



85
2ºj.

Universidad Nacional Autónoma de México

FACULTAD DE INGENIERIA

SISTEMA DE BASE DE DATOS DISTRIBUIDAS
PARA UNA RED DE MICROCOMPUTADORAS

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A :

Fco. Javier Rojas Tlalpachicatl

Director:
M. EN C. EFRAIN PARDO ORTIZ

México, D. F.

1992.

TEJIS CON
FALLA DE ORIGEN



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Indice

Introducción	i
Objetivo General	v

Capítulo I

Introducción redes y bases de datos distribuidas

1.1 Definición de redes	I-1
1.1.1 Ventajas y desventajas de las redes locales	I-4
1.1.2 Aplicaciones	I-6
1.2 Base de datos distribuidas	I-7
1.2.1 Sistemas distribuidos	I-11
1.3 Sistemas comerciales que existen actualmente en el mercado	I-15
1.3.1 Sistema Tandem EMCOMPASS	I-15
1.3.1.1 Distribución de datos	I-16
1.3.1.2 Procesamiento de preguntas	I-17
1.3.1.3 Administración de transacciones distribuidas	I-18
1.3.1.4 Una aplicación de bases de datos distribuidas	I-20
1.3.2 IBM. SNA e ISC	I-23
1.3.2.1 CICS/ISC	I-23
1.3.2.2 Función de transporte	I-25
1.3.2.3 Procesamiento de transacciones asincronas	I-27
1.3.2.4 Procesamiento de transacciones distribuidas	I-28
1.3.3 Otros Sistemas	I-29

Redes locales de micros.

2.1 Elementos de una red	II-1
2.1.1 Estaciones individuales	II-1
2.1.2 Disco duro del servidor (Disk Server)	II-2
2.1.3 Servidor de archivos (File Server)	II-3
2.1.4 Servidor de archivos dedicado y no dedicado	II-4
2.1.5 Servidores de archivos distribuidos	II-4
2.1.6 Servidores de impresión	II-5
2.2 Topología de redes	II-7
2.2.1 Topología de bus o línea	II-7
2.2.2 Topología de ring o anillo	II-9
2.2.3 Topología de star o estrella	II-11
2.3 Medios de transmisión	II-12
2.3.1 Par torcido	II-14
2.3.2 Cable coaxial	II-15
2.3.3 Fibra óptica	II-16
2.3.4 Línea de vista	II-18
2.4 Protocolos de comunicación	II-19
2.4.1 Poleo	II-19
2.4.2 CSMA	II-20
2.4.3 Token Passing	II-20
2.5 Normas para redes locales	II-21
2.5.1 Línea física	II-22
2.5.2 Encadenamiento	II-22
2.5.3 Red	II-23
2.5.4 Transporte	II-24
2.5.5 Sesión	II-25
2.5.6 Presentación	II-25
2.5.7 Aplicación	II-26

Capítulo III

Base de datos distribuidas.

3.1 Ventajas y desventajas de una base de datos centralizada vs. distribuida	III-1
3.1.1 Control centralizado	III-1
3.1.2 Independencia de datos	III-2
3.1.3 Reducción de redundancia	III-3
3.1.4 Estructuras físicas complejas para un acceso eficiente	III-4
3.1.5 Integridad, recuperación y control de concurrencia	III-5
3.1.6 Privacidad y seguridad	III-6
3.1.7 Coordinación entre localidades	III-7
3.2 Estructura de las bases de datos distribuidas	III-7
3.2.1 Arquitectura de una base de datos distribuida	III-8
3.3 Tipos de fragmentación	III-11
3.3.1 Fragmentación horizontal	III-12
3.3.2 Fragmentación vertical	III-13
3.3.3 Fragmentación mixta	III-14
3.4 Transparencia de los datos en la red	III-14
3.4.1 Niveles de transparencia	III-15
3.4.2 Transparencia para aplicaciones sólo de lectura	III-16
3.4.3 Transparencia para aplicaciones de lectura y escritura	III-19
3.5 Autonomía de los datos	III-20
3.6 Recuperación de sistemas distribuidos	III-22
3.7 Control de concurrencia distribuido	III-26
3.7.1 Control de concurrencia basado en horas de entrada (Timestamp)	III-27
3.7.2 Abrazos mortales distribuidos o puntos muertos	III-31
3.7.2.1 Detección de abrazos mortales utilizando control centralizado o jerárquico	III-33

3.7.2.2 Detección de abrazos mortales distribuidos	III-35
3.7.2.3 Prevención de abrazos mortales	III-37
3.8 Distribución de carga	III-38
3.9 Estrategias de acceso	III-40

Capítulo IV

Diseño y especificación de la base de datos distribuida para el sistema

4.1 Diseño y selección del modelo de base de datos	IV-1
4.2 Especificación del sistema	IV-4
4.3 Diseño del esquema global ó conceptual	IV-10
4.3.1 Primera forma normalizada	IV-11
4.3.2 Segunda forma normalizada	IV-15
4.3.3 Tercera forma normalizada	IV-17
4.4 Diseño modelo lógico	IV-22
4.5 Diseño de las bases de datos físicas	IV-23
4.6 Diseño de la fragmentación de los datos	IV-25
4.7 Diseño y asignación de los fragmentos	IV-30

Capítulo V

Desarrollo e implantación del sistema.

5.1 Selección del lenguaje de programación	V-1
5.2 Interfaz con el usuario	V-5
5.3 Selección del sistema operativo de red	V-12

Capítulo VI

Documentación y manual del sistema

6.1 Documentación del sistema	VI-1
6.1.1 Diccionario de datos	VI-2
6.1.1.1 Estructuras del sistema	VI-2
6.1.1.2 Diccionario de datos del sistema	VI-3
6.1.1.3 Entidades	VI-5
6.1.2 Documentación del usuario	VI-7
6.1.2.1 Descripción del sistema	VI-7
6.1.2.2 Requerimientos de Hardware	VI-8
6.1.2.3 Instalación del sistema	VI-10
6.1.2.4 Arranque del sistema	VI-11
6.1.2.5 Módulo de ramos	VI-13
6.1.2.6 Módulo de agentes	VI-32
6.1.2.7 Módulo de claves	VI-37
6.1.2.8 Módulo de estados	VI-41
6.1.2.9 Módulo de utilerías	VI-46
Conclusiones	CC-1

Apéndices

Apéndice A Algebra relacional

A.1 Algebra relacional	A-1
A.1.1 Selección	A-2
A.1.2 Proyección	A-2
A.1.3 Producto cartesiano	A-3
A.1.4 Unión	A-3
A.1.5 Diferencia de conjuntos	A-4
A.2 Operaciones Adicionales	A-4
A.2.1 Intersección de conjuntos	A-4
A.2.2 Producto theta	A-5
A.2.3 Producto natural	A-5
A.2.4 División	A-6

Apéndice B Sistema operativo Netware

B.1 Novell antecedentes	B-1
B.2 Sistema operativo	B-2
B.2.1 Netware para Macintosh	B-2
B.2.2 Netware para VMS	B-3
B.2.3 Netware v2.15	B-4
B.2.4 System Fault Tolerant (SFT v2.15)	B-7
B.2.5 Netware 386 v3.0	B-8
B.3 Topología	B-9
B.4 Tipo de cable	B-9
B.5 Tipo de servidor	B-10
B.6 Tarjetas de red	B-10
B.7 Interconectividad	B-10
B.7.1 Lan vs Lan	B-10
B.7.2 Lan vs Mainframe	B-11
B.7.3 Comunicación X.25	B-11
B.8 Comentarios	B-12

Apéndice C Sistema operativo VAX/VMS Services for MS_DOS

C.1 Digital antecedentes	C-1
C.2 Sistema operativo	C-1
C.2.1 VAX/VMS Services for MS_DOS	C-2
C.3 Topología	C-3
C.4 Tipo de cable	C-4
C.5 Tipo de servidor	C-4
C.6 Tarjetas de red	C-4
C.7 Interconectividad	C-5
C.7.1 Lan vs Lan	C-5
C.7.2 Lan vs Mainframe	C-5
C.7.3 Comunicación X.25	C-6
C.8 Comentarios	C-6

Apéndice D Sistema operativo Hewlett-Packard

D.1 Antecedentes	D-1
D.2 Sistema operativo	D-1
D.3 Topología	D-3
D.4 Tipo de cable	D-3
D.5 Tipo de servidor	D-3
D.6 Tarjetas de red	D-4
D.7 Interconectividad	D-4
D.7.1 Lan vs Lan	D-4
D.7.2 Lan vs Mainframe	D-4
D.8 Comentarios	D-4

Apéndice E Sistema operativo Lan Manager

E.1 Antecedentes 3COM	E-1
E.2 Sistema operativo	E-2
E.2.1 3+Share	E-2
E.2.2 3+Open Lan Manager	E-2
E.3 Topología	E-5
E.4 Tipo de cable	E-5
E.5 Tipo de servidor	E-5
E.6 Tarjetas de red	E-6
E.7 Interconectividad	E-6
E.7.1 Lan vs Lan	E-6
E.7.2 Lan vs Mainframe	E-7
E.8 Comentarios	E-7

Bibliografía	BB-1
--------------------	------

Introducción

La evolución de la computación ha sido tan impresionante estos últimos años, que muchas empresas en su área de informática se han tenido que ir ajustando a los cambios de acuerdo a sus posibilidades o necesidades.

Por ejemplo, muchas organizaciones centralizadas tienen estilos de computación que se complementan a su estructura empresarial. Sin embargo, existen otras que son altamente dinámicas, son descentralizadas y sus estilos varían en gran medida.

Las organizaciones descentralizadas necesitan de conceptos e ideas nuevas. Algunas de éstas ven a las redes de computadoras y bases de datos distribuidas, como la solución a sus necesidades.

Las redes locales se consideran como los sistemas de mayor adaptabilidad, que cualquier otro sistema de comunicación que se haya desarrollado.

A las redes locales se les puede conectar cualquier número de usuarios y cualquier tipo de aplicación, además de mantener una excelente relación costo/beneficio. Pueden llegar a crecer de pequeños sistemas hasta ser muy complejos.

Entre los principales factores que contribuyen a su gran flexibilidad son : El diseño de una red local esta basado en una arquitectura distribuida. Esto es, sus componentes son ensamblados en forma modular, además de ser baratos. Estos son seleccionados de acuerdo al tipo de necesidad que se vaya

requiriendo. Cuando la red comienza a crecer en tamaño y poder, el agregar nuevos aditamentos al sistema resulta ser muy fácil.

Otro factor, es el software estandar de red. Uno de los problemas que existian en las redes locales era básicamente crear una interfaz entre el software de aplicación con el hardware existente. Con la aparición de MS-DOS v3.1 y Netbios el problema se ha superado. Hoy en día la mayoría de las redes locales y software de aplicación utilizan estos estandares.

Como último factor, es la independencia de hardware de las redes locales. Poder integrar cualquier clase de equipo existente en la red local, sin tener la necesidad de desecharlo o bien adquirir aditamentos de diferentes distribuidores.

Las redes locales tienen la gran facilidad de establecer comunicación entre sí, tanto en lugares cercanos como en lugares remotos.

La comunicación de la información puede dificultarse cuando existen grupos que producen datos que estan localizados en diferentes departamentos o localidades del grupo que necesita esa información. El éxito o falla de una empresa puede reflejarse en el flujo de información que existe entre las unidades descentralizadas.

Las bases de datos distribuidas son un concepto que ya es viejo, pero que en esta última década ha tomado fuerza, sobre todo porque existe la necesidad de "guardar" información en localidades cercanas a donde se necesitan (Descentralizar).

Las bases de datos distribuidas habilitan datos, aplicaciones y que el procesamiento se realice en los lugares donde se requirió la información.

Un sistema de base de datos distribuidas no aparece de la noche a la mañana. Requiere planeación, diseño e implementación paso-a-paso.

La idea de realizar esta tesis, es primeramente cumplir con uno de los requisitos que establece la Universidad Nacional Autónoma de México dentro de su Facultad de Ingeniería, para titularse, el segundo es la motivación de investigar dos temas de computación que han tomado gran fuerza en la actualidad; redes de computadoras y bases de datos distribuidas.

La organización de la tesis se compone de 6 capítulos.

El capítulo I, nos brinda una introducción de las redes locales y bases de datos distribuidas. Cuáles son las ventajas y desventajas de una red local, su definición y en qué campos de la industria se aplican. En cuanto a bases de datos distribuidas, cuáles son los sistemas que existen actualmente en el mercado. Se dan ejemplos de ellos y en qué campo se están utilizando.

El capítulo II trata a detalle qué es una red local de microcomputadoras, sus principales aditamentos, topologías existentes, características de los medios de transmisión con los que se puede trabajar, protocolos de comunicación que se utilizan y las normas que existen para redes locales.

En el capítulo III se realiza un estudio de los conceptos que existen en una base de datos distribuida, ventajas con respecto a una base de datos centralizada, la estructura que tiene una base de datos distribuida, los 3 tipos de fragmentación que se utilizan, los niveles de transparencia en la red, porque debe existir autonomía de los datos, cuando un sistema falla cuáles son los mecanismos de recuperación, cómo se resuelve y bajo qué métodos el control de concurrencia y cómo se distribuye la carga de una base de datos distribuida.

El capítulo IV es una de los más importantes, se realiza detalle el análisis y diseño del modelo que se va a utilizar para distribuir la base de datos. Se dan las especificaciones del sistema, se explicó la razón de utilizar el modelo de base de datos relacional, se elabora el diseño del esquema conceptual y en el, se dan a detalle las 3 formas de normalización. Se efectua el diseño del modelo lógico y las bases de datos físicas, para finalmente diseñar la fragmentación de los datos, así como también cómo se asignarán y en que localidades.

El capítulo V explica porque se seleccionó como lenguaje de programación Clipper, la razón de utilizar el sistema operativo de Novell y se discute la interfaz del usuario.

El capítulo VI es la documentación del sistema, las estructuras del sistema, sus entidades y el diccionario de datos del sistema. Se indican cuales son los requerimientos de hardware, cómo se instala el sistema y los pasos que se siguen para ponerlo en marcha, además de explicar a detalle cada uno de los módulos del sistema.

Finalmente se establecen las conclusiones del trabajo. Y además se cuenta con 5 apéndices, donde el primero trata brevemente qué es el álgebra relacional, cuáles son sus operaciones primarias y las que se derivan de estas. Los 4 apéndices restantes son un pequeño análisis de los diversos sistemas operativos de red que existen actualmente en el mercado nacional.

Objetivo general

Se desarrollará un sistema que funcione en una red local de microcomputadoras, que distribuya inteligentemente la base de datos física para optimizar las transacciones de comunicación y automatizar la recuperación de información.

CAPITULO I

Introducción redes y bases de datos distribuidas.

1.1 Definición de redes

Debido al enorme incremento de las nuevas tecnologías en el mundo computacional, surge una que es el desarrollo de las redes locales.

Hoy en día, por ejemplo, existe un dramático y continuo decremento en los costos de hardware, que vienen acompañados por un enorme incremento en las capacidades del mismo.

Los microprocesadores llegan a trabajar a grandes velocidades, poseen un conjunto de instrucciones muy poderoso y tienen una gran capacidad de memoria que se puede llegar a comparar con las computadoras de mediana escala.

Todos estos factores han hecho posible que el número de sistemas se incremente tanto en oficinas, como en fábricas, centros de operación, etc. Además de que las redes locales tienen la facilidad de poder interconectarse.

Entre las principales razones podemos contemplar las siguientes :

- Intercambio de datos entre sistemas.
- Compartir recursos muy caros.
- Proveer respaldos de aplicaciones.

Las redes locales (*Lan's*) son definidas como "Redes de Comunicación, que proporcionan interconexión a una variedad de dispositivos de comunicación y además, comparten recursos dentro de una área limitada."¹

Existen tres elementos de gran significado en su definición.

1.-Una red local es una red de comunicación, no una sola computadora. Es por eso que se requiere de software especializado y protocolos de comunicación para conectar varias computadoras.

2.-Los dispositivos de comunicación son cualquier clase de equipo, que utilice un medio de transmisión para comunicarse. Por ejemplo :

- Computadoras
- Terminales
- Periféricos
- Sensores
- Teléfonos

3.-El ámbito geográfico de una red local es muy pequeño. Es decir las redes locales están destinadas a permanecer en un edificio. Sin embargo, existen redes que enlazan edificios. Por ejemplo, en un campus universitario o en una base militar.

. Algunas características típicas de una red local son :

- Alto promedio de transmisión (0.1 a 100 Mbps).
- Distancias cortas (0.1 a 25 Km.).
- Bajo promedio de error (10⁻⁸ a 10⁻¹¹).

¹ Stallings William Ph.D. Local Area Networks An Introduction.
Ed. Macmillan Publishing Company 1984.

Los dos primeros parámetros sirven para diferenciar a sus dos primos, que son : Sistemas de Multiproceso y Redes de Gran Trayecto.

Otras diferencias que existen entre las redes locales y sus dos primos, es el gran impacto que tienen en cuanto a su diseño y operación. Las redes locales generalmente tienen pocos errores en la transmisión de datos. Además de que los costos de comunicación son bajos a comparación de las redes de gran trayecto.

Existe una distinción entre las Redes Locales y los Sistemas Multiproceso, el cual es el grado de acoplamiento o conexión. Los Sistemas Multiproceso son estrechamente difíciles de acoplar, mientras que las Redes Locales tienden a tener características opuestas (Fig. 1.0).

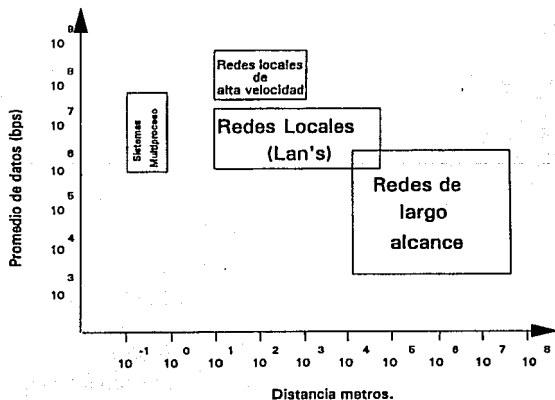


Figura 1.0. Configuración entre un sistema multiproceso, redes locales y redes de largo alcance.

1.1.1 Ventajas y desventajas de las redes locales

En la tabla 1.0 se muestran las principales ventajas y desventajas de una red local de microcomputadoras.

VENTAJAS	DESVENTAJAS
<p>Mejora Respuesta/Rendimiento</p> <p>Confiabilidad</p> <p>Disponibilidad</p> <p>Supervivencia</p> <p>Compartir recursos</p> <p>Soporte varios vendedores</p> <p>Flexibilidad</p> <p>Integración de procesamiento de datos</p> <p>Automatización de oficinas</p>	<p>Interoperatividad no garantizada (Software y datos)</p> <p>Es más difícil administrar la red y establecer estandares.</p> <p>Las bases de datos distribuidas Incrementan los problemas de :</p> <p>Integridad</p> <p>Seguridad</p> <p>Privacidad</p>

Tabla 1.0. Ventajas y desventajas de una red local de microcomputadoras.

Tal vez la ventaja más grande, es la que se relaciona a la evolución de los sistemas. En una instalación donde no existe una red, como por ejemplo en un centro de cómputo, todo el procesamiento de los datos se realiza en uno o pocos sistemas y en donde cualquier cambio tiende a ser traumático.

Una red local proporciona confiabilidad, disponibilidad y supervivencia del procesamiento de datos con sistemas múltiples interconectados. La pérdida de un sistema por lo consiguiente, tiene un impacto mínimo. Se ha mencionado que una red puede compartir recursos, ésto no sólo incluye periféricos, sino también datos. Los datos pueden ser mantenidos en un solo lugar y controlados con una facilidad específica. Sin embargo, via red son disponibles a muchos usuarios.

Dentro de una red local de microcomputadoras existen también algunos imponderables o desventajas, por ejemplo; la red no garantiza que dos dispositivos puedan ser usados cooperativamente. Un concepto conocido como *interoperatividad*. Por ejemplo; Dos procesadores de texto de diferentes vendedores pueden ser instalados en la red y tal vez pueden intercambiar datos. Pero probablemente usen diferente formateo y diferentes caracteres de control. Por lo tanto, no es posible tomar un archivo de un procesador y editarlo en el otro procesador directamente. Antes de efectuar tal paso, es necesario que el archivo sea convertido al formato adecuado por medio del software correcto.

En una red los datos pueden ser distribuidos, con lo cual se puede llegar a comprometer la integridad de la información (dos usuarios tratan de acceder una base de datos al mismo tiempo), seguridad y privacidad.

1.2.1 Aplicaciones

El rango de aplicaciones para una red local es muy grande. Se debe entender, que no todas las redes son capaces de soportar aplicaciones que se encuentren disponibles en el mercado.

Entre las aplicaciones tenemos :

▶ Procesamiento de datos :

- Acceso a datos
- Procesamiento de transacciones
- Transferencia de archivos
- Procesos batch

▶ Automatización de oficinas :

- Documentos / Procesadores de palabra
- Correo electrónico
- Facsímile

▶ Automatización de fábricas :

- CAD / CAM
- Control de inventarios / órdenes de entrada / compras

▶ Administración de energía :

- Calor
- Ventilación
- Aire acondicionado

▶ Control de proceso.

▶ Teleconferencias

▶ Seguridad :

- Sensores / alarmas
- Cámaras y monitoreo

▶ Televisión :

- Presentaciones de video

1.2 Base de datos distribuida

La tecnología de las bases de datos distribuidas constituye un avance relativamente reciente dentro del campo general de las bases de datos. "Una base de datos distribuida es por lo común, una base de datos no almacenada en su totalidad en un solo lugar físico, sino que se distribuye a lo largo de una red de computadoras geográficamente separadas y que se conectan por medio de enlaces de comunicación."²

Esta definición acentúa dos aspectos muy importantes para las bases de datos distribuidas.

- ▶ Distribución : Los datos no residen en el mismo lugar (Procesador). Esto es básicamente lo que distingue a una base de datos distribuida de una base de datos centralizada.
- ▶ Correlación lógica : Los datos tienen algunas propiedades que los hacen permanecer juntos, ésto es lo que distingue a una base de datos distribuida de un conjunto de bases de datos locales o archivos que residen en diferentes lugares de una red de computadoras.

² Date C.J. Introducción a los sistemas de base de datos. Editorial SITESA Addison_Wesley Iberoamericana 1986.

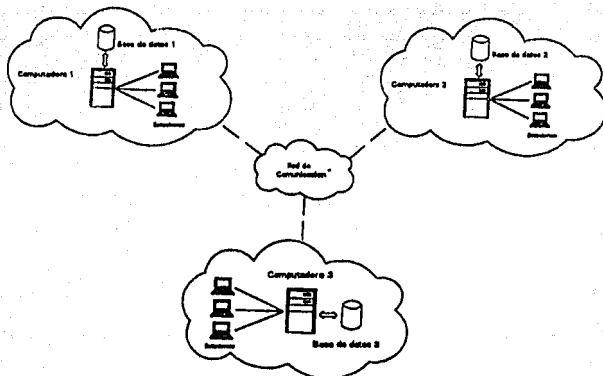


Figura 1.1. Base de datos distribuida en una red de computadoras.

A manera de ejemplo podríamos considerar el siguiente caso. Un sistema bancario con tres sucursales (Figura 1.1) en diferentes localidades, donde la base de datos mantiene las cuentas de los clientes y se distribuye en las sucursales del banco, además de que el registro de la cuenta de cada cliente se almacena en la sucursal más cercana a su domicilio o lugar de trabajo de este. En otras palabras, los datos se almacenan en el sitio donde se usan con mayor frecuencia, pero también están disponibles a través de la red de comunicación. Si se observa podremos notar las ventajas de tal distribución, combina la eficiencia del procesamiento local (sin excesivos costos de comunicación) de la mayoría de las operaciones, además de que permite compartir datos como si fuera un sistema centralizado.

Desde luego existen también desventajas, los costos de comunicación pueden llegar a ser altos y existir muchas dificultades técnicas para lograr un sistema de este tipo.

Desde el punto de vista tecnológico, es muy importante la existencia de algunas aplicaciones que accesen los datos en más de una estación. Estas aplicaciones son llamadas *Aplicaciones Globales* o *Aplicaciones Distribuidas*. La existencia de Aplicaciones Globales pueden ser consideradas como una ventaja de las bases de datos distribuidas con respecto a un conjunto de bases de datos locales.

En una Aplicación Global como podría ser una transferencia de fondos de la cuenta de una sucursal a la cuenta de otra sucursal, requiere actualizar las bases de datos de dos sucursales. Se debe notar que esta aplicación es algo más que ejecutar dos actualizaciones locales en dos sucursales distintas. Se debe tener la seguridad de que ambas actualizaciones sean realizadas o no. El asegurar los requerimientos para una aplicación global es una tarea difícil.

En el ejemplo anterior las computadoras se encuentran en diferentes localidades geográficamente. Sin embargo, una base de datos distribuida puede ser construida en una red local de microcomputadoras.

En la figura 1.2 se muestra el ejemplo con la misma aplicación pero con una configuración diferente.

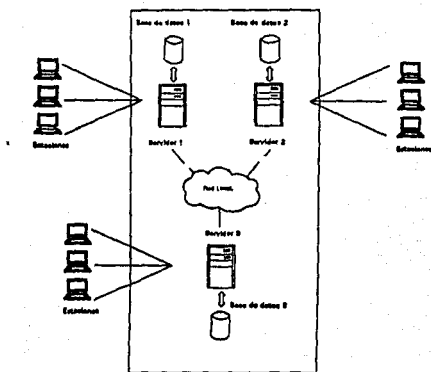


Figura 1.2. Base de datos distribuida en una red local de micros.

En esta aplicación el mismo proceso con las bases de datos de cada sucursal han sido implementadas en un edificio común, pero ahora se conectan a través de una red local de largo alcance. Las estaciones de trabajo o terminales de cada sucursal son conectadas a su respectiva computadora (servidor) por medio de líneas telefónicas. Cada proceso y su base de datos constituye una red local.

La estructura física de las conexiones ha cambiado con respecto al ejemplo 1. Sin embargo, las características de sus arquitecturas tienen algo en común. Es decir, las mismas computadoras ejecutan la misma aplicación que accesa a las mismas bases de datos. Una aplicación que fué local en el ejemplo 1, es local en el ejemplo 2. La localidad no se define con respecto a la distribución geográfica de las computadoras, sino con respecto a la computadora que tiene sus propias bases de datos.

Cabe mencionar, que una base de datos distribuida que se implementa en una red local es altamente eficiente en cuanto a su procesamiento, además de ser muy confiable.

1.2.1 Sistemas distribuidos

El procesamiento distribuido es una frase que describe a sistemas de información con una gran diversidad de características. Existen diferentes definiciones de sistema distribuido o sistema de procesamiento distribuido que son ofrecidas por personas con diferentes motivaciones y puntos de vista. Sin embargo, aún no existe una definición concreta en cuanto a sistema distribuido. Esta frase se aplica a sistemas con diferentes estructuras, medios ambientes de software, características de interconexión, grados de cooperación y dispersión de datos.

Existen otros términos, como son: distribución de datos, distribución de aplicaciones, distribución de funciones y procesos cooperativos. A falta de una definición concreta, podremos auxiliarnos de la siguiente :

› "Un sistema distribuido posee varios procesadores autónomos (computadoras) con sus propios sistemas operativos donde los datos almacenados pueden procesarse, dada la gran interacción que existe entre las bases de datos para lograr un objetivo en conjunto. Los procesadores coordinan sus actividades e intercambian información sobre la red de comunicaciones."³

³ Sloman Morris and Kramer Jeff. Distributed Systems and Computer Networks. Ed. Prentice-Hall 1987

Las razones básicas para desarrollar un sistema distribuido según Wiederhold son las siguientes :

• Funcionales .- Un sistema realiza diferentes funciones y es conveniente desarrollar y manejar estas funciones por separado. Si la actividad dentro de una función es mayor a la interacción entre funciones, será más manejable un sistema en el que las funciones se distribuyan a procesadores especializados.

• Geográficas .- Si existen usuarios en diferentes lugares, llega a ser más eficiente el procesamiento local.

• Desempeño .- La recuperación en nodos locales evita retrasos de comunicación. Los nodos altamente especializados pueden proporcionar alto desempeño para ciertas funciones, sin tener que llegar a alterar otros aspectos de un sistema de datos. Los procesadores de respaldo terminal de base de datos o los sistemas de comunicación, son ejemplos de estos nodos especializados.

• Autonomía .- Cuando en una organización se han delegado responsabilidades para diferentes áreas, implica un mayor control sobre la información correspondiente. Esto se logra por medio de la distribución de datos de manera que los privilegios de acceso, las responsabilidades de control de calidad, la confiabilidad de las comunicaciones y la conveniencia de acceso, correspondan al patrón de responsabilidad.

• Confiabilidad .- Si se aísla parte del sistema, gran parte del mismo puede seguir funcionando mientras un procesador o sus dispositivos de almacenamiento están descompuestos.

• Crecimiento .- Si aumentan las demandas en el sistema, agregar un nuevo nodo puede proporcionar capacidad adicional sin tener que reemplazar todo el sistema anterior.

- Redes - Para que los múltiples procesadores se integren en un sistema, es necesario que exista una red de comunicación entre los nodos o diferentes localidades.

- Distribución - Es muy importante distribuir el procesamiento de tal forma, que el procesamiento local reduzca las demandas sobre la red de comunicación. "4

El objetivo básico de un sistema distribuido es, que el usuario lo perciba como un sistema centralizado. Es decir, el usuario por lo general no necesita saber en donde se encuentra físicamente almacenada alguna parte de los datos (de tal modo que las aplicaciones sean independientes de la manera en cómo se distribuyen los datos, lo que permite cambiar la distribución sin afectar esas aplicaciones). Así pues, el hecho de que la base de datos esté distribuida, deber ser importante sólo a nivel interno y no a los niveles externo y conceptual.

Como se dijo anteriormente, un proceso distribuido ocurre, cuando la aplicación se procesa en forma descentralizada en contraste con los procesos compartidos o multiproceso, los cuales requieren que todo el procesamiento se efectúe desde una central y en una máquina. El procesar aplicaciones por ejemplo; en estaciones de trabajo tipo PC y en una red de microcomputadoras, es un claro ejemplo de proceso distribuido. Cada microcomputadora tipo PC tiene una copia de determinado software y el sistema operativo de la red local sincroniza el uso de los recursos que se comparten por las múltiples aplicaciones. La figura 1.3 ilustra este medio ambiente.

⁴ Wiederhold Gjo. Diseño de Bases de Datos. Ed. McGraw-Hill 1985.

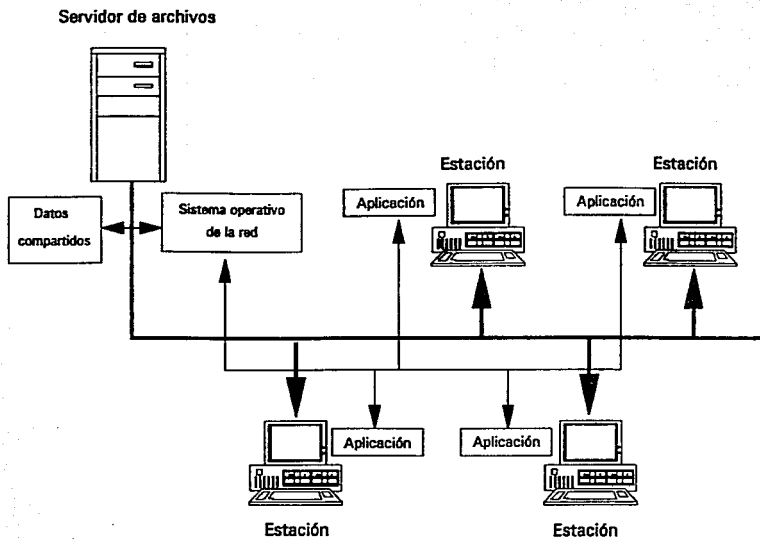


Figura 1.3. Medio ambiente de procesamiento distribuido en una red de micros.

1.3 Sistemas comerciales que existen actualmente en el mercado

El campo de las bases de datos distribuidas está experimentando una rápida evolución, no sólo en los niveles de investigación y prototipos, sino también a nivel de productos industriales y aplicaciones de la vida real. Existe todavía confusión en el mundo industrial sobre las características que deberá poseer un sistema que implemente bases de datos distribuidas.

En esta sección se describirán dos de los principales sistemas que existen; el sistema *Tandem ENCOMPASS* y el sistema *IBM SNA/ISC*. Además, se hará mención a otros sistemas comerciales.

1.3.1 Sistema Tandem ENCOMPASS

Las razones por las cuales se seleccionó éste sistema fuerón;

- 1.-*ENCOMPASS* : Es un sistema muy avanzado.
- 2.-La arquitectura de la computadora *Tandem* : es muy entera, además de su sistema operativo, red de comunicación y manejador de bases de datos que son altamente integrables y homogéneos, hacen que éste sistema sea fácil de entender.

La característica más relevante de la computadora *Tandem* es que está compuesta de varios (al menos dos) CPU's independientes, los cuales se conectan por medio de un bus y comparten el acceso a discos comunes. Una característica que distingue a ésta computadora con respecto al multiprocesamiento clásico, es que cada CPU tiene una memoria independiente y los CPU's son muy similares a computadoras independientes que se comunican .

En una computadora *TANDEM* no solo los CPU's son réplicas, sino también otros componentes como los discos y los canales (buses). EL objetivo principal de estas réplicas, es proporcionar una alta disponibilidad del sistema en caso de que algún módulo falle.

Como la arquitectura de la computadora *TANDEM* es distribuida, el sistema operativo (*GUARDIAN*) otorga facilidad de comunicación entre los procesos que se ejecutan en diferentes CPU's. Todas las comunicaciones entre los procesos es vía mensajes.

Varias computadoras pueden ser conectadas para integrar una red local. Cabe mencionar que el envío de mensajes es el mismo, aún cuando los procesos residan en computadoras diferentes. Es muy importante entender que la comunicación básica entre dos procesos es la misma. Si estos residen en el mismo CPU, en dos CPU's diferentes del mismo sistema, en dos lugares diferentes de la red o en dos diferentes lugares de la red geográfica. El rendimiento de la comunicación es muy diferente para cada uno de los casos mencionados anteriormente.

Esta arquitectura hace posible el manejo de distribución de datos, manejadores de bases de datos distribuidas, procesamiento de preguntas y administración de transacciones distribuidas.

1.3.1.1 Distribución de datos

El sistema *ENCOMPASS* soporta transparencia en su fragmentación, ya que es posible realizarla de manera horizontal. Una relación global puede ser particionada horizontalmente en un número limitado de fragmentos (la fragmentación se tratará más adelante en el capítulo III).

Una relación global puede ser fragmentada sólo por valores de la llave primaria. Por ésta razón el atributo de la fragmentación (o al menos parte de esta) puede ser incorporada en una llave primaria.

ENCOMPASS no soporta fragmentación vertical y duplicación de datos. Un usuario por ejemplo, tiene que actualizar diferentes copias del mismo archivo porque a éstas, el sistema las considera diferentes. El administrador de transacciones que se describe más adelante facilita esta tarea.

La definición de la fragmentación de datos y localización de los mismos no se encuentra en el diccionario de datos o catálogo principal del sistema. Una utilería aparte se utiliza para manejar estos aspectos.

1.3.1.2 Procesamiento de preguntas

Tandem proporciona un lenguaje relacional llamado *ENFORM* el cual tiene una interfaz de registro a la vez. *ENFORM* no conoce la distribución de los datos.

Un usuario que conoce la localización de los datos puede influir en dónde se debe hacer el procesamiento de la siguiente manera : *ENFORM* está compuesto de dos partes. Un *FrontEnd* (procesador que permite la comunicación de varios equipos) que analiza las preguntas y formatea los reportes y un *BackEnd* (procesador que no solo se dedica a proceso en línea sino también en batch) que hace los accesos a la base de datos (selecciones y proyecciones). El usuario puede localizarse en el *BackEnd*, donde muchos de los datos que se requieren residen, independientemente de donde se encuentre el *FrontEnd*.

1.3.1.3 Administración de transacciones distribuidas

En un sistema *Tandem* la estructura computacional de una transacción está basada en un enfoque *demanda-servidor* (*request-server*). La figura 1.4 muestra un ejemplo de ésta estructura; dos procesos de control de demandas, varias terminales y un servidor de procesos que ejecuta los accesos a la base de datos. El proceso de demanda y servidor se comunican vía mensajes y pueden residir en diferentes lugares de la red.

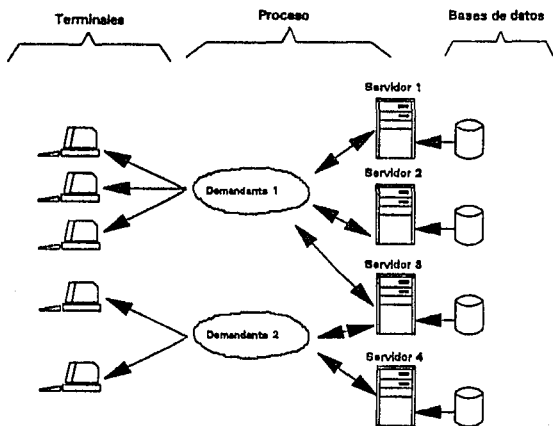


Figura 1.4. Demandantes y servidores en un sistema Tandem.

El proceso de demanda ejecuta las siguientes funciones : interfaz de las terminales, validación de campos, mapeo de datos y transacciones de control. La interfaz de terminal se encarga de los protocolos, los cuales son usados por varias terminales. La

validación de campos es usada para el control de los datos de entrada (chequeo de los campos, existencia de los campos requeridos, etc.). La función de mapeo de datos efectúa la conversión de datos de externos a internos y viceversa. Las transacciones de control incluyen un alto nivel de aplicación lógica : Interpretación de entrada, demanda de ejecución de un proceso en el servidor para la manipulación de la base de datos y la interacción con la terminal.

Los siguientes aspectos caracterizan la estructura *demanda-servidor* del sistema *Tandem*.

- Un proceso de demanda deberá ser multilectura (capaz de manejar más de una demanda a la vez).
- Un servidor es siempre un solo hilo. Es decir, ejecuta una función completamente antes de iniciar una nueva.
- Un servidor de proceso deberá ser de contexto libre. No deberá solicitar nada de las demandas anteriores. Toda la información que se necesite en ese momento deberá ser recibida.
- El proceso de *demanda-servidor* puede residir en diferentes lugares de la red.
- El mismo servidor de proceso puede ejecutar funciones en nombre de diferentes procesos de demanda.
- La comunicación entre un servidor y las demandas es a través de mensajes.
- Cabe mencionar, que para reducir los costos de comunicación se deberá buscar un lugar adecuado para el servidor, el proceso de demandas con las terminales y el servidor con las bases de datos.

1.3.1.4 Una aplicación de bases de datos distribuidas

ENCOMPASS ha sido usado para la construcción del sistema de control de información de manufactura llamado *ENPACT*, el cual es usado por la división de manufactura de *Tandem*. *ENPACT* implementa una base de datos distribuida.

Los datos de la industria, son almacenados en diferentes partes de la red. Actualmente los datos se almacenan en ocho lugares diferentes :

- Austin, Texas
- Chigaco, Illinois
- Cupertino, California (2 lugares)
- Neufahrm, Alemania
- Reston, Virginia
- SunnyVale, California
- Watsonville, California

Una nueva planta está planeada para los siguientes nueve meses. *ENPACT* además de implementar un distribución geográfica, implementa una base de datos distribuida muy dispersa.

Las funciones que son controladas por *ENPACT* son típicas del control de manufactura que incluye partes maestras, factura de materiales, control de inventarios, compras/recepción, programas, planeación de material que se necesita y transferencia de material entre plantas.

Los datos que se usan en esta aplicación se dividen en dos categorías; Datos Globales y Datos Locales. Los Datos Globales son comunes a todos los lugares, por ejemplo; un archivo maestro que contiene las partes del catálogo de *Tandem* y la factura del archivo de materiales. En general los datos que describen la estructura de cada producto son datos globales. Los datos locales

sólo son relevantes al lugar donde se almacenan. Ejemplos típicos de datos locales son; el inventario, programas, compras / recepción.

Las bases de datos han sido distribuidas de la siguiente manera. Cada archivo global, es una réplica completa en cada lugar, cada archivo de datos de almacenamiento local, es fragmentado horizontalmente, ésto con el fin de que almacene la parte más relevante de él. Por ejemplo la información del inventario de Texas sólo es almacenado en Austin, por lo tanto los archivos locales de cada lugar tienen la misma estructura aunque ellos almacenan datos diferentes.

Esta distribución fué realizada para satisfacer dos objetivos básicamente; disponibilidad continua y autonomía de cada lugar. Un usuario de cualquier lugar tiene los datos disponibles para informarse, puesto que existe una copia del archivo global y una porción relevante de un archivo local. El principal problema de estos dos objetivos, ha sido la actualización de las transacciones de los datos globales. Una solución para resolver este problema, ha sido propuesta por los diseñadores de *ENPACT* y consiste en actualizar las demandas de los servidores donde una copia del archivo global existe. Sin embargo, ésta solución tiene tres inconvenientes.

- Los servidores que ejecuten las actualizaciones en un archivo global tienen que conocer donde está cada réplica del archivo, esto provoca que la transparencia de cada lugar se pierda.
- El tiempo de respuesta de una actualización global es muy grande.
- Deberá existir disponibilidad de cada lugar de la red para llevar a cabo una actualización global; ésto puede causar que la autonomía de cada lugar se pierda.

El primer problema ha sido resuelto usando "archivos duplicados" los cuales describen la localidad de cada réplica y de cada archivo global. De esta manera es posible incorporar información distribuida en los programas de aplicación.

La solución para los otros dos problemas ha sido aceptando temporalmente datos inconsistentes de cada archivo global. Una actualización se hace inmediatamente a la copia local, independientemente de la disponibilidad de los otros lugares, de esta manera se obtiene un tiempo de respuesta corto y autonomía del lugar. La actualización es grabada en un "archivo de suspenso" y un proceso asíncrono, ejecuta las actualizaciones remotas tan pronto como sea posible por la demanda de la activación de los servidores remotos correspondientes.

Existe todavía un problema más grave y es la actualización concurrente que puede suceder en dos lugares diferentes provocando así dos copias de datos inconsistentes. La actualización de las transacciones de los datos deberá leer en éste caso datos incorrectos y podrá producir actualizaciones erróneas. Para prevenir este problema, las siguientes restricciones en las transacciones de actualización han sido establecidas: cada registro global puede ser "propietario" de un solo lugar y sólo el dueño del lugar puede iniciar la actualización del registro. Hay que darse cuenta que ésta solución sólo trabaja para actualizar transacciones y para transacciones de sólo-lectura pueden llegar a existir datos inconsistentes.

Una área donde se han encontrado muchas dificultades, es el control de operación. Operaciones como agregar un nuevo lugar al sistema, inactividad de transacciones son muy difíciles y herramientas especializadas se necesitan para este propósito.

1.3.2 IBM, SNA e ISC

La gran variedad de productos de software de las computadoras IBM, incluyen sistemas de administración de bases de datos y manejadores de transacciones, los cuales constituyen un medio ambiente más complejo que el que ofrece una computadora Tandem con su sistema de bases de datos ENCOMPASS. IBM ha desarrollado y seguirá desarrollando herramientas para la construcción de sistemas distribuidos de bases de datos. Lo más importante de estas herramientas es SNA (System Network Architecture) e ISC (Inter System Communication).

SNA incluye software para la construcción de redes IBM y terminales que tengan características específicas. En este caso aquellas que incluyen bases de datos distribuidas y que soportan sesiones lógicas en procesos de diferentes lugares y protocolos de comunicación entre programas. No se discutirán las características de SNA porque SNA define esencialmente la arquitectura de redes de comunicación y no procesamiento de transacciones distribuidas y bases de datos distribuidas.

ISC es un conjunto de protocolos que permiten comunicar varios sistemas. Los protocolos ISC han sido implementados básicamente para manejar transacciones de CICS y manejar sistemas de bases de datos IMS.

1.3.2.1 CICS / ISC

CICS es llamado monitor-Tp o manejador de transacciones. Su principal función, es controlar la ejecución de aplicaciones en línea, las cuales son solicitadas por usuarios finales en las terminales. Cuando un usuario solicita un código de aplicación CICS activa el programa de aplicación correspondiente. CICS, ejecuta además funciones parecidas a las de un sistema operativo.

Las razones básicas para usar *CICS* son :

- *CICS* permite definir la configuración de una terminal. Los códigos de la aplicación pueden ser solicitados de las terminales para los correspondientes programas de aplicación.
- *CICS* es capaz de controlar la ejecución de transacciones más eficiente que un sistema operativo.

La descripción anterior nos muestra que *CICS* no es un manejador de bases de datos. Su función es simplemente ejecutar programas de aplicación, que el usuario solicita desde su terminal. Sin embargo, los programas de aplicación que controla *CICS*, son en general programas que accesan bases de datos. Como por ejemplo, *IMS*. Por esta razón existe una fuerte integración entre el manejador de transacciones y el sistema de bases de datos. Es posible que *CICS/IMS* transmita a *IMS* una primitiva desde X lugar hacia el lugar de origen, en este caso se asume que el originador posee un interprete de la base de datos, ya que *CICS* no interpreta directamente una primitiva.

CICS es una administrador de aplicaciones el cual accesa bases de datos bajo el control de diferentes manejadores de bases de datos. Por lo cual *CICS/ISC* puede ser usado para construir una base de datos distribuida heterogénea. Generalmente, los dos manejadores de bases de datos de *IBM*, que pueden interactuar con *CICS* son: *IMS* y *SQL/DS*.

CICS puede también llamar a otros productos que no son de *IBM* como, *Cullinane's IDMS* y *DataCom/DB* de *Applied Data Reseach*. Sin embargo, construir una base de datos distribuida heterogénea con productos que no son de *IBM* es más difícil.

IMS es un manejador de bases de datos que accesa un registro a la vez, tiene además componentes que ejecutan manejo de transacciones.

SQL/DS es un manejador de bases de datos relacional, que usa lenguaje *SQL (Structure Query Language)*. *SQL/DS* puede ser usado no sólo por el usuario final como un lenguaje de preguntas o consultas sino también por programadores para escribir programas en línea.

Una transacción distribuida puede ser implementada usando *CICS/ISC* de tres maneras diferentes las cuales son llamadas : función de transporte (*Function Shipping*), procesamiento de transacciones asincrónicas (*Asynchronous transaction processing*) y procesamiento de transacciones distribuidas (*Distributed transaction processing*).

1.3.2.2 Función de transporte

La función de transporte consiste en enviar un dato simple accedando una primitiva en un lugar remoto. Por ejemplo, si una transacción en el lugar 1 requiere datos de un archivo en el lugar 2, éstos pueden ser obtenidos vía función de transporte. Si el archivo en el lugar 2 se especifica a *CICS/ISC* la operación será ejecutada en forma transparente al programa.

La implementación de la primitiva *DL1*, por ejemplo, sería de la siguiente manera : una transacción *A*, en el lugar 1 requiere acceder una base de datos de *IMS* en el lugar 2 (Figura 1.5). Los pasos a seguir serían los siguientes :

- Se establece una sesión (desde cualquier software de comunicación). Esto establece la comunicación con *CICS* en el lugar 2.

- La primitiva *DL1* es enviada a *CICS* en el lugar 2.
- *CICS* en el lugar 2 crea una transacción especializada para manejar las demandas. Las transacciones creadas por *CICS* para ejecutar demandas remotas son llamadas transacciones espejo (*Mirror transaction*).
- El resultado espejo se ejecuta por el sistema en el lugar 2 como si fuera una transacción local.
- El resultado se envía de regreso. Si la transacción espejo no ejecuta actualizaciones, ésta es cancelada y la sesión finaliza.

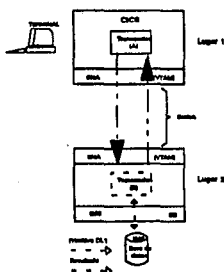


Figura 1.5. Función de transporte en CICS/ISC.

La función de transporte es muy ineficiente en términos de uso de recursos y tiempo de respuesta de transacciones individuales. Esto es, porque el establecer una sesión y una transacción espejo, requiere de operaciones pequeñas como acceder un archivo que se puede llevar cientos de milisegundos o segundos.

1.3.2.3 Procesamiento de transacciones asincrónicas

Con el procesamiento de transacciones asincrónicas una transacción local comienza a ejecutar una transacción en un lugar remoto. Así mismo, la transacción remota puede activar otras transacciones. Muy a menudo se utiliza para ejecutar una tercera transacción. Por ejemplo; la figura 1.6, muestra que la transacción A en el lugar 1 activa la transacción B en el lugar 2, la cual a su vez activa la transacción C en el lugar 1. La terminal que ejecuta la transacción A recibe el resultado de la transacción C.

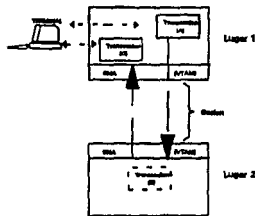


Figura 1.6. Procesamiento de transacciones asincrónicas en CICS/ISC.

El modo de operación es más eficiente que la función de transporte porque requiere aproximadamente los mismos recursos que el proceso de demanda que la función de transporte requiere, los costos de estos recursos son compensados con la ejecución de una transacción remota, la cual involucra más, que acceder un archivo. La principal desventaja es que la atomicidad de una base de datos distribuida no se garantiza. Por ejemplo, el sistema no se puede recuperar si la transacción B (Figura 1.6) falla antes de comenzar la transacción C.

1.3.2.4 Procesamiento de transacciones distribuidas

En el procesamiento de una transacción distribuida (figura 1.7) una transacción local puede comenzar una transacción remota como en el inciso anterior. Sin embargo, al iniciar una transacción deberá esperar a que la transacción que inició el proceso ejecute la función que solicitó. Las dos transacciones se ejecutan de manera sincrónica. Existe además, un intercambio de mensajes mientras las dos transacciones se ejecutan.

La principal desventaja del procesamiento de una transacción distribuida es que no tiene transparencia, porque los programas de aplicación deben especificar el lugar en el cual la transacción remota se localiza cuando ésta se va a activar.

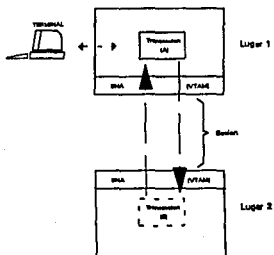


Figura 1.7. Procesamiento de transacciones distribuidas en CICS/ISC.

1.3.3 Otros Sistemas

El mundo industrial se está enfocando en los sistemas que manejan bases de datos distribuidas, entre los sistemas tenemos los siguientes :

- ▶ D_NET por Applied Data Reseach (ADR).
- ▶ IDMS_DDS por Cullinane Corporation.
- ▶ IDMS_DBB50 por International Computers Limited (ICL).
- ▶ VDN por Nixdorf Computer AG.
- ▶ UDS_D por Siemens AG.
- ▶ NET_WORK por software AG.

Algunos de estos paquetes todavía están en desarrollo y aún no se liberan. Como todos estos productos están cambiando rápidamente no se compararán sus características, sólo se usarán como ejemplo.

Todos los sistemas excepto *VDN* han sido desarrollados a partir de sistemas centralizados que utilizan manejadores de bases de datos centralizadas. En el caso de *VDN* el sistema local y distribuido fué desarrollado conjuntamente. Los sistemas locales de *IDMS_DDS*, *IDMS_DBB50* y *UDS_D* son sistemas *Codasy1* de *IDMS* y *UDS*. Los sistemas locales de *D_NET* y *NET_WORK* son *DATACom/Db* y *ADABAS* que pertenecen a la clase de sistemas relacionales con listas inversas. El sistema local de *VDN* es *Codasy1* y relacional.

Distribución transparente; los sistemas *EMCOMPASS* e *IBM CICS/ISC* no usan el diccionario de datos para obtener localización transparente. Otros sistemas como *IDMS_DDS* y *D_NET* estan concentrados en su diccionario de datos. En estos sistemas toda la información sobre la distribución de la base de datos está en el diccionario de datos.

Unidades de distribución; La unidad básica de distribución en muchos sistemas es el archivo, solo *VDN* y *ENCOMPASS* permiten la partición de un archivo en pequeñas unidades. Los sistemas *Codasy1* tienen restricciones adicionales como el mapeo de unidades lógicas a un solo lugar. Por ejemplo *IDMS_DDB50* debe ser almacenado en un solo lugar.

Duplicación de datos: Solo tres sistemas (*D_NET*, *VDN* y *NET_WORK*) soportan la duplicidad de datos. En *D_NET* un archivo puede ser duplicado sin embargo solo el archivo maestro sera actualizado. En *NET_WORK* las copias de los archivos se usan igual que en *D_NET* solo que pueden ser forzados a ser consistentes. *VDN* permite la duplicación de fragmentos los cuales se pueden traslapar.

Acceso a bases de datos remotas; La posibilidad de acceder un archivo remoto es provista por muchas redes de computadoras. En *CICS/ISC* es posible por ejemplo, ejecutar una primitiva remota (Función de carga). La principal desventaja, es que las primitivas que manipulan las bases de datos accesan poca información. Una manera de solucionarlo es construir primitivas más poderosas.

CAPITULO II

Redes Locales de Micros.

2.1 Elementos de una red

En una red local de micros es posible que varias microcomputadoras puedan compartir información y recursos dentro de una área limitada, generalmente menor a una milla. Una red local requiere básicamente que las estaciones individuales (microcomputadoras) sean físicamente conectadas con : tarjetas de interfaz, cables coaxiales, par torcido o cualquier otro medio de transmisión y que cierto software esté instalado en el disco duro del servidor de archivos. El software va a permitir, compartir periféricos, datos, programas de aplicación y hacer uso de cualquier servicio que la red permita.

2.1.1 Estaciones Individuales.

Las estaciones, son computadoras personales que estan conectadas a la red, tiene la gran habilidad de poder trabajar con su propio software (Software que puede estar en la misma computadora) o bien, son capaces de poder acceder información que se encuentra en el disco duro del servidor (*Disk Server*) figura 2.0.

Cada estación de la red tiene una tarjeta de interfaz, a la cual se le conecta un cable que va al servidor de la red. Existen además, unos programas (Shell y protocolo) que junto con el sistema operativo establecen la conexión correcta.

Servidor de Archivos (Disk Server)

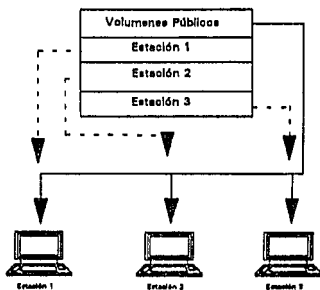


Figura 2.0. Estaciones conectadas con tarjetas de interfaz y cables al servidor de archivos.

2.1.2 Disco duro del servidor (Disk Server)

Todas las redes locales utilizan un disco duro (*Disk Server*) que contiene información que puede ser compartida por las estaciones en la red. Cada estación individual ve al disco duro como un drive más. Por ejemplo, si el disco duro del servidor fuera designado como drive F, cada estación vería un drive más en su configuración individual, si la estación sólo tiene un drive A, al conectarse a la red tendría el drive A y F, donde el drive F generalmente con mayor capacidad de almacenamiento.

Este procedimiento es igual al que las computadoras personales (Ibm-Pc, Hewlett-Packard, Etc.) manejan para almacenar sus archivos.

2.1.3 Servidor de archivos (File Server)

Los servidores de archivos son mucho más eficientes y sofisticados que el disco duro del servidor. Un servidor de archivos contiene un software muy especializado como lo es el sistema operativo.

El servidor de archivos coordina todas las estaciones de trabajo y regula la forma en cómo son compartidos los recursos, además de controlar qué impresora se usa primero y cuáles archivos pueden ser accedidos por determinada estación.

Todos los archivos de la red son almacenados en el disco duro del servidor, además de discos flexibles o bien discos duros de cada estación. La figura 2.1 ilustra a un servidor de archivos en una red local.

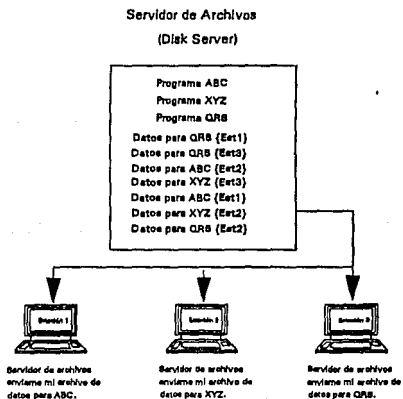


Figura 2.1. Estaciones de trabajo conectadas al servidor de archivos, el cual suministra una gran eficiencia en una red local de microcomputadoras.

2.1.4 Servidor de archivos dedicado y no dedicado

En el mundo de los negocios irónicamente la dedicación no siempre implica un costo menor, por lo general puede resultar muy caro.

Un servidor de archivos dedicado, es una microcomputadora con un disco duro, usado exclusivamente como servidor de archivos. Por dedicado se entiende, que toda la memoria y recursos de procesamiento van a ser usados por el servidor de archivos, el cual incrementa en gran medida la velocidad de la red y la eficiencia de la misma. Cuando un servidor de archivos es no dedicado, éste puede ser usado como estación de trabajo además de cumplir con funciones de servidor. Esto significa que la memoria RAM puede ser particionada para que exista disponibilidad de poder ejecutar programas. Se podría ejemplificar de la siguiente manera; si la estación de la red desea un archivo y el servidor de archivos está siendo utilizado como otra estación y además se está cargando un programa a memoria, deberá esperar a que el servidor termine su proceso para posteriormente ser atendida su petición de archivo. Cabe mencionar, que un servidor de archivos dedicado es mucho más rápido y caro que un servidor no dedicado.

2.1.5 Servidor de archivos distribuido

Existen muchos lugares donde un servidor de archivos llega a ser más que suficiente. Esto es a lo que se le conoce como "Servidor Centralizado". Su función es parecida a la de una microcomputadora, puesto que todos los requerimientos o solicitudes de archivos son manejadas desde una sola entidad, y donde cada estación espera su turno de ser atendida. Si la red local de micros fuera designada para manejar varios departamentos en un compañía, es mucho más eficiente contar con varios servidores de archivos adicionales.

Estas unidades adicionales son conocidas como "*Servidores de Archivos Distribuidos*", los cuales se pueden repartir la carga de la red. Por ejemplo; en una compañía que tiene un departamento de contabilidad, las estaciones usan los mismos programas y accesan los mismos datos. Puede llegar a ser muy ineficiente el enviar la información a un servidor de archivos que se encuentra a una distancia muy grande. Un servidor de archivos distribuidos que se localice en el departamento de contabilidad, puede aumentar la velocidad de acceso además de reducir la carga al resto de la red. Es decir, como el servidor se localiza en el mismo departamento de contabilidad los archivos que son solicitados recorren una distancia más corta y así la información llega más rápido.

Una de las enormes ventajas de los servidores de archivos distribuidos es, que si el servidor de archivos dejará de funcionar la red seguirá trabajando sin ningún problema, ya que con suficiente espacio en disco, un servidor de archivos distribuido puede dar servicio a la red en su totalidad temporalmente.

Así como los servidores de archivos tienen un gran número de ventajas, la seguridad puede llegar a ser más difícil, el administrador de la red deberá ahora tener la seguridad de que todos los servidores de archivos estén protegidos de usuarios no autorizados.

2.1.6 Servidores de Impresión

Dentro de una red local, no sólo se puede disponer de un servidor de archivos, sino se puede instalar un servidor de impresión, el cual permite habilitar que docenas de estaciones compartan varios tipos de impresoras (impresoras laser, impresoras de impacto, impresoras térmicas, etc).

El tener un servidor de impresión así como el software para manejar las impresoras, no implica que una estación de la red no pueda tener su impresora conectada. Esta impresora conectada al puerto paralelo de la estación, puede ser llamada *Impresora Local* y no una *Impresora de la Red*, la cual podrá tener fines especiales, como por ejemplo, impresión especial de formas continuas.

El software que permite compartir las impresoras, deberá tener un *spooler* de impresión ("*Cola de Impresión*"), el cual es un *buffer* donde los archivos que van a ser impresos son almacenados hasta que la impresora se encienda o esté lista. Una vez que los archivos se comienzan a imprimir, la cola de impresión va caminando. Es decir, el siguiente archivo de la línea es colocado al inicio de la cola, una vez que se termina de imprimir el archivo actual. Existen colas de impresión que pueden llegar a ser muy sofisticadas, por ejemplo; pueden tener la habilidad de mover un archivo al inicio de la cola para que éste sea impreso inmediatamente.

2.2 Topología de redes

El término topología en el contexto de redes de comunicación, se refiere a la manera en la cual los puntos terminales o estaciones de una red están interconectadas. Una topología está definida "por la manera en que se colocan sus ligas de comunicación y elementos de switcheo, en ella se determinan las rutas que los datos pueden usar entre cualquier par de estaciones."¹

La topología de una red tiene un serio impacto en su eficiencia, costo, expansión y confiabilidad. La selección del medio de transmisión y tipo de configuración física están íntimamente relacionados con la selección de la topología.

Existen 3 topologías de redes básicamente :

- Topología de *bus* o línea.
- Topología de *ring* o anillo.
- Topología de *star* o estrella.

y de una combinación de ellas se pueden construir topologías más complicadas.

2.2.1 Topología de Bus o línea

En la topología de *bus* o línea figura 2.2, la red es simplemente el medio de transmisión (no switches y no repetidores). Todas las estaciones se conectan a través de interfaces directamente a la línea. Una transmisión emitida por cualquiera de las estaciones viaja de su origen en ambas direcciones de la línea, por lo cual cada una de las estaciones tiene acceso a dicha información. Sin embargo, solo la estación destino, es capaz de procesarla.

¹ Stallings William, Ph.D. Local Networks An Introduction. Ed. Macmillan Publishing Company 1984.

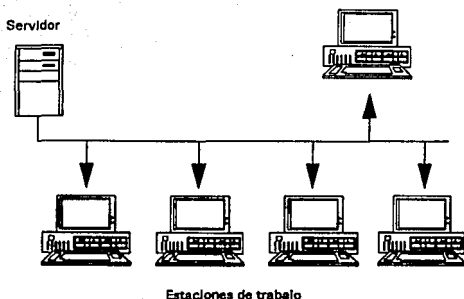


Figura 2.2. Topología de Bus o línea.

Debido a que esta topología no requiere de enrutamiento, la comunicación es más rápida que una topología de estrella o anillo. Sin embargo, como todas las estaciones comparten el mismo medio de transmisión sólo una estación puede transmitir a la vez por lo tanto se requiere de un control de acceso para determinar cuál estación es la que va a transmitir enseguida.

Una de las enormes ventajas de esta topología es el bajo costo, además de que si una estación deja de trabajar o falla, no afecta al resto de la red.

Su principal desventaja, es la necesidad de establecer una distancia mínima entre sus estaciones para evitar señales de interferencia, esta arquitectura no posee grandes características de seguridad, ya que como los mensajes son enviados a través de una línea de datos común ésta puede ser interceptada por usuarios no autorizados a la red.

2.2.2 Topología de ring o anillo

En la topología de anillo figura 2.3, la red consiste en un conjunto de repetidores unidos punto a punto en un bucle cerrado. Cada repetidor tiene participación en dos ligas, es decir, es capaz de recibir información por la estación anterior a ella y pasar *bit* por *bit* tan rápido como es recibida por la otra estación. La transmisión en esta topología es unidireccional a diferencia de las redes bus.

Cada estación de la red se conecta a un repetidor y los datos son transmitidos en paquetes. Por ejemplo, si la estación X desea transmitir un mensaje a la estación Y el mensaje se divide en paquetes, en donde cada paquete contiene una porción de datos más algún tipo de información de control incluyendo la dirección de Y. Los paquetes viajan a través del anillo, uno a la vez y circulan hacia los repetidores, la estación Y reconoce su dirección y copia el paquete conforme va llegando.

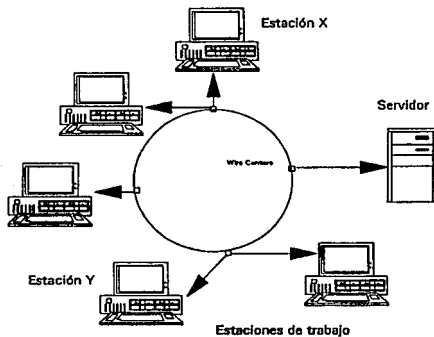


Figura 2.3. Topología de ring o anillo.

Una de las características de la topología de anillo es la necesidad de que todas las estaciones tengan igual acceso a la red. En una red (*Token Ring*) a un paquete se le conoce como *token*, el cual es transmitido de la estación a la red. El *token* contiene la dirección de la estación emisora y la dirección de la estación que recibe el mensaje, cuando la estación que recibe el mensaje regresa el *token* a la estación que lo envió, la cual posteriormente puede enviar el *token* a la siguiente estación en la red.

Es importante hacer notar que el administrador de la red tiene una estación designada como nodo de monitoreo en la red. El cual maneja todas las funciones de diagnóstico de la red, si ésta llegará a fallar entonces otra estación podrá ser designada para esta tarea.

Existen muchas ventajas en la topología de anillo por ejemplo, utilizando software de *Bypass* se puede llegar a mantener la integridad de la red aún cuando una o varias estaciones no estén funcionando por fallas.

La topología de anillo tiene una desventaja, si varias estaciones se cablean o conectan juntas puede llegar a ser extremadamente difícil agregar nuevas estaciones, cuando esto sucede la red se tiene que detener por completo, mientras un nuevo nodo es agregado y los cables son reconectados. Sin embargo, muchas redes *Token Ring* son equipadas ahora con un aditamento llamado "*Wire Centers*". Estos conectores proveen al administrador de la red la facilidad de agregar o quitar estaciones, sin tener que detener por completo la red.

2.2.3 Topología de star o estrella

En la topología de estrella figura 2.4 cada estación está conectada a una computadora central (igual al sistema de teléfonos). Las comunicaciones entre dos estaciones de la red es vía circuito conmutado. Una estación que desea enviar datos, primero deberá enviar una solicitud al nodo central, pidiendo una conexión con la estación destino. Una vez que se ha establecido dicha comunicación, los datos pueden ser intercambiados entre ambas estaciones. Dicha conexión es una conexión dedicada punto a punto durante el tiempo que la comunicación dura.

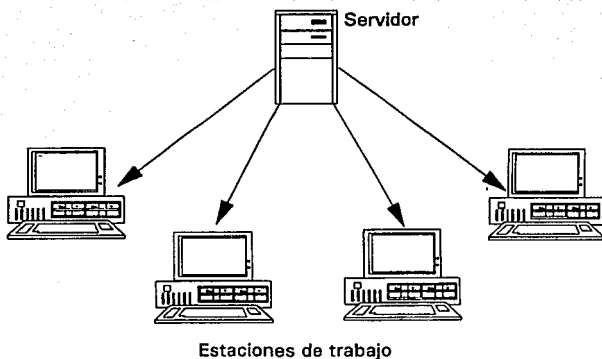


Figura 2.4. Topología de star o estrella.

En esta topología las comunicaciones son totalmente centralizadas. Todas las comunicaciones son controladas por el servidor de archivos, el cual conoce todos los caminos o rutas de datos existentes.

Es muy fácil agregar una nueva estación a la red, puesto que sólo se necesita un cable de la computadora central a la nueva estación, así como una tarjeta de interfaz para poder establecer la comunicación con la red. Una ventaja en esta topología es que el administrador puede otorgar a ciertos nodos un estatus más alto que a otros con lo cual le permite reconocer más rápido a una estación con prioridad mayor a la otra. Finalmente una arquitectura de estrella permite centralizar todos los diagnósticos de la red, enviar los mensajes a través del servidor de archivos, establecer un fácil análisis de todos los mensajes de las estaciones y producir reportes indicando que archivos a utilizado cierto nodo.

La enorme desventaja de esta topología es que si el servidor de archivos fallara, la red dejará de funcionar hasta que ésta sea reparada.

2.3 Medios de transmisión

El medio de transmisión es la ruta física que existe entre el transmisor y receptor de una red local. La configuración más común es una liga punto a punto entre dos dispositivos transmisor/receptor, los cuales por medio de interfaces apropiadas pueden manejar señales digitales o analógicas. Puede existir un dispositivo intermedio el cual sirve para compensar la atenuación o deterioros de la señal de transmisión. Las ligas punto a punto (figura 2.5) son usadas en topologías de anillo que se conectan a repetidores adyacentes y en una topología de

estrella, los dispositivos se conectan a un switch central. Las ligas punto a punto son también usadas para conectar dos redes locales que pueden estar en dos edificios o localidades diferentes.

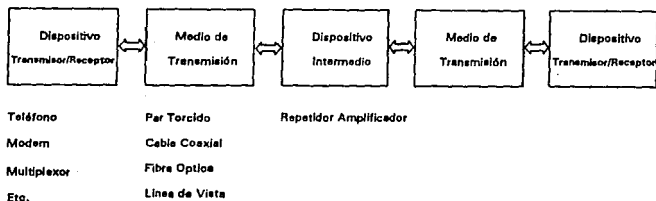


Figura 2.5. Configuración Punto a Punto.

Las ligas multipunto (figura 2.6) son usadas para conectar múltiples dispositivos como en la topología de bus, los dispositivos se conectan al medio de transmisión en varios puntos; repetidores (señales digitales) o amplificadores (señales analógicas). Estos son utilizados para extender la longitud del medio de transmisión.

El medio que puede ser utilizado en una red incluye el par torcido, cable coaxial y fibra óptica.

La selección del medio es función de factores tales como : la velocidad de transmisión, longitud de la red, señales radiadas por el medio ambiente, costo y mantenimiento.

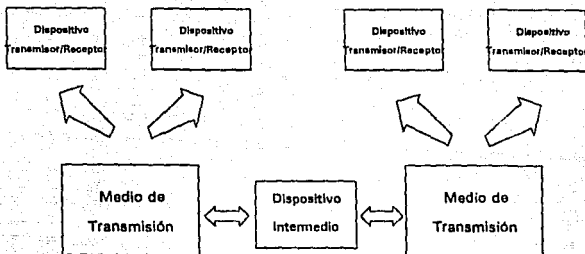


Figura 2.6. Configuración Multipunto.

2.3.1 Par Torcido

Uno de los medios de transmisión más comunes tanto para datos analógicos y digitales es el par torcido figura 2.7.

El par torcido consiste en dos alambres aislados y arreglados en forma de espiral. Los alambres son de cobre o acero revestido de cobre. El cobre facilita la conductividad y el acero proporciona resistencia. Para señales analógicas se requieren amplificadores cada 5 o 6 km., mientras que para señales digitales los repetidores se usan cada 2 o 3 km. Es muy común usar el par torcido para transmitir señales analógicas como son la voz.

El par torcido puede ser usado tanto para aplicaciones punto a punto o multipunto. En una aplicación multipunto el par torcido es más barato pero con la desventaja de poseer un rendimiento bajo.



Figura 2.7. Par torcido 2 pares.

Se pueden manejar señales en un rango de 15 Km. o más y se puede transmitir a baja velocidad cientos de Kbps., el par torcido además de ser mucho más barato que el cable coaxial o la fibra óptica, posee limitaciones de conectividad. Sin embargo, para una red local con un número limitado de usuarios el par torcido es una opción ideal además de ser fácil de instalar.

2.3.2 Cable coaxial

Tal vez uno de los medios de transmisión más versátiles para redes locales es el cable coaxial. Existen 2 tipos de cables coaxiales para el uso de redes locales. El primero es el que comunmente se utiliza en una antena de televisión llamado CATV (*Community antena television*) con una impedancia de 75Ω , el segundo es el cable con una impedancia de 50Ω utilizado para señales digitales llamado banda base (*base band*). El cable de 75Ω se usa para señales analógicas y se le conoce como banda ancha (*broadband*).

El cable coaxial (figura 2.8) tiene una construcción parecida a la del par torcido, consiste en dos conductores que pueden operar sobre un amplio rango de frecuencias, está compuesto de un conductor de cobre envuelto por un aislante, la cubierta exterior está compuesta de cobre o aluminio que actúa también como conductor además de brindar protección. Un cable coaxial tiene un diámetro de 0.4 a 1 pulgada.

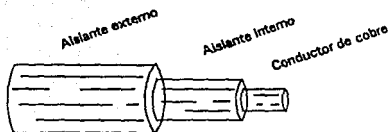


Figura 2.8. Cable coaxial.

El cable coaxial es aplicable a configuraciones punto a punto o multipunto. El cable de 50 Ω o banda base, puede llegar a soportar 100 dispositivos por segmento, con sistemas muy grandes se pueden ligar o conectar los segmentos con repetidores. El cable de 75 Ω puede soportar cientos de dispositivos, el uso de este cable a velocidades muy altas (50 Mbs) puede introducir problemas técnicos, además de limitar el número de dispositivos de 20 a 30.

La inmunidad al ruido del cable coaxial depende de la aplicación e implementación. En general es superior al par torcido para altas frecuencias. El costo de instalación de este medio de transmisión es intermedio al par torcido y fibra óptica.

2.3.3 Fibra óptica

El desarrollo más excitante en los medios de transmisión para una red local es sin lugar a dudas la fibra óptica (figura 2.9) a diferencia del par torcido o cable coaxial, en el cual la señal es transmitida como señal eléctrica la fibra transmite la información como una señal codificada en forma de luz. Es posible transmitir a altas velocidades (en rango del GHz.). La fibra óptica es inmune a interferencias electromagnéticas o radio frecuencias y es capaz de transmitir señales a varios kilómetros

(7 - 8 Km.) sin repetidores y sin ninguna pérdida. Este medio de transmisión es virtualmente inmune a recepciones no autorizadas.

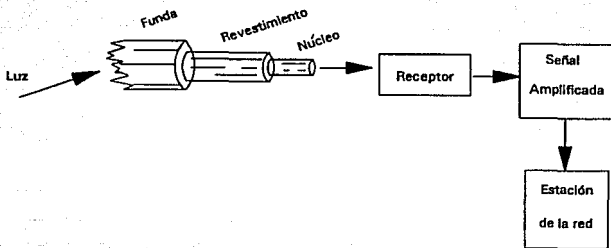


Figura 2.9. Fibra Óptica.

El cable está hecho de un cristal puro dentro del cual hay una fibra ancha que forma el núcleo. Estas fibras son envueltas por un revestimiento que es una capa con un índice de refracción más bajo que el del cristal del núcleo.

Una red que utiliza como medio de transmisión la fibra óptica usa un rayo laser o un led (diodo emisor de luz) para enviar una señal a través del núcleo del cable. Los repetidores ópticos son a menudo usados a lo largo del camino, para amplificar la señal que va al destino, es decir que ésta llegue con la suficiente fuerza.

En la recepción final del cable el mensaje es convertido a una señal analógica o digital según sea el caso por un fotodiodo. El cable puede consistir de una sola fibra monomodo (*monomode*), varias fibras multimodo (*multimode*) o una variación de multimodo de índice degradado (*graded index*) en el cual el índice de refracción va lentamente de el centro de la fibra hacia afuera. La fibra monomodo tiene un ancho de banda muy grande,

pero como su núcleo es extremadamente pequeño dificulta el empalme de varias fibras. También el monomodo requiere de un laser más caro que una luz emitida por un led.

La fibra multimodo tiene un ancho de banda muy pequeño pero es más fácil de empalmar. La fibra multimodo de índice degradado es el medio más caro, pero facilita las transmisiones a grandes velocidades sobre distancias muy grandes.

En el presente la utilización de la fibra óptica como medio de transmisión es muy caro además de ser una tecnología muy sofisticada. Puede llegar a dificultarse el sumar una nueva estación a la red después de la instalación inicial.

El uso más común de la fibra óptica es en configuraciones punto a punto.

2.3.4 Línea de Vista

La línea de vista consiste en una transmisión de ondas electromagnéticas a través de la atmósfera, y una trayectoria de línea de vista es requerida entre transmisor y receptor. Puede manejar frecuencias de microonda, infrarojo o laser. Es muy susceptible a interferencias ambientales. Existe además un alto riesgo de inseguridad, corrupción de información o errores y está limitada a disponibilidad de frecuencias.

2.4 Protocolos de comunicación

El concepto de red está basado en la relación de un grupo de microcomputadoras conectadas, de tal forma que se pueden comunicar entre sí. La comunicación consiste básicamente en el envío de paquetes de información del servidor de archivos a las estaciones de trabajo y viceversa. La forma en la que esta información es mandada se le llama "*Protocolo de comunicación*". y es el sistema operativo de la red el que se encarga de enviar la información.

Cada topología utiliza diferentes protocolos de comunicación:

<u>Topología</u>	<u>Protocolo</u>
<i>Star o Estrella</i>	<i>Poleo</i>
<i>Bus o línea</i>	<i>CSMA</i>
<i>Ring o Anillo</i>	<i>Token Passing</i>

2.4.1 Poleo

Este protocolo está asociado con la topología de estrella. Su funcionamiento puede ser comprendido si imaginamos un reloj con 12 números y una manecilla. La manecilla está girando rápidamente alrededor de los 12 números y cada vez que pasa por un número pregunta si ese número tiene información que mandar, si es así, el mensaje es leído por el servidor de archivos. Si no existe mensaje la manecilla pasa al siguiente número. Si el servidor de archivos tiene información para alguna estación de trabajo, la manecilla se coloca en el número adecuado y el mensaje es enviado. Este protocolo elimina la posibilidad de que una estación de trabajo interfiera la comunicación de otra estación de trabajo.

2.4.2 CSMA

El protocolo CSMA (Carry Sense Multiple Access) está asociado a la topología de bus o línea. Su funcionamiento se puede comparar al de una carretera. Cuando existe poco tráfico, la carretera es segura y fácil de utilizar; un conductor se para en la entrada, echa un vistazo, si la carretera está libre entonces entra. Pero si la carretera está congestionada, el conductor tiene que esperar para entrar a la carretera.

El protocolo CSMA tiene a las estaciones de trabajo y al servidor de archivos escuchando la línea de comunicación todo el tiempo. Una estación de trabajo sólo se puede comunicar cuando la línea está libre. Si una estación trata de enviar un mensaje mientras la línea está ocupada, ésta tiene que esperar hasta que la línea quede libre.

2.4.3 Token Passing

Este protocolo es usado en la topología de anillo y anillo modificado. Un "TOKEN" es un sistema especial de acarreo de información que es transmitido en la red y que circula de una estación de trabajo a la siguiente de una manera controlada semejante a como la hace un trenecito de juguete en una vía circular. Las estaciones de trabajo que no tienen nungun mensaje que enviar lo dejan circular por su camino. Las estaciones que si tienen mensajes lo "agregan" al "TOKEN" y éste continua a la siguiente estación.

2.5 Normas para redes locales

Hoy en día una de las tendencias de la comunidad de comunicaciones es definir un conjunto de estándares para el diseño del hardware y software. Estos estándares consisten en una serie de capas, las cuales tienen funciones de comunicaciones. Los estándares son llamados *ISO/OSI* (*The international standards organization/Open system Interface*) modelos de referencia. El modelo de referencia permite que varios vendedores de hardware y software puedan producir productos que se logren comunicar entre sí.

Este modelo fue desarrollado con la intención de proveer una estructura para el diseño de normas de protocolos, es decir, que realicen varias funciones a todos los niveles de la red y no como una norma en la construcción de equipos.

El *ISO/OSI* consiste en 7 capas y su estructura es la siguiente :

	<i>Función</i>	<i>Capa</i>
USUARIO {	<i>Aplicación</i>	7
	<i>Presentación</i>	6
	<i>Sesión</i>	5
	<i>Transporte</i>	4
SERVICIO DE RED {	<i>Red</i>	3
	<i>Encadenamiento</i>	2
	<i>Línea Física</i>	1

Cada capa puede ser vista como un programa o proceso que se comunica con el correspondiente proceso en otro dispositivo.

De hecho la comunicación dispositivo a dispositivo no se realiza de forma horizontal en cada capa, sino que se realiza verticalmente de las capas superiores a inferiores del dispositivo que transmite y viceversa para el receptor.

Durante la transmisión, cada capa transforma el o los datos enviados por la capa superior, adicionando a los datos una cabeza (*Header*) conteniendo información de control usada por el protocolo en su nivel, antes de pasar el mensaje a otro nivel. Inversamente en el lado receptor las cabeceras (*Headers*) son procesadas y separadas de los datos al ir subiendo las capas. Ninguna cabecera de las capas inferiores alcanza la capa superior.

2.5.1 Línea física

La línea física es la responsable de la transmisión de bits sobre el circuito físico. A esta capa le conciernen todos los aspectos de señales y modulación, la transformación entre las señales eléctricas (Voltajes y corrientes) y la representación de la información en forma digital (1 y 0). Además de que se definen todas las especificaciones tanto eléctricas como mecánicas para la interconexión de componentes.

2.5.2 Encadenamiento

Provee el servicio de entrega de paquetes, además de que se asegura que lleguen correctamente al nodo receptor. Los bits que se transmiten forman un conjunto que contiene información de control codificada incluyendo el mensaje que se va a transmitir.

Tiene las siguientes funciones :

- Secuencia (Sincronía)
- Control de flujo.
- Transparencia de datos.
- Conexión desconexión.
- Control de error.

Con la sincronía se puede determinar cuándo y dónde comienzan los datos.

El control de flujo se refiere a una estación que envía un cierto número de conjunto de bits o paquetes (*Frames*) y deja de enviar hasta que recibe una confirmación ("*ack*") del primer conjunto de bits (*Frame*).

El control de error se determina cuando cierto paquete al recibirlo tiene un error, la estación que lo recibe puede enviar un ("*Nak*") confirmación negativa, con la cual se especifica que hubo un error. Con la ayuda de esta confirmación, la estación que comenzó la transmisión se da cuenta del error y el paquete que contiene el error es retransmitido.

2.5.3 Red

Su función es facilitar la transmisión/recepción de un paquete de su nodo fuente a destino. Dentro de la red produce la estrategia de enrutar un paquete a su destino.

Tiene dos responsabilidades básicamente :

- Control de flujo.
- Ruteo de mensajes.

El control de flujo se refiere al paso de los mensajes a las áreas de memoria para así mantener el mensaje en el nodo receptor. Cabe mencionar que los mensajes de entrada y salida que pueden ser mantenidos en el nodo varía de acuerdo a los sistemas operativos locales. Es por eso que es necesario tener una taza de transmisión para que los mensajes que se envían puedan encontrar un espacio adecuado en los buffers del receptor.

En el ruteo de mensajes se trata de encontrar una ruta propia adecuada entre el nodo receptor y transmisor. Por ejemplo, en una topología de bus o en una red con nodos adyacentes de comunicación la ruta es trivial. Sin embargo, en redes donde pueden intervenir nodos a través del camino o donde existe múltiples rutas de un nodo a otro, la función de ruteo trata de encontrar la mejor ruta de las rutas alternas.

2.5.4 Transporte

Provee un mecanismo confiable para el intercambio de datos entre procesos en diferentes sistemas. Asegura además que los datos sean entregados libres de error, en secuencia, sin pérdidas o duplicaciones. A esta capa del OSI se le conoce como la calidad del servicio, donde esto se define en términos de la proporción de datos y errores durante las transferencias de datos.

La capa de transporte incluye protocolos para el manejo de redes con diferentes calidades en su servicio. Cada tipo de red requiere de un manejo de errores para recuperarse. Esta capa además, es una línea donde las funciones de concentración y desconcentración ocurren. Las funciones anteriores son importantes ya que ayudan a la topología a acomodarse para la carga de datos.

2.5.5 Sesión

La capa de sesiones es la responsable de controlar una comunicación "Sesión" entre dos entes comunicativas en nodos remotos. Es además la responsable de crear y mantener un registro de la sesión que se establece. Una sesión es una conexión específica entre dos puntos. La sesión garantiza una interacción correcta y sincronizada. Los servicios de la sesión son :

- Servicios de administración : Función relacionada al enlace de los usuarios, esto se logrará a través del establecimiento de registros de sesión.

- Servicios de diálogo : Estas funciones son relativas al intercambio de datos, sincronización de los datos a intercambiar y repetición de los datos intercambiables cuando se genera un error.

2.5.6 Presentación

La capa de presentación es la responsable de dar servicios a la capa de aplicación para el correcto manejo de los datos que se intercambiaron, el despliegue de los mismos así como su correcta interpretación sintáctica. Tiene la seguridad de dar una interface uniforme entre las aplicaciones interconectadas en los diferentes nodos. Los servicios de esta capa incluyen :

- Compresión de datos.
- Transformaciones de código.
- Suministrar el despliegue de columnas.
- Traducción de comandos.
- Formateo.

2.5.7 Aplicación

Las funciones ejecutadas en esta capa son las que el usuario especifica. Es muy difícil generalizar acerca de los protocolos, puesto que los programas de cada usuario son diferentes además de que establecen diferentes necesidades.

En este nivel el usuario podrá encontrar programas de administración de base de datos, correos electrónicos, programas para servidores de archivo, impresión, sistemas operativos, transferencias de fondos, transferencia de archivos (File transfer) etc.

Cabe mencionar que en esta capa se trata de especificar un lenguaje común de interface para el usuario, así como un conjunto de definiciones para la invocación de estos elementos de servicio para correr un programa de aplicación.

CAPITULO III

Base de datos distribuida.

3.1 Ventajas y desventajas de una base de datos centralizada vs. distribuida

Las bases de datos distribuidas no son simplemente implementaciones de bases de datos centralizadas. Estas presentan diferentes características de los sistemas tradicionales centralizados, es por esto que es necesario analizar las particularidades de las bases de datos tradicionales como son : control centralizado, independencia de datos, reducción de redundancia, estructuras físicas complejas para un acceso eficiente, integridad, recuperación, control de concurrencia, privacidad y seguridad.

3.1.1 Control centralizado

La posibilidad de tener un control centralizado sobre las fuentes de información en una empresa u organización, fué considerada como una de la motivaciones más grandes al introducir bases de datos. La función fundamental de un administrador de bases de datos (*Data Base Administrator DBA*) fué el garantizar la seguridad de los datos de una determinada empresa, las cuales requerian de una responsabilidad centralizada.

En bases de datos distribuidas la idea de un control centralizado tiene menor importancia. En general, en una base de datos distribuida es posible tener una estructura de control jerárquico basada en un administrador de base de datos global que tiene la responsabilidad central de todas las bases de datos, mientras que los administradores de bases de datos locales, tienen la responsabilidad central de todas las bases de datos locales. Hay que recalcar que los administradores locales deben tener un alto grado de autonomía.

"Dependiendo del diseño del sistema distribuido, cada administrador local podrá tener un grado de autonomía diferente, que se conoce como autonomía local. La posibilidad de contar con autonomía local es una ventaja importante en las bases de datos distribuidas"¹

3.1.2 Independencia de datos

La independencia de los datos es también considerada como una de las principales características al introducir bases de datos. Esencialmente la independencia de los datos significa que la organización de los datos es transparente a los programas de aplicación. Los programas que son escritos tienen una "conceptual" vista de los datos llamada "*Esquema conceptual*." La principal ventaja de la independencia es que los programas no son afectados si existen cambios en la organización física de los datos.

¹ Ceri S., Pelagatti G. Distributed data bases. Principles & Systems. Ed. McGraw-Hill 1985

En base de datos distribuidas, la independencia de los datos tiene la misma importancia que en las bases de datos tradicionales; sin embargo, un nuevo aspecto es agregado llamado "Distribución Transparente". La distribución transparente son todos los programas que pueden ser escritos como si la base de datos no fuera distribuida. La corrección de un programa no es afectada por el cambio de un lugar a otro; sin embargo, la velocidad de ejecución si se puede llegar a afectar.

3.1.3 Reducción de redundancia

En las bases de datos tradicionales la reducción de los datos se trata de evitar principalmente por dos razones :

- La inconsistencia de diferentes copias de la misma estructura lógica, que se evita al tener una sola copia.
- El espacio de almacenamiento que se elimina con la redundancia.

La reducción de redundancia se obtiene al compartir los datos, esto es permitiendo que varias aplicaciones accesen los mismos archivos y registros.

En bases de datos distribuidas existen varias razones para considerar la redundancia de datos como una característica deseable:

- Las aplicaciones locales pueden llegar a elevar su rendimiento si existe duplicidad de datos, en los lugares donde se necesita.
- La disponibilidad del sistema se puede incrementar, porque si una localidad falla no se detiene la ejecución de aplicaciones en otras localidades.

Cabe mencionar, que las razones que existen en contra de la redundancia en bases de datos tradicionales, son válidas para un medio ambiente distribuido con la condición de que debe existir una evaluación del grado óptimo de la redundancia que requiere un análisis y diseño más profundo.

3.1.4 Estructuras físicas complejas para un acceso eficiente

Estructuras de acceso complejas como; archivos secundarios de índices, cadenas de interarchivos, etc., son algunos de los aspectos que presentan las bases de datos tradicionales. Es muy importante para los manejadores de bases de datos (DBMS) soportar estas estructuras. El propósito de estas estructuras es obtener un acceso eficiente a los datos.

En bases de datos distribuidas las estructuras de acceso no son una herramienta adecuada para un acceso eficiente, que es básicamente el principal problema. Es muy difícil construir y darle mantenimiento a este tipo de estructuras, debido a que no es muy conveniente "navegar" en una base de datos distribuida a nivel registro.

Un plan de acceso puede ser escrito por un programador o bien generarlo de manera automática con un optimizador. El escribir un plan de acceso es muy similar a programar una base de datos centralizada, en el sentido de que la programación específica está basada en la forma como se va a acceder la base de datos. Sin embargo, cuando existe un acceso de varias bases de datos en diferentes localidades ésta se debe efectuar a un nivel de grupos de registros, mientras que para ejecutar procesamiento local deberá ser un solo registro a la vez.

Existen varios problemas que deben ser resueltos en el diseño del optimizador que produce un plan de acceso de manera automática. Es muy conveniente dividir estos problemas en 2 categorías : "Optimizaciones Globales" y "Optimizaciones Locales". Las optimizaciones globales consisten en qué datos deben ser accedados y cuáles lugares y archivos deben ser transmitidos entre localidades. El principal parámetro de optimización para esta estrategia es el costo de comunicación aunque el costo de acceso a las bases de datos locales, también debe ser tomado en cuenta.

Las optimizaciones locales consisten en decidir cuándo ejecutar un acceso a las bases de datos locales de cada lugar. Los problemas que se presentan en la optimización local son típicos de las bases de datos no distribuidas.

3.1.5 Integridad, recuperación y control de concurrencia

En bases de datos, la integridad, recuperación y control de concurrencia se refieren a problemas muy diferentes. Sin embargo, están frecuentemente relacionados. La solución a todos los problemas consiste en suministrar transacciones. "Una transacción" es una "unidad atómica" (*Atomic Unit*) de ejecución, es decir, una secuencia de operaciones que se ejecutan totalmente o no se llevan a cabo."²

La atomicidad de una transacción tiene que ver en gran medida con la integridad de las bases de datos, porque así se asegura que todas las acciones que transforman la base de datos sean consistentes en todos los lugares que se llevan a cabo.

²Cori S., Pelagatti G. Distributed Databases Principles & Systems. Ed. McGraw-Hill 1985

Existen dos grandes problemas en la atomicidad de una transacción : fallas y concurrencia. Las fallas pueden detener el sistema en medio de la ejecución de una transacción, lo cual viola los requerimientos de la atomicidad. La ejecución de diferentes transacciones puede permitir que una de las transacciones que se ejecuten sea inconsistente.

La recuperación puede llegar a causar grandes problemas en la atomicidad de una transacción en la presencia de una falla. En bases de datos distribuidas este aspecto es muy importante sobre todo, cuando alguno de los puntos o lugares involucrados llegan a fallar. El problema de la recuperación se analizará en el punto 3.6 del capítulo III.

El problema de control de concurrencia puede ser visto como un problema típico de sincronización. En base de datos distribuidas, la sincronización es aún más difícil que en sistemas centralizados.

3.1.6 Privacidad y seguridad

En las bases de datos tradicionales el administrador tiene un control centralizado, el cual puede asegurar que el acceso a los datos sea en forma autorizada. Sin embargo, la centralización es más vulnerable a violaciones de seguridad y privacidad.

En bases de datos distribuidas, los administradores locales tienen los mismos problemas que un administrador tradicional. Sin embargo, existen dos aspectos muy importantes : Existe un alto grado de autonomía para cada lugar, los propietarios de cada localidad se sienten más protegidos porque ellos pueden imponer sus propias protecciones, además de depender de un administrador central. Los problemas de seguridad son intrínsecos a los sistemas distribuidos en general, porque la comunicación puede representar una desventaja con respecto a la protección.

3.1.7 Coordinación entre localidades

Una de las principales desventajas de las bases de datos distribuidas es la mayor complejidad que se requiere para lograr una coordinación adecuada entre las diferentes localidades.

El incremento de tal complejidad se refleja básicamente en el costo de desarrollo de software, puesto que es más difícil estructurar un sistema distribuido. Existe además una gran posibilidad de errores. Como las localidades o nodos operan en paralelo, es más difícil garantizar que los algoritmos sean correctos y los errores pueden llegar a ser de lo más sutiles.

3.2 Estructura de las bases de datos distribuidas

Un sistema de bases de datos distribuidas consiste en un conjunto de localidades, donde cada una de ellas mantiene una base de datos local. Cada localidad puede procesar transacciones locales, que son aquellas que accesan información que reside en esa localidad. Además, una localidad puede participar en la ejecución de transacciones globales. Es decir, aquellas que accesan información de varias localidades. La ejecución de una transacción global requiere comunicación entre las localidades que interactúan.

Las localidades del sistema se pueden conectar físicamente por medio de diferentes topologías como son : estrella, anillo, línea (ver capítulo II) donde cada una de ellas va de acuerdo a las siguientes características :

- Costo de instalación : El costo de conectar físicamente a las localidades.
- Costo de comunicación : El costo en tiempo y dinero que implica la comunicación de dos o más localidades.
- Confiabilidad : Cuál es la frecuencia de fallas de comunicación o una localidad.
- Disponibilidad : La posibilidad de acceder información a pesar de fallas en las localidades y en las líneas de comunicación.

Las localidades en un sistema de base de datos distribuidas pueden llegar a estar geográficamente muy separadas, es decir, a cientos o miles de kilómetros (*red de largo alcance*) o bien, pueden ocupar una área reducida (*red local*).

Un sistema de base de datos distribuidas que se implementa en una red local, tiene mayores ventajas que un sistema que se implementa en una red de largo alcance. Básicamente como todas las localidades en una red local están próximas entre sí, las líneas de comunicación son de más alta velocidad y la tasa de errores es menor que en una red de largo alcance. En una red de largo alcance las líneas de comunicación son generalmente a través de líneas telefónicas que son relativamente muy lentas. Se utilizan también microondas y canales de satélite lo cual llega a resultar muy costoso. Los medios de transmisión que se utilizan comúnmente para efectuar las conexiones en una red local son: cable coaxial, par torcido y fibras ópticas.

3.2.1 Arquitectura de una base de datos distribuida

La figura 3.0, muestra los niveles conceptuales más importantes en la arquitectura de cualquier base de datos distribuida. En el nivel más alto se localizan el "*Esquema Global*" (*Global Esquema*) el cual define todos los datos que

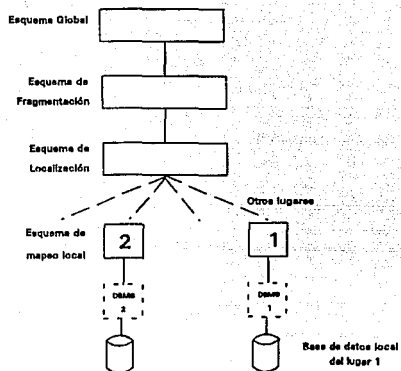


Figura 3.0. Arquitectura de una base de datos distribuida.

contiene la base de datos distribuida. Una de las ventajas del esquema es, que puede ser definido como si la base de datos no fuera distribuida. Sin embargo, el modelo de datos que se va a utilizar deberá estar perfectamente definido para poder efectuar el mapeo de los otros niveles de la arquitectura. Si el modelo de datos que se utiliza es por ejemplo; el modelo relacional, entonces la definición del esquema global consistirá de un conjunto de "Relaciones Globales" (*Global Relations*).

Cada "Relación Global" puede ser dividida en varias porciones que no se traslapen, llamadas Fragmentos. Al mapeo que existe entre *relaciones globales* y los fragmentos se le llama "Esquema de Fragmentación" (*Fragmentation Schema*). Las relaciones de una a muchos y varios fragmentos, pueden corresponder a una *relación global*. Un fragmento se indica con el nombre de la *relación global* y un índice (índice del fragmento), por ejemplo; R_1 indica el fragmento 1 de la *relación global* R .

Los fragmentos son porciones lógicas de relaciones globales, las cuales están físicamente localizadas en una o varias localidades de la red. El "Esquema de Localización" (Allocation Schema) define el o los lugares donde se localiza un fragmento. El tipo de mapeo que se define en este nivel, determina cuando la base de datos distribuida es redundante o no redundante, de tal manera que para un mapeo de uno a muchos tendremos redundancia y si se realiza un mapeo de uno a uno se elimina la redundancia. Los fragmentos de la relación global (R) se localizan siempre en el mismo lugar (j) lo cual constituye una "Imagen Física" (Physical Image) de la relación global R en (J). Puede existir además un mapeo donde una imagen física exista en otro lugar. Por ejemplo; el fragmento R_1 de la relación global R puede existir en la localidad 1 y 2. Para distinguir estos dos fragmentos se utiliza un superíndice. Por ejemplo; R^J el cual indica la imagen física de la relación global R en el lugar (J). Donde el fragmento R_1 de la relación R existe o tiene una imagen física en la localidad 2.

La figura 3.1, ejemplifica las relaciones mencionadas en el párrafo anterior. Existe una relación global (R) separada en cuatro fragmentos R_1, R_2, R_3, R_4 . Estos fragmentos están bajo un esquema de localización redundante en tres lugares diferentes de la red donde en cada lugar existe una imagen física R^1, R^2, R^3 .

Por último, en el nivel más bajo existen los manejadores de bases de datos que manipulan las imágenes físicas. A este mapeo se le conoce como "Esquema Local de Mapeo" (Local Mapping Schema).

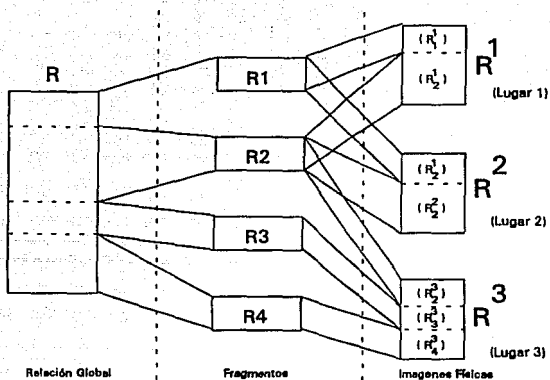


Figura 3.1. Fragmentos físicos de una relación global.

3.3 Tipos de fragmentación

La descomposición de una *relación global* en fragmentos se puede llevar a cabo al aplicar dos tipos de fragmentación :

- Fragmentación horizontal
- Fragmentación vertical

La fragmentación horizontal divide la *relación global* (R) asignando cada tupla de R a uno o más fragmentos. La fragmentación vertical divide la *relación global* de una manera más especial.

En cada tipo de fragmentación, los fragmentos serán definidos por medio de álgebra relacional.

Existen además, reglas que se deben seguir al definir un fragmento :

- Condición de complementación .- Los datos de una relación global deben ser mapeados en fragmentos.
- Condición de reconstrucción .- Siempre será posible reconstruir una relación global a partir de sus fragmentos.
- Condición de separación .- Se ha comentado que es conveniente separar los fragmentos para que la duplicidad de los datos sea controlada por el esquema de localización, esta condición es de gran ayuda, básicamente en la fragmentación horizontal mientras que la fragmentación vertical rara vez permite que esta condición sea violada.

3.3.1 Fragmentación horizontal

La *relación global* (R) se divide en varios subconjuntos $R_1, R_2, R_3, \dots, R_n$. Cada subconjunto consta de varias tuplas de la *relación global* (R). Esto es de gran ayuda en las bases de datos distribuidas, puesto que un subconjunto puede contener datos que tengan propiedades geográficas comunes. Cada tupla de la *relación global* (R) debe pertenecer a uno de los fragmentos, de tal forma que pueda reconstruirse la *relación global* original.

Un fragmento se define como una selección de la *relación global* (R). Es decir, se utiliza un predicado P_i para construir el fragmento :

$$R_i = \sigma_{P_i}(R)$$

La reconstrucción de la selección se calcula al efectuar la Unión de todos los fragmentos :

$$R = \bigcup_{i=1}^n R_i$$

3.3.2 Fragmentación vertical

La fragmentación vertical de una *relación global* $R(r)$ es la subdivisión de sus atributos en grupos $(r_1, r_2, r_3, \dots, r_n)$. Los fragmentos se obtienen al proyectar la *relación global* sobre cada grupo. Cada fragmento R_i de R se define :

$$R_i = \Pi r_i (R)$$

Los diferentes grupos que pueden ser generados, son de gran ayuda en bases de datos distribuidas donde cada uno de éstos puede contener atributos que poseen datos con propiedades geográficas comunes. La fragmentación es correcta si cada atributo es mapeado al menos en un atributo de un fragmento. Es posible reconstruir la *relación global* original a partir de los fragmentos realizando el producto natural.

$$R = R_1 \times R_2 \times R_3 \times R_n$$

De manera más general, la fragmentación vertical se lleva a cabo al agregar un atributo especial llamado *id-tupla* a la *relación global*. Un *id-tupla* es una dirección física o lógica de una tupla. Como cada tupla en R debe tener una dirección física única, el atributo *id-tupla* es una llave de la *relación global*.

Finalmente, al considerar el problema de la separación de fragmentos, observaremos que el atributo *id-tupla* o llave deberá estar repetida en todos los fragmentos para así poder permitir la reconstrucción. El tener este tipo de atributo hace que la separación de fragmentos no sea tan importante como en la fragmentación horizontal. Por ejemplo; si incluimos un mismo atributo en dos fragmentos verticales, podremos conocer cuáles datos están duplicados y en qué columna.

3.3.3 Fragmentación mixta

Dado que los fragmentos, son obtenidos al aplicar las dos operaciones de fragmentación (*horizontal y vertical*) sobre una *relación global*, es posible aplicar estas operaciones en forma recursiva.

Por ejemplo; una *relación global* (R) que se fragmenta verticalmente por primera vez (R_1) puede ser fragmentada en seguida en forma horizontal (R_2) o bien viceversa. La reconstrucción se obtiene al aplicar las reglas en forma inversa.

3.4 Transparencia de los datos en la red

"Es esencial que el sistema reduzca al mínimo la necesidad de que el usuario se de cuenta de como está almacenada la *relación global*"³. La transparencia de los datos en la red, se relaciona de manera directa con la autonomía local. Tal transparencia de los datos en la red, es el grado hasta el cual los usuarios del sistema pueden ignorar los detalles del diseño del sistema distribuido.

³ Korth Henry F., Silberchatz . Fundamentos de bases de datos.
Ed McGraw-Hill 1987

No es muy conveniente que los usuarios se refieran a una copia en específico de un elemento de información. El sistema deberá ser el que determine cuál copia se debe acceder cuando se solicite una modificación. De manera similar, los usuarios no deben conocer cómo está fragmentada la información.

Puede resultar difícil hacer transparente la base de datos para los usuarios que la actualizan, que para los usuarios que sólo la consultan. El problema que se presenta, es asegurar que todas las copias y fragmentos sean actualizados.

3.4.1 Niveles de transparencia

Es muy importante hacer notar la diferencia que existe entre los conceptos de fragmentación de datos y localización de datos, puesto que éstos nos permitirán distinguir dos de los principales niveles de transparencia de datos :

Los principales niveles de transparencia de datos son :

- Fragmentación transparente (*Fragmentation Transparency*).
- Localización transparente (*Location Transparency*).
- Mapeo local transparente (*Local Mapping Transparency*).

La fragmentación es el grado más alto de transparencia que muestra cómo los usuarios o programas de aplicación trabajan con las *relaciones globales*.

La localización transparente es el grado más bajo de la transparencia y requiere que el usuario o los programas de aplicación trabajen con los fragmentos además de las *relaciones globales*.

El mapeo local transparente es la independencia que existe entre cada manejador de bases de datos. Por ejemplo, en un sistema homogéneo de bases de datos distribuidas, es posible que el esquema independiente de cada lugar sea definido usando el mismo modelo de datos como el manejador de bases de datos, lo cual reduce la complejidad del mapeo local.

Existe otro tipo de transparencia llamado "*Reproducción Transparente*" (*Replication Transparency*). La reproducción significa que el usuario ignora la duplicidad de fragmentos, sin embargo, cuando se trabaja con una copia en particular, el sistema se encarga de ejecutar las acciones necesarias sobre las otras copias.

3.4.2 Transparencia para aplicaciones sólo de lectura

Para una aplicación sólo de lectura, los niveles de transparencia que se presentan son básicamente los tres que se mencionaron en el inciso anterior. Por ejemplo, supongamos que desde una estación de trabajo en una red local de micros se desea buscar el nombre de una persona especificando sólo un número de identificación, una vez que el sistema lo encuentra, éste será desplegado en la estación de trabajo. A través de este ejemplo podremos observar cómo se van decrementando los niveles de transparencia.

Nivel 1 .- Fragmentación transparente.

Como primer paso, el programa aplicativo acepta desde una terminal el número de identificación, paso seguido accesa la base de datos en cualquiera de los 3 lugares (Fig. 3.2). El lugar de acceso lo determina el sistema.

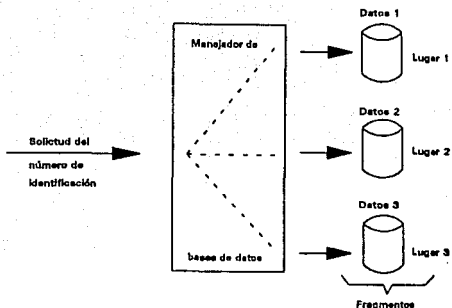


Figura 3.2. Fragmentación Transparente.

Como se podrá observar la aplicación es completamente inmune a cualquier cambio que se pueda aplicar al esquema global.

Nivel 2 .- Localización transparente.

En este nivel la búsqueda del nombre se efectúa utilizando un fragmento. Por ejemplo, se puede iniciar con el fragmento número 1, y si el manejador de base de datos regresara un número negativo, éste nos indicaría que no fué encontrado el nombre, paso seguido la búsqueda tomaría el fragmento número 2 y buscaría el nombre (Fig. 3.3). Cabe mencionar que se le pueden dar diferentes alternativas a la búsqueda.

Como se observa, la aplicación es claramente independiente de los cambios en el esquema de localización, pero no de los cambios en el esquema de fragmentación, debido a que la aplicación ignora cuales son las copias que existen de cada fragmento. Además de que las copias se pueden cambiar de un lugar a otro y también se permite la creación de nuevas copias sin afectar las aplicaciones.

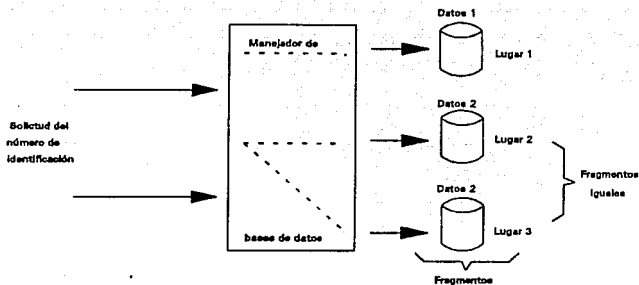


Figura 3.3. Localización Transparente.

La localización transparente puede llegar a ser muy eficiente siempre y cuando se conozca la estructura de los fragmentos.

Nivel 3.- Mapeo local transparente

En este nivel se asume que la aplicación se refiere a objetos que usan nombres los cuales son independientes de los sistemas locales individuales; sin embargo, se tiene que especificar en que lugar reside el objeto figura 3.3.

El aspecto más importante en este nivel son las primitivas que se van a usar para el acceso de la base de datos local, sobre todo si en el sistema existen varios manejadores de bases de datos.

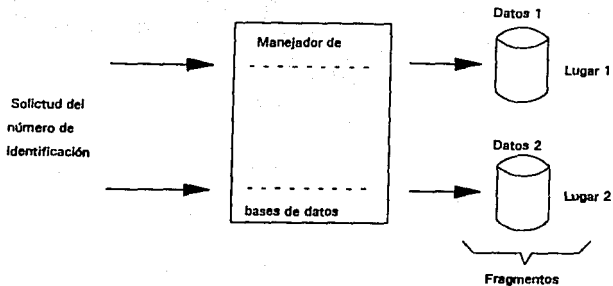


Figura 3.4. Mapeo Local Transparente.

3.4.3 Transparencia para aplicaciones de lectura y escritura.

El principal problema de una transacción de actualización es garantizar su atomicidad. Es decir, la transacción se debe ejecutar totalmente y no quedar incompleta.

Los mismos niveles de transparencia que existen en aplicaciones sólo de lectura pueden ser considerados para aplicaciones de escritura o actualización. Sin embargo, una actualización se debe ejecutar en todas las copias mientras que una lectura se puede llevar a cabo sobre una sola copia.

Si el manejador de base de datos no permite la localización y reproducción transparente, entonces los programas de aplicación tendrán que ejecutar todas las actualizaciones necesarias.

Nivel 1.- Fragmentación transparente.

En este nivel los programas de aplicación ejecutan las actualizaciones como si la base de datos no fuera distribuida. Es además responsabilidad del sistema, el ejecutar todas las operaciones que implícitamente requieren tanto, el esquema de fragmentación como de localización.

Nivel 2.- Localización transparente.

En este nivel los programas de aplicación tienen que acceder explícitamente los fragmentos en forma global.

Nivel 3.- Mapeo local transparente.

En este nivel la aplicación tiene que trabajar explícitamente con los fragmentos locales.

3.5 Autonomía de los datos

"La autonomía local es el grado hasta el cual el diseñador o administrador de una localidad puede ser independiente del resto del sistema"⁴. En cada uno de los elementos de información de cada localidad de un sistema de bases de datos distribuidas, se debe asegurar no utilizar el mismo nombre para dos datos diferentes.

Existen varias soluciones a este problema, una de estas es tener un asignador central de nombres, el cual en un momento dado puede representar una gran desventaja.

⁴ Korth, Henry F. Silberchatz. Fundamentos de bases de datos. Ed. McGraw-Hill 1987.

- El asignar se puede volver un cuello de botella.
- Si el asignador dejara de funcionar, posiblemente detendría a todas las localidades del sistema.
- Se reduce la autonomía local, ya que todo depende de un punto central.

Otra solución es exigir que cada localidad ponga como prefijo su identificador de localidad a cualquier nombre que genere. Con esto se logra garantizar que dos localidades nunca generarán el mismo nombre, además de no requerir de un control central.

Esta solución resuelve el problema de asignación de nombres y de autonomía local pero no de transparencia de la red, ya que se han agregado identificadores de localidad a los nombres. Por ejemplo, si ahora tenemos una relación llamada XXX ahora al agregar el identificador de localidad se llamará :

LOCALIDAD_NUM.XXX

LOCALIDAD_NUM = Identificador de localidad.

XXX = Relación.

Para resolver este problema, cada copia y fragmento debe tener un nombre único. Es muy importante que el sistema pueda identificar cuáles copias son copias del mismo elemento de información. Por ejemplo, si se adopta la convención de agregar ".f1", ".f2", ... ".fn" a los fragmentos de un elemento de información y ".R1", ".R2", ... ".Rn" a las copias tendríamos :

LOCALIDAD_NUM.XXX.F8 ; R2

LOCALIDAD_NUM = Localidad
XXX = Relación
F8 = Fragmento
R2 = Copia

la cual se referiría a la copia 2 del fragmento y éste lo habrá generado la localidad (Localidad_Num).

3.6 Recuperación de sistemas distribuidos

"Una transacción es una unidad de programa cuya ejecución conserva la consistencia de la base de datos. Una transacción debe ejecutarse en forma atómica; es decir, o se ejecutan completamente todas las instrucciones de la transacción o no se ejecutan"⁵. En un sistema de bases de datos distribuidas es más difícil garantizar la propiedad de atomicidad de una transacción. Esto se debe a que es posible que varias localidades participen en la ejecución de una transacción. Si falla alguna de ellas o alguna línea de comunicación, el cálculo puede resultar erróneo.

La función del *manejador de transacciones* de un sistema de base de datos distribuidas es asegurar que se conserve la atomicidad entre las diferentes localidades del sistema. Cada localidad cuenta con su propio manejador local. Y los distintos manejadores cooperan para manejar las transacciones globales.

⁵ Davenport R.A. "The state of the art of distributed databases". North-Holland Publishing company 1978.

Cada una de las localidades del sistema contiene dos subsistemas : un manejador de transacciones y un coordinador de transacciones.

Manejador de transacciones.- Su función es mapear la ejecución de aquellas transacciones (o subtransacciones que accesan datos almacenados en esa localidad).

Coordinador de transacciones.- Su función es coordinar la ejecución de las diversas transacciones (tanto locales como globales) que se iniciaron en esa localidad.

Cada uno de los manejadores de transacciones se encarga de :

- Mantener una bitácora para la recuperación.
- Participar en un esquema de control de concurrencia apropiado para coordinar la ejecución en paralelo de las transacciones que se ejecutan en esa localidad.

Un sistema distribuido puede llegar a tener las mismas fallas que un sistema centralizado (por ejemplo; aterrizaje de la unidad de disco, pérdida de memoria, fallas en el suministro de la energía, etc.) En una configuración distribuida es necesario prever otros tipos de fallas como son :

- La falla total de una localidad.
- Interrupción en una línea de comunicación.
- Pérdida de mensajes.
- Fragmentación de la red.

Para que el sistema se pueda recuperar de cualquier falla, éste se debe de poder reconfigurar de manera que todo el procedimiento se pueda reanudar y que se recupere una vez que el sistema haya sido reparado.

A veces no es posible distinguir cuándo sucede una de las fallas mencionadas anteriormente. Por lo general es posible detectar que existe una falla, pero resulta difícil identificar de que tipo se trata.

Por ejemplo, si una localidad (X) se da cuenta de que existe una falla, en ese momento se debe iniciar el procedimiento de reconfiguración del sistema para que se le permita continuar con sus operaciones normales.

- "Si en la localidad que está fuera de servicio se almacena información repetida, debe actualizarse el catálogo de manera que las consultas no hagan referencia a la copia que se encuentra en dicha localidad.
- Si en el momento de la falla existían transacciones activas en la localidad que quedó fuera de servicio deben abortarse.
- Si la localidad que quedó fuera de servicio es el distribuidor central de algún subsistema, es preciso elegir un nuevo distribuidor. Los distribuidores pueden ser por ejemplo; asignadores de nombres, coordinadores de concurrencia o detectores de paralización global".⁶

El esquema de reconfiguración que se adopte debe estar planteado para que funcione correctamente, aún cuando la red quede fragmentada. Es necesario que se eviten situaciones como :

- Elección de dos o más distribuidores centrales en un fragmento dado de la red.
- Actualización de un dato en más de un fragmento de la red.

⁶ Korth Henry F. Silbershatz Abraham. Fundamentos de bases de datos. Ed. Mc-Graw Hill 1987.

Se debe tener mucho cuidado al reintegrar una localidad o una línea de comunicación. Cuando se trata de una localidad se debe iniciar un procedimiento de actualización de sus tablas de sistema, para que reflejen los cambios que tuvieron lugar mientras estaba inactiva. Si la localidad tenía copias de datos, se deben obtener los valores de todas las copias. Esto es más difícil de lo que parece, ya que es posible que se actualicen los datos que se están procesando mientras el sistema se recupera.

Una solución es por ejemplo, suspender todas las operaciones del sistema temporalmente mientras que la localidad que quedó fuera de servicio se reintegre a él. Esto en muchos casos no puede aceptarse puesto que acarrea bastantes problemas y es preferible representar a las tareas de recuperación como una serie de transacciones. En este caso el subsistema de control de concurrencia y el manejador de transacciones pueden encargarse de realizar de manera confiable la reintegración de la localidad.

Si se recupera una línea de comunicación, es muy posible que se unan de nuevo dos fragmentos de la red. Es muy importante informar cuando una línea se ha restablecido ya que muchas de las operaciones entre localidades de la red se pueden restablecer. Esto se puede realizar al enviar un mensaje a todas las localidades.

3.7 Control de concurrencia distribuido

"Uno de los conceptos más importantes en los sistemas modernos es sin duda el de *multiprogramación*. Si se ejecutan varias transacciones al mismo tiempo, puede compartirse procesador entre ellas"⁷. Este esquema mejora la eficiencia total del sistema puesto que se trabaja más en una menor cantidad de tiempo. Los beneficios que se obtienen de la multiprogramación son básicamente, un mejor aprovechamiento del procesador y además la productividad total de las transacciones se eleva. La productividad es la cantidad de trabajo que se realiza en un intervalo de tiempo dado. Existen 2 esquemas que son complementarios a la multiprogramación y son los sistemas por lote (Batch) y los sistemas interactivos (Sistemas de tiempo compartido).

Como se observa, en un esquema de multiprogramación es posible ejecutar varias transacciones de manera concurrente. Sin embargo, es necesario que el sistema controle la interacción entre las transacciones concurrentes para evitar que destruyan la consistencia de la base de datos. Este control se logra por medio de varios mecanismos a los que se le denomina esquema de control de concurrencia. Se analizarán dos esquemas : el primero llamado "TimeStamps" y el segundo "Abrazos Mortales" (Deadlocks).

Cuando dos o más transacciones accesan simultáneamente una base de datos, ésta puede llegar a ser inconsistente. Existen dos tipos de conflictos en la concurrencia las cuales se generan cuando las transacciones crean simultáneamente nuevas versiones.

⁷ Gardarin Georges. " Integrity, consistecy, concurrency, reliability in distributed database managment systems". Distributed Databases North-Holland Publishing Company, INRIA 1980.

- 1) Pérdida de operaciones cuando una nueva versión es creada por una transacción que utiliza una versión vieja.
- 2) Existe inconsistencia en los datos cuando la integridad de la base de datos es constantemente violada.

La ejecución de transacciones simultáneas deberá estar limitada, para así prevenir la pérdida de operaciones e inconsistencia de los datos.

3.7.1 Control de concurrencia basado en horas entrada (TimeStamp)

En un sistema distribuido algunas veces es necesario saber si el evento A, en algún lugar ocurre antes o después del evento B, en un lugar diferente. Determinar el orden de los eventos es simple en un sistema centralizado, puesto que es posible usar el mismo reloj que determine qué evento ocurre en un tiempo dado. En un sistema distribuido no se puede llegar a tener perfectamente sincronizados los relojes de cada lugar.

Varios algoritmos para el control de concurrencia distribuido y prevención de abrazos mortales (*DeadLocks*) necesitan establecer el orden de aparición de cada evento. La determinación del orden de los eventos consiste en asignar a cada evento A, el cual ocurre en un sistema distribuido una hora entrada $TS(A)$ que tiene las siguientes propiedades.

- 1) $TS(A)$ sólo identifica al evento A (diferentes eventos tienen diferentes horas entradas).
- 2) Para cualquiera de los dos eventos A y B si A ocurre antes que B entonces $TS(A) < TS(B)$.

La principal inconveniencia de la definición anterior es el significado de la relación "ocurre antes" que no está precisamente definida si los dos eventos A y B ocurren en dos diferentes lugares, puesto que no se tiene un "reloj global" para medir el tiempo exacto en el que los eventos hayan ocurrido.

Una definición más precisa de la relación "ocurre antes" es la siguiente de acuerdo con Ceri Stefano:

"El evento A ocurre antes que el evento B en el lugar J " de acuerdo al significado de la expresión se puede determinar el orden de ocurrencia de un evento en un sólo lugar. La relación "ocurre antes" se denota como : \rightarrow la cual puede ser generalizada para un medio ambiente distribuido con las siguientes reglas :

- 1) Si A y B son dos eventos en el mismo lugar y A ocurre antes que B entonces $A \rightarrow B$.
- 2) Si el evento A consiste en enviar un mensaje y el evento B consiste en recibir el mismo mensaje entonces $A \rightarrow B$.
- 3) Si $A \rightarrow B$ y $B \rightarrow C$ entonces $A \rightarrow C$.⁸

Dos eventos A y B se les puede llamar pseudosimultáneos si $A \rightarrow B$ o $B \rightarrow A$. Por ejemplo, en la figura 3.5, los eventos A y D , B y E , C y F son pseudosimultáneos. La relación exacta de dos eventos pseudosimultáneos no puede ser determinado y no es relevante.

La segunda propiedad se puede establecer de acuerdo con Ceri Stefano de la siguiente manera :

- 2) Para cualquiera de los dos eventos A y B si :
 $A \rightarrow B$ entonces $TS(A) < TS(B)$.

⁸ Ceri S. & Pelagatti G. Distributed databases Principles & Systems. Ed. McGraw-Hill 1985.

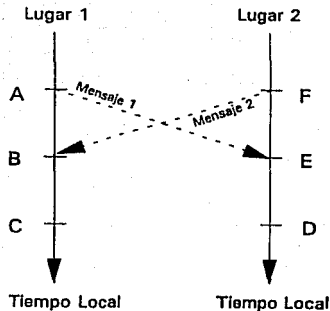


Figura 3.5. Relación "Ocurre Antes" en un sistema de dos localidades.

De la gráfica anterior si A y B son pseudosimultáneos, es posible que $TS(A) < TS(B)$ o que $TS(B) < TS(A)$. La asignación de horas entradas consiste en la determinación de un orden total, el cual consiste con la definición de orden parcial definida por la relación \longrightarrow . Se puede establecer que el evento A es más joven (viejo) que el evento B si $TS(A) > TS(B)$ ($TS(A) < TS(B)$).

Para establecer correctamente la generación de horas entrada. Se deben satisfacer dos requerimientos principalmente : El primero es que cada localidad añada a una hora entrada única local su identificador en la posición de menor significado. Al usar la posición de menor significado además de la de mayor significado se tiene la posibilidad de que todas las horas entrada que son generadas por una localidad, sean más grandes que todas las horas entradas que son generadas por otros lugares.

El segundo requerimiento es más complejo de satisfacer. Cada localidad tiene un contador el cual se incrementa constantemente, ésto es las transacciones que reciben. Las horas entrada están correctamente ordenadas entre ellas mismas. El contador incrementa cada vez que se genera una hora entrada. Sin embargo, la sincronización entre los contadores de las diferentes localidades resulta difícil bajo este segundo requerimiento. Es posible que el contador en una localidad, por ejemplo; la 1 sea usada para generar más horas entradas y que el contador en una localidad diferente, por ejemplo; 2 se utilice para un manejo local de horas entrada, por lo que el 1 avanzará mucho más rápido.

Afortunadamente, los contadores de dos lugares se pueden mantener alineados simplemente al incluir en cada mensaje el valor del contador. Si una localidad recibe un mensaje con una hora entrada entonces el contador se deberá incrementar $TS + 1$. De esta manera 2 contadores tratan de ayudar a establecer una sincronización aproximada.

Si se considera la figura 3.5 y asumimos que el contador en la localidad 1 es inicialmente 0 y el contador de la localidad 2 es inicialmente 10 y representamos la hora entrada generada en el lugar, cuando el contador local tiene un valor X como $\langle X, Y \rangle$. Entonces :

$$TS(A) = \langle 0, 1 \rangle$$

$$TS(D) = \langle 10, 2 \rangle$$

Los eventos A y D son psudosimultáneos y su orden por lo tanto es arbitraria. Cuando el mensaje $M2$ llega a la localidad 1, lleva la hora entrada de D . De aquí que el contador local de la localidad 1 se incrementa en :

$$TS(B) = \langle 11, 1 \rangle$$

Cuando el mensaje $M1$ llega a la localidad 2, éste lleva una hora entrada más pequeña que la de la localidad 2, el contador de la localidad 2 simplemente se incrementa :

$$TS(E) = \langle 11, 2 \rangle$$

Se puede notar que $TS(E)$ es distinta de $TS(B)$ sólo en el identificador de localidad.

Los dos eventos pseudosimultáneos han obtenido horas entrada cercanas.

Se puede apreciar que siempre deben ser usados contadores y no relojes en la implementación de horas entrada. Sin embargo, en algunas aplicaciones puede ser conveniente usar relojes además de contadores.

3.7.2 Abrazos mortales distribuidos o puntos muertos

"Un abrazo mortal ocurre si dos o más transacciones solicitan de manera mutuamente exclusiva los mismos recursos"⁹. Por ejemplo; la transacción A se inicia y coloca una llave sobre el archivo A mientras que la transacción B se inicia y coloca una llave sobre el archivo B . En seguida la transacción A solicita el archivo B y la transacción B solicita el archivo A . Como se observa tanto el archivo A como el B tienen llaves. De aquí, se puede concluir que ninguna puede completarse debido a que ninguna transacción va a liberar su archivo, por lo tanto se creó un bucle (*Loop*) infinito. Este tipo de situaciones se pueden prevenir utilizando esquemas de detección. Los esquemas de detección y prevención examinan todos los requerimientos de todos

⁹ Liebowitz Burth H. Carson John H. "Distributed Processing. Second Edition 1978. IEEE Catalog No. EHO127-1.

los procesos, lo cual dá como resultado que un proceso puede continuar únicamente si un abrazo mortal fué detectado.

En base de datos distribuidas, el problema de abrazos mortales resulta más complicado, debido a que se tienen que coordinar varias computadoras.

La resolución de abrazos mortales involucra la selección de una o más transacciones que van a ser abortadas y restauradas. Cuando una transacción es abortada, ésta libera sus recursos y así otras pueden continuar. El criterio que se debe tomar para seleccionar la transacción que va a ser abortada deberá de tratar de minimizar los costos de esta operación. Existen varios criterios como son :

- Abortar la transacción más joven.
- Abortar la transacción con menos recursos.
- Abortar la transacción que tiene el costo más barato.
- Abortar la transacción que tiene más tiempo de espera.

La redundancia que a menudo se presenta en un sistema de bases de datos distribuidas incrementa la probabilidad de abrazos mortales.

Existen varios métodos para prevenir los abrazos mortales :

- Detección de abrazos mortales usando control centralizado o jerárquico.
- Detección de abrazos mortales distribuidos.
- Prevención de abrazos mortales.

3.7.2.1 Detección de abrazos mortales utilizando control centralizado jerárquico

El detector de abrazos mortales distribuidos tiene la responsabilidad de construir las gráficas de espera distribuidas y descubrir bucles en ella. El detector recibe información de todas las localidades. Además, en cada localidad existe un detector de abrazos mortales, el cual tiene la responsabilidad de detectar todos los abrazos mortales potenciales que existen en su localidad.

El detector global de abrazos mortales recibe los mensajes de todas las localidades, construye la gráfica de espera, busca los bucles y selecciona las transacciones que van a ser abortadas. Esta operación se realiza periódicamente o cada vez que existe un cambio en los bucles potenciales de abrazos mortales.

La detección de abrazos mortales centralizada es muy simple, pero tiene 2 principales desventajas : Primero; es muy vulnerable a fallas, sobre todo donde se establece el detector centralizado. Segundo; requiere de costos de comunicación muy elevados sobre todo si el punto central se encuentra muy alejado de una localidad.

En el método jerárquico se construye un árbol con los detectores de abrazos mortales. En los niveles del árbol se tienen detectores locales de abrazos mortales (*LDD Local deadlock detected*) y en los siguientes niveles se tienen detectores no locales de abrazos mortales (*NLDD No Local deadlock detected*). La figura 3.6 muestra el manejo de detectores jerárquicos de abrazos mortales para un sistema que consta de 5 localidades.

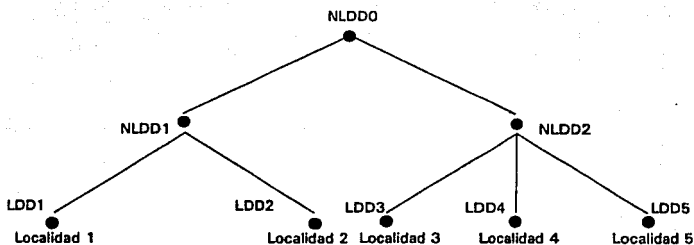


Figura 3.6. Arbol de detectores de abrazos mortales.

Cada detector local (*LDD1.. LDD5*) se comporta como un detector local de abrazos mortales del método centralizado : Determina los abrazos mortales locales y transmite información acerca de los posibles bucles al *NLDD* que se encuentra en el nivel inmediato superior. Cada detector no local de abrazos mortales ejecuta la detección de abrazos mortales que corresponden a su jerarquía, de esta manera se encarga de transmitir una gráfica reducida de los posibles abrazos mortales, a los niveles superiores de la jerarquía. Por ejemplo, en la figura 3.6, los abrazos mortales que involucran *LDD3*, *LDD4* y *LDD5* son descubiertos por *NLDD2*, mientras que un abrazo mortal que involucra *LDD2* y *LDD3* sólo es detectado por *NLDD0* (El nivel más alto).

El rendimiento de este método depende de como se seleccione la jerarquía.

3.7.2.2 Detección de abrazos mortales distribuidos

En el mecanismo de detección de abrazos mortales distribuidos no existe distinción entre detectores locales y no locales de abrazos mortales. Cada localidad tiene la misma responsabilidad. Las localidades intercambian información acerca de la transacciones y así determinan un abrazo mortal general.

Este mecanismo se basa en la transmisión de información acerca de los bucles potenciales, la transmisión es parecida a la que realiza el algoritmo centralizado. Los bucles potenciales de abrazos mortales se detectan localmente con gráficas de espera; sin embargo, todos los puertos de salida y entrada se coleccionan únicamente en un nodo llamado (EX).

La principal diferencia entre la detección de abrazos mortales centralizados y distribuidos, es que en la primera, los bucles de abrazos mortales son enviados a una localidad, mientras que en la segunda, no existe tal localidad. Además, en la detección distribuida los detectores necesitan seguir una regla para determinar a que localidad van a enviar el bucle. Esta regla deberá minimizar la cantidad de información transmitida.

La regla que utiliza este mecanismo consiste en transmitir la información de posibles abrazos mortales " a través " del bucle de abrazos mortales. Por ejemplo, en la figura 3.7, los detectores locales de abrazos mortales han detectado un posible abrazo mortal que involucra a T1, T2, y EX.

La localidad 1 puede escoger que bucle de abrazo mortal va a enviar a las siguientes localidades :

- 1) A la localidad donde se este esperando por T1
- 2) A la localidad donde se este esperando por T2

La opción que se elija deberá ir hacia atrás del bucle (1) o hacia adelante del bucle (2).

Debido a que las localidades 1 y 2 son simétricas, se deberá evitar que el abrazo mortal sea descubierto un par de veces. Para evitar esto, el mecanismo se basa en la siguiente regla : " Los bucles potenciales de abrazos mortales son transmitidos solo si el identificador de la transacción para la transacción de la cual EX está en espera es mayor, que el identificador de la transacción para la transacción que espera EX."¹⁰ Al aplicar esta regla a la figura 3.7, nos daremos cuenta de que únicamente la localidad 2 puede transmitir el bucle potencial de abrazo mortal, porque en la localidad 2 EX está en espera de T2 y T1 está en espera de EX. En la localidad 1 se tiene la situación contraria, la cual no puede transmitir el bucle potencial de abrazo mortal.

Este mecanismo trabaja por iteraciones sucesivas. En cada iteración de una localidad se recibe información de los posibles abrazos mortales de otras localidades, esta información basada en lo posible abrazos mortales se procesa y se envían a todas las localidades del sistema.

Por cada iteración los detectores locales de cada localidad ejecutan las siguientes acciones :

- 1) Construyen gráficas de espera. Incluyendo al nodo EX.

¹⁰ Carl S. And Pelagatti G. Distributed databases. Principles & Systems. Ed McGraw-Hill 1985.

- 2) Por cada mensaje que se recibe, se efectúan las siguientes modificaciones a las gráficas de espera.
 - Cada transacción que se recibe en el mensaje es agregada a la gráfica de espera.
 - Para cada transacción con un mensaje que inicia con EX, se crean nuevas rutas para la siguiente transacción que llegue.
- 3) Se buscan los bucles que no incluyan al nodo EX.
- 4) Se buscan los bucles que incluyan al nodo EX. Estos bucles son abrazos mortales potenciales. Los cuales son transmitidos a las diversas localidades.

3.7.2.3 Prevención de abrazos mortales

Con la prevención de abrazos mortales una transacción es abortada y restablecida, si existe el riesgo de que un abrazo mortal pueda ocurrir. De esta manera, los abrazos mortales nunca ocurren y con la cual se elimina la detección y resolución de abrazos mortales.

La prevención de abrazos mortales se realiza de la siguiente manera : si una transacción T1 solicita un recurso el cual está siendo utilizado por otra transacción T2, entonces se debe recurrir a una "prueba de prevención"; si la prueba indica que existe riesgo de abrazo mortal, entonces a T1 no se le permitirá entrar en espera de ese recurso.

El método de prevención de abrazos mortales asegura que nunca ocurrirá un abrazo mortal, puesto que T1 puede ser abortada y restaurada (*método no preferente*), o T2 puede ser abortada y restaurada (*método preferente*).

El método no preferente para la prevención de abrazos mortales basado en horas entrada es el siguiente : si T_i solicita colocar una llave en un dato, el cual ya tiene una llave puesta por T_j , entonces T_i puede entrar en estado de espera si T_i es más vieja que T_j . Si T_j es más joven que T_i , entonces T_i abortará y será restaurada con la misma hora entrada. La idea principal de este método es que siempre es mejor restablecer la transacción más joven.

El método preferente trabaja con la regla opuesta : Si T_i solicita colocar una llave en un dato el cual tiene una llave puesta por T_j , entonces T_i puede entrar en estado de espera si es más joven que T_j . De otra manera, T_j es abortada y el recurso es concedido a T_i . La idea principal de este método es que siempre una transacción más joven va a esperar por los recursos de una más vieja.

3.8 Distribución de carga

El hecho de tener componentes de hardware de mismo tipo en una red, permite que el desempeño del sistema se incremente enormemente. En un sistema distribuido son muy claros los beneficios que se obtienen al almacenar los datos en la localidad que los usa frecuentemente.

La distribución de carga en las localidades de un sistema distribuido es una importante característica. Ya que ésta toma ventajas de las diferentes computadoras y recursos en cada localidad, además de maximizar el grado de paralelismo de la ejecución de las aplicaciones.

En la distribución de carga se deben tomar en cuenta las siguientes consideraciones :

Procesamiento local : Distribuir los datos para maximizar el procesamiento local, utilizando el simple principio de almacenar los datos tan cerca como sea posible de las aplicaciones que los usan.

Disponibilidad y confiabilidad : Deberá existir un alto grado de disponibilidad para aplicaciones sólo de lectura, sobre todo cuando se tienen múltiples copias de información. El sistema debe ser capaz de seleccionar una copia alterna cuando ésta no esté disponible bajo condiciones normales.

Confiabilidad : En cada una de la copias que existen de la misma información. Esto es para poderse recuperar de copias corruptas o dañadas físicamente. La destrucción física no solamente implica fallas en las líneas de comunicación o computadoras, sino también en destrucción por fuego, sabotaje, temblores, etc.

Costo de almacenamiento y disponibilidad : La distribución de la base de datos se deberá reflejar en los costos y disponibilidad del sistema. Es posible, que se tengan localidades especializadas en la red para almacenar datos, o a la inversa localidades que no soporten almacenamiento masivo. El costo del almacenamiento de datos no es muy relevante, comparado con los costos de CPU, I/O y transmisiones; sin embargo, debe ser considerado.

3.9 Estrategias de acceso

Existe un gran número de estrategias de acceso, que pueden ser utilizadas para consultar una base de datos. La selección de una estrategia puede hacerse empleando la información que exista en memoria, ya sea con poco o ningún acceso a disco. Hay que recordar que los accesos a disco son lentos, comparados con la velocidad del procesador central de una computadora.

A una consulta se le puede calcular su respuesta de varias maneras. Si por ejemplo, se utiliza un lenguaje de consulta *SQL*, está se puede expresar de varias maneras. Sin embargo, muchos de los usuarios no son capaces de escribir una estrategia eficiente. Así pues el sistema se debe encargar de transformar la consulta de la forma en que la escribió el usuario, a una forma equivalente que pueda calcularse de manera eficiente.

CAPITULO IV

Diseño y especificación de la base de datos distribuida para el sistema.

4.1 Diseño y selección del modelo de base de datos

Un buen diseño de una base de datos es un prerequisite muy importante para el desarrollo de aplicaciones. El diseño de una base de datos distribuida es una tarea muy complicada y difícil, puesto que implica el uso de muchos aspectos técnicos, así como también un enorme cambio en la organización o empresa donde se desea implantar una base de datos. Desde el punto de vista técnico, existen nuevos problemas, como son la interconexión de redes de computadoras y la óptima distribución en ella de los datos y aplicaciones de acuerdo con las necesidades de cada lugar. Desde el punto de vista organizacional, la descentralización es crucial debido a que muchos sistemas distribuidos sustituyen a grandes sistemas centralizados, lo cual nos indica que el distribuir una aplicación tiene un gran impacto en la organización.

Los objetivos de todas las bases de datos son los mismos. "El objetivo es básicamente implementar un firme control y utilización correcta de los datos. Estas mejoras van acompañadas del incremento de la disponibilidad e integridad de los datos"¹. La disponibilidad significa compartir los datos por múltiples usuarios. El mejorar la integridad de los datos es un requisito para que los usuarios confíen en su calidad y precisión.

¹ Bray H. Olin. "Distributed database design considerations". Trends and application 1976, Computer Networks IEEE.

Una base de datos posee basicamente 3 componentes :

- Información estructurada (Estructuras de datos) o esquemas. Estos describen qué datos están contenidos en la base de datos.
- Existen los datos como tales, el contenido de la base de datos.
- Varios programas o especificación de procesos, los cuales controlan las operaciones en la base de datos.

Existe otro componente estructural, el cual podría ser llamado un subesquema, en él se describe como las aplicaciones de los usuarios ven la base de datos.

Durante el desarrollo de este capítulo se verán algunos de los problemas que se presentan al diseñar una base de datos centralizada. Los principios de diseño que se aplican a una base de datos centralizada, también se aplican a una base de datos distribuida. El modelo de base de datos a utilizar para el diseño será el modelo relacional.

Se ha elegido el modelo relacional debido a que éste, en la última década a tenido una gran aceptación en la comunidad de bases de datos, desde su perspectiva histórica es relativamente reciente. Su principal ventaja es la simplicidad que presenta, es decir la facilidad de comprensión. Debido a que se puede representar como un conjunto de tablas. Ahora bien, aunque las tablas son un concepto simple e intuitivo, existe una correspondencia directa entre el concepto de una tabla y el concepto matemático de una relación.

Dos de los principales problemas que se presentan durante el diseño de una base de datos centralizada son :

- Diseño del "*esquema conceptual*". En él se describe la base de datos (Son todos los datos que utilizan los programas aplicativos).
- Diseño de la "*base de datos física*". Aquí se determina el mapeo del esquema conceptual a las áreas de almacenamiento, y los métodos de acceso apropiados.

En las bases de datos distribuidas, se presentan de igual manera dos problemas. Estos son; el diseño del esquema global o conceptual y el diseño de las bases de datos físicas que corresponden a cada localidad y que van a integrar el sistema distribuido. Las técnicas que se aplican para resolver los dos problemas anteriores, son las mismas que en bases de datos centralizadas. Sin embargo, la distribución de las bases de datos trae consigo dos nuevos problemas :

- Diseño de la fragmentación. Determinar como las relaciones globales son subdivididas en fragmentos verticales, horizontales o mixtos.
- Diseño de la asignación de fragmentos. Determinar como los fragmentos son mapeados a las imágenes físicas. De esta manera también se determina cuales son los fragmentos duplicados.

4.2 Especificación del sistema

Hoy en día los diversos ramos de la industria, empresas particulares e instituciones públicas y privadas están comenzando a utilizar recursos de cómputo en sus instalaciones con diferentes fines. Estos son :

- Integrar a la organización en un medio común. (Hablando tecnológicamente)
- Facilidad de manejo de información.
- Mejorar la productividad del personal.
- Etc.

Además de establecer modelos que ayuden a la administración, actualización y depuración de sus datos. Contemplando que las cantidades de información que se van a manejar, podrán ser enormes, para lo cual su manejo en forma manual se vuelve lento e insuficiente.

La tecnología de las bases de datos es una ciencia de la computación y la información que ha tenido un rápido desarrollo. La cantidad de información encomendada a las bases de datos se puede llegar a medir. Sin exagerar, ésta puede llegar a contener varios miles de millones de *bytes*, de aquí podemos concluir que muchas organizaciones dependen de la operación continua y eficaz de un sistema de base de datos.

Un sistema de base de datos es en esencia, un sistema de mantenimiento de registros basado en computadores, es decir, un sistema cuyo propósito general es registrar y mantener información.

El prototipo que se va a utilizar para implementar el sistema de base de datos distribuidas en un red local de micros, será controlar el flujo de volantes que existe en cada uno de los ramos de la Subdirección de Emisión Daños. Este prototipo es aplicable a cualquier compañía de seguros.

La Subdirección de Emisión Daños se divide en 5 ramos principalmente:

- Incendio .- Cubre todo lo relacionado a edificios, casas y en general a bienes-inmuebles. La cobertura puede ser en sus interiores como exteriores (Materiales de construcción, acabados, muebles, etc).
- Autos .- Cubre en general a camiones y automóviles.
- Transportes .- Cubre todos lo relacionado a barcos, aviones y en general a transportación de carga. Se puede asegurar además la carga que se va a transportar.
- RCG .- (Responsabilidad Civil General). Cubre daños a personas. Por ejemplo en un avión se aseguran todos los pasajeros que lo aborden.
- Diversos .- Cubre todos los ramos de la ingeniería y objetos personales (Computadoras personales, relojes, etc.)

Cada uno de los ramos maneja un número de volante, el cual circula a través de cada uno de los departamentos que integran el ramo. El número de volante es independiente en cada ramo, es decir, el número se puede repetir en todos los ramos.

El número de volante solo tiene uso interno en la compañía de seguros, ya que su finalidad es emitir una póliza a la persona o institución que desee asegurar sus bienes. Cabe mencionar, que una póliza puede ser de tres tipos : nueva, renovación o endoso.

Los endosos son movimientos que modifican los términos iniciales de una póliza. Existen tres tipos de endosos : A, B y D.

- Endoso B.- Modifica términos que no alteran sumas aseguradas o primas de pólizas. Por ejemplo, cambio de nombre del asegurado, domicilio, teléfono etc.
- Endoso A.- Se alteran sumas aseguradas y primas de la póliza para aumento. Por ejemplo, se tuvo un error al registrar un automóvil. El cual se dio de alta como austero y resulta que en realidad es de lujo y equipado.
- Endoso D.- Es la parte contraria de lo que establece el endoso de tipo A. Las sumas aseguradas y primas de la póliza se decrementan.

Cada ramo cuenta con diferentes departamentos, por donde circula el volante. Los departamentos con que cuentan todos los ramos son :

- Orden de Trabajo.- Es el departamento encargado de recibir solicitudes de todos los ramos, asignar un número de volante y canalizarlo adecuadamente al ramo que verdaderamente le corresponde.
- Cálculo.- El personal de éste departamento se encarga de tarificar los documentos (pólizas). Aquí, se determinan primas y comisiones.

- Registro.- Este departamento está encargado de registrar en libretas oficiales, el nombre de los asegurados, para así proporcionarles el número de póliza o endoso correspondiente.
- Reaseguro.- Este departamento se encarga de distribuir en los contratos las sumas aseguradas de cada documento (pólizas o endosos).
- Codificación 1.- Este departamento se encarga de agrupar los documentos en lotes para realizar cifras y así éstas puedan ser capturadas en el sistema automatizado (Ver UDS).
- UDS (*Unit Data Systems*).- En éste departamento se capturan los documentos (pólizas o endosos) de cada ramo, para que puedan ser contabilizados, además de procesar la impresión de pólizas y endosos. Cabe mencionar, que éste departamento cuenta con un sistema automatizado capaz de emitir (imprimir y contabilizar) pólizas o endosos a los clientes de la aseguradora.
- Codificación 2.- Este departamento de encarga de separar los lotes que fueron procesados en el departamento de codificación 1 para posteriormente enviarlos al departamento de mecanografía.
- Mecanografía.- Este departamento realiza la mecanografía de documentos (pólizas o endosos) que no pueden ser realizados a través del sistema automatizado.
- Revisión.-Este departamento se encarga de checar que el documento carezca de errores.

Existen tres departamentos, en donde la persona que procesa un volante puede resultar crítico. Estos tres departamentos son :

- Cálculo
- Mecanografía
- Revisión

Por cada volante que se procesa, existe un agente de seguros, el cual trata directamente con la persona o institución que desea adquirir un seguro.

Finalmente, cada volante que circula por los departamentos tiene la siguiente información.

- Número de volante.
- Fecha de ingreso o cuándo comenzó a circular por los departamentos (Orden de trabajo). Además de la fecha en la cual estuvo en un determinado departamento (Cálculo, Registro, etc).
- Número de póliza.
- De que tipo es : Póliza
Renovación
Endoso (A, B o D)
- Nombre de la persona que contrata el seguro.
- Dirección de la persona : Calle
Número del domicilio
Colonia
Estado
CP.
Teléfono
- Datos del agente : Clave del agente
Nombre del agente
Oficina
Estado
- Si el volante paso por uno de los tres departamentos críticos, entonces se debe registrar :
El nombre de la persona del departamento.
Clave de la persona que proceso el volante.
Departamento al que pertenece.

- El movimiento final que registraron cada uno de los volantes. Este movimiento puede ser de 5 formas diferentes :

Registrar el número de póliza o renovación.

Registrar el número de endoso.

Registrar el número de volante como : No Procede o cancelado.

Registrar un Memo Agente, el cual es un documento que afecta datos internos de la empresa (Clave de agente nombre de agente, etc.).

Registrar el movimiento como duplicado.

La idea principal de este sistema de información, es que los datos mencionados en los párrafos anteriores sean almacenados en un base de datos distribuida que se va ha implementar en una red local de microcomputadoras. Actualmente, este procedimiento se lleva a cabo manualmente en unas formas especiales y existen varias personas que se encargan de registrar los datos, principalmente las fechas. El mantener tanta información de esta manera puede llegar a resultar muy latoso, sobre todo si se desea consultar un determinado número de volante de algún ramo.

El crear un sistema que nos auxilie en el almacenamiento de información y consulta de ésta, es interesante. Sin embargo, se vuelve de mayor utilidad cuando los datos almacenados nos ayudan a la toma de decisiones. Para complementar el sistema se llevará a cabo un pequeño análisis de productividad. El cual nos proporcionará información, cómo la de saber cuantos días tardan los volantes en pasar de un departamento a otro, determinar la cantidad de días que un volante tuvo, desde su ingreso a un ramo, hasta que finalizo su proceso, determinar la cantidad de días que tardan los volantes en ser procesados por departamento, determinar el número de volantes que existen en un departamento a partir de una fecha determinada. Calcular el número de volantes que se tardaron de 8 a 4 días, 5 a 10 días, y 15 días en adelante en ser procesados.

4.3 Diseño del esquema conceptual o global

Un diseño conceptual refleja las necesidades de información que deben ser satisfechas tanto en el presente como en el futuro. "El diseño de la base de datos será tan bueno como lo sean las definiciones de los campos y sus relaciones. De hecho el proceso de diseño es una definición de la organización en términos de sus entidades importantes y sus relaciones intrínsecas"².

La principal tarea, en el diseño conceptual es la recolección de información, para así poder realizar el proceso de normalización de los datos.

El proceso de normalización consiste en agrupar a los campos de datos en tablas que representan a las entidades y sus relaciones. La teoría de normalización se basa principalmente en la observación de un cierto conjunto de relaciones que tienen mejores propiedades en un medio de inserción, actualización y supresión, a diferencia de otros conjuntos de relaciones conteniendo los mismos datos.

La razón principal de usar el procedimiento de normalización es asegurar que el modelo conceptual de la base de datos funcionará.

El primer paso de la normalización consiste en transformar los campos de datos en una tabla de dos dimensiones. Durante el primer paso se eliminan las ocurrencias repetidas de los campos, de tal manera que se obtiene un archivo fijo.

² Atre Shakuntala. Database : Técnicas de bases de datos: Estructuración en diseño y administración. Ed. Trillas 1988.

El segundo paso de la normalización es establecer claves que estén relacionadas con los campos de datos. Si los campos de datos sólo dependen de una parte de la clave, la clave y los campos se pueden separar en registros independientes. La división de la primera tabla normalizada, en una serie de tablas en las que cada campo sólo depende de la clave completa, se llama segunda forma normalizada.

El tercer paso consiste en separar los campos de las segundas relaciones normales, que aunque dependan de una sola clave, deben tener una existencia independiente en la base de datos. Esto se hace de forma tal, que la información sobre estos campos pueda introducirse separadamente a partir de las relaciones en las que se encuentra implicada.

4.3.1 Primera forma normalizada

En esta primera forma, una relación es una "tabla". En cada intersección de un renglón y una columna sólo puede haber un valor en la tabla. En la figura 4.0, se presenta una relación "no normalizada". En los cruces se llega a presentar más de un valor. Por lo cual no se pueden determinar de manera única los valores de los atributos de acuerdo a una clave primaria.

Si la tabla de la figura 4.0 que es una relación "no normalizada" la transformamos a una relación normalizada figura 4.1. Ahora observaremos que existen valores repetidos como son : Número_Volante, Número_Póliza, Tipo, Nombre_Asegurado, Domicilio_Asegurado, Colonia_Asegurado, CP, Estado, Teléfono, Clave_Agente, Estado_A, Oficina, Nombre_Agente, Orden_trabajo, Cálculo, Registro, Reaseguro, Codificación_1, UDS, Codificación_2, Mecanografía, Revisión, Nombre_Persona, Clave_Persona, Departamento y Movlimiento.

Identif. Datos	Identif. Forma	Tipo	Nombre_Agrupado	Nombre_Agrupado	Nombre_Agrupado	OP	Fecha	Fecha_Agenda	Fecha_F	Identif. Agendas	Identif. Agendas	Estado	Estado	Programa	Contenido 1	URL	Identificador	Identificador	Nombre_Fuente	Forma_Fuente	Identificador	Identificador
01	0001	Form	Agenda Escolar	Programa 001	Actividad 00000001	0001	0000	000	00	1	Actividad 1	00000	00000	00000		00000		00000	Actividad 1	Forma 00000	00000	00000
01	0001	Form	Agenda Escolar	Programa 002	Actividad 00000002	0001	0000	000	00	1	Actividad 2	00000	00000	00000		00000		00000	Actividad 2	Forma 00000	00000	00000
01	0001	Form	Agenda Escolar	Programa 003	Actividad 00000003	0001	0000	000	00	1	Actividad 3	00000	00000	00000		00000		00000	Actividad 3	Forma 00000	00000	00000
01	0001	Form	Agenda Escolar	Programa 004	Actividad 00000004	0001	0000	000	00	1	Actividad 4	00000	00000	00000		00000		00000	Actividad 4	Forma 00000	00000	00000
01	0001	Form	Agenda Escolar	Programa 005	Actividad 00000005	0001	0000	000	00	1	Actividad 5	00000	00000	00000		00000		00000	Actividad 5	Forma 00000	00000	00000
01	0001	Form	Agenda Escolar	Programa 006	Actividad 00000006	0001	0000	000	00	1	Actividad 6	00000	00000	00000		00000		00000	Actividad 6	Forma 00000	00000	00000
01	0001	Form	Agenda Escolar	Programa 007	Actividad 00000007	0001	0000	000	00	1	Actividad 7	00000	00000	00000		00000		00000	Actividad 7	Forma 00000	00000	00000
01	0001	Form	Agenda Escolar	Programa 008	Actividad 00000008	0001	0000	000	00	1	Actividad 8	00000	00000	00000		00000		00000	Actividad 8	Forma 00000	00000	00000
01	0001	Form	Agenda Escolar	Programa 009	Actividad 00000009	0001	0000	000	00	1	Actividad 9	00000	00000	00000		00000		00000	Actividad 9	Forma 00000	00000	00000
01	0001	Form	Agenda Escolar	Programa 010	Actividad 00000010	0001	0000	000	00	1	Actividad 10	00000	00000	00000		00000		00000	Actividad 10	Forma 00000	00000	00000

Figura 4.0. Relación "No-Normalizada".

En los cruces de renglones y columnas se presenta más de un valor.

Identif. Datos	Identif. Forma	Tipo	Nombre_Agrupado	Nombre_Agrupado	Nombre_Agrupado	OP	Fecha	Fecha_Agenda	Fecha_F	Identif. Agendas	Identif. Agendas	Estado	Estado	Programa	Contenido 1	URL	Identificador	Identificador	Nombre_Fuente	Forma_Fuente	Identificador	Identificador
01	0001	Form	Agenda Escolar	Programa 001	Actividad 00000001	0001	0000	000	00	1	Actividad 1	00000	00000	00000		00000		00000	Actividad 1	Forma 00000	00000	00000
01	0001	Form	Agenda Escolar	Programa 002	Actividad 00000002	0001	0000	000	00	1	Actividad 2	00000	00000	00000		00000		00000	Actividad 2	Forma 00000	00000	00000
01	0001	Form	Agenda Escolar	Programa 003	Actividad 00000003	0001	0000	000	00	1	Actividad 3	00000	00000	00000		00000		00000	Actividad 3	Forma 00000	00000	00000
01	0001	Form	Agenda Escolar	Programa 004	Actividad 00000004	0001	0000	000	00	1	Actividad 4	00000	00000	00000		00000		00000	Actividad 4	Forma 00000	00000	00000
01	0001	Form	Agenda Escolar	Programa 005	Actividad 00000005	0001	0000	000	00	1	Actividad 5	00000	00000	00000		00000		00000	Actividad 5	Forma 00000	00000	00000
01	0001	Form	Agenda Escolar	Programa 006	Actividad 00000006	0001	0000	000	00	1	Actividad 6	00000	00000	00000		00000		00000	Actividad 6	Forma 00000	00000	00000
01	0001	Form	Agenda Escolar	Programa 007	Actividad 00000007	0001	0000	000	00	1	Actividad 7	00000	00000	00000		00000		00000	Actividad 7	Forma 00000	00000	00000
01	0001	Form	Agenda Escolar	Programa 008	Actividad 00000008	0001	0000	000	00	1	Actividad 8	00000	00000	00000		00000		00000	Actividad 8	Forma 00000	00000	00000
01	0001	Form	Agenda Escolar	Programa 009	Actividad 00000009	0001	0000	000	00	1	Actividad 9	00000	00000	00000		00000		00000	Actividad 9	Forma 00000	00000	00000
01	0001	Form	Agenda Escolar	Programa 010	Actividad 00000010	0001	0000	000	00	1	Actividad 10	00000	00000	00000		00000		00000	Actividad 10	Forma 00000	00000	00000

Figura 4.1. Relación en la primera forma normal.

La figura 4.1, representa una relación en la primera forma normalizada. Puesto que en la intersección de una columna y un renglón sólo existe un valor y no un grupo de valores. La figura 4.2, muestra la relación en forma gráfica.

Durante la primera forma normalizada se debe establecer una llave primaria, la cual se construye en base a los atributos de la relación. En base a esta llave primaria, que aquí se ha determinado como : *Número_Volante* + *Orden_Trabajo* + *Clave_Agente*, se pueden conocer los demás valores de la relación (Tipo, Nombre_Asegurado, Domicilio_Asegurado, Colonia_Asegurado, CP, Estado, Teléfono, Nombre_Agente, Movimiento, etc). Cabe mencionar que todos los atributos que no son llave en la relación son funcionalmente dependientes de la llave primaria. Es decir, a partir de la llave primaria se pueden determinar sus valores de manera única.

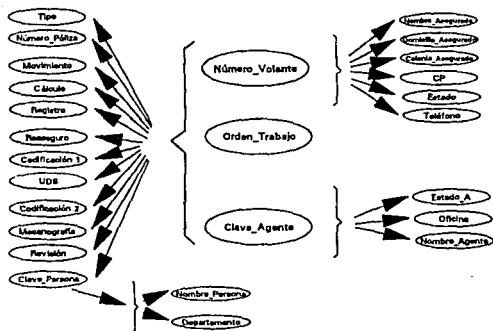


Figura 4.2. Relación en la primera forma normaliza. A partir de 3 valores (Número_Volante, Orden_Trabajo y Clave_Agente se puede conocer cualquier valor de la relación.

Si observamos la figura 4.1, nos daremos cuenta de que existen problemas durante la primera forma normalizada. Uno de los principales problemas que se presentan son las fallas de inserción. Los registros no se pueden dar de alta totalmente :

- Es posible que un nuevo agente ingrese a la compañía de seguros.

Para el caso anterior no es posible dar de alta el registro, debido a que todavía no se conocen los demás datos de la relación. Cabe mencionar que como el agente acaba de ingresar a la compañía, posiblemente no haya contactado algún cliente, al cual le pueda ofrecer un seguro. Por lo tanto, aún no existen números de volantes que se puedan dar de alta o bien asignar números de pólizas.

Existen otras dos fallas que se presentan en la primera forma normalizada :

- Fallas de actualización.
- Fallas de eliminación.

Por ejemplo, en una falla de actualización, si un asegurado cambia de domicilio entonces se tendrán que actualizar todos los registros que contemplarán a ese asegurado. Si el mecanismo de actualización fallara, entonces existirán unos registros con la dirección nueva y otros con la dirección vieja, lo cual causará problemas de inconsistencia. No solamente al cambiar este atributo sino cualquier otro que estuviera repetido como se muestra en la figura 4.1.

Entre los principales problemas de eliminación que existen en la primera forma normalizada se puede mencionar el siguiente: no poder dar de baja a un agente que renuncio o fue despedido sin tener que afectar al registro en su totalidad.

4.3.2 Segunda forma normalizada

Una relación se encuentra en la segunda forma normalizada, cuando todo atributo que no sea llave es completamente dependiente de manera funcional de la llave primaria. Es decir, cualquier atributo que no es llave necesita de la llave primaria para ser identificado en forma única. Se dice, que toda relación que esta en la segunda forma normalizada también representa una relación en la primera forma.

Los tres principales problemas: inserción, actualización y eliminación que aparecieron en la primera forma se pueden corregir. Por ejemplo, para el caso de la inserción y eliminación, si separamos la Clave_Agente, Estado_A, Oficina, Nombre_Agente, en una nueva relación llamada AGENTE figura 4.3. Y teniendo como llave la Clave_Agente. La falla de inserción y eliminación se rectifica, ya que ahora el agente puede ser dado de alta o baja, sin tener que estar relacionado con algun cliente de la compañía de seguros. Para el caso de la baja o eliminación se debe considerar que si el agente tenía volantes en circulación o bien terminados, su clave en la relación general (RAMO Figura 4.5) debe ser modificada. Para esto se toma en consideración que si la clave es modificada o bien no se indica en la alta, se dice que ese registro es directo con Clave_Agente=XXX.

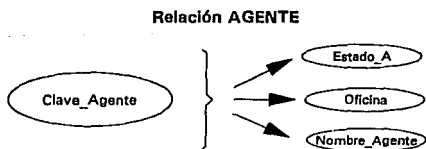


Figura 4.3. Relación en la segunda forma normaliza, la llave primaria de la relación Agente es la Clave_Agente.

Para el caso de la actualización de algún dato de un asegurado, se separan los atributos de Nombre_Asegurado, Domicilio_Asegurado, Colonia, CP, Estado y Teléfono en una nueva relación llamada *ASEGURADO*, teniendo como llave el Número_Volante figura 4.4. Al efectuar tal separación los problemas de inconsistencia que existían en la primera forma son eliminados.

Finalmente la relación de la primera forma normalizada que se presenta en la figura 4.2 fue dividida en 3 relaciones (*RAMO*, *AGENTE* y *ASEGURADO*). Se dice que estas relaciones se encuentran en la segunda forma normalizada, ya que se han eliminado algunas fallas de almacenamiento que se encontraron en la primera forma normalizada.

Relación ASEGURADO

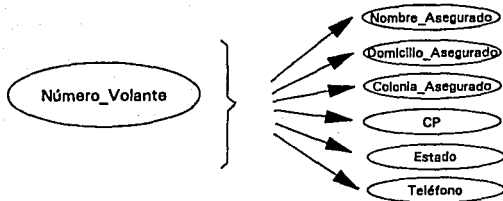


Figura 4.4. Relación en la segunda forma normalizada, la llave primaria de la relación Asegurado es el Número_Volante.

Relación RAMO

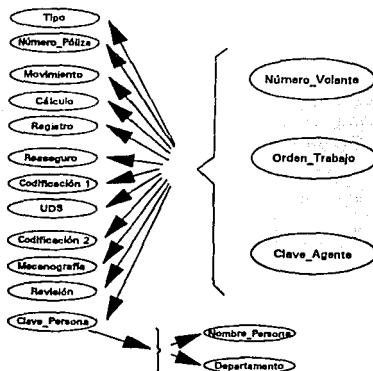


Figura 4.5. Relación en la segunda forma normalizada, la llave primaria de la relación Ramo es Número_Volante + Orden_Trabajo + Clave_Agente.

4.3.3 Tercera forma normalizada

Una relación está en la tercera forma normalizada, si no existe dependencia funcional transitiva entre los atributos que no son clave. Esta dependencia existe cuando un atributo que no es clave se puede determinar con uno o más atributos que no sean. Por ejemplo, el Nombre_Persona y el departamento se pueden determinar a partir de la Clave_Persona. De aquí, se concluye que el Nombre_Persona y el Departamento dependen funcional y transitivamente de la Clave_Persona.

Una atención especial requiere este campo de Clave_Persona. En el se contemplaran las claves de los tres departamentos criticos (cálculo, mecanografía y revisión) ya que un volante puede circular o no circular por uno, dos, todos, o ningún departamento. Este campo contempla 6 caracteres donde los dos primeros corresponden al departamento de calculo, los caracteres 3 y 4 al departamento de mecanografía, mientras que los caracteres 5 y 6 al departamento de revision.

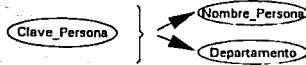
Al crear dos relaciones más figura 4.6, se eliminan las fallas de almacenamiento. Por decir puede existir una nueva Clave_Persona, Nombre_Persona y Departamento que aun no tenga asignado un Número_Volante. En cuanto a la actualización el Nombre_Persona y Departamento aparecerá una sola vez. Para el caso de la eliminación resulta ser el mismo que el de la actualización, solo se tendra que dar de baja un registro en una sola relación.

Durante la tercera forma normalizada se ha creado una nueva relación llamada CLAVES. Es importante hacer notar que el proceso de normalización de una relación, que se encontraba en la primera forma y que después de una serie de reducciones paso a ser de la tercera forma normalizada, sigue siendo la misma, puesto que no se perdió información. "Uno de los puntos que se debe enfatizar es que el proceso de reducción o normalización no es una función de los valores de los datos que aparecen en las relaciones de algún momento determinado: es una función de las relaciones entre los atributos".³

Finalmente como resultado de este proceso de normalización, se han establecido en la tercera forma normalizada cuatro relaciones figura 4.7. Estas relaciones llamadas Ramo, Asegurado, Agente, y Claves representan el modelo conceptual o global.

³ Atre Shakuntala. Técnicas de bases de datos : Estructuración en diseño y administración. Ed. Trillas.

Relación CLAVES



Relación RAMO

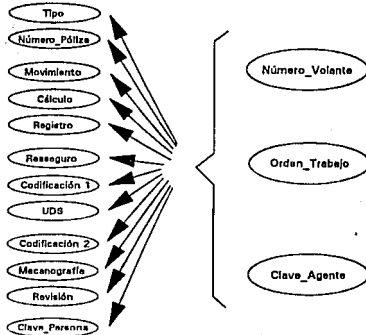


Figura 4.6. Relaciones en la tercera forma normalizada, para la relación CLAVES la llave primaria de Clave_Persona, mientras que para la relación RAMO la llave primaria es Número_Volante + Orden_Trabajo + Clave_Agente.

- RAMO : (Número_Volante, Orden_Trabajo, Clave_Agente, Tipo, Número_Póliza, Movimiento, Cálculo, Registro, Reaseguro, Codificación_1, UDS, Codificación_2, Mecanografía, Revisión, Clave_Persona)

- ASEGURADO (Número_Volante, Nombre_Asegurado, CP, Estado, Teléfono, Domicilio_Asegurado, Colonia_Asegurado)

- AGENTE (Clave_Agente, Estado_A, Oficina, Nombre_Agente)

- CLAVES (Clave_Persona, Nombre_Persona, Departamento)

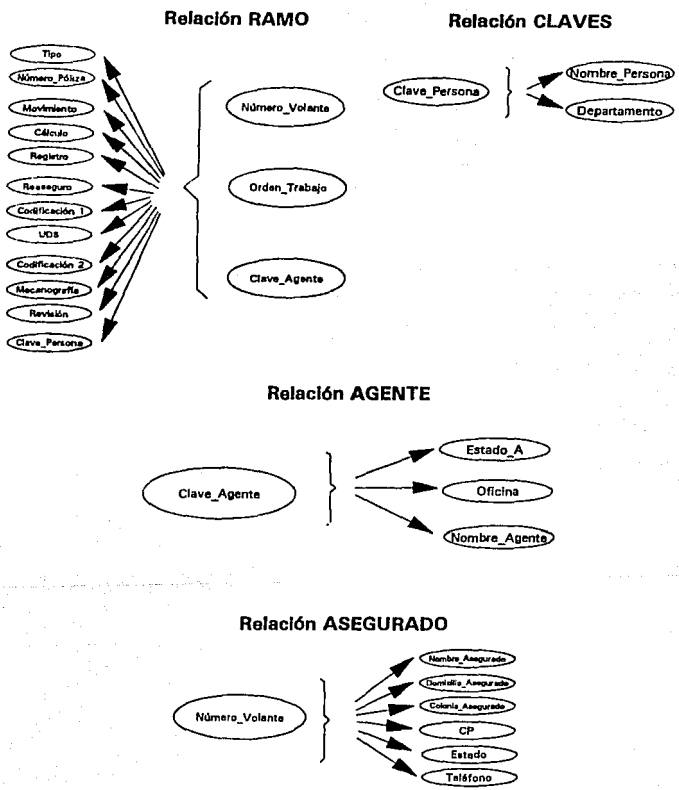


Figura 4.7. Cuatro relaciones en la tercera forma normalizada.

4.4 Diseño del modelo lógico.

El modelo lógico es aquel que se obtiene a partir del modelo conceptual. Existen principalmente 3 modelos de datos :

- Relacional
- Jerárquico
- Red

Como se indico al inicio de este capitulo, el modelo que se utilizará para el desarrollo del sistema de bases de datos distribuidas será el modelo relacional. La transformación del modelo conceptual al modelo relacional es de manera directa.

El modelo relacional consta de varias tablas. Estas tablas definen relaciones y atributos. Y uno o varios atributos de esta relación representarán a la llave primaria.

De las cuatro relaciones *RAMO*, *ASEGURADO*, *AGENTE* y *CLAVES* que se definieron en el modelo conceptual, existen una serie de atributos que son particulares a cada relación.

La relación *RAMO* que la podriamos llamar la principal, está constituida por los atributos (*Número_Volante*, *Orden_Trabajo*, *Clave_Agente*, *Tipo*, *Número_Póliza*, *Movimiento*, *Orden_Trabajo*, *Cálculo*, *Registro*, *Reaseguro*, *Codificación_1*, *UDS*, *Codificación_2*, *Mecanografía*, *Revisión*, *Clave_Persona*). La llave primaria de esta relación se compone de tres atributos *Número_Volante* + *Fecha_Ingreso* + *Clave_Agente*. El *Número_Volante* es único, el cual tiene una *Orden_Trabajo* y una sola *Clave_Agente* por lo tanto la cadena que se puede formar es única.

La relación *ASEGURADO* tiene como atributos principales: (*Número_Volante*, *Nombre_Asegurado*, *CP*, *Estado*, *Teléfono*, *Domicilio_Asegurado*, *Colonia_Asegurado*). La llave primaria ésta constituida por un solo elemento, *Número_Volante*. Los valores que puede tomar esta llave son únicos, no existen 2 renglones de la tabla que tengan el mismo *Número_Volante*.

La relación *AGENTE* tiene como atributos principales: *Clave_Agente*, *Estado_A*, *Oficina*, *Nombre_Agente*. La llave primaria es única. La llave primaria de esta relación la constituye únicamente la *Clave_Agente*. Los valores que toma esta llave son únicos ya que no se repiten en ningún otro renglón de la relación.

Finalmente, la relación que forma la tabla de *CLAVES* tiene como llave primaria la *Clave_Persona*, y sus dos atributos que no son llave son: *Nombre_Persona* y *Departamento*. Esta llave también es única.

Las cuatro relaciones anteriores representan el modelo lógico de la base de datos. Existen algunos atributos de llave primaria que son redundantes en algunos de los puntos de vista del usuario. Por ejemplo, el *Número_Volante*, *Clave_Persona* y *Clave_Agente* que aparecen en 2 de las 4 relaciones establecidas. Sin embargo, esto no representa ninguna desventaja, ya que a partir de ellas se conoce la liga que existe entre las demás relaciones.

4.5 Diseño de las bases de datos físicas

El modelo físico, es la estructura de la base de datos que se va almacenar en dispositivos físicos. Es muy importante, conocer las características del funcionamiento del modelo físico antes de implantar la base de datos.

El modelo físico incluye los métodos de acceso (secuencial, secuencial indexado, directo, etc.) así como también la estructura física de los archivos.

El modelo físico describe la estructura de la base de datos en el nivel físico. En él se incluyen los tamaños y características de los campos de la base de datos. Además, se incluye una descripción del manejo de las áreas físicas de almacenamiento tales como segmentos, bloques y *buffers*.

Generalmente, los diseñadores de bases de datos tratan de optimizar el modelo físico en cuanto a tiempo de acceso a las bases de datos y espacio de almacenamiento que éstas ocupan. Sin embargo, éstos dos aspectos no resultan ser muy relevantes, si los requerimientos comerciales resultan ser otros. Por ejemplo, es posible que una base de datos requiera de diversos tipos de entrada o accesos a la base de datos. Para proporcionar éste tipo de acceso, puede ser necesario que un archivo se invierta, por lo cual existe un gasto de tiempo y/o espacio.

Para diseñar un modelo físico se requiere que el diseñador conozca tres aspectos fundamentales :

- Conocer las funciones del manejador de base de datos (DBMS).
- Conocer las características de los dispositivos de acceso directo.
- Conocer la aplicación a desarrollar.

En cuanto al primer aspecto, el diseñador tiene que ser un experto en el conocimiento del manejador de base de datos (DBMS).

El segundo aspecto, involucra conocer perfectamente las características de los dispositivos de acceso directo. Muchas veces se piensa que una serie de registros se encuentran físicamente uno tras otro. Sin embargo, puede existir que el "siguiente registro" se encuentre físicamente muy separado del anterior. En este aspecto se deben conocer las generalidades de la base de datos como son : la disposición del registro en disco, tamaño de bloques, tamaño de los *buffers* , características de entrada y salida, etc.

El tercer aspecto, consiste en las aplicaciones. Se deben conocer las relaciones de los campos de los datos y de las entidades. Es decir el modelo externo.

El diseño del modelo físico de una base de datos resulta ser un tema muy delicado. Sin embargo, para el desarrollo de los accesos a las bases de datos se usarán archivos indexados, debido a que permiten gran flexibilidad de acceso en interrogaciones de tipo muy complejo.

4.6 Diseño de la fragmentación de los datos

El diseño de la fragmentación es uno de los principales problemas que se presentan al distribuir una base de datos. El objetivo de la fragmentación es básicamente determinar que los fragmentos no se traslapen.

Los fragmentos, son considerados como "unidades lógicas de distribución". Tales fragmentos pueden consistir de un grupo de tuplas (en el caso de fragmentación vertical) o bien atributos (en el caso de fragmentación horizontal). Cada grupo de tuplas o atributos tienen las mismas propiedades, lo cual constituye un fragmento.

A continuación, se determinarán los fragmentos horizontales, verticales y mixtos para las relaciones *RAMO, CLAVES AGENTE y ASEGURADO*.

Considerando que la base de datos distribuida para la compañía de seguros tiene inicialmente 4 zonas de importancia (Norte, Sur, Este y Oeste), y además en cada estado de la república existe una sucursal. Estas sucursales han sido agrupadas de acuerdo con su localización geográfica. Por ejemplo, los estados que abarcan la parte norte del país, como son : Chihuahua, Baja California Norte, etc. fueron agrupados en la zona norte. Mientras que los demás estados fueron agrupados en otras 3 zonas.

En la tabla 4.0, se muestra como fueron agrupados los estados de la república mexicana. Esta agrupación se realiza, solo con la finalidad de tener un ejemplo de cómo se pueden distribuir las localidades.

Al analizar la relación *Asegurado*, observaremos que esta puede asumir 1 valor diferente para cada estado, Sin embargo los estados fueron agrupados en 4 zonas como se muestra en la tabla 4.0. De ahí concluimos que sólo, 4 valores son los que puede tomar el registro estado. Por lo tanto, los 4 predicados que se generan de esta relación son :

- P_1 : Estado = Norte
- P_2 : Estado = Sur
- P_3 : Estado = Oeste
- P_4 : Estado = Este

Al determinar la implicaciones que existen en los predicados, se redujeron los fragmentos de la siguiente manera:

y_1 : (Estado=Norte) and (Estado=Sur) and (Estado=Oeste)
and (Estado=Este)

y_2 : (Estado=Norte) and not (Estado=Sur) and not (Estado=Oeste)
and not (Estado=Este)

y_3 : not (Estado=Norte) and (Estado=Sur) and not (Estado=Oeste)
and not (Estado=Este)

y_4 : not (Estado=Norte) and not (Estado=Sur) and (Estado=Oeste)
and not (Estado=Este)

y_5 : not (Estado=Norte) and not (Estado=Sur) and not (Estado=Oeste)
and (Estado=Este)

y_6 : not (Estado=Norte) and not (Estado=Sur) and not (Estado=Oeste)
and not (Estado=Este)

Y además se sabe que :

Estado = Norte \leftrightarrow not (Estado=Sur) and not (Estado=Oeste) not
(Estado=Este)

Estado = Sur \leftrightarrow not (Estado=Norte) and not (Estado=Oeste) not
(Estado=Este)

Estado = Oeste \leftrightarrow not (Estado=Norte) and not (Estado=Sur) not
(Estado=Este)

Estado = Este \leftrightarrow not (Estado=Norte) and not (Estado=Sur) not
(Estado=Oeste)

De aquí concluimos que y_1 y y_6 son contradictorias y y_2 , y_3 ,
 y_4 , y_5 se reducen a los predicados P_1 , P_2 , P_3 y P_4 .

Por lo tanto la fragmentación primaria horizontal de la
relación *Asegurado* es la siguiente :

$$\text{Asegurado}_1 = \sigma_{\text{Estado=Norte}} \text{Asegurado}$$

$$\text{Asegurado}_2 = \sigma_{\text{Estado=Sur}} \text{Asegurado}$$

$$\text{Asegurado}_3 = \sigma_{\text{Estado=Oeste}} \text{Asegurado}$$

$$\text{Asegurado}_4 = \sigma_{\text{Estado=Este}} \text{Asegurado}$$

La condición de reconstrucción se puede verificar fácilmente
con la operación de Unión U.

$$\text{Asegurado} = \text{Asegurado}_1 \cup \text{Asegurado}_2 \cup \text{Asegurado}_3 \cup \text{Asegurado}_4$$

Al analizar la relación *Ramo* observaremos que los datos se pueden agrupar en dos relaciones. En la primera se contemplan los datos de la póliza (tipo, número, movimiento), mientras que en la segunda relación, los datos que corresponden a las fechas de los departamentos por los que circula el volante. La fragmentación que se requiere para separar la relación esta dada por la fragmentación vertical.

Esta fragmentación toma una gran desventaja cuando la aplicación usa una de las dos relaciones, siempre y cuando la relación de la que se partio es muy grande. La desventaja, que presenta es que si se requieren los dos fragmentos entonces se tendrá que efectuar el producto natural de ambas relaciones.

Para nuestro caso la relación se agrupa en dos. La fragmentación vertical de la relación *Ramo* es la siguiente :

$$Ramo_1 = \Pi_{(Número_Volante, Tipo, número_póliza, movimiento, clave_agente)}^{Ramo}$$

$$Ramo_2 = \Pi_{(Número_Volante, Orden_Trabajo, cálculo, registro, reaseguro, codificacion_1, UDS, codificacion_2, mecanografia, revision, clave_persona)}^{Ramo}$$

La condición de reconstrucción se puede realizar por medio del producto natural.

$$Ramo = Ramo_1 \times Ramo_2$$

La relación *Agente* no sera fragmentada, sino se duplicará en cada una de las localidades que se vayan a contemplar en el sistema.

Cabe mencionar que se hara necesaria la creación de un catálogo de claves de estados, donde se registrarn la clave del estado así como su nombre correspondiente, este catálogo será duplicado al igual que la relación *Agente*.

4.7 Diseño y asignación de los fragmentos

Una de las mejores formas de asignar fragmentos a una localidad, es considerar a cada fragmento como un archivo. Sin embargo, esta consideración no es siempre la mejor por las siguientes razones :

- Los fragmentos no se modelaron como archivos individuales.
- Existen más fragmentos, de los que la relación original global tenía.
- En sistemas de archivos las aplicaciones que hacen uso de archivos remotos, se les considera como un "acceso remoto a un archivo", mientras que en bases de datos distribuidas existe un manejo más sofisticado de los datos.

Algunas de estas consideraciones aún no han sido resueltas sobre todo la última, la cual se evalua por medio de la distribución de sus datos y la correcta optimización de sus aplicaciones.

En una aplicación donde los datos se distribuyen es muy importante que las aplicaciones se optimizen para cada uno de los fragmentos de datos, esto implica de alguna manera resolver la optimización de consultas y preguntas (Query), lo cual resulta ser difícil debido que se es un subproblema de la asignación de los datos.

En la determinación de la asignación de los fragmentos, es muy importante distinguir, cuando se diseña una asignación redundante y no redundante.

El determinar una asignación no redundante es fácil con la ayuda de método más apropiado (*Best-Fit*), el cual se basa en una medida asociada con cada posible asignación, y el lugar con el mejor patrón o medida es siempre seleccionado.

La duplicación de fragmentos llegar a ser más complejo debido que :

- El grado de la duplicación de cada fragmento llega ser una variable del problema.
- El modelar aplicaciones de lectura es complicado, debido a que ahora se tienen varias localidades que pueden acceder los fragmentos.

Para determinar la duplicidad de los fragmentos se pueden utilizar cualquiera de los dos métodos siguientes :

- Determinar el conjunto de las localidades donde el beneficio de asignar una copia del fragmento es más alto que el costo, y asignar una copia del fragmento a cada elemento de este conjunto; este método selecciona "todos los lugares benéficos".
- Determinar primero la solución del problema de no-duplicidad, y progresivamente introducir copias repetidas comenzando con la más benéfica. El proceso se termina cuando la no "adición de copias " es benéfica.

Ambos métodos tienen algunas desventajas. En el caso de "todos los lugares benéficos" cuantificar los costos y beneficios de cada fragmento individualmente es más crítica que en el caso no redundante. El método de la "adición de copias" es completamente heurístico.

Para el diseño del sistema de base de datos distribuidas se contemplarán casos de redundancia y no-redundancia.

De acuerdo al análisis de los párrafos anteriores la base de datos se va a distribuir en diferentes localidades de la la red de microcomputadoras. Las relaciones que se contemplan estarán almacenadas en forma local y en forma repetida.

La relación de *Agentes* y el catálogo de *estados y servidores* existirán en forma duplicada en todos los servidores de la red. Mientras que la relación *Asegurado, Ramos y Claves* serán locales a cada uno de los servidores.

La figura 4.8 muestra en forma gráfica la asignación de los fragmentos de la red general de microcomputadoras. En ella se contemplan los servidores de archivos que corresponden a cada una de las redes locales, así como las relaciones y catálogos que se van almacenar.

Como nota aclaratoria se supone que las redes locales se encuentran conectadas y que existen 4 servidores archivos, en cada uno de estos servidores se contemplan los estados de la república mexicana (Ver Tabla 4.0). Sin embargo, visto en forma más general podría llegar a existir un servidor de archivos por cada estado de la república.

NORTE	SUR
Baja California Norte	Campesche
Baja California Sur	Chiapas
Chihuahua	Chiapas
Coahuila	Guerrero
Durango	Morelos
Nuevo León	Oaxaca
Guanajuato	Quintana Roo
Sonora	Yucatán
Tamaulipas	

OESTE	ESTE
Aguascalientes	Distrito Federal
Colima	Hidalgo
Estado México	Puebla
Guerrero	San Luis Potosí
Jalisco	Tlaxcala
Michoacán	Veracruz
Nayarit	
Querétaro	
Zacatecas	

Tabla 4.0. Agrupación de los estados de la República Mexicana en 4 zonas.

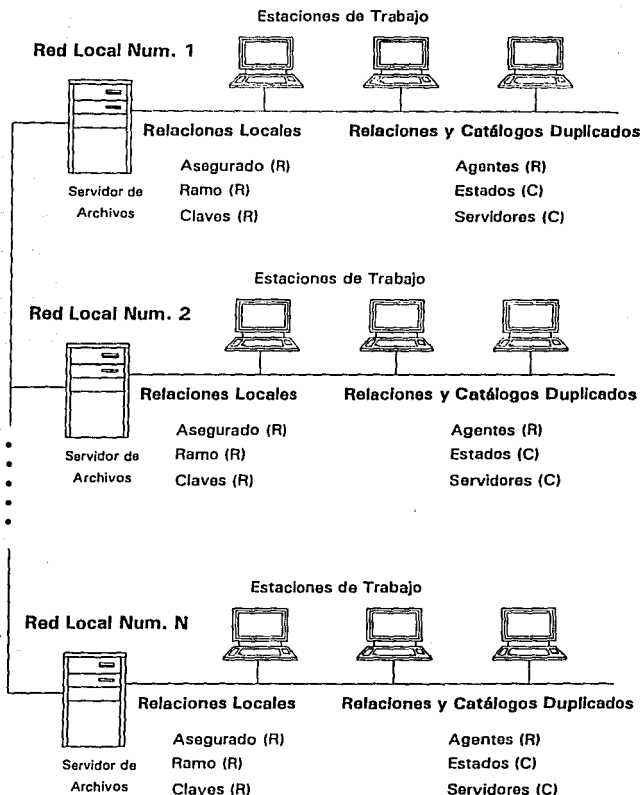


Figura 4.8. Diseño y asignación de fragmentos en la red de microcomputadoras.

CAPITULO V

Desarrollo e implantación del sistema.

5.1 Selección del lenguaje de programación

Desde la década de los 60's se han diseñado y aplicado miles de lenguajes programación. Muchos de estos lenguajes han tenido que ver con proyectos de investigación y solo un número pequeño de ellos han alcanzado uso generalizado. Por ejemplo, el lenguaje que más se utiliza para aplicaciones profesionales es el *COBOL*, empleado en todas las actividades profesionales de procesamiento de datos. En la programación científica, *FORTRAN* es el lenguaje más común, mientras que *PASCAL* y *C* se usan en la programación de sistemas. En el futuro, el lenguaje *ADA*, patrocinado por el Departamento de la Defensa de los Estados Unidos, quizá sustituya a *FORTRAN* y *PASCAL* en muchas aplicaciones y gran parte de la programación en *COBOL*. Además lenguajes como *LISP* y *PROLOG* se están usando en aplicaciones de inteligencia artificial.

En general los lenguajes de programación se incluyen en una de las cinco categorías siguientes :

- *Lenguajes Ensambladores* .- Son lenguajes de máquina. La programación en código ensamblador requiere que el programador traduzca el diseño de su programa a secuencias de acciones de la máquina. Los lenguajes ensambladores son específicos de la familia de computadoras en la que se aplican.

- *Lenguajes para aplicación de sistema* .- Esta clase evolucionó a partir de los lenguajes de ensamblador, cuando se reconoció la dificultad para programar en código ensamblador. Los lenguajes de aplicación de sistemas proporcionan algunas ventajas como son tener acceso directo a las operaciones y direcciones de la máquina. En teoría lo que se puede programar en lenguaje ensamblador se puede programar en lenguaje para aplicación de sistemas. Los lenguajes *BCPL* y *C* son lenguajes de aplicación de sistemas.

- *Lenguajes estáticos de alto nivel* .- Son lenguajes que brindan al programador algunas proposiciones de control y declaraciones de variables, pero no brindan la posibilidad de controlar las operaciones de la máquina generadas por el compilador. Se caracterizan por tener asignación estática de memoria. La cantidad de memoria que vayan a requerir se calcula y reserva con anterioridad a la ejecución del programa. Los ejemplos, más conocidos son *COBOL* y *FORTRAN*.

- *Lenguajes de alto nivel estructurados por bloques* .- Son lenguajes que se desarrollaron a partir de los lenguajes estáticos. Proporcionan al programador una serie de construcciones de control y la posibilidad de clasificar los objetos de los programas dentro de un tipo determinado. Tienen la particularidad de ofrecer en forma limitada la asignación de memoria llamada estructura de bloque, apartando memoria al ejecutar un bloque y ésta se libera al finalizar el bloque, con la limitante que el programador debe calcular con exactitud cuánta memoria va a necesitar. Algunos ejemplos de estos lenguajes son *ALGOL* y *PASCAL*.

Los lenguajes como *ADA* y *MODULA* también se incluyen en esta categoría, aunque sus reglas de alcance no son estrictamente las de los lenguajes estructurados por bloques.

- *Lenguajes dinámicos de alto nivel* .- Esta clase de lenguajes se distinguen, porque todo el control de la memoria se hace dinámicamente. Es decir, la ejecución de las proposiciones individuales del lenguaje puede hacer que se asigne y retire la memoria. En general la estructura de estos lenguajes tiende a ser bastante distinta de la estructura de los lenguajes de alto nivel estáticos o estructurados. Ejemplos de esta clase de lenguajes son *APL*, *PROLOG*, y *LISP*.

"Es importante que los ingenieros de software comprendan las ramificaciones de las distintas construcciones de los lenguajes de programación. Sin esta comprensión, no es posible tomar decisiones técnicas razonables a cerca de los lenguajes de programación que se deben elegir para un proyecto."¹

Para el desarrollo de la aplicación de esta tesis, se ha elegido a Clipper como lenguaje de programación, cabe mencionar que Clipper no es un manejador de base de datos, si no un manejador de archivos. Sin embargo, su riqueza en herramientas de programación hacen posible que se resuelva cualquier problema de base de datos.

¹Sommerville Ian. Ingeniería de Software.
Segunda Edición. SITESA Sistemas técnicos de Edición
Addision - Wesley Iberoamericana 1988.

Clipper es una paquete que tiene la ventaja de contar con una arquitectura abierta, poder usar y construir librerías independientemente del lenguaje que se use. En otras palabras el poder de clipper no esta limitado al contenido de la caja donde viene el software, si no que es un verdadero medio ambiente de desarrollo con una gran capacidad de expansión.

La arquitectura abierta de Clipper también ofrece el intercambio de datos, esto significa que la gente que desarrolle puede usar la tecnología de almacenamiento de datos que mejor convenga a sus propósitos. Las aplicaciones que se desarrollen en Clipper pueden ser programadas para leer y escribir datos en cualquier formato.

Como el compilador de Clipper esta escrito en lenguaje C, el desarrollo de rutinas y librerías en lenguaje C, para su integración lo hacen más versatil, además de que se incluyen rutinas para manejo de esquemas de candados (Locking) a nivel registro y a nivel archivo, tiene la facilidad de soportar redes locales de PC, que lo hacen altamente eficiente en ambientes multiusuario.

Para la programación del sistema de control de volantes, se han tenido que elaborar algunas rutinas en C. Estas rutinas contemplan el manejo a los API (Application Program Interfase) del sistema operativo de red.

Si al lenguaje Clipper lo clasificamos en una de las categorías mencionadas anteriormente, observaremos que puede caer en dos. Como lenguaje para aplicaciones de sistema y como lenguaje de alto nivel estructurado por bloques.

5.2 Interfaz con el usuario

La interfaz del usuario es la vara con la que el usuario mide la calidad de un sistema. Si un sistema tiene una interfaz mal diseñada, está expuesto al rechazo. Una interfaz mal diseñada puede hacer que el usuario cometa errores catastróficos. Si la información se presenta en forma confusa o engañosa, el usuario puede mal interpretar el significado de un elemento de información y generar una serie de secuencias peligrosas.

Las interfaces del usuario se pueden clasificar en interfaces interactivas en línea y en interfaces fuera de línea. En la interfaz en línea, el usuario interactúa en forma directa con el sistema de computación. En una interfaz fuera de línea, el usuario prepara por separado una entrada legible para la máquina y después la presenta al sistema.

En las interfaces interactivas del usuario se pueden aplicar varios estilos de diálogo basados en aspectos mnemotécnicos en lenguaje cuasinatural, guiados por formas, diálogos tipo menú, y de pregunta y respuesta. Los tres primeros se clasifican como diálogos iniciados por el usuario y los dos últimos como iniciados por la computadora. Los diálogos iniciados por la computadora son muy útiles para usuarios sin práctica, mientras que los iniciados por el usuario son mejores si estos tienen experiencia y trabajan regularmente con computadoras.

"Una interfaz pobre o mal diseñada puede tener graves consecuencias. En el mejor de los casos, una interfaz inadecuada producirá un costo de adiestramiento elevado y una mayor proporción de errores de usuarios y en el peor de los casos, dará como resultado que se abandone el sistema por completo."²

Como se mencionó anteriormente las interfases se clasifican como sigue :

- Interfaces iniciadas por el usuario
- Interfaces iniciadas por la computadora

Para el primer caso el usuario inicia el diálogo con la computadora. En el segundo caso el diálogo es iniciado por la computadora, el usuario responde y basandose en la respuesta la computadora toma la acción correspondiente.

El principio fundamental del diseño de las interfaces es que no se deben adecuar a las necesidades y posibilidades del usuario individual. No se debe forzar a los usuarios a que se adapten a una interfaz que es conveniente aplicar o que es adecuada para el diseñador del sistema. Diseñar una interfaz a la medida del usuario significa que se debe expresar en términos familiares para éste y que los objetos manipulados sean análogos directos en el ambiente en que trabaja.

Por ejemplo, en un sistema utilizado por el departamento de secretarias, los objetos manipulados deben ser cartas, documentos, agendas, carpetas, etc., al menos en lo que a las secretarias concierne. En la práctica, cada uno de esos objetos se aplicarán por medio de diferentes archivos, pero no forzar a la secretaria a manejar conceptos de computación.

Existen otros tipos de interfaces que se presentan en base a imágenes (iconos) de objetos de oficina familiares : calculadoras, papeleras, lápices, borradores, etc. Para seleccionar una de ellos, el usuario señala la imagen apropiada y la operación se inicia.

Este tipo de interfaz es muy adecuada para usuarios sin experiencia, porque no se intimidan con la terminología asociada con la máquina. Pero tienen el inconveniente de que las operaciones completamente nuevas que carecen de un análogo no automatizado, no se pueden representar con un icono familiar. Por ejemplo, ¿Cómo se puede elaborar una imagen de comparaciones entre archivos múltiples?

Otro principio fundamental es la consistencia de la interfaz. Se debe mantener la consistencia dentro de un sistema y entre los subsistemas que se ejecuten en la misma computadora. La consistencia significa que los mandatos del sistema deben tener el mismo formato, los parámetros se deben pasar a todos los mandatos de la misma manera y la puntuación de éstos debe ser similar.

El tercer principio del diseño de la interfaz es que ésta tenga dispositivos integrados de ayuda. Los cuales deben proporcionar diferentes niveles de ayuda e información, desde aspectos básicos sobre cómo empezar con el sistema, hasta una descripción completa de las cualidades del sistema y la manera de utilizarlas.

Las interfaces iniciadas por el usuario se clasifican de la siguiente manera :

- *Interfaces controladas por formas* .- El usuario introducen mandatos o preguntas que el computador interpreta.
- *Interfaces orientadas a formas* .- El usuario llama a una imagen de una forma a su pantalla y la llena escribiendo en las partes apropiadas de la pantalla.

Las interfaces controladas por mandatos implican la introducción del usuario de un mandato en el sistema, el cual tomará una acción dependiendo del mandato. Esta clase de interfaz es un derivado directo de las interfaces fuera de línea.

Para muchas aplicaciones esta forma de interfaz del usuario es muy adecuada. Gran parte de la información procesada por el sistema se reúne, introduce y despliega como texto y no se necesitan otras posibilidades que las requeridas para introducir y sacar textos.

En las interfases del usuario orientadas a las formas se basan en que el usuario llame la imagen de una forma a su terminal. Entonces el usuario rellena las posiciones apropiadas de la pantalla. Tales sistemas se basan en un terminal con un cursor que se puede mover a cualquier lugar de la pantalla.

Este tipo de interfaz a menudo es la mejor elección cuando se automatiza un sistema de trabajo administrativo basado en formas estándar. El diseño de la interfaz se puede basar en las formas que ya están en uso.

En las interfaces iniciadas por la computadora son aquellas en las cuales el sistema guía el progreso del diálogo entre el usuario y el computador. La información se despliega, el usuario responde de manera simple y, a partir de esa respuesta, la computadora inicia una acción o despliega más información.

Las interfaces iniciadas por la computadora se clasifican en dos clases :

- Sistemas de menús en donde se presenta al usuario una lista de posibilidades y él elige una de ellas.
- Sistemas de pregunta y respuesta en donde la computadora hace una pregunta al usuario e inicia una acción basada en la respuesta.

El problema general con las interfaces iniciadas por la computadora es que a medida que el usuario se familiariza con ellas, puede sentirse frustrado por la falta de control sobre el sistema.

En una interfaz tipo menú, el usuario debe seleccionar una de varias posibilidades e indicar su elección a la máquina. Hay varias formas de indicar esta elección. El usuario puede digitar el nombre o el identificador de la selección; puede señalarla con un ratón o con otro dispositivo destinado a ello; puede usar teclas para mover el cursor y situarlo sobre la alternativa, en otros tipos de terminales sensibles al contacto, puede indicar la selección con un dedo.

En las interfaces de preguntas y respuestas son interfaces del usuario en las que el desarrollo del diálogo entre la persona y la computadora está dirigida por las respuestas del usuario a las preguntas que hace la máquina. Este tipo de interfaz se utiliza sobre todo en sistemas de aprendizaje por computadora, pero también se han desarrollado sistemas de entrevistas médicas que usan esta clase de interfaces.

El diseño de interfaces de pregunta y respuesta depende mucho de la aplicación que se realiza. Parece que tienen más éxito cuando el número de repuesta posibles del usuario a una pregunta es limitado.

Para la programación del sistema de la tesis , se van a utilizar 3 tipos de interfaces :

- Interfases orientadas a formas
- Interfaces de menús
- Interfaces de pregunta y respuesta

Las interfaces orientadas a formas se utilizan, debido a que los usuarios tienen que capturar datos como son : tipo de póliza, número de póliza, movimiento que tuvo la póliza, clave de agente, clave de persona, etc. Las formas que se presentan son en el módulo de ramos , cuando se da de alta un volante, se modifica o se elimina. Como se muestra en la figura 5.0

Captura de datos en una forma

20 de Junio,1991		SISTEMA DE CONTROL DE VOLANTES		13:11						
-PRINCIPAL-										
ALTA DE VOLANTE										
Número de Volante : 00057		Orden de trabajo : 20-06-91								
Asegurado : CENTAUROS	←	Número de Póliza :	←							
Domicilio : PROVIDENCIA #424	←	Tipo Póliza :	B	←						
Colonia : INSURGENTES	←	Clave Agente :	0291	←						
CP : 34567	←	Movimiento :								
Teléfono : 7563478	←									
Estado : DF	←									
		<table border="1"> <tr> <td>MOVIMIENTO</td> </tr> <tr> <td>Numero de Poliza</td> </tr> <tr> <td>Numero de Endoso</td> </tr> <tr> <td>No Procede o Cancelado</td> </tr> <tr> <td>Memo Agente</td> </tr> <tr> <td>Duplicado</td> </tr> </table>			MOVIMIENTO	Numero de Poliza	Numero de Endoso	No Procede o Cancelado	Memo Agente	Duplicado
MOVIMIENTO										
Numero de Poliza										
Numero de Endoso										
No Procede o Cancelado										
Memo Agente										
Duplicado										
[↑↓] Cursor		[←] Seleccionar		[Esc] Cancelar						

Figura 5.0

Una forma utilizando menús para seleccionar una opción

Mensajes al usuario sobre la selección de teclas

En la figura 5.0 se muestra una forma solicitando los datos de asegurado, esta forma puede ser de tres tipo alta, baja o modificación.

La interface de menús se contempla en todos los módulos del sistema, un ejemplo de esta interface, es la que se presenta en la figura 5.1.

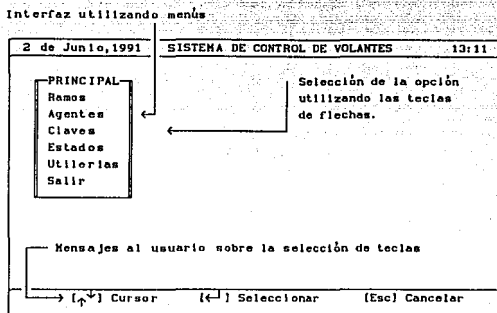


Figura 5.1

En la figura 5.1 se muestran los 5 módulos del sistema : Ramos, Agentes, Claves, Estados y Utilerias. El usuario solo tiene que mover el cursor hacia la opción deseada y presionar la tecla de [←] o [Enter]. En ese momento el sistema le responderá con otro menú, con una forma o bien con una pregunta donde el usuario tendrá que responder.

Para el tercer tipo de interfaz que se realiza a base de preguntas y repuestas, el sistema en varias partes pregunta y el usuario tiene que responder, a veces tecleando los datos necesarios o bien seleccionado la información de un menú. Un típico ejemplo, es la configuración de los renglones de la impresora, donde el usuario tiene que responder a cuántos renglones desea la hoja.

Durante el capítulo VI se podrán observar a detalle cada uno de los módulos, con sus pantallas y sus respectivas opciones. Así como también cuales son la respuestas a cada una de la opciones que solicita el sistema.

5.3 Selección del sistema operativo de red

Como resultado de la aparición de microcomputadoras (PC's), las empresas han podido aprovechar nuevas facilidades, incrementado su productividad en aspectos de oficina, control de proyectos, y otros entornos.

Sin embargo, el gran incremento que provoca la utilización de estas herramientas se ha visto limitado con la capacidad de compartir resultados, falta de control y seguridad en la información. Otro factor que ha limitado la productividad de la tecnología de PC's, es el costo de los aditamentos y periféricos.

Todos estos factores han dado lugar al surgimiento del concepto de "Redes locales de microcomputadoras", a través del cual se puede alcanzar un mejor aprovechamiento de los recursos de cómputo, compartiendo sus componentes y permitiendo controlar eficazmente la información y herramientas utilizadas en la aplicación de soluciones.

Para el desarrollo de esta tesis, se ha elegido a las redes locales de microcomputadoras, por ser las que en un futuro se convertirán en el estandar de la industria, claro que sin dejar los equipos de cómputo grandes como lo son las mainframes o minicomputadoras.

Actualmente los sistemas operativos de red que mayor tendencia han tenido en el mercado nacional son de 4 fabricantes :

- 3 Com
- Digital
- Hewlett Packard
- Novell

En los apéndices del b al e se describen a detalle cada uno de los sistemas operativos de los fabricantes mencionados anteriormente.

Al analizar los diferentes sistemas operativos, observaremos que Novell tienen actualmente un 75% del mercado nacional, convirtiendose en el fabricante número 1.

Novell es una empresa que se dedica a las redes locales basada en máquinas microcomputadoras y ofrece una gran variedad de opciones para el manejo de la red, con software y hardware de buena calidad. Novell es una empresa principalmente dedicada a soluciones de software, situación que motivo a que el desarrollo de sus programas buscará la independencia de hardware.

La razón principal por la que se eligio Novell como sistema operativo es su gran independencia que existe con el hardware, además de su gran facilidad para desarrollar aplicaciones en cualquier lenguaje de programación que se conozca.

Sus API's (Application Program Interfase) que son cerca de 100, tienen una enorme gama de rutinas que pueden ser utilizadas, como herramientas del sistema. Novell tiene la versatilidad de ser además 100 % compatible con NetBios. Si por ejemplo, se desarrolla una aplicación bajo los API's de Novell, esta no correrá en ninguno de los otros 3 sistemas operativos. Sin embargo, si se utiliza Netbios, el sistema o aplicación que se desarrolle correrá bajo cualquier medio ambiente.

Novell tiene la capacidad de que cualquier aplicación que corra en una PC en forma de monousuario, correrá en su sistema operativo en forma múltiusuario, siempre y cuando se tomen las consideraciones necesarias para su correcta implementación. Como son el manejo de candados (Lookings) a nivel registro y archivo sobre todo en aplicaciones que manejen bases de datos.

CAPITULO VI

Documentación y manual del sistema.

6.1 Documentación del sistema

El término documentación del sistema se refiere a todos los documentos que pertenecen a la aplicación del éste. desde la especificación de los requisitos del mismo, hasta el plan de aceptación final. Los documentos que describen el diseño, aplicación y pruebas del sistema son esenciales, si se quiere comprender y dar mantenimiento al programa. Los información que compone tal documentación deben contemplar :

1. La definición de requisitos.
2. Por cada programa del sistema, una breve descripción de como se descompone y una especificación de ese nuevo componente
3. Por cada unidad (programa o rutinas)una descripción de su operación. Esta se puede llevar acabo utilizando comentarios dentro del programa.
4. Un plan de pruebas amplio que describa como se prueba cada unidad del programa.
5. Un plan de pruebas que muestre como se realizó la prueba de integración, esto es la prueba de todas las unidades y programas juntos.

6.1.1 Diccionario de datos

El diccionario de datos es un documento que proporciona detalles de todas y cada una de la entidades relevantes para el sistema que se describe.

Uno de los objetivos básicos de un diccionario de datos es permitir el manejo y la documentación de los datos. Puesto que la base de datos sirve a varios usuarios, es vital que cada uno de ellos entienda precisamente qué son los datos y qué significan.

6.1.1.1 Estructuras del sistema

Nombre	Descripción
As_1I	Registro de datos del asegurado para Incendio
As_1A	Registro de datos del asegurado para Autos
As_1D	Registro de datos del asegurado para Diversos
As_1R	Registro de datos del asegurado para RCG
As_1T	Registro de datos del asegurado para Transporte
As_2I	Registro de información correspondiente al volante para Incendio
As_2A	Registro de información correspondiente al volante para Autos
As_2D	Registro de información correspondiente al volante para Diversos
As_2R	Registro de información correspondiente al volante para RCG
As_2T	Registro de información correspondiente al volante para Transporte

Nombre	Descripción
As_3I	Registro de fechas por departamento para Incendio
As_3A	Registro de fechas por departamento para Autos
As_3D	Registro de fechas por departamento para Driversos
As_3R	Registro de fechas por departamento para RCG
As_3T	Registro de fechas por departamento para Transporte
Cat_Age	Catálogo de Agentes
Cat_Clv	Catálogo de claves de personas de los departamentos de Cálculo, Mecanografía y revisión.
Cat_Est	Catálogo de estados
Cat_Ser	Catálogo de servidores de archivos
Ramos	Registro que conserva el último pendiente para los 5 ramos.
Bit_Fld	Registro de bitacora de operaciones no realizadas

6.1.1.2 Diccionario de Datos del sistema

Nombre	Tipo	Long	Descripción
• N_Volante	Char	5	Número de Volante
Asegurado	Char	48	Nombre del asegurado
Domicilio	Char	30	Domicilo del asegurado
Colonia	Char	20	Colonia del asegurado
CP	Char	5	Codigo Postal
Teléfono	Char	7	Teléfono del asegurado
T_Póliza	Char	1	Tipo de póliza
N_Póliza	Char	7	Número de póliza
Movimiento	Char	10	Movimiento del documento
O_Trabajo	Date	8	Fecha de Orden de Trabajo

Nombre	Tipo	Long	Descripción
Cálculo	Date	8	Fecha de Cálculo
Registro	Date	8	Fecha de Registro
Reaseguro	Date	8	Fecha de Reaseguro
Cod_1	Date	8	Fecha de Codificación 1
UDS	Date	8	Fecha de UDS
Cod_2	Date	8	Fecha de Codificación 2
Mecano	Date	8	Fecha de Mecanografía
Revisión	Date	8	Fecha de Revisión
● C_Agente	Char	4	Clave de agente
NombreA	Char	30	Nombre del agente
Oficina	Numeric	2	Oficina del agente
C_EstadoA	Char	2	Clave estado de agente
● C_Persona	Char	2	Clave de persona
NombreP	Char	30	Nombre de la persona
C_Dep	Numeric	1	Número de departamento
● C_Estado	Char	2	Clave de estado
N_Estado	Char	25	Nombre de estado
● Server	Char	20	Nombre del servidor de archivos
Drive	Char	1	Drive de asignación en la red
Ruta	Char	20	Ruta de asignación
Incendio	Char	5	Ramo de Incendio
Autos	Char	5	Ramo de Autos
Diversos	Char	5	Ramo de Diversos
RCG	Char	5	Ramo de RCG
Transporte	Char	5	Ramo de Transporte
B_Local	Char	20	Server donde hubo falla
B_Archivo	Char	12	Nombre del archivo que fallo
B_Recno	Numeric	6	Número de registro
B_Campo	Char	15	Nombre del campo
B_Tipo	Char	1	Tipo de campo
B_Val_Old	Char	20	Valor anterior
B_Val_New	Char	20	Valor nuevo
B_Fecha	Date	8	Fecha de falla
B_Ok	Logical	1	Registro a ser actualizado
B_Operación	Char	1	Operacion A:Alta B:Baja C:Cambio

6.1.1.3 Entidades

Entidades : As_1I, As_1A, As_1D, As_1R, As_1T

N_Volante
Asegurado
Domicilio
Colonia
Cp
C_Estado
Teléfono

Entidades : As_2I, As_2A, As_2D, As_2R, As_2T

N_Volante
T_Póliza
N_Póliza
Movimiento
C_Agente

Entidades : As_3I, As_3A, As_3D, As_3R, As_3T

N_Volante
O_Trabajo
Calculo
Registro
Reaseguro
Cod_1
UDS
Cod_2
Mecano
Revisión
C_Persona

Entidades : Cat_Age

C_Agente
NombreA
Oficina
C_EstadoA

Entidades : Cat_Clv

C_Persona
NombreP
C_Dep

Entidades : Cat_Est

C_Estado
N_Estado

Entidades : Cat_Ser

Server
Drive
Ruta

Entidades : Ramos

Incendio
Autos
Diversos
RCG
Transporte

Entidades : Bit_Fld

B_Local
B_Archivo
B_Indice
B_Recno
B_Campo
B_Tipo
B_Val_Old
B_Val_New
B_Fecha
B_Operador
B_Ok

6.1.2 Documentación del usuario

La documentación proporcionada a los usuarios suele ser el primer contacto que estos tienen con el sistema, tal documentación debe proporcionar una versión inicial precisa del sistema. La información se debe estructurar de forma que el usuario pueda leerla con el grado de detalle apropiado a sus necesidades.

En ella se deben contemplar las siguientes características :

1. Una descripción funcional sobre lo que puede hacer el sistema.
2. Una descripción que explique cómo instalar el sistema y adecuarlo para configuraciones particulares del hardware.
3. Una breve descripción que explique en términos sencillos, cómo iniciarse en el sistema y cómo opera el sistema.

6.1.2.1 Descripción del sistema

La idea básica del sistema de control de volantes, es administrar correctamente el flujo de volantes que existe a través de los diferentes ramos de la subdirección de Emisión daños de cualquier compañía de seguros, aplicando los conceptos de bases de datos distribuidas y redes locales de microcomputadoras.

El sistema contempla las principales herramientas para llevar a cabo este objetivo. En cualquiera de los módulos se contemplan altas, bajas y cambios en cualquier módulo, además de una serie de reportes de productividad que permiten la ayuda en la toma de decisiones. Entre los principales reportes se contemplan, el flujo de volantes que existe en cada uno de los departamentos, los volantes que no han terminado su proceso, los volantes que fueron

cancelados, cual es el último movimiento de volantes, que no han concluido su proceso, cual es el tiempo de proceso por departamento, el tiempo promedio en Emisión, es decir cuantos días se esta tardando la Emisión, cual es el tiempo de proceso por departamento, cual es el volumen de entradas y salidas y el volumen en proceso. Todos estos reportes son impresos de tal foma que el usuario pueda entenderlos fácilmente.

La ventaja principal del sistema es que trabaja bajo un ambiente de redes Novell, en donde los usuarios pueden compartir recursos e información en forma transparente.

El sistema contempla además una serie de utilerías en caso de que un servidor de archivos no este funcionando, un archivo se encuentre dañado, o bien un registro no este disponible en un momento dado. Cada uno de estos movimientos que no se puede efectuar es registrado en una bitacora, para que posteriormente esta se pueda recuperar y las actualizaciones se lleven acabo en el punto donde no se lograron concluir.

6.1.2.2 Requerimientos de hardware

El sistema requiere básicamente de una red local, que opere bajo sistema operativo Netware de Novell con versión 2.0A o mayor (No importa la topologia que se use). Estaciones de trabajo 100% compatibles con IBM, y con un mínimo de memoria de 512K. El número de estaciones de trabajo, esta limitado a la versión de sistema operativo que se use.

El servidor de archivos deberá tener las siguientes características :

- 2 MB de memoria en RAM (version 2.0A a v2.15)
- 2.5 Mb de memoria en RAM (version 2.2)
- Disco duro mínimo de 40 MB.
- Impresora de 132 Columnas.
- 1 Tarjeta de Interfaz para red

Las estaciones de trabajo deberán cumplir con lo siguiente :

- 512Kb de memoria en RAM.
- Una unidad de disco flexible.
- Una tarjeta de interfaz para red.
- Sistema operativo Dos v3.3
- El software de red.
- Impresora (Opcional)

Para instalar la red local de microcomputadoras se deberá contar con lo siguiente :

- Cable suficiente para conectar las estaciones de trabajo.
- Sistema operativo de red, Netware versión 2.0A o mayor.
- Los aditamentos adecuados para la topología seleccionada. Por ejemplo, si se esta instalando una red con tarjetas de interfaz ethernet, se deberán tener los respectivos terminadores ethernet de 50Ω, conectores tipo "T" (En caso de que se use cable delgado) o los respectivos transceivers (Cable grueso).

6.1.2.3 Instalación del sistema

Para efectuar la instalación en el servidor de archivos, primeramente la red deberá estar funcionando como tal, paso seguido desde cualquier estación de trabajo se deberá acceder a la red, con la cuenta de un usuario que tenga los privilegios y derechos suficientes para modificar directorios, crear archivos, dar de alta usuarios, y en general que tenga atributos de supervisor de la red.

Una vez que el usuario se haya firmado en la red, éste podrá cambiar al drive donde se encuentra el programa de instalación y teclear simplemente *instala*.

A continuación se describe el proceso :

● Paso 1 :

Cambiar al drive donde se encuentra el programa de instalación :

A:>

● Paso 2 :

Teclear el siguiente nombre.

A:>*instala*

● Paso 3 :

Paso seguido el programa le mostrará los drives lógicos de la red, y usted seleccionará uno de ellos.

A continuación el programa buscará el subdirectorío "Volante", si el subdirectorío no existe, la instalación dará comienzo.

En caso de que el subdirectorío exista el programa le enviará un mensaje, avisándole si desea instalar en ese subdirectorío, si su respuesta en afirmativa la instalación dará comienzo, en caso contrario, el programa de instalación creará un subdirectorío que no exista.

● Paso 4 :

El programa le solicitará los discos siguientes.

● Paso 5 :

El programa finalizará la instalación y ahora el sistema se podrá utilizar, simplemente tecleando "Volante".

6.1.2.4 Arranque del sistema

Ahora que el sistema se ha instalado en la red de microcomputadoras, el usuario podrá trabajar con él. El archivo más importante es el llamado *Volante.Exe*, para ejecutar este programa primeramente se tiene que cambiar al subdirectorío donde se encuentra instalado el sistema y después teclear "VOLANTE".

Por ejemplo :

```
F:> Cd \Volante
```

Al cambiar de directorío la microcomputadora le responderá :

```
F:\VOLANTE>
```


Ahora solo tiene que teclear "Volante" para arrancar el sistema y en seguida presionar la tecla de <←> o <Return>, después de unos instantes se verá la pantalla principal del sistema (Fig 6.0).

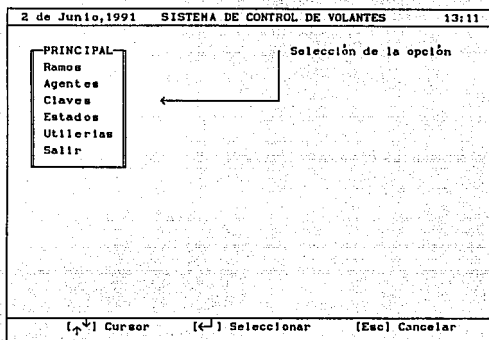


Figura 6.0

En la primera pantalla (Fig 6.0), se muestran los diferentes módulos del sistema. El primero de ellos se refiere a todos los ramos que contempla la subdirección de Emisión Daños. El segundo módulo contempla los agentes que venden seguros a todos los clientes de la compañía. El tercero se refiere a las claves de las personas de los departamentos de cálculo, mecanografía y revisión. El módulo de estados contempla los nombres de los estados de la república mexicana, así como su clave correspondiente. El módulo de utillerias contempla una serie de herramientas necesarias para administrar el sistema de control de volantes. Y por último la opción de salir, la cual abandona completamente el sistema solicitando la confirmación.

La opción de *Ramos* es la que se encarga de controlar las altas, cambios, bajas, reindexar archivos y generar reportes de los 5 ramos de la subdirección de Emisión Daños. Al seleccionar esta opción la pantalla nos responderá de la siguiente manera: (fig 6.1).

6.1.2.5 Módulo de ramos

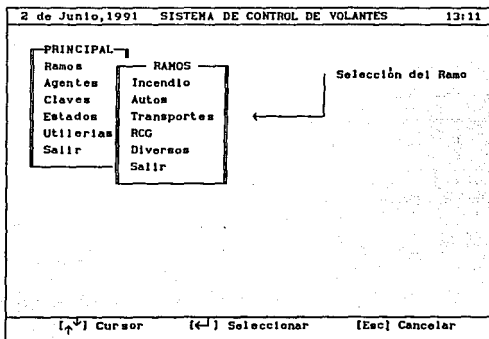


Figura 6.1

Para seleccionar el *RAMO*, solo basta colocar el cursor en el ramo deseado y pulsar la tecla de <←> o <Return>. Cada uno de los ramos tiene bases de datos diferentes, debido a que los números de volantes son independientes entre ramos. Paso seguido el sistema le responderá como en la figura 6.2.

En caso de que no exista un archivo correspondiente al ramo o bien este no se pueda utilizar, el sistema no permitirá trabajar con el ramo seleccionado, indicándole al usuario cual es el archivo que falta o en su defecto que el archivo esta ocupado (Tiene un candado "Looking").

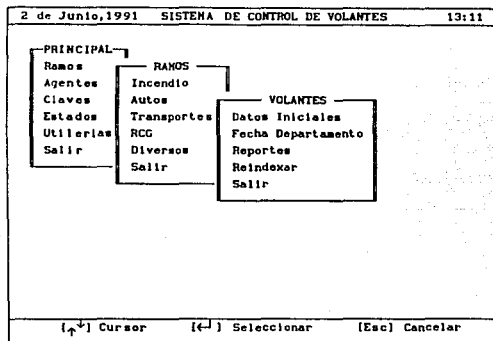


Figura 6.2

Una vez seleccionado el ramo de trabajo, se tendrá la posibilidad de agregar nuevos volantes (Datos Iniciales), cambiar fechas de departamentos y datos iniciales, generar reportes o bien, en caso de que los archivos de índices se hayan dañado, reindexar los archivos de el ramo con el que se este trabajando.

Si la opción que se selecciona es la de datos iniciales el sistema le responderá, como el de la figura Fig. 6.3.

2 de Junio, 1991		SISTEMA DE CONTROL DE VOLANTES		13:11	
-PRINCIPAL-					
DEPARTAMENTO INCENDIO					
Num. Volante	Ord. Trabajo	Tipo Poliza	Num. Poliza	Clv. Agen	
00001	14-06-91	P	6000735	0000	
00002	14-06-91	P	6000731	1023	
00003	14-06-91	R		XXXX	
00004	14-06-91	D	6000400	4567	
00005	14-05-91	D		3456	
00006	17-05-91	A	6000734	1456	
00007	17-05-91	P	6000800	5671	
00008	17-05-91	P	6000813	XXXX	
00009	17-05-91	P		XXXX	
00010	18-05-91	B	6000712	0000	
00011	18-05-91	R	6000804	0862	
00012	19-05-91	P	6000810	7713	
00013	20-05-91	A		1313	

[↑↓] Cursor [←] Seleccionar [Esc] Cancelar

Figura 6.3

En esta parte del sistema, se podrán agregar, borrar o modificar registros.

Existen algunas teclas que merecen mención especial, por ejemplo:

- [Alt S] Búsqueda de un número de volante. Solicita el número de volante y se posiciona en el volante solicitado en caso de que no exista el cursor se coloca en el más cercano.
- [Del] Elimina un registro de la base de datos, solicitando primeramente su confirmación.
- [Ins] Inserta un nuevo registro.
- [Esc] Regresa al menú anterior cancelando cambios.
- ← ↑ ↓ → Mueve cursor en los direcciones correspondientes.

- [Enter] Presenta los datos que van a ser cambiados.

Para agregar un nuevo registro se deberá presionar la tecla de <Ins>. Paso seguido el sistema le responderá de la siguiente manera : (Fig. 6.4)

20 de Junio, 1991		SISTEMA DE CONTROL DE VOLANTES		13:11	
PRINCIPAL					
ALTA DE VOLANTE					
Número de Volante : 00057			Orden de trabajo : 20-06-91		
Asegurado : CENTAUROS			Número de Póliza :		
Domicilio : PROVIDENCIA #42			Tipo Póliza : B		
Colonia : INSURGENTES			Clave Agente : 0000		
CP : 34567			Movimiento :		
Teléfono : 7563478					
Estado : DF					
[↑] Cursor		[←] Seleccionar		[Esc] Cancelar	

Figura 6.4

Como se observa el número de volante y fecha de orden de trabajo es generado automáticamente por el sistema, los demás datos deberán ser capturados por el usuario.

Para que el registro sea dado de alta en la base de datos, se tiene que especificar el nombre del asegurado, en caso contrario nunca se grabará en la base de datos.

Existen 4 campos que merecen mención aparte y son: Estado, Tipo de Póliza, Clave agente y Movimiento ya que estas despliegan catálogos con información establecida en otros módulos o bien son parte del sistema (Tipo póliza y movimiento).

Para el caso de estado, el catálogo que se muestra tiene información de los estados y su clave correspondiente. Fig. 6.5

20 de Junio, 1991 SISTEMA DE CONTROL DE VOLANTES		13:11
PRINCIPAL		
ALTA DE VOLANTE		
Número de Volante : Asegurado : CENTAUR Domicilio : PROVIDE Colonia : INSURGE CP : 34567 Teléfono : 7563478 Estado :	ESTADO BAJA CALIFORNIA NORTE → BN BAJA CALIFORNIA SUR → BS CAMPECHE → CA CHIHUAHUA → CH COLIMA → CL COAHUILA → CO CHIAPAS → CP DISTRITO FEDERAL → DF DURANGO → DU ESTADO DE MEXICO → EM GUANAJUATO → GU GUERRERO → GU HIDALGO → HD JALISCO → JL	-91 P 0000
{↑↓} Cursor {←} Seleccionar {Esc} Cancelar		

Figura 6.5

Al visualizar el catálogo de estados el usuario tendrá la posibilidad de seleccionar el estado deseado, en caso de no seleccionar ninguno el sistema asignará automáticamente "XX" para la clave de estado.

En caso de que el usuario capture una clave de estado válida, el catálogo de estados no se desplegará, en caso contrario se mostrará.

El campo tipo de póliza (fig. 6.6) nos muestra las 5 opciones que se pueden seleccionar de acuerdo al tipo de documento que se necesite. El usuario tiene además la posibilidad de capturar las letras P (Póliza), A (Endoso A), B (Endoso B), D (Endoso D) y R (Renovación) sin tener que seleccionarlas dentro del menú.

20 de Junio, 1991		SISTEMA DE CONTROL DE VOLANTES		13:11
PRINCIPAL				
ALTA DE VOLANTE				
Número de Volante : 00057		Orden de trabajo : 20-06-91		
Asegurado : CENTAUROS		Número de Póliza :		
Domicilio : PROVIDENCIA #42		Tipo Póliza :		B
Colonia : INSURGENTES		Clave Agente		TIPO
CP : 34567		Novialento		Póliza
Teléfono : 7563478				Endoso A
Estado : DF				Endoso B
				Endoso D
				Renovación
<input type="text"/>				
[↑↓] Cursor		[←] Seleccionar		[Esc] Cancelar

Figura 6.6

El campo de Clave de agente nos muestra un catálogo con las claves de todos los agentes (fig 6.7) que existen en la compañía de seguros. Si el usuario captura la clave y esta se encuentra registrada en el catálogo, entonces no se desplegará el catálogo. Si usuario no captura o selecciona la clave, el sistema asume en forma automática que es un registro directo, y el sistema asignará a ese registro "XXXX".

20 de Junio, 1991		SISTEMA DE CONTROL DE VOLANTES		13:11
PRINCIPAL				
ALTA DE VOLANTE				
Número de Volante : 00057		Orden de trabajo : 20-06-91		
Asegurado : CENTAUROS	Número de Póliza :			
Domicilio : PROVIDENCIA #42	Tipo Póliza : B			
Colonia : INSURGENTES	Clave Agente :			
CP : 34567	Movimiento		0000	
Teléfono : 7563478			0001	
Estado : DF			0010	
			0015	
			0020	
			0025	
			0030	
			0031	
			0035	
[↑↓] Cursor			[←] Seleccionar	
[Esc] Cancelar				

Figura 6.7

El campo de movimiento nos muestra un catálogo con las 5 opciones posibles que puede tener un documento (Fig 6.8).

Para seleccionar el número de póliza o número de endoso, es necesario que primero se capture el número de póliza. Si se selecciona una de las otras opciones el sistema desplegará en el campo de captura la información necesaria.

Una vez que se ha terminado de capturar toda la información, para que esta sea actualizada en los archivos correspondientes, es necesario que el usuario presione la tecla de [PgDn], y en seguida los datos serán grabados, enseguida el sistema nos mostrará una pantalla como la de la figura 6.3.

20 de Junio, 1991		SISTEMA DE CONTROL DE VOLANTES		13:11
--PRINCIPAL--				
ALTA DE VOLANTE				
Número de Volante : 00057		Orden de trabajo : 20-06-91		
Asegurado : CENTAUROS	Número de Póliza :			
Domicilio : PROVIDENCIA #42	Tipo Póliza : B			
Colonia : INSURGENTES	Clave Agente : 0291			
CP : 34567	Movimiento :			
Teléfono : 7563478				
Estado : DF				
			MOVIMIENTO	
			Número de Póliza	
			Número de Endoso	
			No Procede o Cancelado	
			Memo Agente	
			Duplicado	
[↑↓] Cursor		[←] Seleccionar		[Esc] Cancelar

Figura 6.8

Al presionar la tecla de [Del] el sistema nos mostrará el registro que se desea borrar (Fig 6.9). Si el usuario desea eliminar este registro, solamente tendrá que confirmar la eliminación.

El sistema se encargará de darlo de baja en forma automática.

Es recomendable que cada vez que se efectue la eliminación de un registro o varios, se reindexe la base de datos, para así asegurar la eliminación física de los registros de la base de datos.

20 de Junio, 1991		SISTEMA DE CONTROL DE VOLANTES		13:11
-PRINCIPAL-				
MODIFICACION DE VOLANTE				
Número de Volante : 00057		Orden de trabajo : 20-06-91		
Asegurado : CENTAUROS		Número de Póliza :		
Domicilio : PROVIDENCIA #42		Tipo Póliza : B		
Colonia : INSURGENTES		Clave Agente : 0000		
CP : 34567		Movimiento :		
Teléfono : 7563478				
Estado : DF				
¿ Se elimina ?				
SI No				
[↑↓] Cursor				[Esc] Cancelar

Figura 6.9

Una de las características que hacen que el sistema sea muy versátil, es su fácil manejo, por ejemplo, si una base de datos tiene 3000 registros y el cursor se encuentra en el registro número 1, para llegar a editar el registro 2995 se tendrán que dar varias veces [PgDn] o bien hacerlo en forma directa al presionar las teclas de [Alt S], en ese momento el sistema preguntará el número de volante al que deseamos ir. Para llegar hasta él, simplemente tecleamos el número y el sistema se posicionará en el, en caso de que no llegará a existir el número de volante. entonces el sistema se colocará el cursor en el más cercano (Fig 6.10).

2 de Junio, 1991		SISTEMA DE CONTROL DE VOLANTES		13:11	
-PRINCIPAL-					
DEPARTAMENTO INCENDIO					
Num. Volante	Ord. Trabajo	Tipo Poliza	Num. Poliza	Clv. Agen	
00001			000735	0000	
00002			000731	1023	
00003				XXXX	
00004	14-06-91	D	6000400	4567	
00005	14-05-91	D		3456	
00006	17-05-91	A	6000734	1456	
00007	17-05-91	P	6000800	5671	
00008	17-05-91	P	6000813	XXXX	
00009	17-05-91	P		XXXX	
00010	18-05-91	B	6000712	0000	
00011	18-05-91	R	6000804	0862	
00012	19-05-91	P	6000810	7713	
00013	20-05-91	A		1313	

Número de Volante : 5671

[↵] Cursor [←] Seleccionar [Esc] Cancelar

Figura 6.10

Para modificar las fechas de volantes se tendrá, primeramente que seleccionar la opción de Fecha Departamento (Fig 6.2), una vez seleccionada esta opción, el sistema nos mostrará una pantalla semejante a la de la figura 6.11.

En la figura 6.11, el usuario tendrá la posibilidad de seleccionar un volante a la vez, para así modificar las fechas de los departamentos. La selección de este volante se lleva a cabo con solo presionar la tecla de [Enter] o [↵].

Una vez presionada la tecla de enter el sistema nos mostrará una pantalla semejante a la de la figura 6.12. En donde se muestran todos los departamentos con sus respectivas fechas y movimientos que ha tenido o bien que todavía no tiene.

22 de Junio, 1991		SISTEMA DE CONTROL DE VOLANTES		13:11	
-PRINCIPAL-					
DEPARTAMENTO INCENDIO					
Num. Volante	Ord. Trabajo	Cálculo	Clv. Pers.	Registro	
00001	14-06-91	17-06-91	c1	19-06-91	
00002	14-06-91	17-06-91	c2	20-06-91	
00003	14-06-91	17-06-91	c3	18-06-91	
00004	14-06-91			18-06-91	
00005	14-05-91	18-06-91	c1		
00006	17-05-91	18-06-91	c2		
00007	17-05-91				
00008	17-05-91	20-06-91	c1	21-06-91	
00009	17-05-91				
00010	18-05-91				
00011	18-05-91				
00012	19-05-91			21-06-91	
00013	20-05-91	21-06-91	c3		

[↑] Cursor [←] Editar [Del] Borrar [Ins] Insertar

Figura 6.11

Para modificar la fecha de un volante, solamente se tendrá que teclear la nueva fecha en el departamento deseado. Esta operación se tendrá que realizar en cada uno de los departamentos donde se desea cambiar la fecha.

Cabe mencionar que en los departamentos de cálculo, mecanografía y revisión el sistema solicitará la clave de la persona que tiene ese volante como se muestra en la figura 6.13.

El campo de movimiento tiene 5 opciones que son desplegables a través de un menú, como se muestra en la figura 6.13.

Al finalizar la captura, para que esta sea grabada en los archivos correspondientes, el usuario solo tendrá que presionar la tecla de [PgDn], y en seguida los datos serán actualizados, paso seguido el sistema mostrará una pantalla como la de la figura 6.11.

PRINCIPAL		FECHAS ACTUALIZACION	
Num. Volante	Número de Volante: 00001	rs.	Registro
00001	Orden de Trabajo : 14-06-91		19-06-91
00002	Cálculo : 17-06-91		20-06-91
00003	Registro : 19-06-91		18-06-91
00004	Reaseguro :		18-06-91
00005	Codificación 1 :		
00006	UDS :		
00007	Codificación 2 :		
00008	Hecanografía :		
00009	Revisión :		21-06-91
00010	MOVIMIENTO :		
00011			
00012			21-06-91
00013			

[↑] Cursor [←] Editar [Del] Borrar [Ins] Insertar

Figura 6.12

PRINCIPAL		FECHAS ACTUALIZACION	
Num. Volante	Número de Volante: 00001	rs.	Registro
00001	Orden de Trabajo : 14-06-91	Clave	-06-91
00002	Cálculo : 17-06-91	MG	-06-91
00003	Registro : 19-06-91	Az	-06-91
00004	Reaseguro :	TA	-06-91
00005	Codificación 1 :	RQ	
00006	UDS :	MA	
00007	Codificación 2 :	JR	
00008	MOVIMIENTO		
00009	Número de Póliza		-06-91
00010	Número de Endoso		
00011	No Procede o Cancelado		
00012	Nemo Agente		
00013	Duplicado		21-06-91
00014			

[↑] Cursor [←] Editar [Del] Borrar [Ins] Insertar

Figura 6.13

Una vez que la información se ha comenzado a capturar, el sistema estará listo para presentar los datos en diferentes formas. Los 9 diferentes reportes (fig 6.14) se pueden imprimir de 2 formas distintas, la primera de ellas, es mandarlos directamente a la impresora y la segunda es grabarlos a un archivo para su uso posterior.

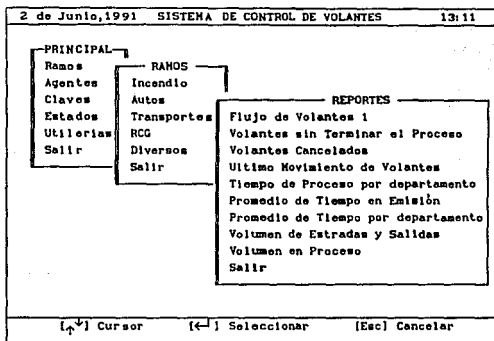


Figura 6.14

Para generar un reporte por impresora, esta deberá estar lista, es decir encendida y on-line.

El primer reporte (Fig 6.15), llamado flujo de volantes 1, presenta datos como son : el número de volante, nombre del asegurado, Tipo de póliza, número de póliza, y las fechas de todos los departamentos, haya o no circulado por estos, así como cual es su movimiento final.

Este reporte puede ser generado en dos formas diferentes, a partir del primer pendiente o bien especificando 2 fechas, una inicial y una final.

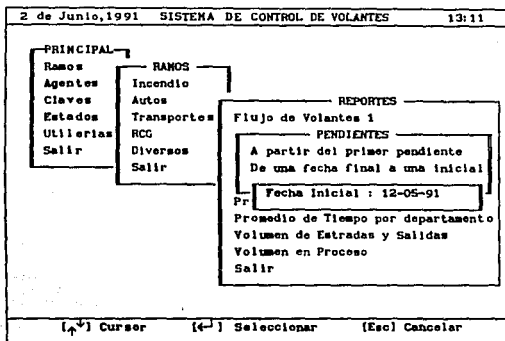


Figura 6.14

Para la primera opción de imprimir todos los volantes que se encuentran después del primer pendiente. Para la segunda opción se tienen que especificar 2 fechas a través de las cuales el reporte sera impreso, esto es, los volantes que se encuentren en el rango establecido.

El reporte de volantes sin terminar el proceso (Fig 6.16) imprime datos como: Número de volante, Nombre del asegurado, Tipo de póliza, Número de póliza, la fecha del departamento de orden de trabajo, la fecha del departamento donde se encuentra actualmente, así como el nombre de ese departamento. Si el departamento corresponde a cálculo, mecanografía o revisión se imprime la clave de la persona que tiene ese volante.

Este reporte nos brinda la posibilidad de imprimirlo en dos formas diferentes, la primera son todos los pendientes que no han terminado su proceso y que tienen de 8 a 14 días en proceso, y la segunda son todos los pendientes que tienen más de 15 días en proceso.

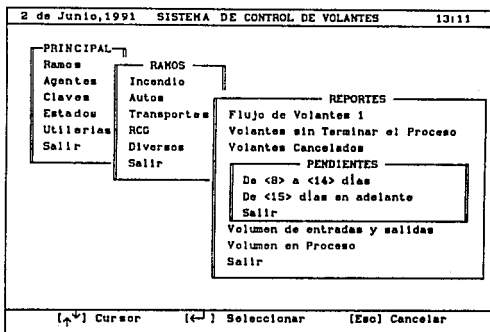


Figura 6.16

El reporte llamado volantes cancelados (fig. 6.17), genera la siguiente información : número de volante, nombre del asegurado, clave del agente, fecha de orden de trabajo, fecha en que se cancelo y el número de días que se tardo antes de que fuera cancelado. Para generar este reporte solo se tiene que especificar una fecha con la cual dara comienzo el reporte.

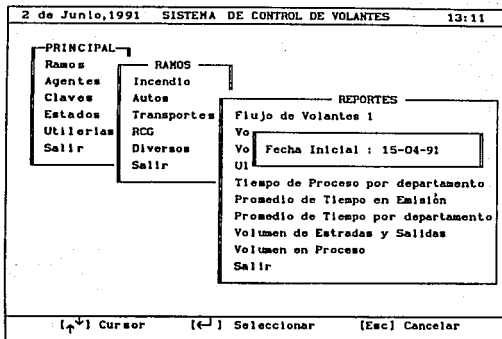


Figura 6.17

El reporte de último movimiento de volantes (fig 6.18) se imprime información como : número de volante, el nombre del asegurado, tipo de póliza, número de póliza, fecha de orden de trabajo, fecha y nombre del departamento donde se encuentra actualmente, y en caso de que el departamento corresponda a cálculo, mecanografía o revisión se agrega la clave de la persona que tiene ese documento.

Al imprimir el reporte se tiene que indicar un número el cual deberá estar entre 0 y 4.

El rango que se establece tiene un significado donde :

Número

Significado

- | | |
|---|---|
| 0 | Se imprimen todos los volantes pendientes. |
| 1 | Se imprimen todos los volante pendientes y los terminados un día anterior. |
| 2 | Se imprimen todos los volantes pendientes y los terminados dos días anteriores. |
| 3 | Se imprimen todos los volantes pendientes y los terminados 3 días anteriores. |
| 4 | Se imprimen todos los volantes pendientes y los terminados 4 días anteriores. |

2 de Junio, 1991 SISTEMA DE CONTROL DE VOLANTES 13:11

PRINCIPAL	
Ramos	RAMOS
Agentes	Incendio
Claves	Autos
Estados	Transportes
Utilerias	RCG
Salir	Diversos
	Salir

REPORTES	
Flujo de Volantes 1	
Volantes sin Terminar el Proceso	
Volantes Cancelados	
Ultimo Movimiento de Volantes	
Tie	
Pro	Cuántos días 0..4 ? 2
Pro	
Volumen de Entradas y Salidas	
Volumen en Proceso	
Salir	

[↑↓] Cursor [←] Seleccionar [Esc] Cancelar

Figura 6.18

Los reportes de tiempo de proceso por departamento, promedio de tiempo en Emisión, promedio de tiempo por departamento, volumen de entradas y salidas, brindan dos posibilidades de impresión (Fig 6.19), una semana anterior a la fecha del día de su microcomputadora, o bien a partir de una fecha deseada.

Si la opción que seleccione es a partir de una fecha determinada, se tendrá que indicar cuál es la fecha.

El reporte de tiempo de proceso por departamento nos da la siguiente información, número de volante, fecha de inicio, fecha de terminación, cuál fue el documento que se generó, y cuántos días se tardó en ser procesado. Así como un promedio en días.

En el reporte de promedio de tiempo en Emisión se podrá conocer cuántos fueron los documentos que se tardaron un 1 día, 2 días, ..., 10 días etc. en ser procesados. El reporte muestra el número de documentos, el número de días y el total de días, así como un promedio indicando a cuántos días se esta tardando la Emisión.

El reporte de tiempo por departamento identifica cuantos fueron los documentos que se tardaron 1 día, 2 días, ..., 10 días, etc., en ser procesados por departamento. El reporte general número de documentos, el número de días y el total de días, así como un promedio, indicando cuántos días se esta tardando la Emisión por departamento.

El reporte de volumen de entradas y salidas nos brinda la posibilidad de determinar cuántos volantes entraron y cuántos salieron por día, e inclusive nos da información de promedios semanales y promedios generales.

El reporte de volumen en proceso no da información por cada uno de los departamentos, es decir cuántos volantes se están procesando por departamento.

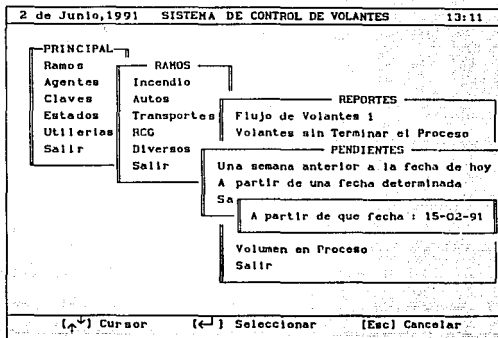


Figura 6.19

Si se detectan fallas en el funcionamiento de un ramo del sistema, posiblemente un archivo de índices este dañado. Estas fallas, se pueden detectar de la siguiente manera : Por ejemplo , si se da de alta un registro con número de volante 1000 y posteriormente el usuario se da cuenta de que uno de los datos no fue capturado correctamente, entonces el usuario procede a realizar el cambio, si al especificar el número de volante el sistema le manda un mensaje de error o bien no lo encuentra entonces eso indicará que los archivos de índices estan dañados. Para corregir esa falla, simplemente reindexe la base de datos y el problema se solucionará (Fig 6.20).

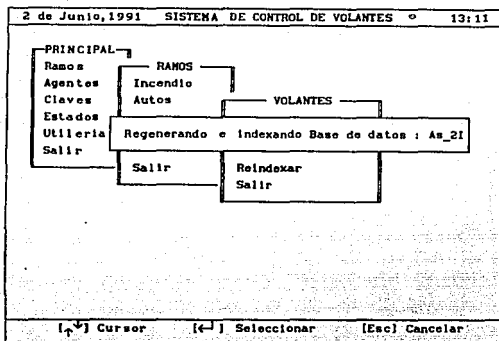


Figura 6.20

6.1.2.6 Módulo de Agentes

El módulo de agentes nos permite capturar la información correspondiente a todos los agentes de seguros que laboran para la compañía. Este módulo es muy importante, ya que ellos son los que se encargan de vender los seguros a los clientes (Fig 6.21).

El módulo de agentes, contempla básicamente 3 funciones: Actualizaciones (Altas, Cambios y Bajas), regenerar los índices (reindexar) de las bases de datos correspondientes y generar reportes de toda la información de la base de datos.

Para seleccionar una de las opciones anteriores solo bastará colocar el cursor en la opción deseada, paso seguido presionar la tecla de [Enter] o [↵], y el sistema tomará la acción correspondiente.

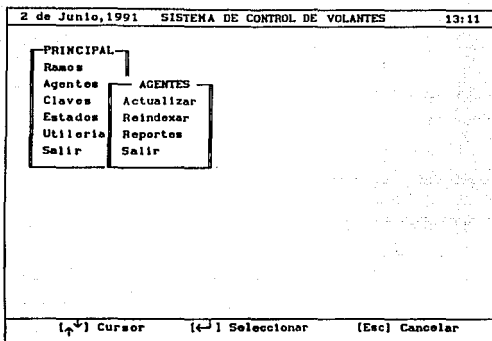


Figura 6.21

Si la opción que se elige es la de actualizar el sistema nos mostrará una pantalla como la de la figura 6.22.

En esta parte del sistema se podrán agregar, borrar o modificar registros.

Existen algunas teclas que merecen, mención especial, por ejemplo :

- [Alt S] Búsqueda de una clave de agente. Solicita la clave del agente y se posiciona en la clave solicitada, en caso de que no exista, el cursor se colocará en el más cercano.
- [Del] Elimina un registro de la base de datos, solicitando primeramente su confirmación.

- [Ins] Inserta un nuevo registro.
- [Esc] Regresa al menú anterior cancelando cambios.
- ← → Mueve el cursor en las direcciones correspondientes.

2 de Junio, 1991		SISTEMA DE CONTROL DE VOLANTES		13:11
-PRINCIPAL-				
AGENTES				
Clave Agente	Nombre Agente	Oficina	Estado	
0000	Gabriel Garcia Enriquez	10	AG	
0001	Ana Hernandez Gonzalez	11	DF	
0010	Carlos Fernandez Campos	01	DF	
0011	Olivia Guerrero Sanchez	01	DF	
0015	Pablo Gonzalez Perez	09	DF	
0018	Manuel Garcia Rios	10	EM	
0020	David Mendoza Goren	34	CA	
0021	Genaro Cervantes Garcia	45	SP	
0023	Isabel Otero Vera	10	JL	
0024	Carlos Almandoz Vargas	01	GU	
0049	Carlos Espinoza Garcia	09	CA	
XXXX	Directo	01	DF	

[←] Cursor [↵] Editar [Esc] Cancelar [Ins] Insertar

Figura 6.23

Para agregar un nuevo registro se deberá presionar la tecla de [Ins]. Paso seguido el sistema colocará el cursor al inicio de la pantalla, y el usuario estará en posibilidades de capturar su información de nuevo.

Esta captura se realiza en forma ordenada, donde primero se especifica la clave del agente, en seguida el nombre del agente, la oficina y el estado que es seleccionado a través de un catálogo (Fig 6.23) de información.

PRINCIPAL		AGENTES		Estado
Clave Agente	N			
0000	G	BAJA CALIFORNIA NORTE	→	BN
0001	A	BAJA CALIFORNIA SUR	→	BS
0010	C	CAMPECHE	→	CA
0011	O	CHIHUAHUA	→	CH
0015	P	COLIMA	→	CL
0018	M	COAHUILA	→	CO
0020	D	CHIAPAS	→	CP
0021	C	DISTRO FEDERAL	→	DF
0023	I	DURANGO	→	DU
0024	C	ESTADO DE MEXICO	→	EM
0049	C	GUNAJUATO	→	GJ
XXXX	D	GUERRERO	→	GU
		HICALGO	→	HD

[↑] Cursor [←] Editar [Esc] Cancelar [Ins] Insertar

Figura 6.23

Cabe mencionar que si la clave del agente ya se encuentra registrada, el sistema nos mandará un error.

Para eliminar un registro solo bastará presionar la tecla de [Del] y en seguida el sistema nos pedirá la confirmación, y en caso de ser afirmativo el registro se eliminará.

Para modificar el dato de un registro, solo bastará posicionarse en el dato deseado y presionar la tecla de [←] o [Enter]. En caso de que se modifique la clave del estado el sistema mostrará un catálogo con todos los estados de la república mexicana (Fig 6.23).

Si la clave del agente es modificada esta afectará otras bases de datos, tanto del servidor de archivos donde se encuentran directamente los datos, como a las otras bases de datos, en los otros servidores. El proceso de modificar la clave del agente se puede notar muy lento, debido a la gran cantidad de verificaciones que se realizan.

La opción de reindexar (Fig. 6.24) regenera los índices de las bases de datos correspondientes al módulo de agentes. Este proceso, es recomendable correrlo cuando se han eliminado uno o varios registros.

2 de Junio, 1991		SISTEMA DE CONTROL DE VOLANTES		13:11
PRINCIPAL				
Ramos				
Agentes		AGENTES		
Claves		Actualizar		
Estados		Regenerando y reindexando base de datos: Cat_Age		
Utileria				
Salir		Salir		
[←] Cusor [←] Seleccionar [Esc] Cancelar				

Figura 6.24

La opción de reportes (Fig. 6.25) imprime todos los datos correspondientes a los agentes: Clave de agente, Nombre de agente, oficina y clave de estado. Este reporte se puede generar a través de 2 formas diferentes : a una impresora directamente o bien grabarlo en un archivo para su uso posterior.

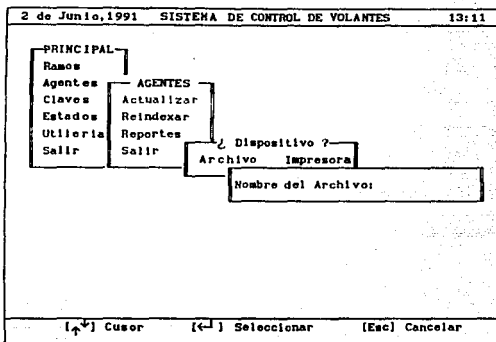


Figura 6.25

6.1.2.7 Módulo de claves

El módulo de claves nos permite actualizar la información correspondiente a las claves de las personas que corresponden a los departamentos de cálculo, mecanografía y revisión (Fig 6.26).

El módulo de claves, contempla básicamente 3 funciones: Actualizaciones (Altas, Cambios y Bajas), regenerar los índices (reindexar) de las bases de datos correspondientes y generar reportes de toda la información de la base de datos.

Para seleccionar una de las opciones anteriores solo bastará colocar el cursor en la opción deseada, paso seguido presionar la tecla de [Enter] o [↵], y el sistema tomará la acción correspondiente.

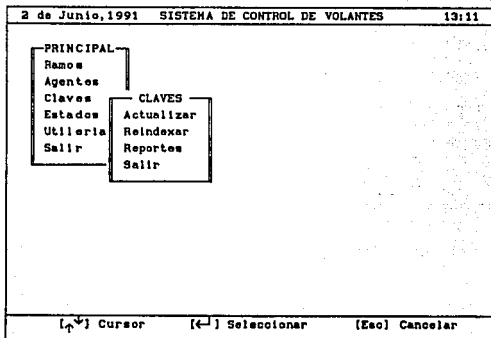


Figura 6.

Si la opción que se elige es la de actualizar el sistema nos mostrará una pantalla como la de la figura 6.27.

En esta parte del sistema se podrán agregar, borrar o modificar registros.

Existen algunas teclas que merecen ,mención especial, por ejemplo :

- [Alt S] Búsqueda de una clave de una persona. Solicita la clave de la persona y se posiciona en la clave solicitada, en caso de que no exista, se posiciona el cursor en la más cercana.
- [Del] Elimina un registro de la base de datos, solicitando primeramente su confirmación.
- [Ins] Inserta un nuevo registro.

- [Esc] Regresa al menú anterior cancelando cambios.
- ← → Mueve el cursor en las direcciones correspondientes.

2 de Junio, 1991		SISTEMA DE CONTROL DE VOLANTES		13:11
PRINCIPAL				
Ramos				
Agentes				
Claves				
Estados				
Utilleria				
Salir				
	CLAVES DE PERSONAS			
	Nombre	Clave	Departamento	
	JAVIER TEJEIDA R.	JT	1 - Departamento	
	CESAR SILVA MTZ.	CS	1 1) Cálculo	
	BENJAMIN MONROY M.	BH	1 2) Mecanografía	
	JOSEFINA CATALAN F.	JC	2 3) Revisión	
	PATRICIA SERRANO E.	PS	2	
	LAURA OROZCO TORRES	LO	2	
	LETICIA ARRIOLA VEGA	LA	3	
	CLAUDIA RUIZ GARCIA	CR	3	
	ADRIANA OTERO VARGAS	AT	3	
[↑↓] Cursor [←] Seleccionar [Esc] Cancelar				

Figura 6.27

Para agregar un nuevo registro se deberá presionar la tecla de [Ins]. Paso seguido el sistema colocará el cursor al inicio de la pantalla, y el usuario estará en posibilidades de agregar su información.

Esta captura se realiza en forma ordenada, donde primero se especifica el nombre de la persona, la clave de la persona y el departamento como puede ser Cálculo, mecanografía o revisión (Fig 6.27).

Cabe mencionar que si la clave del agente ya se encuentra registrada, el sistema nos mandará un error.

Para eliminar un registro solo bastará presionar la tecla de [Del] y en seguida el sistema solicitará la confirmación, en caso

de ser afirmativa el registro se eliminará.

Para modificar un dato de un registro, solo bastará posicionarse en el dato deseado y presionar la tecla de [←] o [Enter]. En caso de que se cambie el departamento el sistema nos mostrará un menú con los 3 departamentos posibles (Fig 6.27).

Si la clave de la persona es modificada esta afectará otras bases de datos, tanto del servidor de archivos donde se encuentran directamente los datos, como a las otras bases de datos en los otros servidores. El proceso de modificar la clave de la persona se puede notar muy lento, debido a la gran cantidad de verificaciones se realizan.

La opción de reindexar (Fig. 6.28) regenera los índices de las bases de datos correspondientes al módulo de claves. Este proceso es recomendable correrlo cuando se han eliminado uno o varios registros.

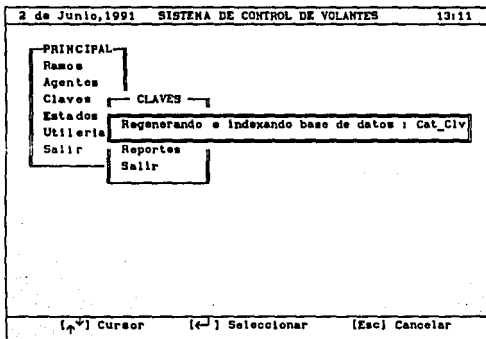


Figura 6.28

La opción de reportes (Fig 6.29) imprime todos los datos correspondientes a las claves de agentes : Nombre de la persona, Clave de la persona y departamento. Este reporte se puede generar a través de 2 formas diferentes : a una impresora directamente o bien grabarlo en un archivo para su uso posterior.

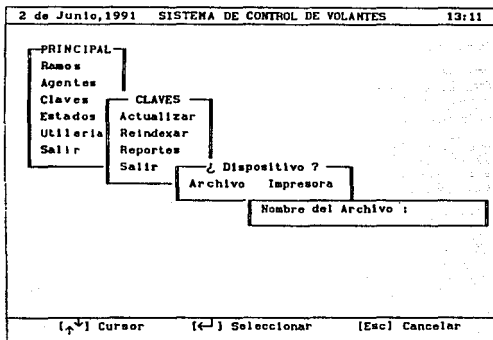


Figura 6.29

6.1.2.8 Módulo de Estados

El módulo de estados (Fig 6.30) nos permite actualizar el catálogo correspondiente a todos los estados de la república mexicana. La información que se tiene en este módulo afecta directamente al módulo de claves de agentes y ramos.

El módulo de Estados, contempla básicamente 3 funciones: Actualizaciones (Altas, Cambios y Bajas), regenerar los índices (reindexar) de las bases de datos correspondientes y generar reportes de toda la información de la base de datos.

Para seleccionar una de las opciones anteriores solo bastará colocar el cursor en la opción deseada, paso seguido presionar la tecla de [Enter] o [↵], y el sistema tomará la acción correspondiente.

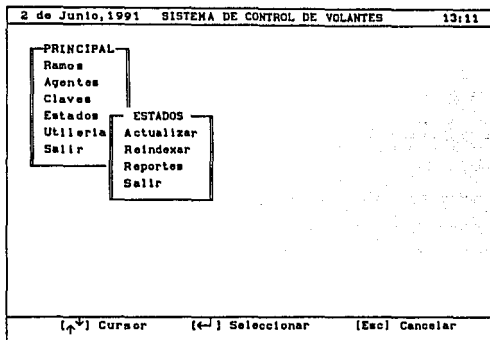


Figura 6.30

Si la opción que se elige es la de actualizar el sistema nos mostrará una pantalla como la de la figura 6.31.

En esta parte del sistema se podrán agregar, borrar o modificar registros.

Existen algunas teclas que merecen, mención especial por ejemplo :

- [Alt S] Búsqueda de una clave del estado. Solicita la clave del estado y se posiciona esa clave, en caso de que no exista, el cursor se colocará en la más cercana.

- [Del] Elimina un registro de la base de datos, solicitando primeramente su confirmación.
- [Ins] Inserta un nuevo registro.
- [Esc] Regresa al menú anterior cancelando cambios.
- ← ↕ → Mueve el cursor en las direcciones correspondientes.

2 de Junio, 1991		SISTEMA DE CONTROL DE VOLANTES		13:11
PRINCIPAL				
Ramos				
Agentes				
Claves				
Estados	ESTADOS	CLAVES DE ESTADOS		
Utilleria	Actualiz	Estado	Clave	
Salir	Reindexa			
	Reportes			
	Salir			
		BAJA CALIFORNIA NORTE	BN	
		BAJA CALIFORNIA SUR	BS	
		CANPECHE	CA	
		CHIHUAHUA	CH	
		COLIMA	CL	
		COAHUILA	CO	
		CHIAPAS	CP	
		DISTRITO FEDERAL	DF	
		DURANGO	DU	
		[↕] Cursor	[←] Seleccionar	[Esc] Cancelar

Figura 6.31

Para agregar un nuevo registro se deberá presionar la tecla de [Ins]. Paso seguido el sistema colocará el cursor al inicio de la pantalla, y el usuario estará en posibilidades de capturar su nueva información.

Esta captura se realiza en forma ordenada, donde primero se inserta el nombre del estado, y en seguida la clave del estado (Fig. 6.31).

Cabe mencionar que si la clave del estado ya se encuentra registrada, el sistema nos mandará un error.

Para eliminar un registro solo se tiene que presionar la tecla de [Del] y en seguida el sistema nos pedirá la confirmación, en caso de ser afirmativa el registro se eliminará.

Para modificar el dato de un registro, solo bastará posicionarse en el, y en seguida presionar la tecla de [←] o [Enter].

Si la clave del estado es modificada esta afectará otras bases de datos, tanto del servidor de archivos donde se encuentran directamente los datos, como a las otras bases de datos, dn los otros servidores. El proceso de modificar la clave del estado se puede notar muy lento, debido a la gran cantidad de verificaciones que se realizan.

La opción de reindexar (Fig. 6.32) regenera los índices de las bases de datos correspondientes al módulo de agentes. Este proceso es recomendable correrlo cuando se han eliminado uno o varios registros.

La opción de reportes (Fig. 6.33) imprime todos los datos correspondientes a los estados: Nombre del estado, clave de estado.

Este reporte se puede generar a través de 2 formas diferentes : a una impresora directamente o bien grabarlo en un archivo para su uso posterior.

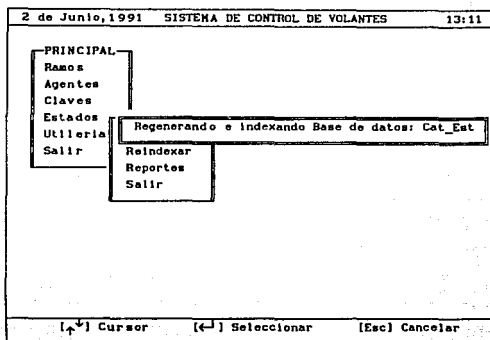


Figura 6.32

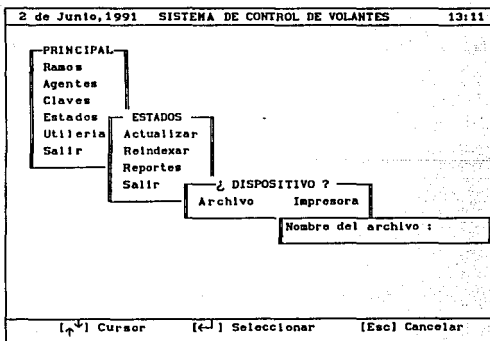


Figura 6.33

6.1.2.9 Módulo de utilerías

Uno de los módulos más interesantes del sistema, es sin lugar a dudas el de las utilerías (Fig. 6.34). en el se presentan una serie de herramientas que nos auxilian en la administración del sistema de control de volantes.

Entre las principales opciones se contemplan : manejo de servidores, mantenimiento de archivos, recuperación de bitacora, enviar mensaje, listar directorio, ver documento, configurar impresora y reindexar bases de datos.

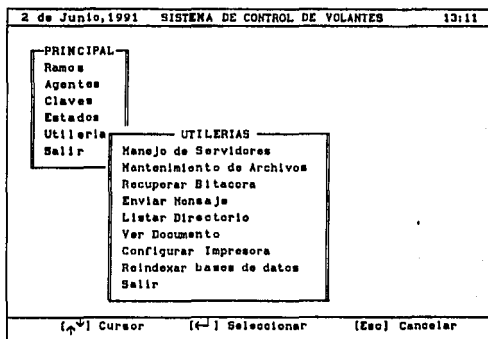


Figura 6.34

Si la opción que se selecciona es la de manejo de servidores el sistema nos mostrará una pantalla semejante a la de la figura 6.35. A través de esta opción el usuario podrá controlar la red de microcomputadoras donde se encuentra instalado el sistema de control de volantes.

Esta opción contempla básicamente 4 funciones: seleccionar servidor, actualizaciones (Altas, Bajas y Cambios), regenerar índices (reindexar) de las bases de datos correspondientes y generar reportes de toda la información de la base de datos.

Para seleccionar una de las opciones anteriores solo bastará colocar el cursor en la posición deseada, y presionar la tecla de [Enter] o [↵].

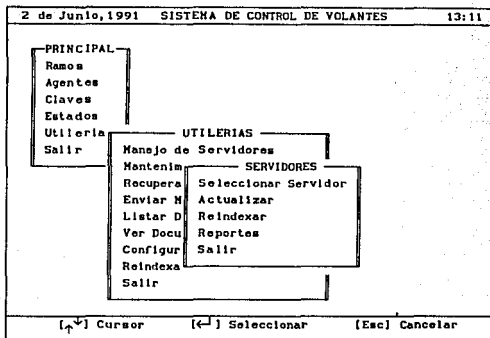


Figura 6.35

Si la opción elegida es seleccionar servidor, esta nos desplegará una pantalla como la de la figura 6.36 con todos los servidores a donde nos podemos conectar siempre y cuando estemos firmados en los servidores de archivos. Si el usuario desea cambiar a otro servidor, solo bastará colocar el cursor en el servidor deseado y el sistema automáticamente tomará las acciones necesarias para modificar las rutas y tomar los archivos respectivos de cada servidor.

Si el usuario desea saber en que servidor de archivos, se encuentra actualmente, solo tiene que presionar las siguientes teclas, desde cualquier parte del sistema : [Alt R]. Y el sistema de control de volantes nos desplegará el nombre del servidor que nos atiende en ese momento.

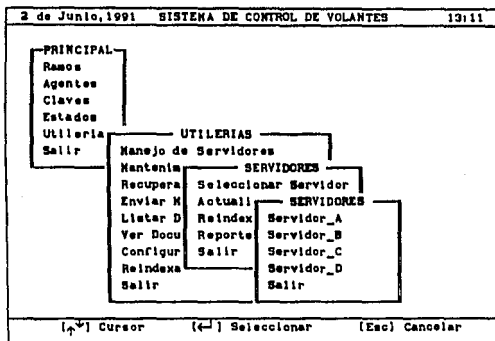


Figura 6.36

Si la opción que se elige es la de actualizar el sistema nos mostrará una pantalla como la de la figura 6.37.

En esta parte del sistema se podrán agregar, borrar o modificar registros.

Existen algunas teclas que merecen mención especial, por ejemplo :

- [Alt S] Búsqueda de un nombre de servidor. Solicita el nombre de servidor de archivos y se posiciona en nombre solicitado, en caso de que no exista, el cursor se colocará en el más cercano.

2 de Junio, 1991 SISTEMA DE CONTROL DE VOLANTES 13:11															
PRINCIPAL Ramos Agentes Claves Estados Utlileria Salir	<table border="1"> <thead> <tr> <th colspan="2">SERVIDORES DE LA RED</th> </tr> <tr> <th>Servidor</th> <th>Drive</th> </tr> </thead> <tbody> <tr> <td>SERVIDOR_A</td> <td>H</td> </tr> <tr> <td>SERVIDOR_B</td> <td>H</td> </tr> <tr> <td>SERVIDOR_C</td> <td>K</td> </tr> <tr> <td>SERVIDOR_D</td> <td>X</td> </tr> <tr> <td>SERVIDOR_E</td> <td>F</td> </tr> </tbody> </table>	SERVIDORES DE LA RED		Servidor	Drive	SERVIDOR_A	H	SERVIDOR_B	H	SERVIDOR_C	K	SERVIDOR_D	X	SERVIDOR_E	F
SERVIDORES DE LA RED															
Servidor	Drive														
SERVIDOR_A	H														
SERVIDOR_B	H														
SERVIDOR_C	K														
SERVIDOR_D	X														
SERVIDOR_E	F														
[↑] Cursor [←] Editar [Del] Borrar [Esc] Cancelar															

Figura 6.37

- [Del] Elimina un registro de la base de datos, solicitando primeramente su confirmación.
- [Ins] Inserta un nuevo registro.
- [Esc] Regresa al menú anterior cancelando cambios.
- ← ↑ → Mueve el cursor en las direcciones correspondientes.

Para agregar un nuevo registro se deberá presionar la tecla de [Ins]. Paso seguido el sistema colocará el cursor al inicio de la pantalla, y el usuario estará en posibilidades de capturar su nueva información.

La captura de un nuevo registro se realiza en forma ordenada, donde primero se especifica el nombre del servidor y a continuación el drive donde se encuentra instalado (Fig. 6.37).

Para eliminar un registro solo bastará colocar el cursor en el registro deseado y presionar la tecla de [DEL], paso seguido el sistema nos pedirá la confirmación, y en caso de ser afirmativa el registro se eliminará de la base de datos.

Para modificar el dato de un registro, solo bastará colocar el cursor en el campo del dato deseado y presionar la tecla de [←] o [Enter].

La opción de reindexar (Fig 6.39), regenerará los índices de las bases de datos correspondientes a los servidores de archivos. Este proceso es recomendable correrlo cuando se han eliminado uno o varios registros.

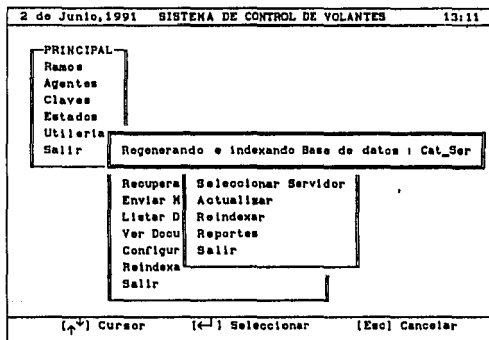


Figura 6.38

La opción de reportes (Fig. 6.39) imprime los datos correspondientes a los servidores : Nombre del servidor y en que drive se localiza. Este reporte se puede generar a través de 2 formas diferentes : a una impresora directamente o bien grabarlo a en un archivo para su uso posterior.

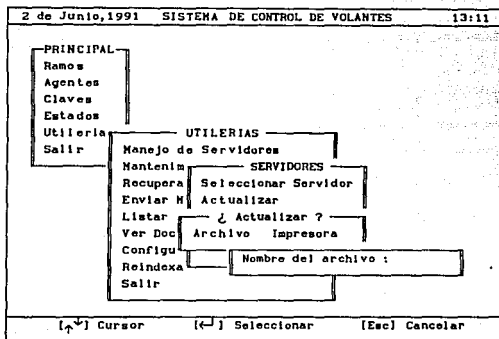


Figura 6.39

Debido al gran incremento de datos en cada uno de los archivos de los ramos, surge la necesidad de tener una opción que los respalde y de mantenimiento, es decir los registros que ya no se utilicen o correspondan a meses anteriores, respaldarlos en disco duro o disco flexibles.

Al respaldar los registro que ya no se utilicen, el sistema se agliza, debido a que no tiene que efectuar búsquedas en archivos con una gran cantidad de información.

La opción de mantenimiento de archivos, es la encargada de realizar los respaldos correspondientes. Esta opción nos permite separar información por meses y por ramos (Fig. 6.40), en un archivo fácil de identificar.

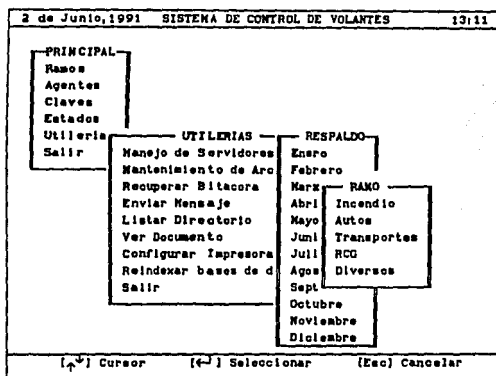


Figura 6.40

Para llevar a cabo la operación, es necesario elegir primeramente el mes de respaldo, una vez seleccionado el mes de respaldo, el sistema preguntará por el ramo. Paso seguido el sistema solicitará los nombres de los archivos y el directorio donde se desea grabar esta información.

El sistema tiene la ventaja de asignar los nombres de los archivos en forma automática y el usuario lo puede modificar a su gusto.

Por ejemplo si se respalda el mes de julio y el ramo de RCG, el sistema asignará los siguientes nombres :

- AS_1R.Jul
- AS_2R.Jul
- AS_3R.Jul

Esto con el fin, de que el usuario pueda distinguir los meses que respaldo.

Recuerde que el disco duro o flexible donde se va a grabar el respaldo, deberá tener el espacio suficiente.

Una de las partes del sistema que permiten conservar la integridad de la base de datos, es la opción de recuperar bitacora, cuando este proceso se realiza todas las fallas que surgieron durante la actualización, cambios o bajas de información son recuperadas.

El usuario solo tiene que escoger la opción y el sistema se encargará de verificar los datos, efectuar los cambios, eliminar registros y en general tomar las acciones correspondientes para cada uno de los archivos donde sucedió la falla.

Otra de las utilerías que contempla el sistema es enviar un mensaje a través de toda la red (Fig. 6.41).

Solo se tiene que especificar el Uid ("*User Identification*") del usuario al que se le va a mandar el mensaje y el mensaje en sí.

Si el usuario al que se le envía el mensaje se encuentra firmado en la red, entonces en su pantalla de desplegará el mensaje que se le envío.

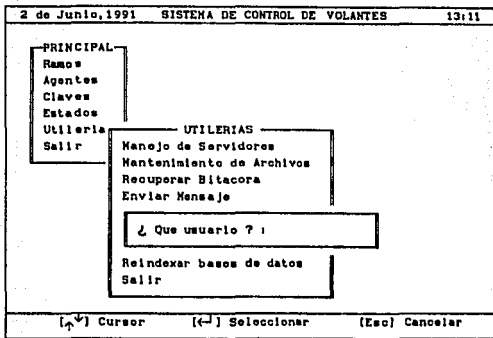


Figura 6.41

La opción de listar directorio nos muestra todos los archivos de un directorio en específico, esta opción se puede utilizar estableciendo mascarar para recuperar información de un solo tipo (Fig. 6.42).

Por ejemplo, en la figura 6.42 se establece una máscara con todos los archivos que tengan por extensión "DBF", si el sistema no detecta ningún archivo con esa extensión, entonces no desplegará nada en pantalla, en el caso contrario nos mostrará una pantalla parecida a la de la figura 6.43

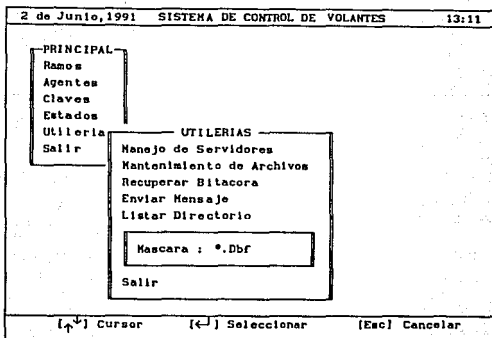


Figura 6.42

2 de Junio, 1991 SISTEMA DE CONTROL DE VOLANTES 13:11

	Mascara : *.Dbf	Archivos : 20		
PR				
Ra	AS_3I.DBF	05-06-91	1395	11:52
Ag	AS_1I.DBF	05-06-91	1675	11:50
Cl	AS_3D.DBF	05-06-91	2571	12:00
Es	RAMOS.DBF	10-05-91	253685	10:12
Ut	CAT_SER.DBF	05-06-91	4886	14:45
Sa	AS_1T.DBF	05-06-91	22554	08:12
	AS_2D.DBF	05-06-91	12365	08:45
	AS_2R.DBF	05-06-91	1234	17:50
	CAT_EST.DBF	05-06-91	8954	20:58
	CAT_AGE.DBF	06-06-91	6395	18:01
	REP_7.DBF	08-06-91	7162	12:23
	AS_2T.DBF	05-06-91	9152	09:30
	AS_1R.DBF	05-06-91	7319	10:12
	AS_3R.DBF	05-06-91	4912	11:11

[↑]Ren. Arriba [↓]Ren. Abajo [PgUp]Pag. Sig. [PgDn] Pag. Ant

Figura 6.43

Si el usuario desea visualizar un archivo de texto simplemente tendra que escoger la opción de visualizar el documento.

Para visualizar el documento es necesario que es usuario le indique al sistema cual es el archivo que desea (Fig 6.44). Paso seguido si el archivo se encuentra será desplegado en pantalla como el de la figura 6.45. En caso de no existir el archivo, el sistema nos enviará un mensaje de error.

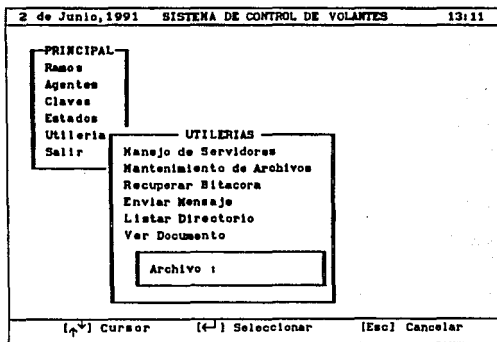


Figura 6.44

```
2 de Junio,1991 SISTEMA DE CONTROL DE VOLANTES 13:11
TEST.PRG
Parameter Param 1
External Drive, Network, Clock_Fa, Server, Send_Msg, Get_S
Public Fields, Letreros, Tipos, Ia_Append, Prev_Rec
Public Cat_Age, Cat_Age1, Cat_Clv, Cat_Est, Cat_Ser
Public As_1, As_1NV, As_1CE
Public As_2, As_2NV, AS_2CA
Public As_3, As_3NV, AS_3CP, As_3OT
Public Local, Imp_Ren, Drive, Ruta
Public Color_Letreros, Color_Sistema, Color_Error

Set Exclusive Off
Set Deleted On
Set Bell On
Set Exact On

[↑]Ren. Arriba [↓]Ren. Abajo [PgUp]Pag.Sig. [PgDn]Pag.Ant
```

Figura 6.45

Si uno de los reportes que se esta imprimiendo sobrepasa las columnas de la hoja o bien los renglones por hoja no son los adecuados, entonces se tendrá que reconfigurar la impresora. El sistema internamente nos brinda la posibilidad de hacerlo, con solo seleccionar la opción de configurar impresora.

Al seleccionar esta opción el sistema nos mostrará un menú con tres opciones diferentes donde 2 de ellas corresponden al tipo de letra que se desea imprimir. Uno de ellos llamado letra normal (letra grande o normal) y el otro letra condensada (letra reducida en tamaño). La tercer opción es la que se encarga de configurar los renglones por página (Fig 6.46), al seleccionarla el usuario tendrá que indicar el número de renglones que desea por página. Al establecer parámetro, todos los reportes que se generen lo tomarán como referencia.

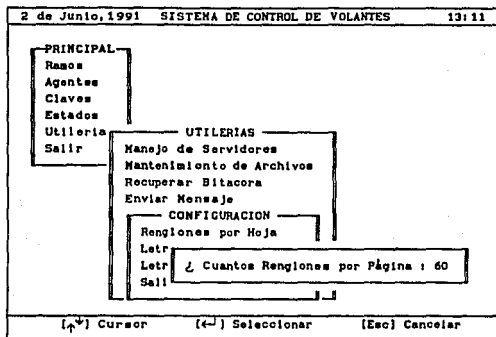


Figura 6.46

La última opción del menú de utilerías es la que se encarga de regenerar los índices de todas las bases de datos del sistema (Fig 6.47).

Como se trata de un proceso largo, el sistema, solicita la confirmación del usuario, en caso de ser afirmativa el sistema comenzará a reindexar las bases de datos.

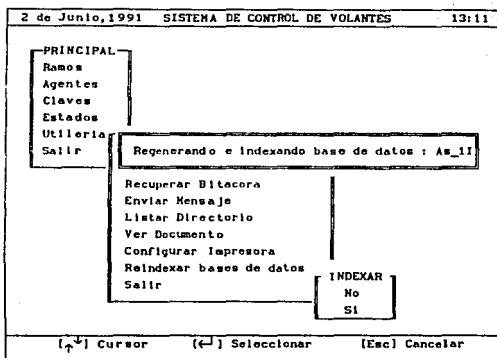


Figura 6.47

Conclusiones

Desde el punto de vista tecnológico las redes locales de microcomputadoras y las bases de datos distribuidas serán altamente consideradas por las empresas e instituciones que desean ir a la vanguardia.

Con la aparición de las redes locales de microcomputadoras, se puede alcanzar un mejor aprovechamiento de los recursos de cómputo, compartiendo componentes, controlar eficazmente la información y herramientas utilizadas en la aplicación de soluciones.

Las bases de datos distribuidas son un concepto que surgió en la década de los 60's pero no es sino hasta los 90's que ha tomado gran importancia en el procesamiento de datos, debido a su gran disponibilidad de almacenar y procesar la información en los lugares donde se haya requerido.

El presente trabajo ha tenido como objetivo contemplar estos dos nuevos aspectos. Y mostrar como se pueden aplicar a un modelo de la vida empresarial.

Como se menciona en el capítulo IV, el modelo es aplicable a cualquier compañía de seguros en su Subdirección de Emisión Daños.

Algo que es bien importante recalcar, es que todas las bases de datos distribuidas se pueden implementar utilizando cualquier tipo de herramientas como lo son los manejadores de bases de datos (*DBMS Data Base Management System*). No importa que se trate de un manejador de base de datos muy poderoso o bien que sea muy limitado (En cuanto a operaciones como son : *Roll-Back, Comit, Etc.*). La diferencia radica, en el desarrollo de rutinas específicas para cada operación o transacción que se vaya a realizar en la base de datos distribuida. Como fue el caso del sistema de la tesis.

La mayoría de las operaciones de la tesis se tuvieron que ir diseñando, implementando, desarrollando y probando durante el análisis del modelo conceptual y sobre la marcha del la programación del sistema. Esto no quiere decir que las rutinas no funcionen, sino que el trabajo que se realizó tuvo que ser prácticamente desde cero. Puesto que *Clipper* a pesar de ser un lenguaje poderoso y versátil no cuenta con ninguna operación para aplicarla en el diseño de un sistema de base de datos distribuidas.

En forma general y como resultado del estudio de las bases de datos distribuidas y redes locales de microcomputadoras, podemos efectuar las siguientes conclusiones y observaciones.

Una base de datos distribuida se puede implementar en cualquier plataforma, minis, micros, mainframes. No importa el sistema operativo que se utilice *Novell, Unix, Vms, etc.* La principal diferencia, es la correcta utilización de las herramientas para el desarrollo del software aplicativo, así como la selección adecuada de los componentes de hardware de comunicación.

El costo de una red no importando la plataforma que se utilice, resulta ser muy alto, debido a que se requieren equipos de comunicación muy sofisticados. Por ejemplo en una red local de microcomputadoras, para realizar enlaces en localidades remotas se necesitan modems, líneas de comunicación, servidores de comunicación y tarjetas de comunicación muy especiales (*Gateways, Bridges, Etc.*).

Las bases de datos distribuidas tienen la enorme ventaja de poder trabajar en un ambiente heterogéneo no importando cuales sistemas operativos se involucren, el principal problema es el desarrollo del software que permita la correcta interfaz entre los datos. Cabe mencionar que no existe en el mercado un sistema que contemple estas características.

La mayoría de los sistemas comerciales se quedan a nivel prototipo o pruebas de laboratorio, debido a que no se han explotado a gran escala o en forma comercial.

La capacidad de una red local de microcomputadoras se puede llegar a comparar con la de una minicomputadora e inclusive con la de una mainframe. Aclarando que para tales pruebas se necesita como servidor de archivos una microcomputadora muy poderosa, lo cual resulta muy caro. Sin embargo, comparando con el valor de una mainframe de tamaño regular, su costo resulta mínimo.

La tendencia actual en cualquier empresa, es invertir la menor cantidad de dinero y tener resultados muy satisfactorios a mediano y corto plazo. La espera de resultados inmediatos casi siempre viene acompañada de una red de computadoras.

Las bases de datos distribuidas todavía tienen una enorme potencialidad. Ya que ahora las empresas podrán descentralizar sus trabajos y así establecer lugares o centros de atención específicos de cada región, dependiendo de cual sea la carga de trabajo por zona.

Se podrán tener varios administradores tanto de red como de base de datos.

Como todo, siempre existe un riesgo, el cual podría ser la falla de una línea de comunicación, falla de equipo, aterrizaje de las cabezas del disco duro, etc. En caso de mainframes, la descompostura del aire acondicionado, falta de no-brake, etc. Todos estos problemas deben ser contemplados a la hora de analizar, diseñar e implementar el sistema.

APENDICES

Apéndice A Álgebra relacional

A.1 Álgebra relacional

El álgebra relacional es un método para operar relaciones. Estas relaciones son consideradas comunmente como conjuntos de registros, donde cada registro se describe en términos de campos de llaves o atributos.

Cuando dos relaciones comparten un atributo, se dice que existe dependencia funcional, o están relacionadas. La dependencia funcional se puede expresar como una relación de 1-a-N.

Existen 5 operaciones fundamentales en el álgebra relacional :

- Selección (σ)
- Proyección (Π)
- Producto cartesiano (\times)
- Unión (\cup)
- Diferencia de conjuntos ($-$)

Además de las 5 operaciones principales existen otras operaciones, a saber :

- Intersección de conjuntos (\cap)
- Producto theta (X_{θ})
- Producto natural (x)
- División (\div)

La selección y proyección son operaciones unarias ya que actúan sobre una relación. Las otras tres relaciones operan sobre parejas de relaciones por lo que se les llama operaciones binarias.

A.1.1 Selección

La selección involucra tomar una tabla o relación y seleccionar los renglones que satisfagan un predicado. El predicado aparece como subíndice de (σ) mientras que la relación que constituye el argumento se da entre paréntesis después de la (σ). Es decir :

$\sigma_{\text{Nombre_Campo} = \langle \text{Valor del Campo} \rangle} (\text{Relación } X)$

Las operaciones que se permiten en el predicado son =, *, <, >, ≤, ≥.

A.1.2 Proyección

La operación de proyección es unaria, la cual copia su relación o tabla quitando ciertas columnas. La proyección se señala con la letra mayúscula (Π). Como subíndice de Π se indican todos los atributos que se desean aparezcan en el resultado. La relación con la que se va a operar se escribe después de Π entre paréntesis.

II Nombre_Campo_1, Nombre_Campo_2, .. , Nombre_Campo_N (Relación X)

A.1.3 Producto cartesiano

El producto cartesiano multiplica dos tablas o relaciones, esto es, si una tabla o relación tiene N renglones y I columnas, y la otra tabla o relación tiene M renglones y J columnas, el producto de las relaciones generará una relación o tabla, con $(N * M)$ renglones y las columnas serán $(I + J)$. El producto cartesiano se señala por una cruz $\{X\}$. Esta operación es binaria. Así el producto cartesiano de las relaciones (r_1, r_2, \dots, r_n) se expresa como :

$$r = r_1 X r_2 \dots X r_n$$

A.1.4 Unión

La unión de dos tablas o relaciones que tienen N y M renglones, se obtiene al concatenar estas en una tabla con un total de $(N + M)$ renglones. La unión generalmente tiene sentido solo si los esquemas globales de dos tablas son iguales, esto es si poseen el mismo número de campos y los atributos de cada campo son iguales. La unión de las relaciones (r_1, r_2, \dots, r_n) se expresa de la siguiente manera :

$$r = r_1 \cup r_2 \cup r_n$$

A.1.5 Diferencia de conjuntos

El operador de diferencia de conjuntos, señalado por (-), permite encontrar las tuplas que estén en una relación, pero no en otra. La expresión $(r - s)$ resulta en una relación que contiene las tuplas que estén en r pero no en S .

A.2 Operaciones adicionales

Las operaciones adicionales del álgebra relacional, se definen en base a los cinco operadores fundamentales, estas operaciones adicionales no hacen más potente al álgebra relacional, pero si simplifican algunas consultas comunes.

A.2.1 Intersección de conjuntos

La primera operación adicional que se va a definir es la intersección de conjuntos (\cap), esta operación se define por una pareja de operaciones de diferencia de conjuntos, es decir :

$$r \cap s = r - (r - s)$$

Por lo tanto la intersección de conjuntos no hace más potente al álgebra relacional. Sencillamente es más conveniente escribir $r \cap s$ que $r - (r - s)$.

A.2.2 Producto theta

El producto theta es una operación binaria que permite combinar la selección y el producto cartesiano en una sola operación; se indica (X_{θ}) donde X es el símbolo de la "unión" y el subíndice θ (letra griega theta) se sustituye con el predicado de selección. El operador producto theta forma el producto cartesiano de sus dos argumentos y después lleva a cabo una elección mediante el predicado θ .

En general, θ puede ser un predicado arbitrario. Dadas dos relaciones, r y s , y un predicado θ .

$$r X_{\theta} s = \sigma_{\theta} (r X s)$$

A.2.3 Producto natural

La operación producto natural simplifica todavía más la notación del álgebra relacional. La definición formal del producto natural se puede expresar de la siguiente manera. Supónganse dos esquemas de relaciones R y S que son, por supuesto, listados de nombres de atributos. Considérese los esquemas como *conjuntos* más que como listados. Esto permite indicar tanto aquellos atributos que están tanto en R como en S por medio de $R \cap S$, y como los que aparecen en R , en S o en ambos por medio de $R \cup S$. Nótese que aquí se está hablando de unión e intersección de conjuntos de atributos, no de relaciones.

Piénsese en dos relaciones $r(R)$ y $s(S)$. El producto natural de r y s , expresado por $r \times s$, es una relación del esquema $R \cup S$. Es la proyección sobre $R \cup S$ de un producto theta donde el predicado requiere que $r.A = s.A$ para cada atributo A en $R \cap S$. Formalmente,

$$r \times s = \Pi_{R \cup S} (r \times r.A_1 = s.A_1 \wedge \dots \wedge r.A_n = s.A_n \text{ } s)$$

Donde $R \cap S = \{ A_1, \dots, A_n \}$.

A.2.4 División

La última operación del álgebra relacional es llamada división (+). La operación de división es apropiada para consultas que incluyen la frase "para todos". La operación de la división formalmente se expresa como, sean $r(R)$ y $s(S)$ relaciones, y sea $S \subseteq R$. La relación $r + s$ es una relación con el esquema $R - S$. Si una tupla t está en $r + s$ si para cada tupla t_s en s existe una tupla t_r en r que satisface estas dos condiciones:

$$\begin{aligned} t_r[S] &= t_s[S] \\ t_r[R - S] &= t[R - S] \end{aligned}$$

Apéndice B Sistema operativo Netware

B.1 NOVELL Antecedentes

Novell Inc. fué fundada a principios de la década de los ochentas, esta compañía se ha caracterizado por estar más enfocada al software que al hardware. Novell ha sido de las compañías que ha desarrollado sus propios sistemas operativos para ambiente de red local.

A mediados de los ochentas adquirió los derechos de la compañía CXI, para el desarrollo del hardware de todos sus productos de gateway y comunicaciones asíncronas.

Los sistemas operativos de Novell han dominado prácticamente todo el mercado de redes locales tanto en los Estados Unidos (Donde posee un 62% de los sistemas operativos instalados) como en el mercado nacional (en el cual se calcula tendrá de un 80% a 90%).

A diferencia de otros sistemas operativos (HP, IBM, 3COM, Etc.), el sistema operativo de Novell (*Netware*), no fué desarrollado basado en los sistemas OS/2 o DOS. El *Netware* es un sistema nativo de Novell el cual soporta estaciones de trabajo operando bajo OS/2 o DOS en forma transparente.

B.2 Sistema Operativo

Novell ha lanzado al mercado varios sistemas operativos cuya diferencia del uno al otro es el número de facilidades que proporcionan. Teniendo como característica principal la de ser independiente de hardware, ya que soportan más de 48 tipos diferentes de topologías. Entre los últimos sistemas operativos liberados se tienen los siguientes :

- Netware para Macintosh
- Netware para VMS
- Advanced Netware v2.15
- System Fault Tolerant (SFT v2.15)
- Netware 386 v3.0

B.2.1 Netware para Macintosh

Netware para Macintosh fue diseñado y desarrollado con la asistencia técnica de *Apple*. Permite a las estaciones Macintosh acceso a los servicios de los servidores de red corriendo bajo cualquier sistema operativo de Netware.

- Permite que estaciones de trabajo Macintosh y Pc's puedan compartir recursos en una red local.
- Los usuarios de Macintosh no tienen que sacrificar amigabilidad y funcionalidad al utilizar Netware.
- Los usuarios con experiencia en *AppleShare* (Sistema Operativo para red *Apple*) no necesitan reentrenamiento para utilizar Netware, ya que la información almacenada en el servidor de la red se puede visualizar utilizando la interfaz de iconos que maneja Macintosh. Las otras estaciones de la red corriendo bajo DOS, OS/2 o Windows 386 visualizan su información de acuerdo con el formato que maneja su sistema operativo.

- Buen manejo de niveles de seguridad que van desde horas específicas de acceso, hasta solicitar a los usuarios que modifiquen su password periódicamente.
- Los recursos de la red pueden ser contabilizados de acuerdo al tiempo de conexión, los bloques leídos del disco o escritos y la cantidad de información almacenada en el disco duro del servidor de la red.
- El software de Netware para Macintosh puede quedar residente en el servidor de la red o en una estación de trabajo.

B.2.2 Netware para VMS

Netware para VMS permite que los sistemas DEC VAX/VMS funcionen como servidores de Netware. Usuarios de PC y VAX comparten los mismos recursos de VMS tales como archivos, bases de datos, impresoras y aplicaciones.

- Conectividad transparente; es decir los archivos localizados en el servidor VAX aparecen como archivos de DOS para los usuarios de PC, mientras que los usuarios de VAX manejan su estandar VMS.
- Facilidades de respaldo de información desde VAX o desde Netware.
- Independencia de hardware, ya que soporta más de 48 topologías diferentes.
- Emulación de terminal para VT100, VT220, VT240, Tektronix 40XX y otras terminales.
- Servicios de Emulación de Terminal (TES), los usuarios de PC pueden acceder los servicios que proporciona VAX y así correr aplicaciones de VMS.
- Las impresoras de VAX están disponibles a los usuarios de PC como si estas estuvieran conectadas directamente a la PC.

- La administración de las colas de impresión se maneja a través de las facilidades de VMS.
- Los niveles de seguridad que utiliza el sistema operativo son los mismos que proporciona VAX/VMS.
- Los accesos remotos a VAX/VMS utilizan las facilidades de DECNET.
- Compatibilidad entre servidores de VMS y PC's en la misma red.
- Permite la integración de estaciones de trabajo bajo ambiente DOS , Windows 386 y OS/2 en forma transparente.
- En un futuro se podrán soportar estaciones de trabajo para Macintosh y UNIX.
- Este sistema operativo requiere de la compra de un equipo VAX.

B.2.3 Netware v2.15

Advanced Netware v2.15 es un sistema operativo de red nativo de Novell, liberado a fines de 1988. Permite la integración de estaciones de trabajo bajo ambiente Macintosh, DOS y OS/2 en forma transparente.

Ventajas :

- Buen manejo de memoria caché.
- Duplica la estructura de directorios (FAT) del servidor para prevenir perdida o corrupción de esta área, además redirecciona los datos de los bloques del disco duro dañados hacia bloques disponibles.
- Propiedad de "HOT FIX" : provee la detección de defectos y su corrección mientras el sistema se encuentra operando. Cuando un bloque defectuoso es detectado durante la operación de escritura, el dato es movido hacia una área de seguridad en el disco y el bloque dañado es marcado como no usable.

- Lectura - Escritura : Verificación de lectura después de escritura, lo cual garantiza que los datos que sean escritos en el disco serán siempre leídos.
- Monitoreo de No-Break : Capacidad de monitorear una fuente ininterrumpible de poder colocada en el servidor de la red.
- Cuando una falla de energía ocurre el sistema operativo da de baja la red sin ningún riesgo.
- Consola virtual : Cualquier estación de trabajo de la red puede funcionar como consola de la red, para monitoreo de la información y usuarios de la red.
- Existen 4 niveles de seguridad básicos de acceso : password, derechos, atributos de directorio y atributos de archivo. Se tiene ocho restricciones adicionales que pueden ser impuestas a cada nivel de directorio.
- Auditoria de la información de red; por ejemplo, colas de impresión, capacidad de memoria utilizada, monitoreo de la actividad de los usuarios.
- Flexible manejo de colas de impresión.
- Capacidad de restricción de " login " a los usuarios de acuerdo a :
 - Dirección de la estación de trabajo.
 - Día y hora.
 - Número de lecturas al disco del servidor.
 - Número de escrituras al disco del servidor.
 - Cambio obligatorio de password.
- Detección de intrusos a la red : cuando un usuario teclea una clave de " password " errónea un número predeterminado de veces en forma consecutiva, el sistema asume que el usuario es un intruso que intenta violar la seguridad de la información y automáticamente inhibe el acceso a la red; tanto al usuario como a la estación de trabajo.
- Registro detallado de errores de " login " de usuarios y errores del sistema.

- Contabilidad de los recursos : permite llevar una contabilidad de los recursos utilizados por los usuarios :
 - Por tiempo de conexión.
 - Bytes leídos o escritos.
 - Número de espacio en disco utilizado.
 - Número de I/O al disco del servidor.
 y hacerles un cargo por ello si es necesario.
- La arquitectura de multiusuario y multitarea permite que muchas operaciones se pueden ejecutar simultáneamente dando como resultado ejecución más rápida de la red.
- Elección de servidor dedicado y No dedicado.
- Soporta 100 usuarios concurrentes.
- Independiente del hardware.
- Soporta diferentes protocolos de transmisión IPX/SPX (Nativo de Novell), Netbios, DLC y TCP/IP.
- Correo electrónico a través de MHS. (Producto adicional)
- Las estaciones de trabajo se pueden habilitar por medio de un *Eprom* en sus tarjetas de interfaz.
- Comunicaciones asíncronas para transferencia de archivos a través del puerto serial COM1 / COM2 de las estaciones de trabajo.
- Incluye emulador de *Netbios* para hardware compatible con IBM.
- Fácil actualización a SFT v2.15
- Filosofía Cliente/Servidor para manejo de bases de datos con *Btrieve* (manejador de bases de datos indexadas desarrollada por Novell.)
- *VAP's (Value Added Process)*, los cuales permiten que ciertas aplicaciones sean ejecutadas en el servidor junto con *Netware*.
- Conectividad con equipo mainframes por medio de *gateways*.
- Soporte a *APPC LU6.2*.

B.2.4 System Fault Tolerant (SFT v2.15)

SFT v2.15 provee todas las facilidades descritas en el Advance Netware v2.15 más otras propiedades que permiten redundancia de la información del servidor, la cual puede ser accesada en caso de que ocurra una falla de hardware.

Netware v2.15 provee alto nivel de seguridad, características de tolerancia a fallas que necesitan instalaciones corporativas y públicas y características de contabilidad de recursos de uso de la red, además de herramientas para desarrolladores de software independientes.

Ventajas :

- **Espejo de Discos** : Soporta dos discos duros idénticos en el servidor controlados por una misma tarjeta controladora de discos. SFT duplica el volumen del disco íntegramente en un segundo disco duro. La escritura al disco original es espejeada por el servidor. El servidor verifica que la escritura se realice en ambos discos. Si el disco original fallara en cualquier momento, el otro disco (disco espejo) trabaja de manera automática sin pérdida de la información. El disco espejo trabaja en ambos casos lectura y escritura.
- **Duplicidad** : SFT permite duplicidad de controladores, drivers, interfases y fuentes de poder. Si un disco duro, controlador, o interfase fallara, automáticamente estos se recuperan sin pérdida de operación.
- **Integridad de datos a través de TTS (Transaction Tracking System)** : Protege la información de ser destruida si ocurre una falla del sistema durante una actualización de la información.

- Reduce el tiempo de acceso a disco.
- La tarjeta coprocesadora de discos da mayor rapidez a los requerimientos de lectura.
- Mejora la operación de la red con tablas de localización de archivos indexados mayores a 2MB.

B.2.5 Netware 386 v3.0

Netware 386 toma todas las ventajas de las potencialidades del procesador 80386 y ofrece nuevos niveles de rendimiento y funcionalidad en las redes de equipos de cómputo.

Ventajas :

- Utiliza todas las ventajas del microprocesador de 32 Bits.
- Incluye todas las características de los sistemas operativos mencionados anteriormente (*Hot Fix*, *Transaction Tracking System* (TTS), discos en espejo, monitoreo de No-break, etc).
- 250 Usuarios concurrentes.
- Más de 35 trillones de bytes(32 TB) para almacenamiento en disco.
- La Interface *OPEN DATA-LINK* es soportada. *OPEN DATA-LINK* es una interfaz estándar la cual permite que múltiples protocolos compartan un adaptador de red simultáneamente.
- Flexibilidad en los servicios de impresión, los cuales son proporcionados a través de una aplicación del servidor de impresoras, que se cargan dentro del servidor de la red. El servidor de impresoras permite a los usuarios imprimir en red o localmente en cualquier impresora de estaciones de trabajo asignadas como compartidas. Cada servidor de impresión soporta 16 impresoras.

- Jerarquización de las colas de impresión a ser servidas. Se pueden crear colas que den servicio de acuerdo a un criterio de primeras entradas primeras salidas o dando prioridad a los trabajos impresos en una forma particular.
- Independencia de protocolos. Un mecanismo de protocolos permite el acceso de protocolos múltiples IPX/SPX, AppleTalk, TCP/IP, SNA NetNEUI, OSI.
- SFT Nivel III permite la duplicidad de todo el server de la red.

Desventajas :

- No se pueden definir modems compartidos, únicamente de SPOOL de impresoras. Los modems compartidos se pueden lograr a través de otro producto llamado ACS (*Asynchronous Communication Server*).
- Los productos existentes para aprovechar el manejo de APPC LU6.2 son limitados por los productos que ofrecen terceros, por ahora se conocen solo con servicios de transferencia de archivos, emulación de terminal (3780,5250,2780 RJE), con acceso a API opcional.

B.3 Topología

Netware es independiente del tipo de topología a utilizar.

B.4 Tipo de Cable

Se puede utilizar cable coaxial, cable telefónico, fibra óptica o una combinación de ellos por medio de diferentes dispositivos de hardware.

B.5 Tipo de Servidor

Servidores Novell 386A, 386Ae o cualquier máquina IBM PC, AT (286 ó 386) o compatible o bien PS/2 modelos 50,60,70 80 con un requerimiento mínimo de memoria de 2MB.

B.6 Tarjetas de Red

Soporta 48 tipos diferentes de tarjetas de red.
(Ethernet, Arcnet, Token Ring, G-Net, Etc.)

B.7 Interconectividad

B.7.1 Lan vs Lan

- Bridges Internos : Soporta hasta 4 tarjetas de red dentro del servidor para interconectar redes locales de la misma topología o diferente.
- Bridges Externos locales : Se realizan en una estación de trabajo, la cual puede ser definida como dedicada o no dedicada. Una estación definida como bridge puede soportar hasta 4 tarjetas de red. Se recomienda este tipo de bridge en lugar del bridge interno, dado que la tarea de ruteo de paquetes lo realiza una estación y no el servidor.
- Bridges Remotos : Existen dos tipos de bridges remotos que pueden ser definidos en Netware :
 - Para transferencia de archivos.
 - Para sesiones en línea.

Para transferencia de archivos : el bridge de una red con un usuario remoto o red con red, se realiza a través del puerto serie COM1, COM2 de una de las estaciones de la red definida como estación de bridge. Esta estación puede ser dedicada o no dedicada. El Software requerido para este tipo de comunicación está incluido dentro del sistema operativo. En el caso de una comunicación red contra equipo mini (HP3000, VAX, NCR, ALTOS, Etc.), el software que se utilizará para la transferencia de archivos puede ser muy variado, como por; ejemplo, *Advanced Link*, *Crosstalk*, etc.

B.7.2 Lan vs Mainframe

- Gateways : Existen varios productos de gateway de Novell que permiten emulación de terminal 3270, 3780 o 5250 en forma local o remota.
- Para el caso de emulación 3270, se pueden tener gateways de 5, 16, 40 y hasta 128 sesiones concurrentes.
- Para emulaciones 5250 de sistemas JX de IBM, se tienen gateways para 5 y 9 sesiones.

B.7.3 Comunicación X.25

Para el caso de comunicaciones sincronas, cuenta con un gateway con protocolo X.25, en el cual es posible conectar una red ya sea a una minicomputadora que tenga implementado este tipo de comunicación a otras redes en forma puntual o a través de la red pública de datos Telepac.

B.8 Comentarios

El Netware es un sistema poderoso, bien estructurado, con buen manejo de su memoria y con ciertas sofisticaciones de seguridad que permiten al administrador de la red, tener un control absoluto sobre los usuarios y los recursos a compartir, sin tener que estar físicamente operando la consola del servidor.

Netware ha estado presente en el mercado por algún tiempo; ha sido exhaustivamente probado por varias organizaciones que se avocan a la evaluación de sistemas, pero principalmente por usuarios finales.

Tanto los resultados de dichas evaluaciones como los comentarios de usuarios finales, colocan a este sistema en una posición sobresaliente del mercado.

A pesar de que Netware no es reconocido como una norma de sistema operativo, ha influido en el diseño de otros sistemas operativos. Por otro lado, Novell ha tenido especial interés en que Netware soporte la arquitectura de 7 niveles de OSI para conexión de sistemas; lo cual garantiza al usuario la capacidad de conectividad de Netware con otros ambientes de diferentes características.

Referente a la conectividad que se puede lograr con Netware, Novell provee una amplia gama de productos de comunicación con los que es posible conectarse a equipos mini, mainframes o multiusuario.

Apéndice C Sistema operativo VAX/VMS Services for MS-DOS.

C.1 DIGITAL Antecedentes

Digital Equipment Corporation es una compañía internacional, líder en conectividad, servicios e integración de sistemas. La filosofía de Digital se basa en el concepto de procesamiento distribuido, lo cual significa tener aplicaciones poderosas donde ellas se necesiten.

Digital posee aproximadamente un 19% de la base instalada de redes locales en los Estados Unidos. En cuanto a redes de largo alcance posee una de las más grandes del mundo; sin embargo, los productos para redes locales de Digital no son tan conocidos.

C.2 Sistema Operativo

La arquitectura de Digital llamada *Personal Computing System Architecture* (PCSA) es la mejor solución para integrar estaciones de PCs en un ambiente de red. Usando PCSA las estaciones de PC pueden ser conectadas a equipos Digital VAX/VMS y así compartir datos y archivos.

En un medio ambiente de PCSA, cualquier equipo Digital —En una red local o en una red de gran alcance— trabaja como un servidor mientras que las PC funcionan como clientes o estaciones de trabajo. EL software del server que permite la integración de PCs en una red local es *VAX/VMS Services for MS-DOS*, de esta manera los usuarios de PC pueden acceder impresoras remotas y discos. Mientras que el software que necesitan las estaciones de trabajo que van a ser conectadas a un equipo VAX o MicroVAX es

DECNET/PCSA que es un superconjunto de DECNET-DOS (incluye todas las herramientas de DECNET/PCSA)

C.2.1 VAX/VMS Services for MSDOS

VAX/VMS Services for MSDOS permite que los recursos, datos y aplicaciones que se encuentran en el servidor de archivos (VAX) puedan ser accedidos por PCs.

Ventajas :

- Los servicios de impresión permiten a los usuarios imprimir en impresoras locales de la red o bien en impresoras que se encuentren en otra red.
- 4 tipos de servicios de archivos : Para acceso a directorios y archivos almacenados en el servidor de archivos.
 - Directorio del sistema.- Contiene el software del sistema PCSA.
 - Directorio de aplicación.- Contiene los archivos ejecutables para aplicaciones de MS-DOS.
 - Directorio común.- Contiene datos y archivos de texto que los usuarios comparten y actualizan.
 - Directorio de usuario.- Contiene los archivos personales de un usuario.
- Servicios de discos virtuales.- Tiene capacidades semejantes a la de un disco duro a menor costo y sin pérdida de velocidad.
- Facilidad de comunicaciones asíncronas para estaciones remotas.
- Incluye software y hardware para realizar respaldos de información.
- Monitoreo de las actividades que se realizan en la red.

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

- Las estaciones de trabajo soportan interfaz a NETBIOS.
- Compatibilidad total con los productos de Digital.
- Con tarjetas DEPCA las estaciones de trabajo pueden habilitarse en forma remota.
- Se tiene un alto grado de seguridad que contempla mecanismos para el control de acceso, seguridad de las bases de datos y auditoría de la información.
- Bitácora de conexiones actuales, que archivos han sido abiertos, sesiones y servicios disponibles.
- Se puede limitar el número de intentos de acceso a la red. Esto con la finalidad de descubrir intrusos.
- El servidor de archivos tiene dos niveles de seguridad :
 - Acceso a los servicios permitidos por el administrador.
 - Acceso a los archivos utilizado VAX/VM.
- Emulación de terminal para VT220, VT240 y 3270.
- Conectividad hacia cualquier equipo que exista en el mercado.

Desventajas :

- No se pueden definir modems compartidos.
- No es independiente de hardware, se requiere un equipo VAX corriendo VMS para cualquier solución disponible actualmente.

C.3 Topología

Digital maneja una sola topología: Ethernet, soportando actualmente tres tipos de tarjetas, dos de ellas fabricadas por Digital.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

C.4 Tipo de Cable

Para redes locales se puede utilizar ThinWire (cable coaxial) y cable telefónico utilizando convertidores a thinWire.

En cuanto a redes de longitud amplia se puede utilizar thinwire, cable telefónico, fibra óptica, microondas, satélite o una combinación de ellos por medio de diferentes dispositivos de hardware.

C.5 Tipo de Servidor

Cualquier equipo VAX o MICROVAX puede funcionar como servidor de archivos. La restricción está en el número de usuarios que pueden trabajar.

Un equipo MICROVAX 2000 soporta 30 usuarios simultaneos, mientras que un equipo MICROVAX 3800 soporta 75 usuarios concurrentes.

C.6 Tarjetas de Red

Digital soporta las siguientes tarjetas : 3COM EtherLink II, Etherlink/MC, DEPCA y MICOM-Interlan N15010-1 y N15010-2. DEPCA es una tarjeta Ethernet controladora de comunicaciones fabricada por Digital.

C.7 Interconectividad

C.7.1 Lan vs Lan

- Bridges Internos : No existen en digital.
- Bridges Externos : Este tipo de bridges se puede realizar de dos maneras diferentes :
 - 1) Por medio del LAN Bridge 100 que es un dispositivo de alto rendimiento que conecta dos segmentos de Ethernet con lo cual se crea una extensión de la red local. Maneja cualquier protocolo que cumpla con la norma IEEE 802.3 tales como DECNET, LAT, XNS o TCP/IP.
 - 2) Por medio del METROWAVE Bridge. Este es un dispositivo que conecta dos redes locales separadas geográficamente. METROWAVE bridge consiste de dos Bridge 100 que utilizan como medio de transmisión microondas. Maneja cualquier protocolo que cumpla con la norma 802.3/Ethernet
- Bridges Remotos : Se pueden implementar por medio del DECrouter 200 para enlaces remotos de nodos via líneas asincrónicas. Para conexiones sincrónicas entre redes remotas se puede utilizar el DECrouter 2000

C.7.2 Lan vs Mainframe

En este aspecto es muy importante señalar la ventaja tecnológica que Digital posee sobre los demás competidores al proveer una mayor funcionalidad para el procesamiento de información con excelentes productos como VIDA, que permite una mayor transparencia en el acceso de datos desde VAXes hacia IBMs.

Gateways : Existen varios productos de gateway de Digital que permiten emulación de terminal 3270, 3780 o 5250 en forma local o remota.

C.7.3 Comunicación X.25

Para el caso de comunicaciones sincronas, cuenta con un gateway con protocolo X.25, llamado X25router 2000 en el cual es posible conectar una red, ya sea a una minicomputadora que tenga implementado este tipo de comunicación o bien a redes que utilizan la red pública de datos Telepac.

C.7.4 Comentarios

La estrategia de Digital para los 90's es incluir computadoras personales en sus grandes redes, para compartir archivos y recursos tales como impresoras y computadoras más poderosas. El software que permite la integración de Pc's es *VAX/VMS Services for MSDOS* el cual hace uso de todos los recursos que proporciona el sistema operativo VMS de Digital.

VAX/VMS services for MSDOS posee los niveles de seguridad más altos del mercado internacional, ya que a través de estos se tiene un control absoluto de la red, recursos y usuarios.

Una de sus principales desventajas es que el usuario administrador de la red tiene que conocer otro sistema operativo, puesto que existen ciertos parámetros como son la definición de colas de impresión, definición de accesos a la red, privilegios y restricciones de los usuarios, que se deben de realizar por medio de VMS y no por MSDOS.

Referente a la conectividad Digital ofrece un gran número de productos de comunicación con lo que es posible conectarse a equipos mini, mainframes o múltiusuario, además de actualmente con la más alta funcionalidad en el mercado.

Apéndice D Sistema operativo Hewlett-Packard

D.1 Antecedentes

Hewlett Packard (HP) es una compañía con un amplio prestigio en el mercado por la calidad de sus productos.

HP se podría catalogar como una empresa más orientada a la tecnología de hardware que al software, pues ha producido sistemas mini muy poderosos, que han tenido muy buena aceptación en el mercado.

A partir de 1979 comenzó a desarrollar productos de proceso distribuido para la interconexión de diferentes sistemas mini, bajo el nombre de *AdvanceNet*. A mediados de los 80's introdujo al mercado un software de red con el nombre de "OfficeShare", el cual permite a usuarios de ambiente DOS tener acceso tanto a un servidor de archivos o a un sistema mini (basicamente HP1000, HP3000, HP9000 y VAX). Recientemente, HP adquirió un 5% de 3COM, con el objeto de hacer desarrollos de conectividad y soporte mutuo de productos.

D.2 Sistema Operativo

Hasta ahora, "OfficeShare", es el único sistema operativo de red que ha sido liberado totalmente por HP.

OfficeShare permite buena conectividad con sistemas minis especialmente de HP, dando al usuario normal de DOS la capacidad de compartir los recursos de estos sistemas desde su estación. Es necesario hacer notar que *OfficeShare* por si sólo no es capaz de tal conectividad, en algunos casos, es necesario que determinado software sea ejecutado en los equipos minis para tal propósito.

Ventajas :

- Fácil enrutamiento de trabajos de impresión.
- Facilidad para definir dispositivos compartidos como graficadores e impresoras.
- Transferencia de archivo a HP1000, HP3000, HP9000 y VAX a través de *AdvancedLink*.
- Uso de disco y archivos de una HP3000 con software de BSP (*Bussines System Plus*)
- Emulación de terminal para HP1000, HP9000/300, HP9000/800 y DEC VAX usando *HP Terminal Program* o *AdvancedLink* para emulación.
- Diferentes estructuras de conexión (HP StarLan, HP StarLan 10, HP ThinLan y HP Serial).

Desventajas :

- No soporta ambiente OS/2, ni ambiente Macintosh.
- Dependencia total de los productos hardware de HP.
- No tiene niveles de seguridad para controlar la confidencialidad de la información. Los usuarios tienen libre acceso a toda la información que reside en el servidor.
- No tiene la utilería de monitoreo de usuarios en línea.
- No detecta intrusos intentando violar información de la red.
- No tiene la utilería de contabilidad de recursos utilizados por los usuarios.

- El sistema se degrada cuando el servidor realiza trabajos de impresión y/o cuando se realizan trabajos de compilación.
- Pérdida de información debido a constantes colisiones de paquetes en el medio de transmisión.
- No contempla la posibilidad de discos en espejo duplicados o servidores en espejo.
- Reportes de falla levantados por usuarios finales que indican las siguientes fallas :
 - El servidor se bloquea sin aparente razón.
 - Constantes problemas en el cableado.
 - Sistema operativo lento.
 - Aplicaciones como EXCEL no se pueden hacer compartidas. Es necesario crear una copia del paquete por cada usuario que los requiera.

D.3 Topología

Soporta únicamente la topología Ethernet.

D.4 Tipo de Cable

Puede utilizar cable coaxial o cable telefónico.

D.5 Tipo de Servidor

- Servidor de archivos : Vectra PC (excepto modelo portátil CS) o una PC IBM compatible con 640 KB en RAM.
- Servidor de Procesos : Servidores HP3000/V con 4 MB en RAM, sistema operativo MPE/V

D.6 Tarjetas de Red

Tarjeta de red HP.

D.7 Interconectividad

D.7.1 Lan vs Lan

- Bridge interno : No soportado.
- Bridge externo : No soportado.
- Bridge remoto : A través de programas de comunicaciones para emulación de terminal.

D.7.2 Lan vs Mainframe

Comunicación a sistemas IBM disponible para protocolos SDLC.

D.8 Comentarios

OfficeShare no se ha caracterizado por ser un sistema operativo poderoso. Es evidente que tiene serias deficiencias las cuales no han sido corregidas por el soporte técnico de HP.

Dentro del mercado nacional, posee un porcentaje mínimo de sistemas operativos instalados.

Apéndice E Sistema operativo Lan Manager.

E.1 Antecedentes 3COM

3 COM Corporation (COMputer COMMunications, COMpatibility) fué fundada en Junio de 1979 por el Dr. Robert M. Metcalfe, uno de los principales coinventores de la tecnología *Ethernet* junto con el Dr. David Boggs.

Ethernet es una de las primeras tecnologías desarrolladas para redes locales y que mayor aceptación tienen en el mercado actualmente. A partir de 1980 la IEEE la designó como una norma y actualmente es aceptada internacionalmente para la construcción de redes con esta topología.

3COM ha aprovechado la aceptación internacional de *Ethernet* como norma de tecnología y ha desarrollado un gran número de productos de software y hardware para redes locales así como de conectividad con otros ambientes. Basados en está tecnología.

3COM posee aproximadamente un 13% de la base instalada de redes locales en los Estados Unidos.

En el mercado nacional 3COM ha realizado alianzas con otras empresa tales como Microsoft para la realización de sistemas operativos y con Hewlett Packard para futuros desarrollos de conectividad y soporte mutuo de productos.

La calidad de los productos de 3COM tienen una buena reputación en el mercado norteamericano y se caracterizan por su alto costo.

E.2 Sistema Operativo

3Com posee básicamente dos sistemas operativos :

- 3+Share
- 3+Open Lan Manager

E.2.1 3+Share

Es el sistema operativo básico de 3COM para redes locales bajo ambiente DOS. Es utilizado principalmente para pequeños grupos de trabajo que requieren compartir recursos. Es un sistema confiable, con buen manejo de colas de impresión, eficientes programas para la administración de la memoria caché, fácil administración y pantallas amigables para el usuario, diseñadas en windows.

Su principal desventaja es que consume aproximadamente 100Kb de memoria RAM en las estaciones de trabajo, además el manejo de las pantallas del sistema operativo se vuelve lento y tedioso sin la ayuda del "ratón".

E.2.2 3+Open Lan Manager

Es un sistema operativo para red desarrollado conjuntamente por MicroSoft y 3COM. Liberado parcialmente en octubre de 1988, basado en el sistema operativo multitarea OS/2.

Permite la integración de estaciones de trabajo bajo ambiente DOS y OS/2 en forma transparente.

Ventajas :

- Fácil definición de SPOOL de impresoras, modems, scanners y graficadores compartidos.
- Buen manejo de colas de impresión, con pantallas "pop up" informando al usuario sobre el estado de su trabajo, además de que el sistema le informa al usuario si la impresora necesita atención manual o bien le falta papel.
- Una cola de impresión puede manejar más de 9 impresoras o bien múltiples colas de impresión pueden manejar una impresora.
- Ruteo de trabajos de impresión : el administrador de la red puede configurar el sistema para que los trabajos de impresión sean enviados a la primera impresora disponible o alguna en específico.
- Soporta impresoras que manejan lenguaje PostScript.
- Excelentes niveles de seguridad, el administrador puede definir que archivos, discos, directorios, subdirectorios, impresoras, dispositivos de comunicación y otros recursos de la red puede acceder un usuario.
- Auditoría de la información de la red; por ejemplo, colas de impresión, capacidad utilizada de memoria, monitoreo de la actividad de los usuarios y en general accesos y usos a los recursos.
- Registro de errores detallados de "Logon" de usuarios y errores del sistema.
- Aproximadamente 250 API's.
- Integración de computadoras Apple Macintosh a la red. Actualmente soporta los protocolos : Xerox Network System (XNS), Data Link Control (DLC), Transmission Control Protocol/Internet Protocol (TCP/IP).

Desventajas :

- Es el equipo de hardware más caro del mercado.
- Alto consumo de memoria en el server y estaciones de trabajo.
- Dependencia de productos hardware de 3COM para el caso de interconexión de redes.
- No existe comunicación asincrónica para usuarios remotos que requieran aplicaciones en línea o de transferencia de archivos.
- Sin "ratón" el manejo del sistema operativo se vuelve lento y tedioso.
- Una de las mayores desventajas que existen en el sistema operativo 3+Lan es que no ha sido probado totalmente en un ambiente real con todas las aplicaciones que se mencionan en la información disponible de 3COM.

La razón es que muchas de las facilidades que se anuncian aún se encuentran en desarrollo. Y se espera que para mediados de este año sean liberadas; sin embargo, muchas de ellas tendrán que ser adquiridas como un producto adicional del sistema operativo.

A continuación se mencionan algunas de las facilidades que no han sido implementadas:

- 3+Backup : Software de backup en el servidor, el cual puede ser automáticamente programado (en desarrollo).
- 3+Fault Tolerant\DM : Software para tener discos espejo con información redundante (en desarrollo).
- 3+Fault Tolerant\SM : Servidores en espejo (en desarrollo).
- 3+Open Lan View : Utilería de supervisión de red de forma gráfica (en desarrollo).

- DPA (*Demand Protocol Access*) permite cargar automáticamente el protocolo de comunicación necesario para correr aplicaciones de otros ambientes (minis, múltiusuario o mainframe) desde las estaciones de trabajo (en desarrollo).
- Filosofía Cliente/Servidor : Para poder ejecutar otras aplicaciones dentro del servidor, desde las estaciones de trabajo en forma paralela, aprovechando la capacidad de múltiprosesos del OS/2 (en desarrollo).

Actualmente el concepto de cliente/servidor se encuentra disponible para procesos de bases de datos usando SQL, que es el manejador de bases de datos nativo para 3COM.

E.3 Topologia

3+Open se encuentra limitado a las topologías de Ethernet y Token Ring. La posibilidad de utilizar otro tipo de topología se encuentra sujeta a que los programas manejadores que proporciona el fabricante de la tarjeta de interfaz que se desee utilizar.

E.4 Tipo de cable

Se puede utilizar cable coaxial, cable telefónico, fibra óptica o una combinación de ellos por medio de hardware de 3Com llamado Multiconnect.

E.5 Tipo de servidor

Para servidores dedicados y optimizados se recomienda utilizar los modelos de 3COM 3S/200 o 3S/400 que vienen preparados con los requerimientos de memoria de 4MB de RAM y procesador 286 o 386.

Se pueden utilizar microcomputadoras con procesador 286 o 386 como servidores dedicados y no dedicados, con la desventaja que tienen que cumplir con los 4 MB de memoria mínimo.

E.6 Tarjeta de red

3Com soporta únicamente sus tarjetas de EtherLink Plus para topología Ethernet o Token Ring para PC y PS/2. En este sentido 3COM está limitado en el tipo de hardware que se puede utilizar.

E.7 Interconectividad

E.7.1 Lan vs Lan

- Bridges Internos : Se pueden tener hasta 4 tarjetas de red en el mismo servidor para interconectar redes locales de la misma topología o diferente, por ejemplo Macintosh-Token Ring y Ethernet.

- Bridges Externos : Este tipo de brigdes se pueden realizar de dos maneras :

- 1) Por medio del repetidor MultiConnect de 3COM. Su principal desventaja es que no restringe el tráfico de paquetes dentro del medio de transmisión.

- 2) Por medio del NETbuilder en sus diferentes modelos. Este producto es inteligente ya que emplea el procesador 68020 a 20 Mhz para hacer el filtrado de paquetes, entre las diferentes redes conectadas y evitar el exceso de tráfico en el medio de transmisión.

Ambas soluciones son efectivas pero muy costosas.

- Bridges Remotos : Se pueden implementar por medio de un dispositivo de hardware llamado NetBuilder IB/3 para enlaces remotos de redes por medio de interfases T-1 o enlaces a 56 Kbps.

E.7.2 Lan vs Mainframe

- Gateways : Hasta 32 sesiones de emulación de terminal 3270 concurrentes o sesiones de APPC pueden ser logradas con la tarjeta de comunicaciones Maxess SNA de 3COM.

E.8 Comentarios

3+OPEN LAN MANAGER es un sistema bien estructurado, poderoso, seguro y con un excelente control sobre los usuarios del sistema y los recursos a compartir.

Algunos resultados mostrados en pruebas de evaluación de 3+OPEN LAN MANAGER comparado con otros sistemas en igualdad de condiciones, indican que es eficiente y con buena respuesta de tiempo de accesos a disco.

Como sistema operativo de red es sin lugar a dudas uno de los mejores en el mercado. Sin embargo, debido a que no está totalmente liberado no se puede hacer uso de las aplicaciones de procesos distribuidos, ni de los que 3COM a denominado DPA para cargar automáticamente el protocolo de comunicación adecuado para correr aplicaciones de otros ambientes, desde las estaciones de trabajo. Es posible que 3+OPEN tome un año más para poder ofrecer un producto completo con todas las facilidades que se han mencionado.

Con respecto a la conectividad 3COM presenta varios productos de comunicación con los que es posible conectarse a otros ambientes como mainframe, múltiusuario, minis o redes Macintosh.

Uno de los puntos claves que se encuentran en este momento en contra de 3+OPEN es su excesivo requerimiento de memoria RAM tanto en el servidor como en las estaciones de trabajo. Para el caso de las estaciones de trabajo bajo DOS es casi imperativo proveerlas con 1MB de memoria ya que unicamente el shell del sistema ocupa 105KB.

En relación al hardware de 3COM, sus productos poseen buena calidad y prestigio dentro del mercado; por lo mismo son de un mayor costo que otros productos de similares cualidades. Es importante señalar que 3COM es una compañía tanto de software como de hardware con una vasta experiencia en ambos campos. Sin embargo, sus productos son altamente dependientes uno del otro, lo cual restringe al cliente en la elección del equipo que desea adquirir.

Bibliografía

Atre Shakuntala. DataBase : Técnicas de bases de datos : Estructuración en diseño y administración. Ed. Trillas 1988.

Booth Grayce M. The distributed system environment some practical approaches. Ed. McGraw-Hill, 1981.

Bray Olin H. "Distributed database design consideration". Trends and Applications 1976 Computers Networks IEEE.

Bray Olin H. Distributed databases management systems. Ed. Lexington Books 1982.

Carlesi C., Ricciardi E. and Thanos C. "Data distribution criteria and query processing strategy in a small distributed environment". Distributed databases North-Holland Publishing company, INRIA 1980.

Ceri S., Pelagatti G. Distributed databases Principles & Systems. Ed. McGraw-Hill 1985.

Champine G.A. "Six approaches to distributed databases". Datamation Magazine May 1977.

Chou Chuen-Pu and Liu Ming T. "A concurrency control mechanism and crash recovery for a distributed databases system (DLDBS)*." Distributed databases North-Holland Publishing company, INRIA 1980.

Chu Wesley W. "Performace of file directory systems for databases in star and distributed networks". AFIPS Conference Proceedings 1976 National Computer Conference.

Chu Wesley W. And Ohlmacher. "Avoiding deadlock in distributed databases". Proceedings of the ACM-National Symposium 1974.

Date C.J. Introducción a los sistemas de bases de datos. Editorial SITESA Addison_Wesley Iberoamericana, 1986.

Davenport R.A. "The state of the art of distributed databases". North-Holland Publishing Company 1978.

Davis Dwight. "Benchmarking the impacts of distributed databases architecture". Netware/October 1988.

Davis D.W. Distributed Systems : Architecture and implementation: An advanced course. New York Springer 1981.

Deakin Rose. Bases de datos. Uso y administración en centros de cómputo. Ed. Trillas 1988.

Dettman Terry. Dos Programmer's Reference
2 nd. Edition Ed. Que 1989.

Draffan W. and Poole J. Distributed databases An advanced Course. Ed. Cambridge Univ. Press 1980.

Duncan Ray. Advanced MSDOS
2 Edition Ed. Microsoft Press 1989.

Durr Michael, Gibbs Mark. Networking Personal Computers, 3 edition Ed. QUE Corporation 1989.

Finkelstein Richard. "Client/Server computing. The best of two worlds". Connect the journal of computer networking. Summer 1989 Vol. 2 Number 4.

Gardarin Georges. "integrity, consistency, concurrency, reliability in distributed databases management systems". Distributed databases North-Holland Publishing company, INRIA 1980.

Holbrook Steve. "Designing distributed applications for netware". Netware/October 1988.

Kernighan Brian W, Ritchie Dennis. El lenguaje de programación C. Edición en español. Ed Prentice Hall 1985.

James Martin. Computer networks and distributed processing software, techniques and architecture. Ed. Prentice Hall 1981.

Hordeski Michael F. Microcomputers Lans : Network design and implementation. Ed. Tab Professional and Reference Books.

Korth Henry F., Silberschatz Abraham. Fundamentos de bases de datos. Ed. McGraw-Hill 1987.

Lauriston Robert. "Seven top lan's: A hard look of ease of use". Pc World June 1989.

Liebowitz Burth H., Carson John. Distributed Processing. Second Edition 1978 Ed. IEEE Catalog No. EHO127-1.

Lobel C. De and Litwin W. Distributed databases. International Symposium on distributed databases. Paris, Francia 1980.

Lorin Harold. Aspects of distributed computer systems. Second Edition Ed. Wiley 1988.

Lyon John M. Introducción al diseño de banco de datos. Ed. Limusa 1986.

Major Drew. "Distributed versus centralized processing in Pc networks". Netware/October 1988.

Mariani M.P. Distributed data processing : technology and critical issues. Amsterdam 1982.

Martella G. and Schreiber F.A. "A data dictionary for distributed databases". Distributed databases North-Holland Publishing company, INRIA 1980.

Nantucket Clipper. Manual 1987.

Norton Peter, Wilton Richard. The New Pter Norton Programmer's guide to IBM PC & PS/2. Ed. Microsoft Press 1988.

Novell Inc. Application programmer's guide to Netware. Novell Inc 1986. February 1986 Edition. Manual de revisión 1.00

Novell Inc. Distributed Application processing in the netware environment. First Printing June 1988.

NovellCo. Seminario de productos Novell Noviembre 1989. Novellico de México S.A. de C.V.

NovellCo. Seminario técnico Novellico Abril 1987. Novellico de México S.A. de C.V.

Parsaye Kamram, Chignell Mark, Khoshafian Setrag, Wong Harry. Intelligent databases. Object-Oriented, deductive hypermedia technologies. Ed. John Wiley & Sons 1989.

Peebles Richard and Manning Eric. "System Architecture for distributed data managment" IEEE January 1978.

Rothnie James B. and Bernstein PA., Fox S. "SDD-1 A system for distributed databases". ACM transaction on data base Systems Vol. 5 No.1. March 1980.

Rothnie James B. and Goodman Nathan. "A survey of reseach and development in distributed database managment". Proceedings of the third International Conference on very large databases, 1977. IEEE.

Schatt Dr. Stan. Understanding local area networks. Ed. W Sams & Company 1987.

Sloman Morris and Kramer Jeff. Distributed systems and computer networks. Ed. Prentice Hall (Uk) 1987.

Sommerville Ian. Ingeniería de Software. Ed. SITESA Addison Wesley Iberoamericana. Segunda Edición 1988.

Spence Rick. Clipper Programming Guide Mocritrend Books 1989.

Squire Enid. Introducción al diseño de sistemas. Ed. Representaciones y Servicios de Ingeniería S.A. Fondo Educativo Interamericano 1984.

Stallings William Ph. D. Local networks an introduction. Ed. Macmillan Publishing Company 1984.

Stoney Jon. "A few well-chosen phrases". New Scientist 20/27 December 1984.

Walter Bernd. "Concepts for a robust distributed database system". Distributed databases North-Holland Publishing company, INRIA 1980.

Wiederhold Glo. Diseño de bases de datos. 2. Edición. Ed. McGraw-Hil 1985.

Zemrowski Kenneth. "Problems of database use in a distributed rdata network" Fifteenth Annual Technical Symposium 1976.