y su protocolo de red (IPX) era extremadamente eficiente en pequeñas redes. Por contra, Unix era más complejo de configurar inicialmente, aunque mucho más flexible y potente.

Además como se ha comentado anteriormente, una de las grandes ventajas de NT consiste en su facilidad de instalación. En general, el proceso de instalación de NT consiste en arrancar con los disquetes suministrados por el fabricante (incluso es posible evitar tener que usarlos si se dispone de una conexión a un CD-ROM o a una red) y luego seguir los

pasos que van apareciendo en secuencia en pantalla para ir pasando por las sucesivas fases de la instalación.

Por contra, el planteamiento que se da en el sistema operativo Unix es radicalmente contrario. A pesar de que en los últimos tiempos hemos asistido a una tendencia en la cual todos los fabricantes han intentado incorporar a sus sistemas una especie de instalación automatizada (especialmente en los sistemas de sobremesa), la realidad es que, en general, instalar un sistema Unix es más complicado que instalar NT. Para empezar, partimos que el hardware utilizable es mucho más amplio: los Unix de cada marca suelen estar pensados para ordenadores de su propia marca (HP-UX para Hewlett-Packard, DG/UX para Data General, etc.), pero siempre existen versiones (Linux, por ejemplo) disponibles para un amplio número de plataformas.

Otra característica que incluyen NT y Netware pero no la mayoría de las versiones de Unix es la capacidad de autodetectar el hardware. Hay que considerar que la gran parte de los sistemas Unix están compilados para una plataforma determinada y por lo tanto el hardware no contemplado en el sistema es muy poco. El caso de Linux es un tanto especial por la cantidad de dispositivos distintos para los cuales está pensado que controle.

En lo referente a la administración, la duda vuelve a ser la misma: ¿qué es preferible un sistema en el cual las tareas de administración son increíblemente sencillas pero limitadas (NT), un sistema complejo pero flexible (Unix) o una solución intermedia (Netware)?

Para contestar a esta pregunta, la cuestión fundamental es saber de cuántos técnicos informáticos con capacidad de atender al sistema se dispone. Si no hay ninguno, NT se vislumbra como la mejor solución posible, ya que cualquier usuario un poco avezado será capaz de realizar la administración sin mayores dificultades.

Por el contrario, cuando se dispone de un técnico informático que pueda destinar todo o parte de su tiempo a la administración del sistema, las cosas ya no están tan claras, y Unix y Netware ganan terreno.

5.4.1. El punto de vista de los sistemas abiertos.

Se define sistema abierto como aquel que cumple tres características: portabilidad, escalabilidad e interoperabilidad. Tal como se verá a continuación, no se puede considerar que ni NT ni Netware lo sean, si bien es cierto que ambos demuestran una clara intención de convertirse en sistemas abiertos.

Portabilidad.

Se define portabilidad como la posibilidad de que aplicaciones de distintos desarrolladores de software funcionen en máquinas de distintos fabricantes. Evidentemente, la primera condición que se debe dar para que esta característica se pueda cumplir es que el sistema operativo en cuestión esté disponible en cada una de las plataformas que se estén estudiando.

Existe una versión de Unix casi para cualquier plataforma del mercado. NT se encuentra limitado a que la capa de abstracción de hardware (HAL, Hardware Abstraction Layer) no se ejecuta más que en cuatro plataformas (MIPS, Alpha de Digital, PowerPC e Intel), siendo portables las aplicaciones desarrolladas para NT a esas cuatro plataformas. De Netware solo hay disponibles versiones para plataforma Intel x86, con lo cual su portabilidad es nula.

El HAL de NT presenta la gran ventaja de que el desarrollador de aplicaciones para NT no tiene que considerar en cual de las plataformas se va a ejecutar su programa, dado que las APIs son idénticas entre todas las copias de Windows NT y para portar una aplicación de una plataforma a otra no hay que hacer otra cosa que recompilarla.

En el caso de Unix, esto también debería ser así pero, al depender el código directamente de la plataforma en la cual se va a ejecutar, suelen existir diferencias, especialmente en lo que respecta a la interpretación del contenido de cada uno de los tipos de datos, que hacen que normalmente sea necesario tocar el programa fuente o incluir trozos de código.

Escalabilidad.

Se define escalabilidad como la posibilidad de utilizar el mismo entorno de software en diferentes gamas de ordenadores, desde ordenadores de sobremesa hasta grandes ordenadores.

Unix es escalable, como lo demuestra su presencia en máquinas 8088 hasta grandes máquinas de proceso paralelo o distribuido. En general, las principales plataformas Unix pueden escalarse hasta 16 CPU o más y algunas, como Sun, alcanzan los 64 procesadores. Según el Gartner Group, Unix puede dar cabida a 1.500 usuarios simultáneos y se prevé que en 1999 alcance los 3.000.

NT, igual que antes, consigue un cierto posicionamiento, que permiten que NT se ejecute en servidores de hasta 8 procesadores. Actualmente, NT soporta unos 400 usuarios simultáneos.

Por último, Netware, como ya hemos indicado anteriormente se queda rezagado al no ejecutarse en otras máquinas que no sean Intel.

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.

Otro aspecto a considerar es que, en entornos estadísticos, suele ser deseable que las máquinas permanezcan en funcionamiento las 24 horas del día, si es posible los 365 días del año, porque suelen aprovecharse el horario que no es de oficina para lanzar las tareas más pesadas desde el punto de vista del tiempo de máquina necesario.

En este aspecto NT deja bastante que desear. En primer lugar, la gestión de DLLs de NT hace que después de instalar muchos de los paquetes existentes e incluso después de reconfigurar cualquier pequeño detalle del sistema, sea necesario reinicializar la máquina. Los propios técnicos de Microsoft recomiendan que los servidores NT sean apagados como mínimo un par de veces al mes para permitir que se lleve a cabo una especie de "autopurgación".

También Netware necesita ser detenido alguna que otra vez al mes como mínimo, aunque hay casos en los cuales no se detiene casi nunca y funciona correctamente.

Por último, el sistema operativo Unix no necesita bajadas (son bastantes los entornos en los que lleva funcionando varios años sin haberse detenido jamás), solo que en estos casos es necesario realizar algunas operaciones de mantenimiento cada cierto tiempo.

5.4.3. Seguridad.

Para garantizar que un sistema es seguro, se utiliza un ranking, que ya es un estándar de facto, expedido por el Centro nacional de seguridad de ordenadores (NCSC, National Computer Security Center). En este ranking se obtienen niveles sucesivos de seguridad, que de menor a mayor son los siguientes: D, C1, C2, B1, B2, B3 y A1 ().

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.

En el acceso a control remoto, la computadora remota toma el control de la computadora de la red en que ya se ha introducido. En otras palabras, la computadora remota se convierte en monitor y teclado remoto del nodo LAN. El proceso se realiza más en la computadora host, que en la computadora remota.

En el proyecto se trabajará con el acceso a nodo remoto, ya que tiene una ventaja importante frente al acceso a control remoto: la simplicidad. El RAS de Windows NT Server proporciona un servicio de acceso a nodo remoto para usuarios que llaman a la red de computadoras. Un beneficio clave de la utilización del acceso a nodo remoto es que el sistema completo de seguridad se aplica a quien llama de forma remota.

Una vez que el RAS está instalado y configurado en un servidor de Windows NT, cualquier usuario que tenga una autorización y un código de acceso apropiados puede comunicarse con el servidor desde cualquier computadora con módem.

Las computadoras se comunicarán de la siguiente manera:

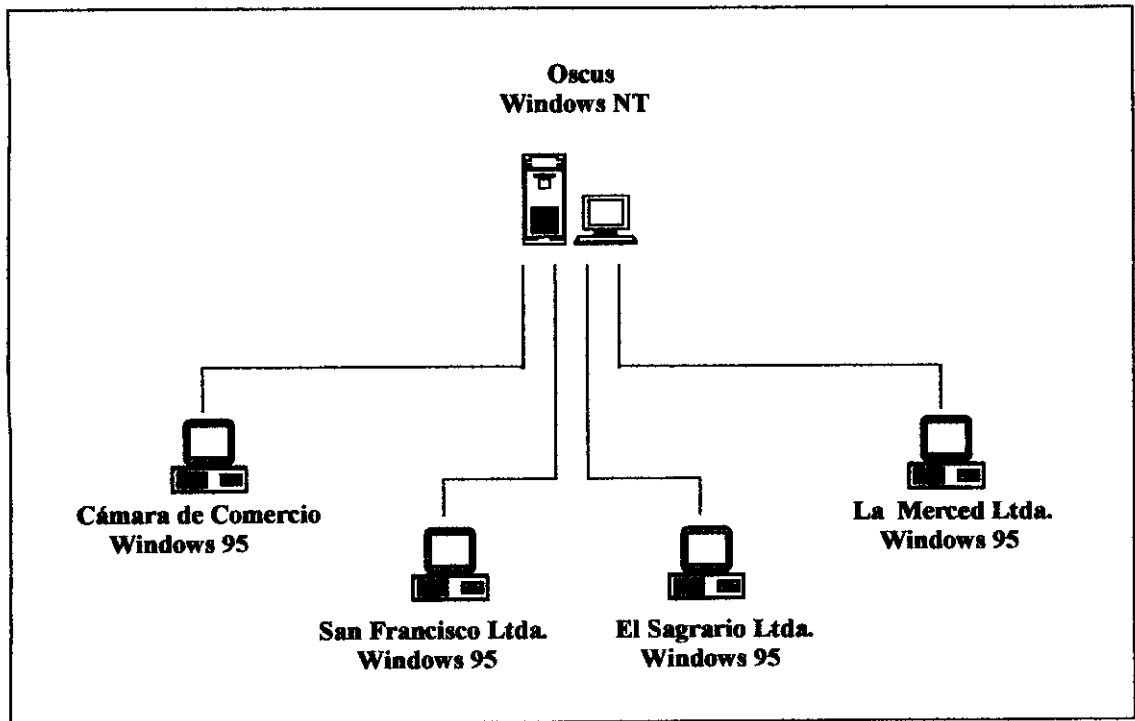


Figura.5.1 Esquema de la Comunicación a Implantar

5.5. Lenguaje de Programación a utilizar

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, han hecho de este lenguaje un líder indiscutible en lo que a desarrollo de aplicaciones se refiere. Con la versión 5.0. la gestión de bases de datos

funciona en un muy alto nivel, pudiendo gestionar bases de datos de tipo Access, Paradox, dBASE, FoxPro, ODBC, etc.

Este paso de gigante ha hecho de Visual Basic uno de los lenguajes favoritos por los desarrolladores de aplicaciones de bases de datos, en especial el hecho de que Visual Basic implemente el lenguaje SQL, uno de los más potentes y sencillos lenguajes de bases de datos.

5.5.1. Lenguaje estructurado SQL

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 los motores de bases de datos, están desarrolladas o basadas en este lenguaje, por lo que su implementación en Visual Basic no es complicada.

El objetivo principal de SQL es la realización de consultas y cálculos con los datos de una o varias tablas; en el proyecto se ha utilizado este lenguaje estructurado ya que nos permite transformar la información almacenada en tablas de diferentes plataformas en archivos planos o de texto y viceversa.

5.5.1.1. Consejos para escribir mandatos en SQL.

He aquí una serie de consejos (a veces normas), que hay que tener en cuenta a la hora de escribir mandatos SQL en nuestras aplicaciones en Visual Basic:

1. Un mandato en SQL se expresa en una cadena de caracteres o String.
2. Dicho mandato se puede escribir en la propiedad RecordSource de un control Data (más adelante, podremos prescindir del control Data para realizar nuestras consultas), con el fin de crear una consulta en la interfaz.
3. Los nombres de los campos especificados (y de las tablas), que contengan más de una palabra, han de encerrarse entre corchetes ([nombre]). Como norma general, se suelen escribir siempre entre corchetes.
4. Para especificar un determinado campo de una determinada tabla, se ha de escribir primero el nombre de la tabla, un punto y, a continuación, el nombre del campo (nombre_tabla.nombre_campo).
5. Al especificar una expresión de búsqueda, si ésta se refiere a una expresión de caracteres, éstos han de encerrarse entre comillas, normalmente simples ('expresión_a_buscar').
6. Para especificar una fecha en una búsqueda, ésta debe encerrarse entre almohadillas o pragmas (#fecha#).
7. Si se utiliza la propiedad RecordSource del control Data, para crear nuestras consultas en SQL, tras introducir el mandato SQL (siempre como una expresión de cadena) es necesario refrescar el control Data (control_data.Refresh).

5.5.1.2. Mandato SQL estándar.

El lenguaje SQL está compuesto por una serie de sentencias y de cláusulas muy reducidas en número, pero muy potentes en efectividad. De entre todas las palabras, existen cuatro que son las más utilizadas, estando compuestas por una sentencia y por tres cláusulas:

```
SELECT lista_campos FROM lista_tablas [WHERE criterios [ORDER BY lista_campos]]
```

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'.

Cláusula WHERE.

La cláusula **WHERE** es opcional, y permite seleccionar qué registros aparecerán en la consulta (sino se especifica aparecerán todos los registros). Para indicar este conjunto de registros se hace uso de criterios o condiciones, que no es más que una comparación del contenido de un campo con un determinado valor (este valor puede ser constante (valor predeterminado), el contenido de un campo, una variable, un control, etc.).

Ejemplos que ilustran el uso de esta cláusula:

```
SELECT * FROM clientes WHERE nombre = 'ALFREDO';
```

Selecciona todos los campos de la tabla 'clientes', pero los registros de todos aquellos clientes que se llamen 'ALFREDO'.

5.6. Base de Datos a utilizar.

Microsoft Access es un sistema de gestión de bases de datos relacionales que permite almacenar y recuperar la información, de acuerdo con las relaciones que se hayan establecido, en las tablas de una base de datos.

Las herramientas que utiliza Microsoft Access para llevar a cabo la gestión de la información almacenada en una base de datos, se componen de objetos.

El termino objeto se puede referir a alguna de las siguientes entidades:

Tabla.- Conjunto de datos acerca de un tema particular. Los datos de la tabla se representan en columnas (campos) y filas (registros) siguiendo el modelo relacional.

Consulta.- Pregunta que se plantea a Access sobre el contenido de una base de datos. Los datos que responden a la pregunta pueden provenir de una o varias tablas. El conjunto de registros que responden a la consulta se denomina hoja de respuesta dinámica.

Formularios.- Diseño personalizado para introducir, modificar y ver los registros de una base de datos (por defecto, los datos se presentan de forma tabular). Al diseñar un formulario se especifica la posición y colores para resaltar datos, presentación de mensajes y la posibilidad de introducir datos en varias tablas a la vez.

Informes.- Objeto que permite imprimir registros con un diseño personalizado. Los elementos de un formulario o de un informe que permite presentar o imprimir los datos de

una forma personalizado se llaman controles. Mediante un informe podrá agrupar registros, presentar los datos de un campo, los resultados de un cálculo, el texto para un título o mensaje, un gráfico e incluso otro formulario o informe.

Macros.- Microsoft Access también nos proporciona la posibilidad de programar para combinar objetos de la base de datos y así crear un sistema de administración de datos que se adapte a las necesidades de cada usuario. Una macro es una lista de acciones que Access efectuará de forma automática.

5.7. Requerimientos de la institución.

De acuerdo al estudio realizado en cada una de las instituciones que intervienen en nuestro proyecto, los requerimiento 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.

5.8.4. Recursos de comunicaciones.

Los recursos de comunicaciones a ser utilizados son:

- Dos líneas telefónicas digitales.
- Dos módems externos tipo estandar.

5.9. Recursos que necesitan las Instituciones.

Para la implementación del software piloto las instituciones deberán cumplir con ciertos requisitos en lo referente a sus recursos humanos, software, hardware y comunicaciones los mismos que detallamos a continuación:

5.9.1. Recursos humanos.

Se necesita una persona por Cooperativa la misma que será encargada de manipular el software desarrollado para la actualización de datos.

5.9.2. Recursos del hardware.

Cada institución necesita poner al servicio del proyecto una computadora que deberá tener como mínimo las siguientes características:

- Un procesador 80486 o superior
- 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 basaremos 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.

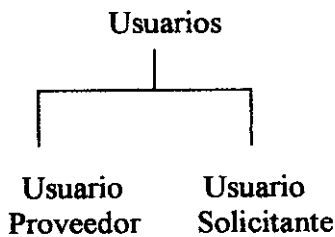
Agente	Acción	Receptor	Resultado
Usuario	Solicitar información	Servidor base	Recibir información
Servidor Base	Procesar información	Usuario	Enviar información
Conexión	Conectar	Servidor base Usuario	Conexión
Comunicación	Comunicar	Conexión	Comunicación
Transmisión	Transmitir	Servidor base Usuario	Enviar datos
Integración	Integrar	Servidor base	Integrar plataformas
Generar	Generar reportes	Usuario	Reporte

5.10.2. Fases del Diseño.

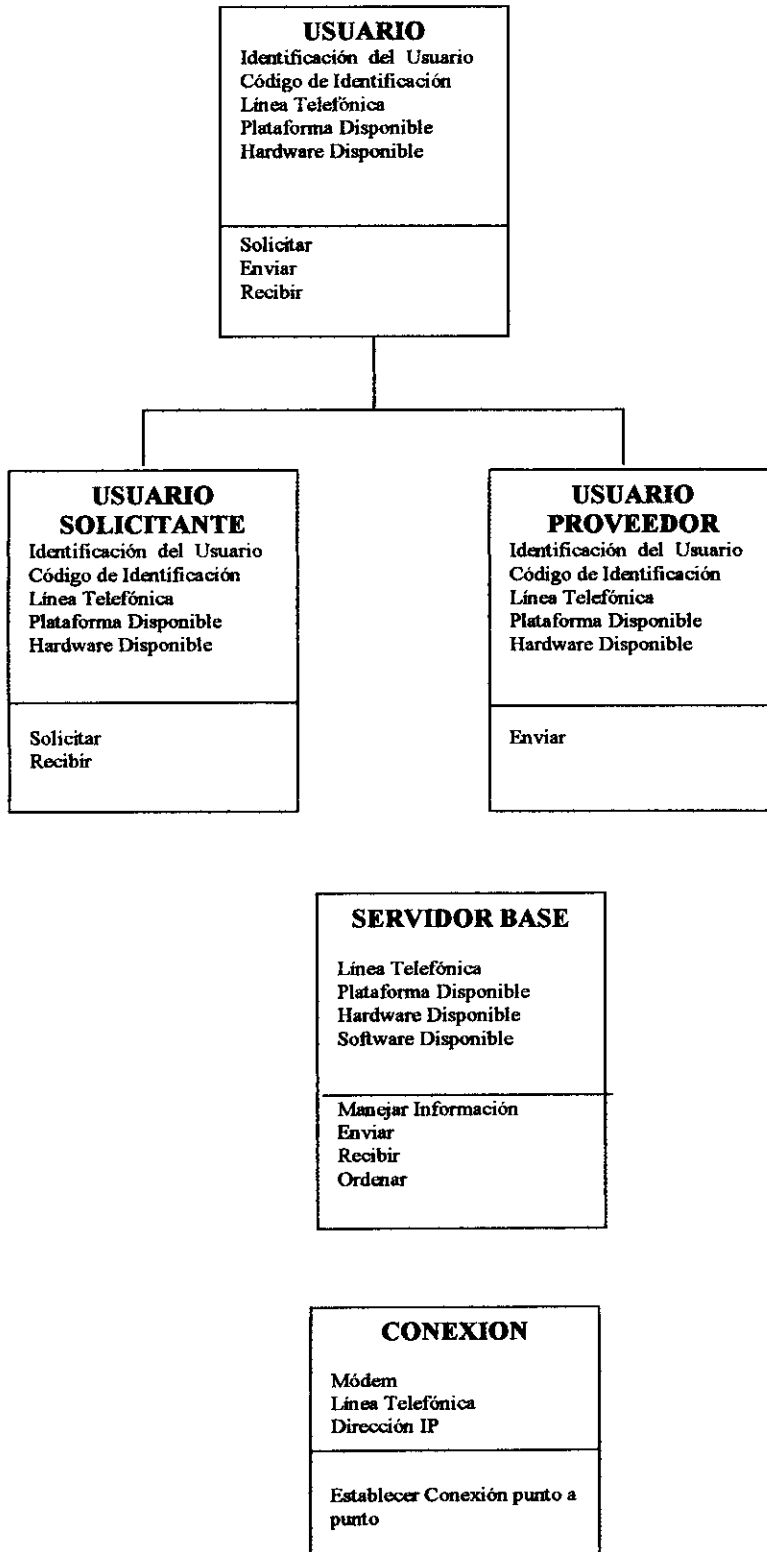
Se refiere al refinamiento de las clases, responsabilidades, atributos y responsabilidades y de los diferentes diagramas existentes.

5.10.2.1. Refinamiento de las Clases.

A continuación luego del análisis estas clases tendrán las siguientes clases análogas.



5.10.2.2. Refinamiento de los Atributos y Responsabilidades.



COMUNICACION Velocidad Conexión RAS
Comunicación Datos

TRANSMISION Medio de Transmisión Método de Transmisión Modos de Transmisión Equipo de Transmisión
Transmitir Información

INTEGRACIÓN Plataforma Nativa Plataforma Base Campos de los Datos Registros Filtros Ordenación
Integrar Sistemas Nativos de Diferentes Plataformas

GENERAR Tipos de Filtros Campos
Generar reporte

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.

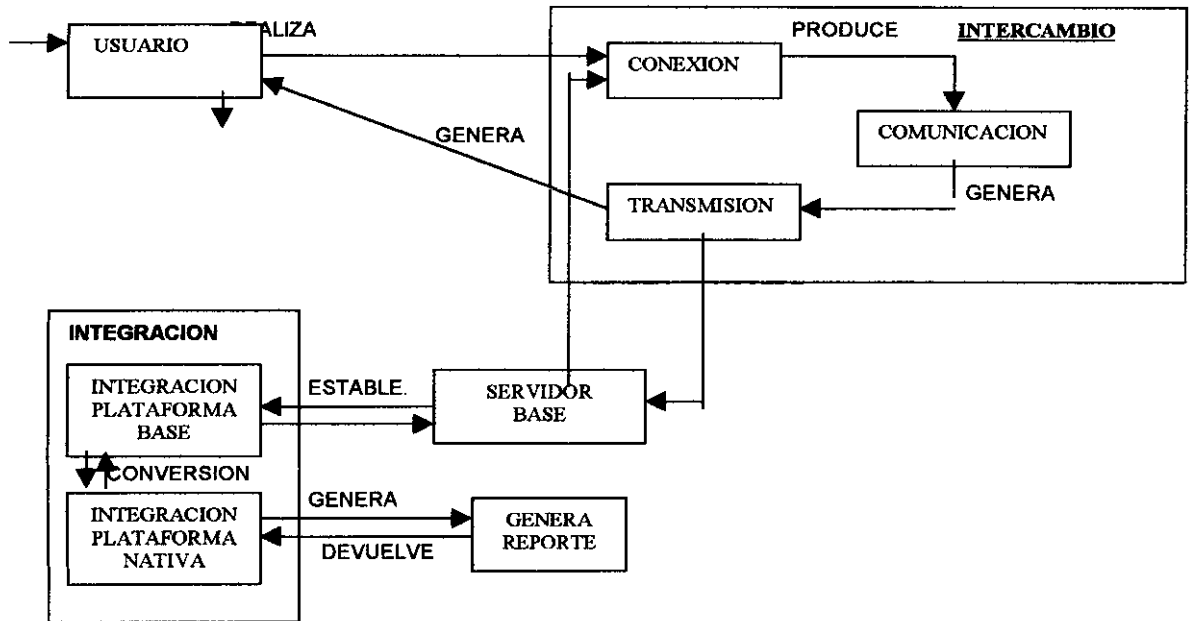


Figura 5.2. Diagrama de transición de estados.

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.

Nodo : Integración plataforma nativa.
Alcanzado desde : Integración plataforma base, generar informes.
Acción : Transforma plataforma nativa.
Salida : Genera informe, integración plataforma base.

5.10.2.5. Identificación de los contratos.

1. **Clase** : Usuario.
Operación : Solicita informe, transmitir información.
2. **Clase** : Servidor base.
Operación : Manipular la información.
3. **Clase** : Conexión
Operación : Realiza la conexión entre los dispositivos
4. **Clase** : Comunicación.
Operación : Comunicación entre usuarios.

- 5. **Clase** : Transmisión.
Operación : Optimización de los datos para su transmisión.

- 6. **Clase** : Integración.
Operación : Cambiar los sistemas nativos a una plataforma base y viceversa.

- 7. **Clase** : Generación
Operación : Crear el informe solicitado por el usuario

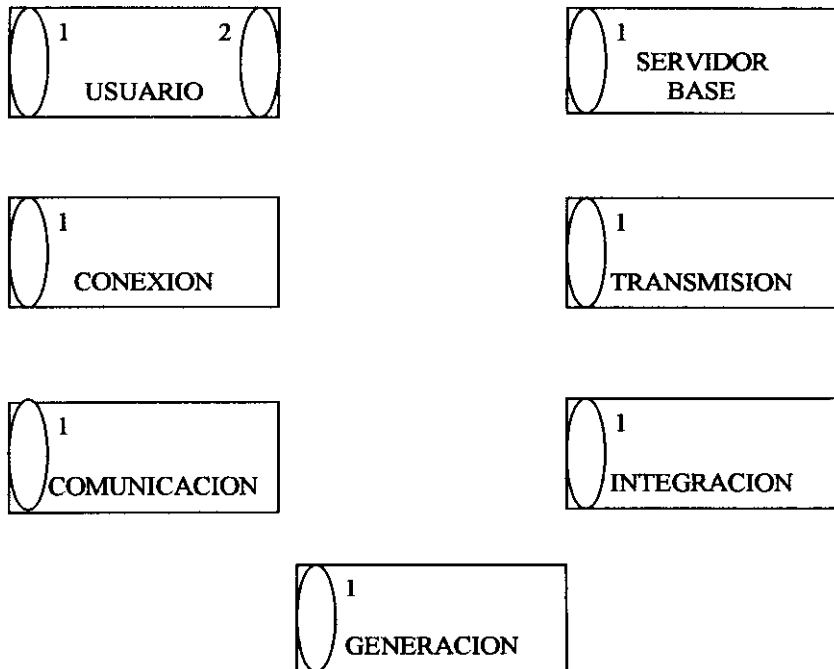


Figura 5.3. Identifica los contratos de cada clase

5.10.2.6. Grafos de colaboraciones.

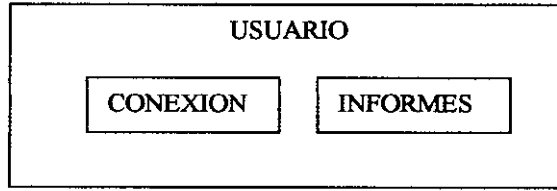


Figura 5.5. Grafo de colaboración de la clase usuario

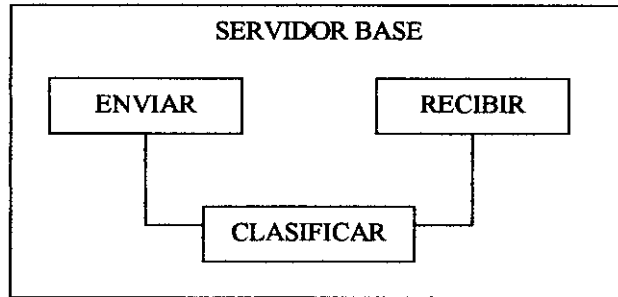


Figura 5.6. Grafo de colaboración de la clase servidor base

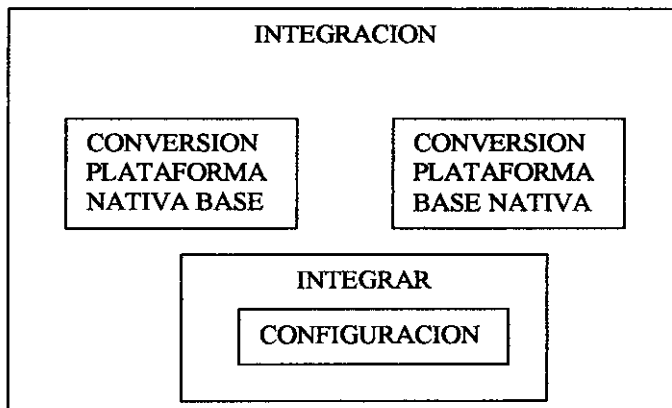


Figura 5.7. Grafo de colaboración de la clase integración.

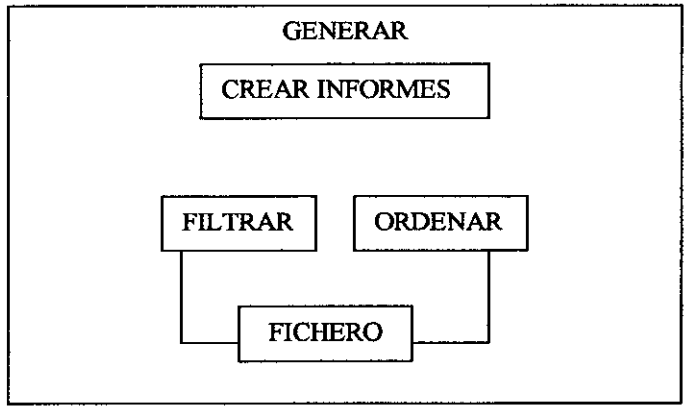


Figura 5.8. Grafo de colaboración de la clase generar.

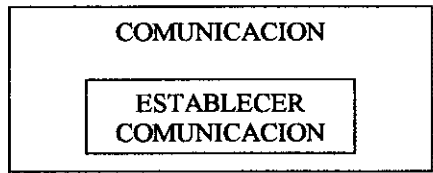


Figura 5.9. Grafo de colaboración de la clase comunicación

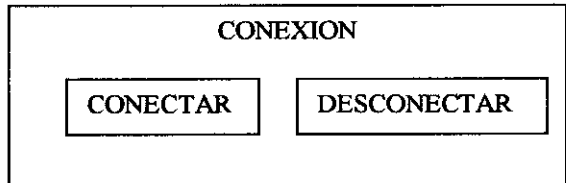


Figura 5.10. Grafo de colaboración de la clase conexión.

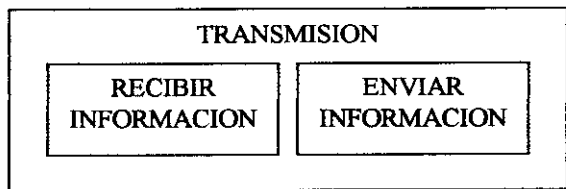


Figura 5.11. Grafo de colaboración de la clase transmisión.

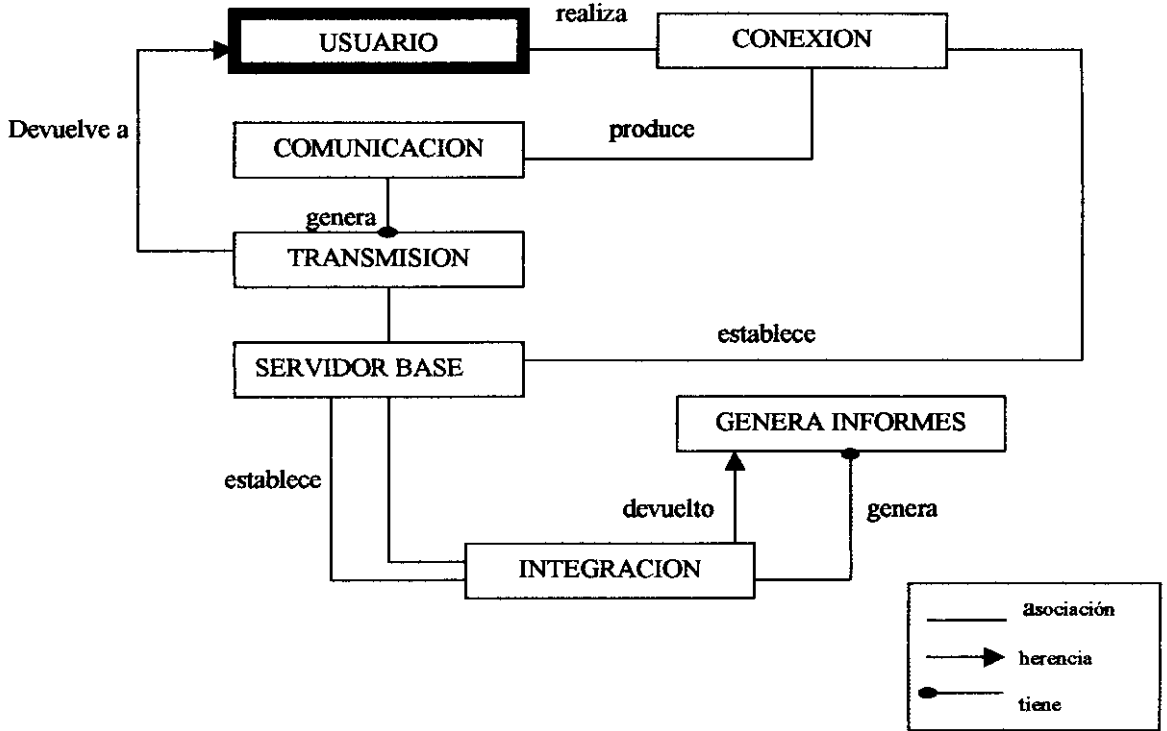
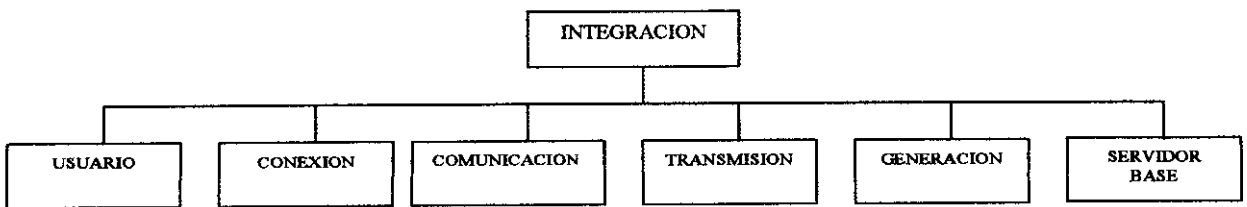
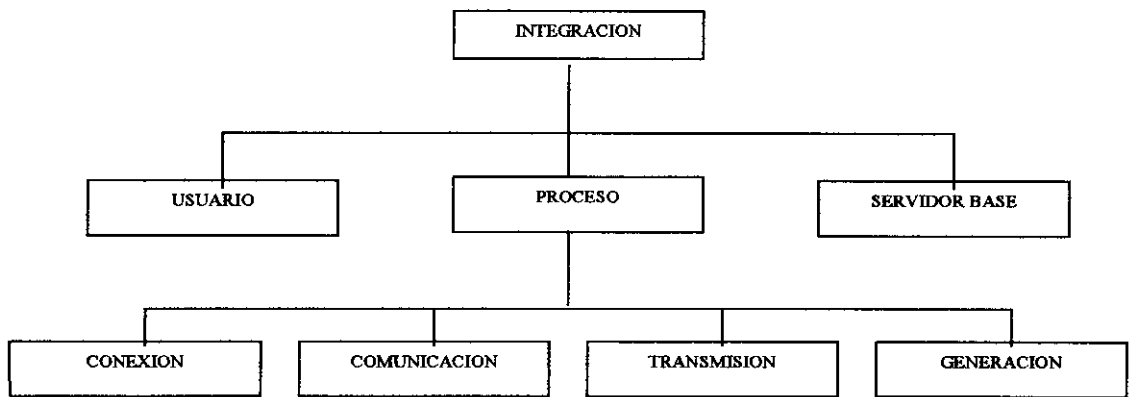


Figura 5.12. Colaboraciones entre las clases.

5.10.2.7. Refinamiento de Jerarquías.





5.10.2.8. Diagrama de Colaboraciones entre Clases.

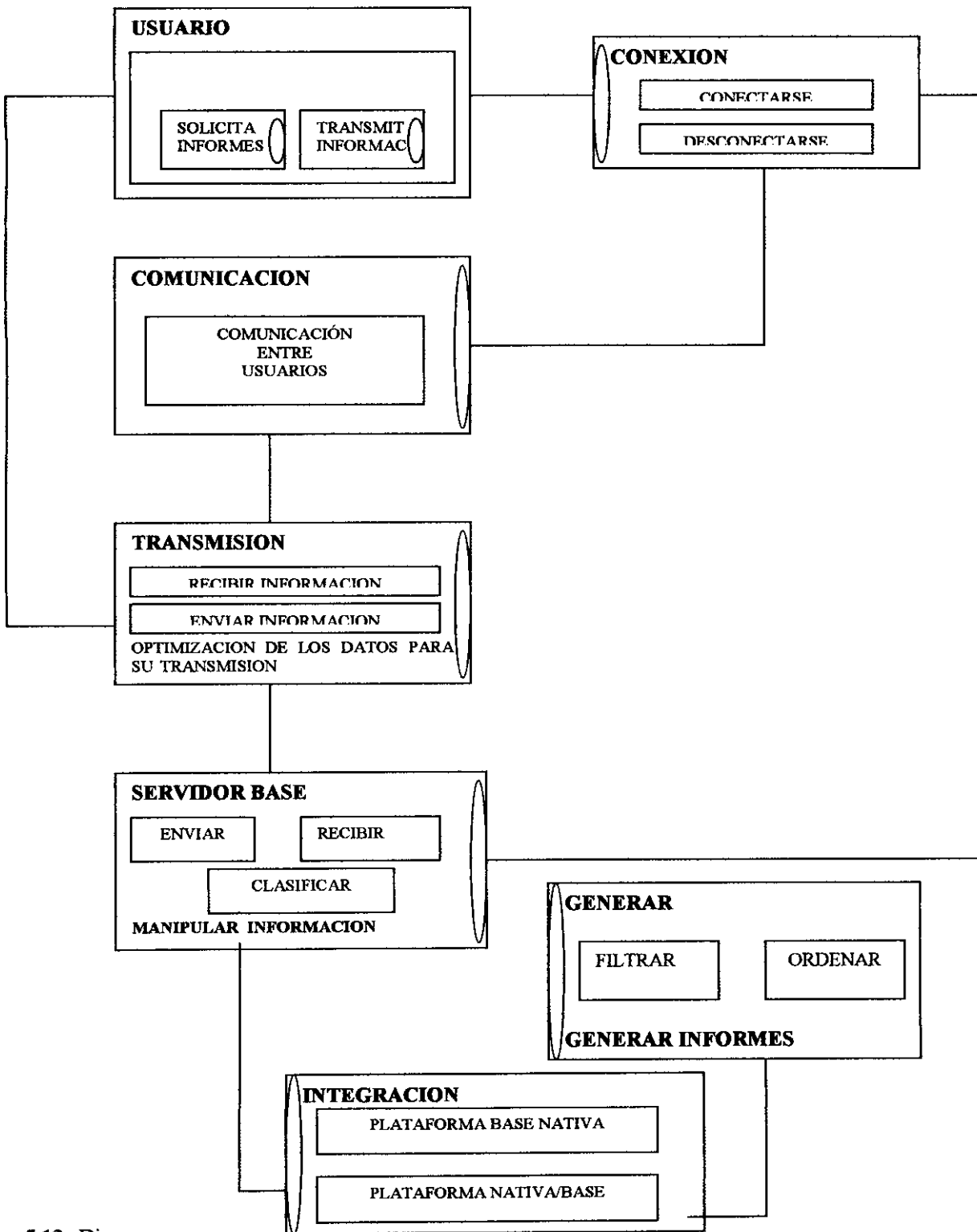


Figura 5.13. Diagrama.

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

- Manipulación de la información de los usuarios.
- Enviar y recibir la información solicitada.
- Ordenación de la información para ser enviada.

Atributos.

- 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.

Clase : **Comunicación**

Luego de establecer conexión se procede a establecer una comunicación y verificación de conexión entre usuarios.

Responsabilidades.

- Comunicar los datos en forma precisa y sin errores.

Atributos.

- Velocidad con la que se va a comunicar, esto se da en baudios por segundo
- Conexión RAS.

Clase : **Transmisión**

Una vez que se ha realizado la comunicación se procede a la transmisión de los datos.

Responsabilidades.

- Transmitir la información requerida.

Atributos.

- Medio de transmisión.

- Método a transmitir.
- Modo de transmitir.
- Tipo de equipo de transmisión.

Clase : **Integración.**

Se refiere al acoplamiento de las bases de datos de las plataformas nativas a la plataforma base y viceversa.

Responsabilidades.

- La integración de una sistema de plataforma nativo a uno de plataforma base e viceversa.

Atributos.

- Tipo de plataforma nativa.
- Tipo de plataforma base.
- Datos de los campos requeridos.
- Registros.
- Tipos de filtros de datos a utilizar.
- Método de ordenación de datos.

Clase : Generar.

Se refiere a la generación de reportes que el usuario a solicitado.

Responsabilidades.

- La generación de reportes.

Atributos.

- Tipo de filtros.
- Campos.

5.10.2.10. Definición de Subsistemas.

SUBSISTEMA : ENLAZAR

CONTRATOS	CLASES A LAS QUE DELEGA
Realiza la conexión entre los dispositivos	Conexión
Comunicación entre usuarios	Comunicación
Optimización de los datos para su transmisión	Transmisión

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

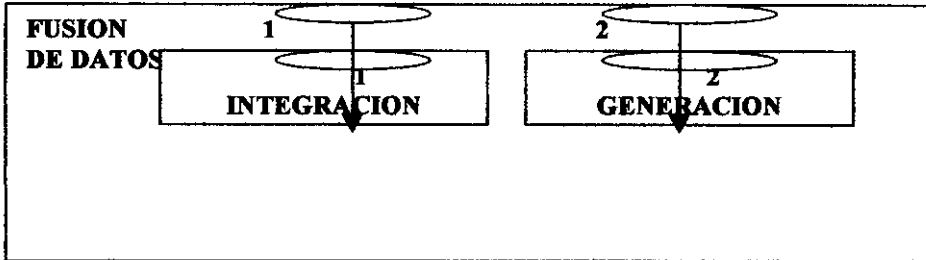


Figura 5.14. Subsistema de fusión de datos

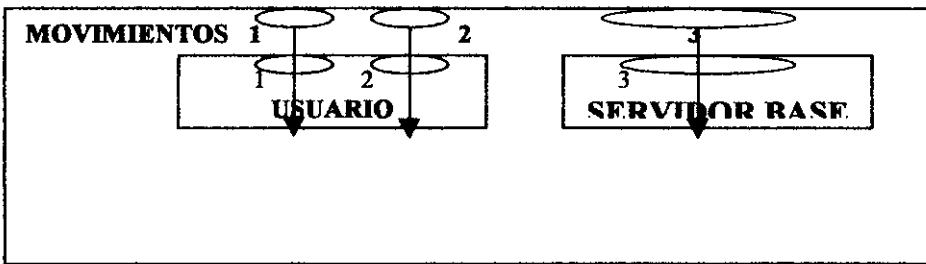


Figura 5.15. Subsistema de movimientos.

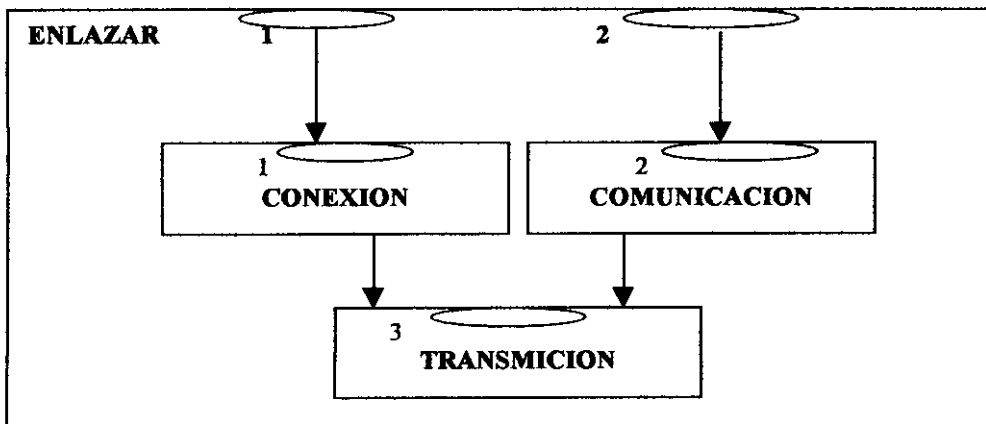


Figura 5.16. Subsistemas de enlazar.

5.11. Descripción de los campos de la base de datos.

La definición de los campos para nuestro software piloto se lo ha realizado en base a una encuesta en cada cooperativa. Adicionalmente a esto se ha creído conveniente crear nuestros propios campos a fin de que el mismo sea parametrizado.

Las bases que se han creado en este proyecto son:

La base de datos llamada DBNATIVA la cual está manejada únicamente por el servidor base la misma que contiene una tabla llamada TBLDATOS siendo sus campos los siguientes:

Nombre del campo	Tipo de datos	Descripción
cedula	Texto	cedula de identidad o ruc del socio
titular	Texto	nombre del socio
cooperativa	Texto	nombre de la cooperativa a la que pertenece el socio

Figura 5.17. Tabla TBLDATOS de la base DBNATIVA

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
server	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
fehact	Fecha/Hora	fecha de ultima actualizacion 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:

- 1.- 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).
- 2.- Luego de establecer la conexión RAS (Servicio de Acceso Remoto) se puede realizar el enlace de los datos fuentes y pasarlos a la base nativa (Servidor) mediante ODBC y lenguaje de programación SQL.

Para realizar esto se toma en cuenta la fecha de ingreso del socio, que es la condición para el filtro de los datos, adicionalmente a esto se ha incrementado un campo status el que nos permitirá descartar los socios que han sido liquidados de la cooperativa.

Al término del enlace se actualiza un campo en el que se almacena la fecha actual con la finalidad de que no se duplique la información y de esta manera únicamente añadir los nuevos socios desde la última fecha de actualización.

3.- Las consultas se las realizan por medio de la cédula, nombre del titular o cooperativa, para esto utilizamos SQL y DBGRID.

4.- Se ha generado varios reportes los mismos que se obtendrán por impresora.

5.13. Características de la aplicación

La pantalla principal cuenta con cinco controles, esta se puede observar el anexo 2, su descripción es la siguiente:

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

Esta opción permite configurar las terminales que van a ser usados en el proyecto, 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.

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 socio 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.

< **A**ctualizar >

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

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

Esta opción permite retornar 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.

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

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

Listado de Anexos.

- 1. Ley de Cooperativas y Reglamento.**
- 2. Entradas / Salidas del Sistema.**
- 3. Código Fuente del Sistema.**

Ley de Cooperativas y Reglamento.

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.

Consulta de Socios

Consultar Datos por : Cédula Nombres Cooperativa

Dato a Consultar :

cedula	titular	cooperativa
▶ 1010101010101	JUAN PEREZ	LA MERCED
3030303030303	CESAR LOPEZ	LA MERCED
1010101010101	JUAN PEREZ	LA MERCED
3030303030303	CESAR LOPEZ	LA MERCED
1010101010101	JUAN PEREZ	LA MERCED
3030303030303	CESAR LOPEZ	LA MERCED
1010101010101	JUAN PEREZ	LA MERCED
3030303030303	CESAR LOPEZ	LA MERCED
1010101010101	JUAN PEREZ	LA MERCED
3030303030303	CESAR LOPEZ	LA MERCED

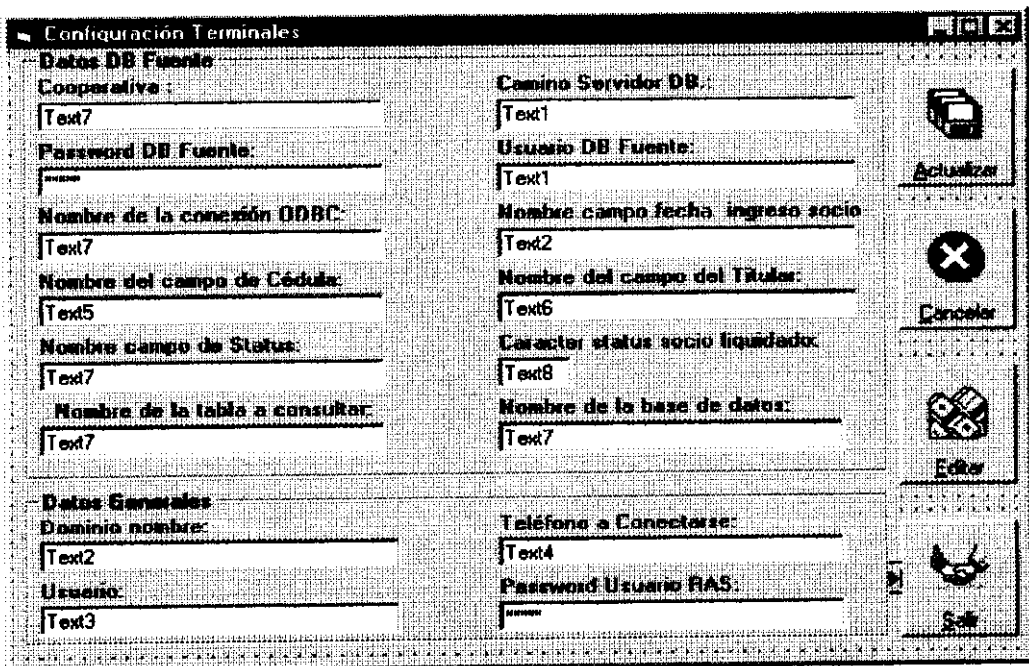
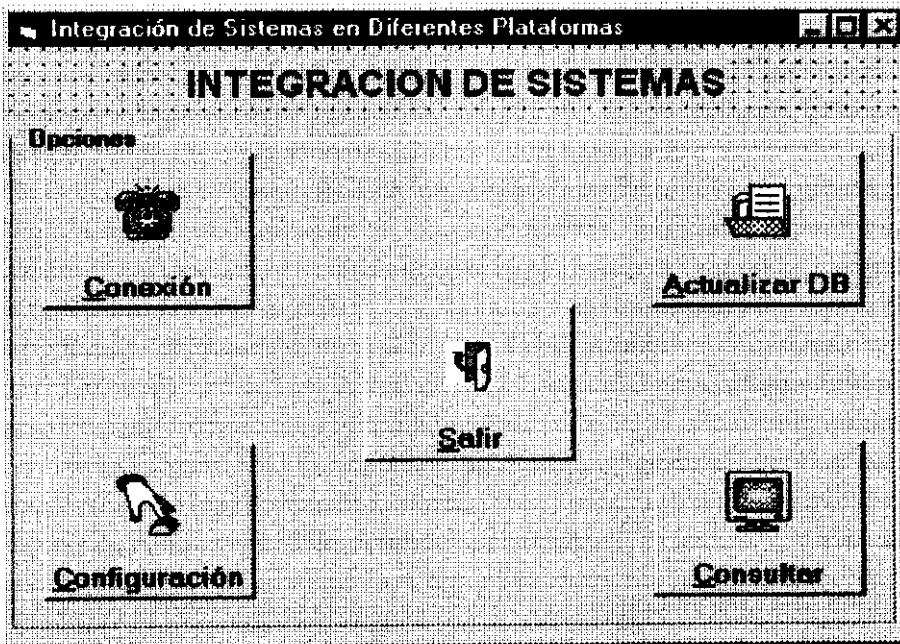
Impresión Consultar Salir

INTEGRACION DE SISTEMAS NATIVOS

Listado General de Socios

Fecha de Reporte : Ambato a, 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



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, odbc1, 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
```

```
odbc1 = rsterminal!nodbc
```

```
ntabla1 = rsterminal!ntabla
```

```
cstatus1 = rsterminal!cstatus
```

```
nstatus1 = rsterminal!nstatus
```

```
camino1 = rsterminal!camino
```

```
ncedula1 = rsterminal!ncedula
```

```
ntitular1 = rsterminal!nombre
```

```
coop1 = rsterminal!cooperativa
```

```
ndatabase1 = rsterminal!ndatabase
```

```
password1 = rsterminal!password
```

```
usuariodb1 = rsterminal!usuariodb
```

```
fehact1 = rsterminal!fehact
```

```
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
' en la cadena de FPD26. 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(camino1 & "dbnativa.mdb")
Set Rsrecord = dbbase.OpenRecordset("tbldatos")
```

```
'sql1 = "SELECT " & ncedula1 & "," & ntitular1 & " INTO [copia_fuente] FROM dbfuente.recordset("" &
nttabla1 & "") where " & nstatus1 & "<>" & "" & UCase(cstatus1) & ""
'dbbase.Execute (sql1)
```

```

'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

```

```

'sql1 = "select " & rsfuente.Fields(aux).Name & "," & rsfuente.Fields(aux1).Name
'sql1 = rsfuente.Name

```

```

'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 lRetVal As Long

```

```

  'RetVal = tapiRequestMakeCall(Trim$("09479416"), App.Title, _

```

```

  ' Trim$("marco"), "")

```

```

  'RetVal=RasDial(0&,vbnulstring, lprasdialparams As Any, ByVal dwNotifierType As Long, lpvNotifier As
  Long, lphRasConn As Long) As Long

```

```

  If lRetVal <> 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 conexión1..."
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 FRMCONFIG.

Option Explicit
Dim i As Integer

```
Private Sub Cmdcancelar_Click()  
    Data1.Recordset.CancelUpdate  
    Data1.Refresh  
    For i = 0 To 14  
        Text1(i).Enabled = False  
    Next i  
    cmdactualizar.Enabled = False  
    cmdcancelar.Enabled = False  
End Sub
```

```
Private Sub cmdactualizar_Click()  
    Data1.Recordset.Update  
    Data1.Refresh  
    For i = 0 To 15  
        Text1(i).Enabled = False  
    Next i  
    cmdactualizar.Enabled = False  
    cmdcancelar.Enabled = False  
End Sub
```

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

```
Private Sub Cmdeditar_Click()  
    Data1.Recordset.Edit  
    'HABILITO EDITBOXES  
    For i = 0 To 15  
        Text1(i).Enabled = True  
    Next i  
    cmdactualizar.Enabled = True  
    cmdcancelar.Enabled = True  
End Sub
```

```
Private Sub Form_Load()  
    Data1.DatabaseName = App.path & "\dbterminal"  
End Sub
```

```
Private Sub Text1_LostFocus(Index As Integer)  
    For i = 0 To 15  
        If Text1(i).Text = "" Then  
            Text1(i).Text = " "  
        End If  
    Next i  
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)
rptodos.ReportFileName = App.path & "\rptconsulta.rpt"
rptodos.SQLQuery = sql
rptodos.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

```

