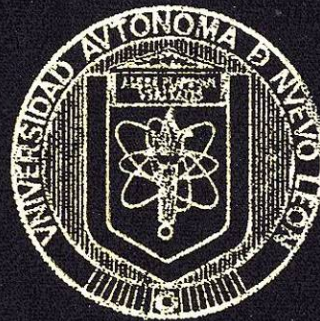


**UNIVERSIDAD AUTONOMA DE NUEVO LEON**

FACULTAD DE CIENCIAS FISICO MATEMATICAS



REQUERIMIENTOS PARA APLICACIONES  
MULTI-USUARIO EN REDES LOCALES

**T E S I S**

QUE EN OPCION AL TITULO DE  
LIC. EN CIENCIAS COMPUTACIONALES

PRESENTA

SERGIO MARGARITO AGUIRRE TAMEZ

SAN NICOLAS DE LOS GARZA, N. L. ENERO DE 1990



TL

TK5105

.5

A38

1990

c.1



1080171499

T-157

C-2

L.C.C.



# UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE CIENCIAS FISICO MATEMATICAS

0. PROLOGO

1. INTRODUCCION

2. CONCEPTOS GENERALES.

3. DISEÑO DE APLICACIONES MULTI-USUARIO EN REDES LOCALES.

3.1. APLICACIONES TÍPICAS

3.2. DIFERENCIAS BÁSICAS ENTRE APLICACIONES MULTI-USUARIO Y  
UNI-USUARIO.

3.3. MÉTODOS BÁSICOS TÍPICOS

3.4. RUTINAS DE ERROR.

4. PROGRAMANDO APLICACIONES MULTI-USUARIO EN FOXBASE+/LAN.

5. CONCLUSIONES.

6. BIBLIOGRAFIA



## REQUERIMIENTOS PARA APLICACIONES MULTIUSUARIO EN REDES LOCALES

### T E S I S

QUE EN OPCION AL TITULO DE  
LIC. EN CIENCIAS COMPUTACIONALES

PRESENTA

SERGIO MARGARITO AGUIRRE TAMEZ

SAN NICOLAS DE LOS GARZA, N. L. ENERO DE 1990



## CONTENIDO

0. PROLOGO.
1. INTRODUCCION.
2. CONCEPTOS GENERALES.
3. DISEÑO DE APLICACIONES MULTI-USUARIO EN REDES LOCALES.
  - 3.1. APLICACIONES TIPICAS DE UNA LAN.
  - 3.2. DIFERENCIAS BASICAS ENTRE APLICACIONES MULTI-USUARIO  
UNI-USUARIO.
  - 3.3. METODOLOGIAS TIPICAS DE DISEÑO.
  - 3.4. RUTINAS DE ERROR.
4. PROGRAMANDO APLICACIONES MULTI-USUARIO EN FOXBASE+/LAN.
5. CONCLUSIONES.
6. BIBLIOGRAFIA.



## **0. PROLOGO**

Lo que me motivó a desarrollar este tema de tesis fue la necesidad que tube de una documentación que hablara sobre aplicaciones multi-usuario cuando intervine en la conversión de un sistema, hecho en FoxBase+, que funcionaba en ambiente local uni-usuario, a un ambiente multi-usuario.

Este documento lo elaboré en el procesador de texto WordPerfect Versión 5.0; gracias a este procesador de texto, se hizo posible introducir en el mismo documento las gráficas que incluye. Para elaborar las gráficas utilicé los paquetes gráficadores Picture Maker y Pro-Design que tienen la capacidad de mostrar en una pantalla la gráfica completa (legiblemente), requisito obligado para utilizar el capturador de pantallas del WordPerfect.

Me siento muy agradecido con mi esposa Norma Patricia por el apoyo que me brindó en el transcurso del desarrollo del tema y desde que nos conocimos. Agradezco también a mi familia, a mis padres y mis hermanos por apoyarme durante toda mi carrera profesional.

Agradezco al Ing. Oscar Recio Cantú, asesor de mi tema, por su tiempo y conocimiento que tan generosamente me otorgó y también mis agradecimientos a toda la gerencia de Sistemas de HYLSA División Aceros Planos por el apoyo científico y tecnológico que se me brindó, y en la cual presto mis servicios desde Septiembre de 1988.

Sergio M. Aguirre Taméz



## **1. INTRODUCCION**

Un Concepto que surgió por el año de 1980 es el de las Redes Locales (en inglés Local Area Network) y ha venido adquiriendo cada vez mayor importancia debido básicamente a la versatilidad que posee para intercomunicar equipos digitales de tipos muy diversos y además la posibilidad que nos ofrece para usar terminales inteligentes, que dicho en otros términos sería el llamado proceso distribuido.

El poder interconectar sistemas de comunicación de voz, sistemas de comunicación de datos y sistemas de video, que siempre se han desarrollado en forma individual, es una de las causas del reciente interés, de usuarios y vendedores, en las Redes Locales.

Todos los dispositivos que pueden intercambiar información de alguna forma con otros dispositivos pueden ser conectados por una LAN. La forma de la información pueden ser datos codificados o no codificados, o pueden ser señales de voz o video. Los dispositivos pueden abarcar unidades de desplegado, archivos y unidades de impresora, sensores y animadores, unidades de copiado, teléfonos, cámaras de video y monitores. También todos los sistemas de computadora, sistemas de control o tableros de conmutadores electrónicos pueden ser conectados por, o incorporados a una Red Local.

El *Ambiente Multi-Usuario* es aquel en el que varios usuarios usan o comparten simultáneamente recursos de hardware, software y



archivos de datos.

El compartir hardware significa, la capacidad de poder usar (o conectarse a) otros recursos hardware (equipos) que no necesariamente están conectados directamente al equipo en uso. Por ejemplo un usuario no tiene conectada a su PC una impresora laser (equipo muy costoso que solo lo tiene la Dirección General de la empresa), pero puede rutear su impresión a donde si la tienen y necesariamente esta incorporada a la Red Local.

Otro concepto del *Ambiente Multi-Usuario* es el de compartir software y se refiere a la capacidad que tienen los usuarios de usar los mismos paquetes de software como son hojas electrónicas, graficadores, procesadores de palabra etc. La realidad es que este concepto no parece tener mucha importancia, pero si lo vemos desde el punto de vista de almacenamiento físico representa un gran ahorro, ya que evita la duplicidad.

Y por último tenemos el concepto de compartido de archivos que se puede decir que es el mas importante aunque en las Redes Locales es la parte que menos ventajas presenta. Este concepto se refiere a la capacidad que tienen los usuarios de hacer actualizaciones al mismo archivo en forma simultanea.

Los paquetes de software no permiten compartir archivos, esto es, por ejemplo dos usuarios no pueden acceder la misma hoja electrónica (archivo en el que se guardan los datos de una hoja

electrónica) ni modificarla.

En los equipos grandes (Mains Frames) las aplicaciones que se desarrollan son, por añadidura del mismo computador: *Multi-Usuarios*, es decir, *Los Requerimientos para que una Aplicación sea Multi-Usuario* son automáticos pero en las Redes Locales no es así y esto es precisamente lo que trataremos.



## **2. CONCEPTOS GENERALES**

En este capítulo presentamos el significado de conceptos relacionados con las Redes Locales, sin embargo un estudio exhaustivo de ellos quedaría fuera del alcance de este material.

### ¿ Que es una Red Local ?

Una Red Local es un sistema de transporte de información entre dispositivos ubicados en una misma localidad, tales como un edificio de oficinas, una planta manufacturera, un área de hospital, el campus de una universidad, o alguna otra area geográficamente limitada. Tales localizaciones, con frecuencia son llamadas "Establecimientos".

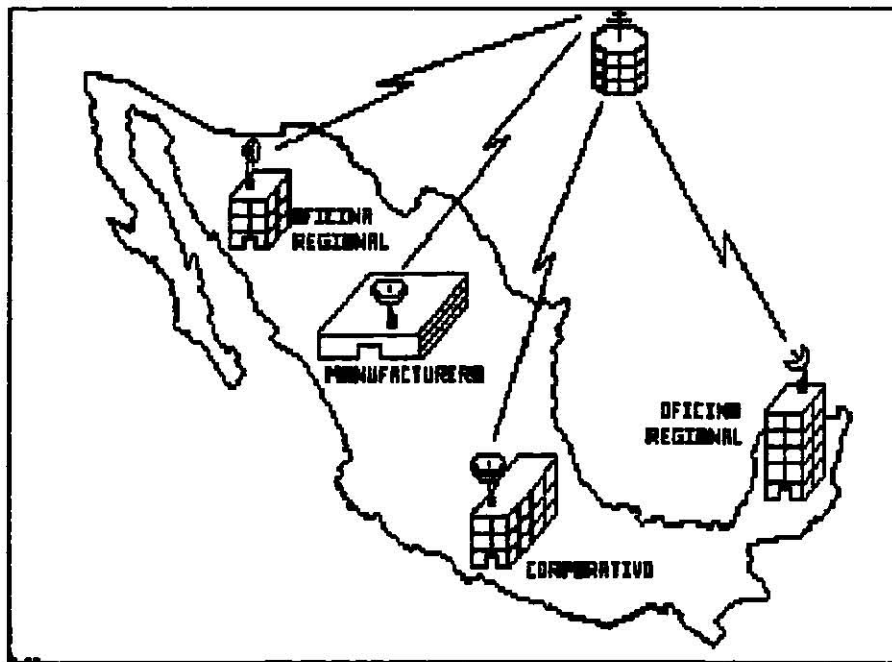


Fig 2.1 Sistema de Información de una Empresa

Sin considerar el tamaño, las empresas tienen la constante necesidad de procesar y usar información precisa y oportuna. Una considerable cantidad de recursos y tecnología tiene que ser

aplicada para satisfacer las necesidades de comunicación entre establecimientos. (Figura 2.1).

Pero hasta recientemente, se empezó a prestar atención al movimiento de información dentro del establecimiento, esto es, un sistema de información dentro del establecimiento.

En resumen una red local es un grupo de dos o más computadoras interconectadas de tal manera que pueden compartir recursos e información.

Estos recursos a compartir se pueden clasificar en dos tipos:

- Hardware.-

- Espacio en disco duro.
- Impresoras (Laser, Banda, Plotter, etc.).
- Otros dispositivos como unidades de cinta magnética, etc.
- Gateways. Para enlazar recursos externos de datos en otra red, o en un computador grande (Host).

- Software.-

- Herramientas de productividad departamental.
- Procesadores de palabra.
- Hoja electrónica de cálculo.
- Bases de Datos.
- Graficadores.
- CAD.

- Aplicaciones departamentales multiusuario.

Estas últimas se tratarán con todo detalle en el siguiente capítulo.

### ¿ Por qué una Red Local ?

Según estadísticas que manejan algunas publicaciones, el 80% de la información que genera una empresa se utiliza dentro de la misma. Es decir, dentro de un edificio, grupo de edificios o campus.

Los sistemas de automatización de oficinas requieren el uso de sistemas de red. La proliferación de pequeños computadores usados para procesamiento distribuido incrementa la necesidad de comunicación en un establecimiento para facilitar el intercambio de datos y aplicaciones.

Otro factor importante del porqué de una LAN, consiste en aprovechar las ventajas que nos ofrece esta tecnología.-

- Máquinas pequeñas obtienen el poder de las grandes.
- Todos pueden compartir el uso de equipos especializados.
- Programas y datos pueden ser almacenados, como si fuera una copia maestra para todos en la red.
- Los datos pueden ser transportados de una manera simple.
- Las microcomputadoras en una red (LAN), pueden compartir un enlace de comunicaciones a HOST.



## ¿ Cuáles son las principales características de las LAN ?

Primeramente de acuerdo al ambiente en las que pueden ser utilizadas, clasificaremos las redes locales en dos:

### - Redes de Oficina.-

Son aquellas en las que sus nodos lo componen básicamente un tipo de microcomputador en la cual comparten sus recursos de hardware y de software.

### - Redes industriales.-

Se caracterizan básicamente en que sus nodos están compuestos de computadores, controladores de proceso y robots de múltiples proveedores. Por lo general estas redes tienen capacidad para manipular información codificada y no codificada.

Las principales características de una red local son:

- La distancia que cubre es hasta de 10 Kms.
- Diseñada para existir en la geografía de una empresa, edificio o grupo de edificios.
- La velocidad de transmisión generalmente es de 1 o más megabits por segundo.
- Maneja tipos de información: codificada (datos y texto) y no codificada (voz, imagen y video).
- Independencia del dispositivo o computador conectado. Esto es a través de cables, adaptadores y protocolos de comunicaciones

comunes.

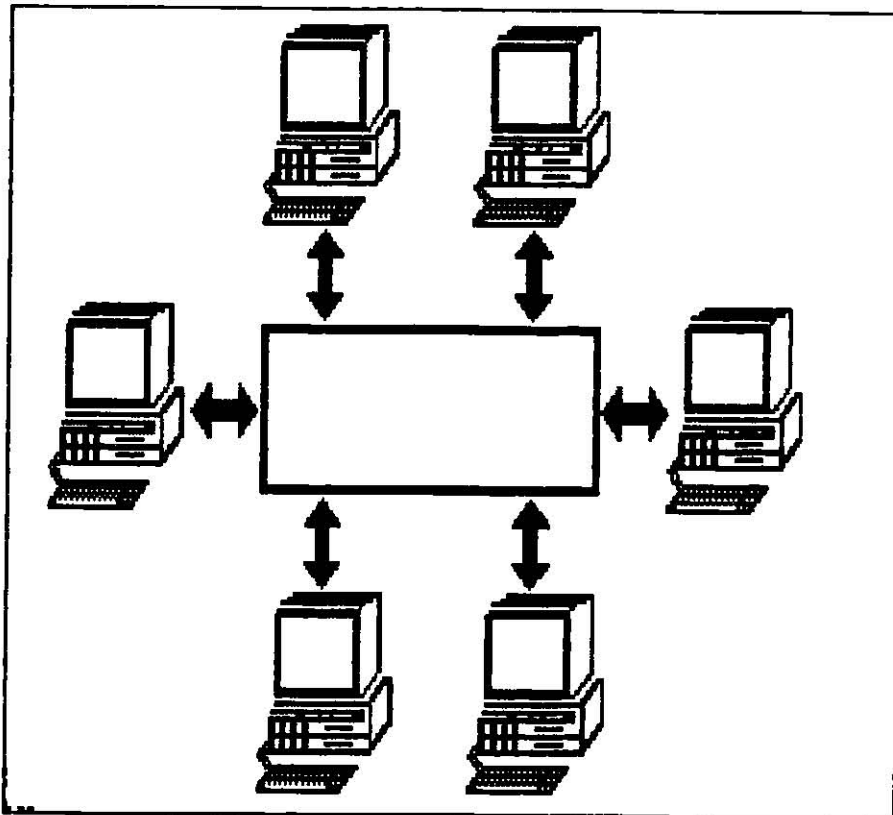
- Su meta es, apoyándose en el punto anterior, la conexión a la red de dispositivos de múltiples proveedores.

### ¿ Que son topologías de una LAN ?

Una topología de LAN es el modelo de interconexión usado entre los diferentes nodos de una red. A continuación se presentan ejemplos de topologías típicas de una LAN.

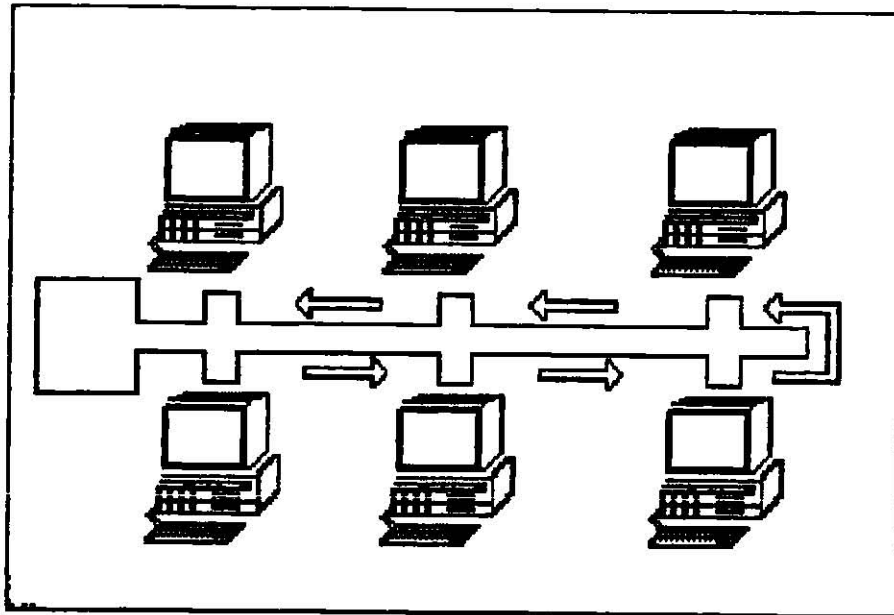
#### Estrella.

Todos los nodos están conectados directamente a un punto central, usualmente el, "File Server".



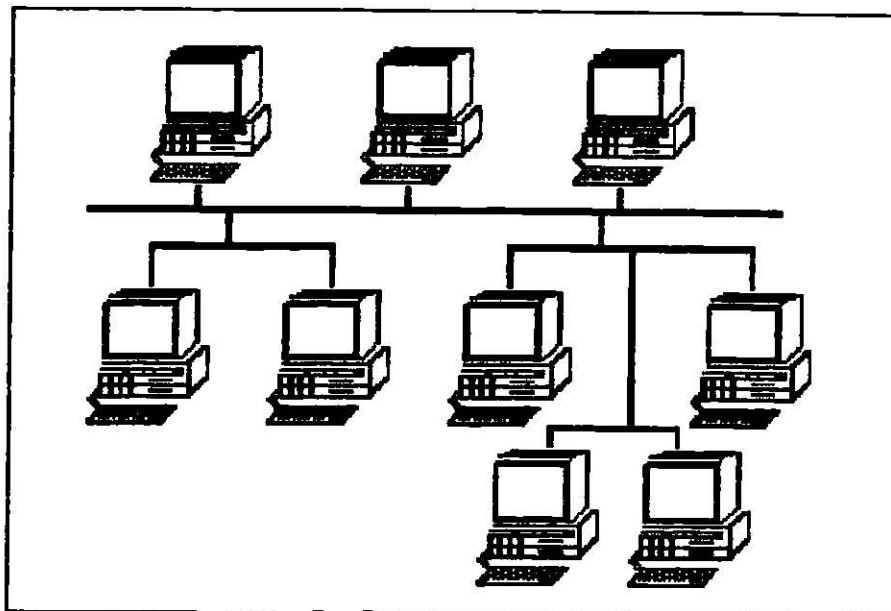
## BUS.

Sus nodos están conectados a través de hardware apropiado de interface a un medio de transmisión de datos lineal o bus, conexión multipunto.



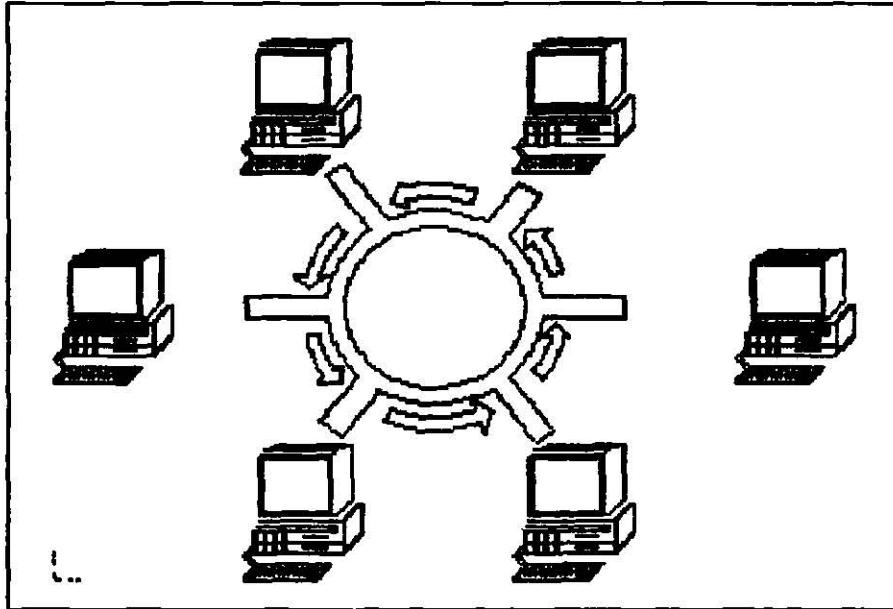
## Arbol.

Generalización de la topología de bus, el medio de transmisión consiste en un bus ramificado sin lazos cerrados.



Anillo.

Nodos conectados con enlaces punto-punto formando un lazo cerrado.



Las topologías estrella, bus, árbol y anillo frecuentemente tienen un punto central de control (estación principal o estación maestra). Esta asociación entre topologías y control, sin embargo, no son necesariamente una limitación de diseño. Una red puede también ser diseñada combinando algunas de las topologías básicas.

¿ Cuáles son los medios de comunicación usados por las LAN ?

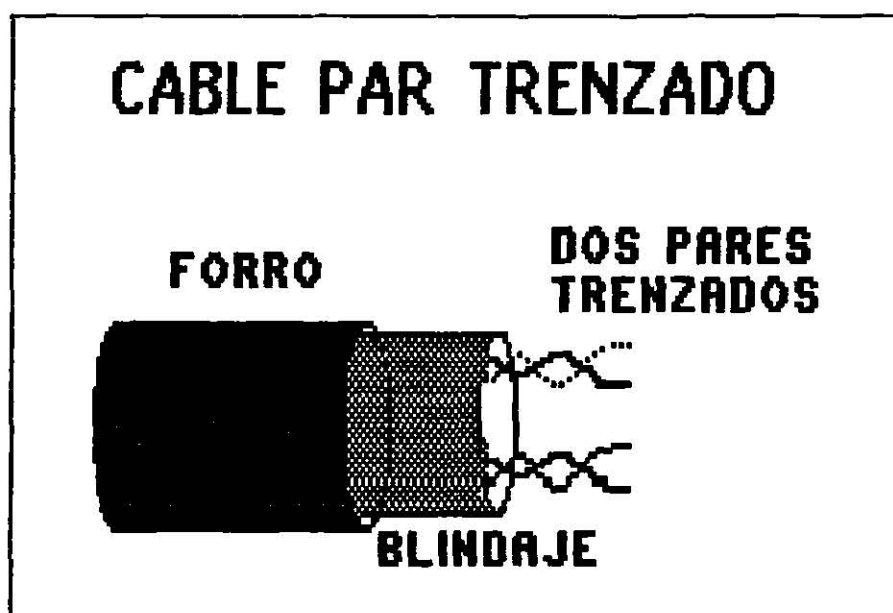
Para la mayoría de las aplicaciones de LAN, los medios de transmisión son cables que acarrean las señales a través del establecimiento. Las compañías construyen sus nuevos edificios considerando la selección de los medios de transmisión, en vez de



más tarde tener que rediseñar la construcción para adecuar el edificio.

Tres de los más importantes medios de comunicación son:

**Par Trensado.**- Consiste en un par de alambres acomodados en forma de espiral protegidos por un blindaje y cubiertos por un forro dieléctrico.

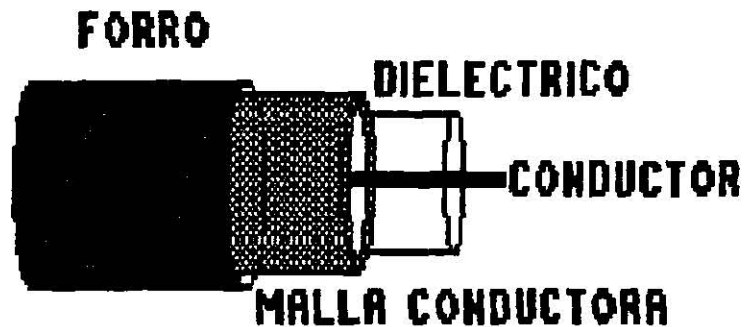


**Características.-**

- Posibilidad de transmitir señales análogas o digitales.
- Cobertura de la señal
  - Análoga hasta 5 Km
  - Digital hasta 25 Km.
- Bajo costo.

**Cable coaxial.**- Consiste en un par de conductores donde uno de ellos con forma cilíndrica rodea al otro conductor, que es un alambre, separados por un dieléctrico.

## CABLE COAXIAL



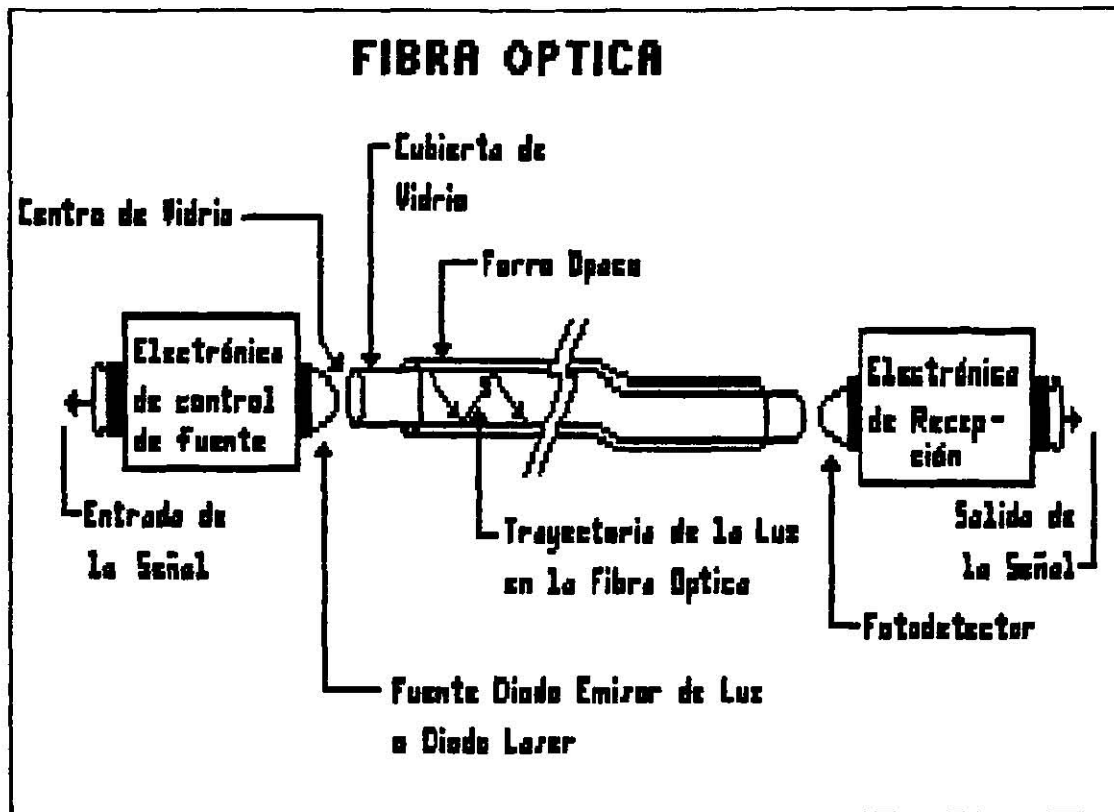
### Características.-

- Transmisión digital (50 ohms)
  - Cobertura hasta de 10 Km.
- Transmisión análoga (75 ohms)
  - Cobertura hasta 50 Km.
- Alto costo.

Fibra óptica.- Consiste en una fibra de plástico o vidrios (silicatos) por la cual puede viajar un haz luminoso. Se pueden transmitir pulsos luminosos o una luz continua modulada en intensidad. Se utiliza básicamente en enlaces punto-punto.

### Características:

- Gran capacidad de transmisión y de cobertura.
- Inmune a ruido (EMI y RFI).
- Aislamiento eléctrico.
- Alto costo.
- Requiere convertidores.



¿ Que es un protocolo de LAN ?

Se refiere al método que usan las estaciones de una red para compartir el cable que las interconecta, de tal manera que los mensajes que ellas envían no interfieren entre sí.

Existen varios métodos disponibles, sin embargo la mayoría de las redes (LAN's) utilizan los esquemas TOKEN PASSING ó CSMA/CD.

Protocolo CSMA/CD.

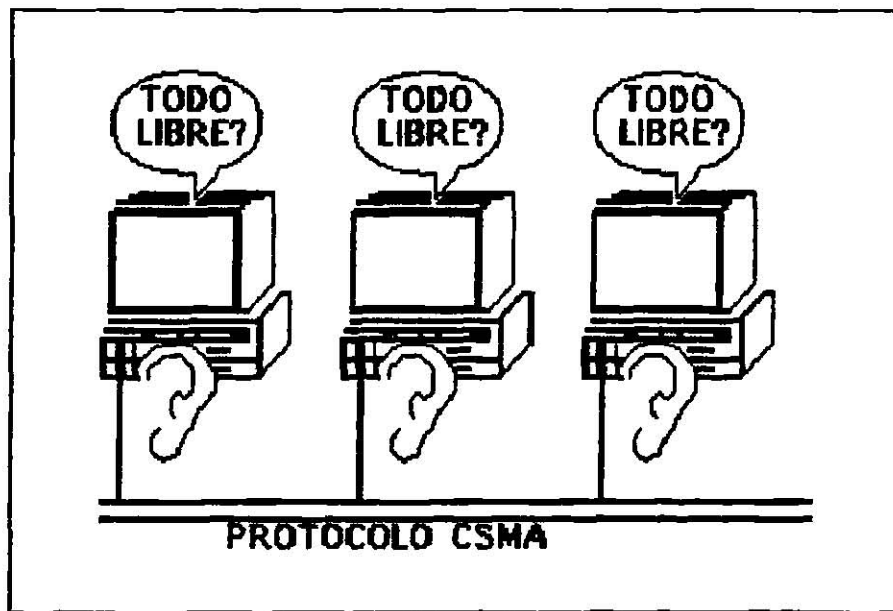
Eventos relevantes bajo este protocolo.

- \* Una estación "escucha" en la línea por una señal del mensaje.
- \* Si la señal no está presente la estación envía su mensaje y

monitorea la línea para detectar alguna posible "colisión".

- Si la "colisión" se presenta la estación detiene la transmisión y avisa a las demás.

- Las estaciones involucradas en la "colisión" esperan un tiempo específico antes de transmitir sus mensajes.



Redes que utilizan este protocolo.-

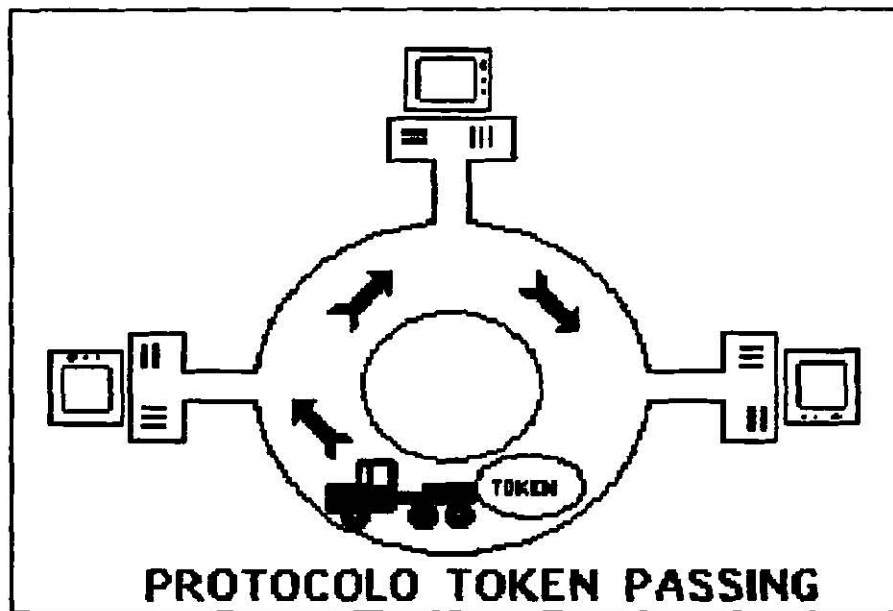
- Ethernet.
- Ungerman Bass.
- IBM PC-Network.

Protocolo Token Passing.

Eventos relevantes bajo este protocolo.-

- Un "token" circula alrededor de la red hasta que una estación esperando transmitir lo intercepta.
- Le agrega un mensaje con la dirección de la estación del destinatario.

- La estación destinatario copia el mensaje del "token" y lo envía a la próxima estación.
- La estación originadora copia el mensaje determina si este llegó intacto leyendolo cuando completa el circuito.
- El "token" es entonces limpiado y regresado a la red.



### ¿ Hay estandares para las LAN's ?

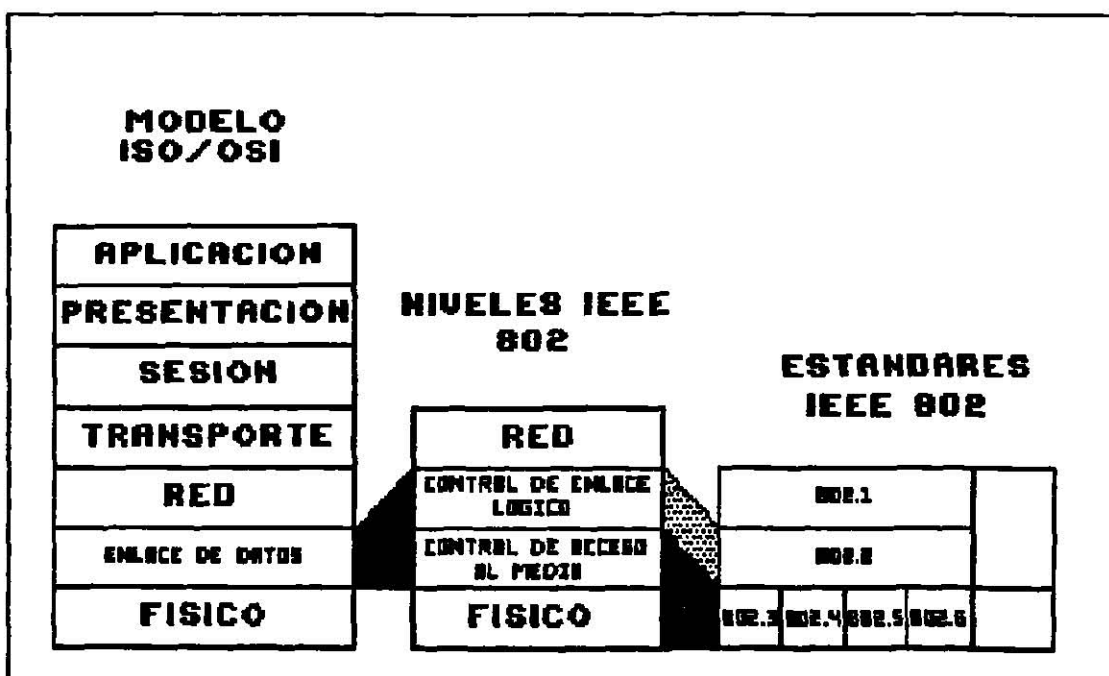
Con el anticipado crecimiento en las redes locales, forçados estandares fueron iniciados por algunas corporaciones. El mejor trabajo que se conoce es el del comite IEEE (Institute of Electrical y Electronics Enginers). En 1980 la IEEE estableció el comite 802 y el árbol de subcomites para revisión y endosamiento de estandares propuestos para varios aspectos de las redes locales (facilidades en transmisiones físicas, enlaces de datos y medios de control de acceso e interfaces con altos niveles de comunicaciones).



Las actividades sobre estandares de redes locales han sido intensas, sin embargo los comites de la IEEE continuan sus trabajos sobre interfaces físicas y lógicas de alto nivel con el objetivo de desarrollar un conjunto completo de estandares propuestos, esperando asegurar compatibilidad entre equipos disimiles y que los fabricantes adopten el estandar 802.

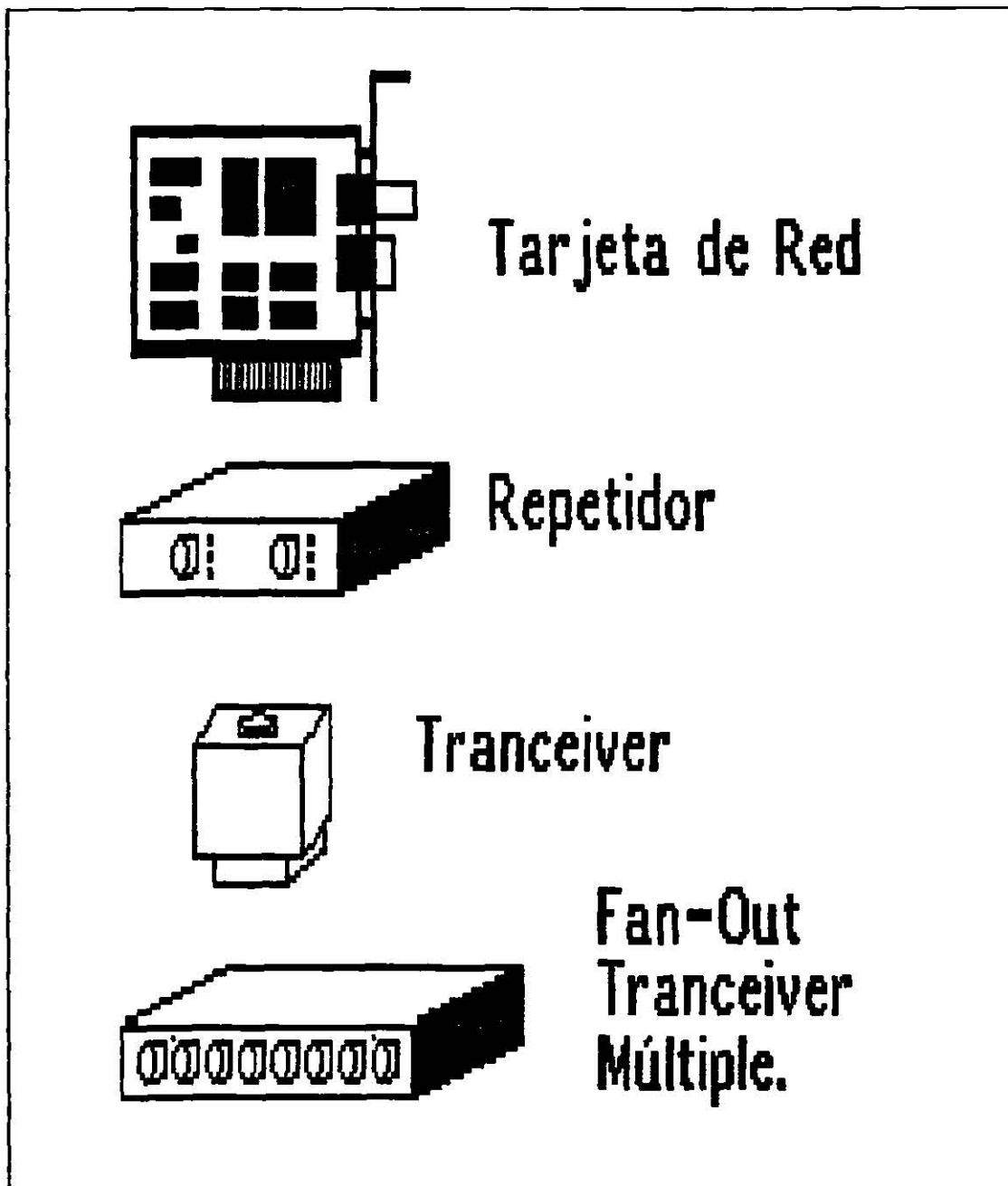
También dentro de la International Organization for Stadares (ISO), los grupos que trabajan con los más bajos niveles de la Open Systems Interconection (OSI) ejecutan actividades de estandarización que referencian modelos enfocados a redes locales.

El Proyecto IEEE 802.X mapea los tres niveles más bajos del modelo ISO/OSI, que son el red, enlace de datos y el nivel físico.

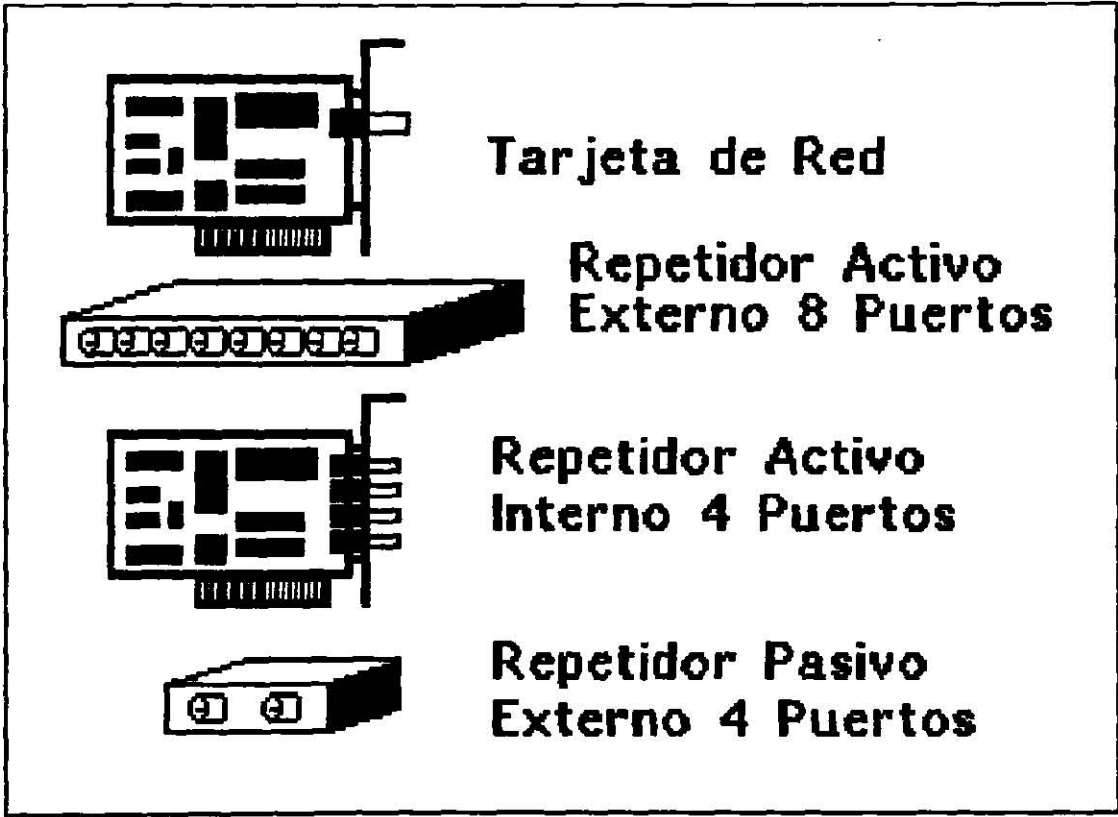


**¿ Que elementos componen a una Lan ?**

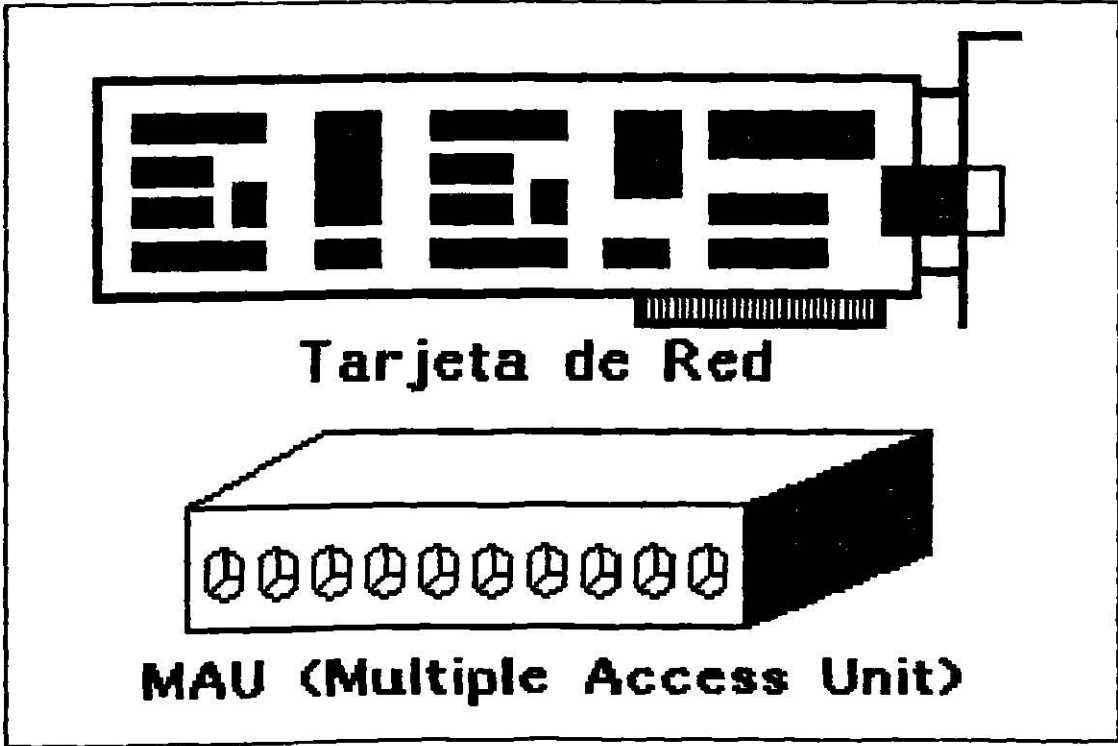
Los componentes de hardware de una red local varían según el tipo, a continuación se muestran los componentes de Ethernet, Arcnet y Token Ring.



Componentes una Red Ethernet.



Componentes de una Arcnet.



Componentes de una red Token Ring

## ¿Que características debe tener un Sistema Operativo de Red?

Las características relevantes de un sistema operativo de red son :

- Compatibilidad con el sistema operativo MS-DOS y OS/2.
- Compartición de discos y periféricos.
- Sistema de seguridad.
- Bloqueo de registros y archivos.
- Manejo de colas de impresión.
- Optimización en el manejo de discos.
- Mensajes.
- Correo electrónico.
- Servidores dedicados y no-dedicados.
- Protección contra copia.
- Ayuda en línea y utilerías.
- Acceso remoto.
- Contabilidad de uso de recursos.
- Estadísticas de uso y de comportamiento de parámetros de configuración.
- Sistema tolerante de fallas.

A continuación se presenta una tabla mostrando las características de cinco sistemas operativos que hay en el mercado.



TABLA 1: CARACTERISTICAS DE CINCO SISTEMAS OPERATIVOS

	IBM PC LAN Program 1.30	IBM OS/2 LAN Server 1.00	3Com+Open 1.0	3Com+Share 1.3.1	Novell SFT NetWare 286 version 2.15
Precio	\$245 por usuario	\$1040 para Server \$830 por OS/2 workstation \$245 por DOS workstation	\$995 para 5 usuarios \$2995 sin limite de usuarios	\$595 para 5 usuarios \$2495 sin limite de usuarios	\$4695 más de 100 usuarios
Mínimos Requerimientos Server					
Hardware	80286 ó 80386	80286 ó 80386	80286 ó 80386	80286 ó 80386	80286 ó 80386
Software	DOS 3.3 o 4.0	OS/2 EE 1.1	OS/2 1.0	DOS 3.1 o mas	
Memory	640K para servicio extendido 350K para servicio básico	5 Mb	4 Mb	640K	1 Mb
DOS Memoria usada de la estación por el Sistema Operativo de la Red	81-190K	81-190K	NBP: 25K XNS:100K	NBP: 25K XNS:100K	45-60K
Repartición de recursos par-a-par	si	si	no	si	no
Disco caching	si	si	si	si	si
Correo electrónico	si	si	opcional	opcional	no
Disco de diagnóstico	no	no	no	no	si
Soporte múltiple de archivo Server	si	si	si	si	si
Server puede actuar como una estación	si	si	si	si	no
Administrador de accesos desde estaciones remotas	si	si	si	opcional	si
Monitoreo de tráfico					
Reporte actual de Log-ons	si	si	si	si	si
Reporte de recursos repartidos	si	si	si	no	si
Error de Log	si	si	si	si	si
Rastreo auditivo	no	si	si	no	si
Cuenta de cargo de usuario	no	no	no	no	si
Seguridad					
Passwords para usuario	si	si	si	si	si
Passwords para recursos	si	si	si	si	si
Restricciones de tiempo	no	no	si	no	si
Restricciones de direcciones de estación	no	no	no	no	si

TABLA 1: CARACTERISTICAS DE CINCO SISTEMAS OPERATIVOS

	IBM PC LAN Program 1.30	IBM OS/2 LAN Server 1.00	3Com+Open 1.0	3Com+Share 1.3.1	Novell SFT NetWare 286 version 2.15
<b>Soporte de Hardware de Red</b>					
Ethernet	no	no	si	si	si
Token Ring	si	si	si	si	si
ARCnet	no	no	si	no	si
<b>Opciones de conectividad</b>					
Apple Talk	no	no	si	si	si
LU 6.2	si	si	si	si	si
SNA	si	si	si	si	si
DECnet	no	no	si	si	no
TCP/IP	no	no	si	si	no
x.25	no	no	si	si	si

### **3. DISEÑO DE APLICACIONES MULTI-USUARIO EN LAN**

Hasta ahora hemos tratado en forma general lo que son las redes locales, pero es de nuestro interés centrar este estudio en el diseño de aplicaciones (programas) computacionales que sean capaces de permitir a los usuarios acceder los mismos archivos simultáneamente, tanto para consultarlos como para hacerles actualizaciones.

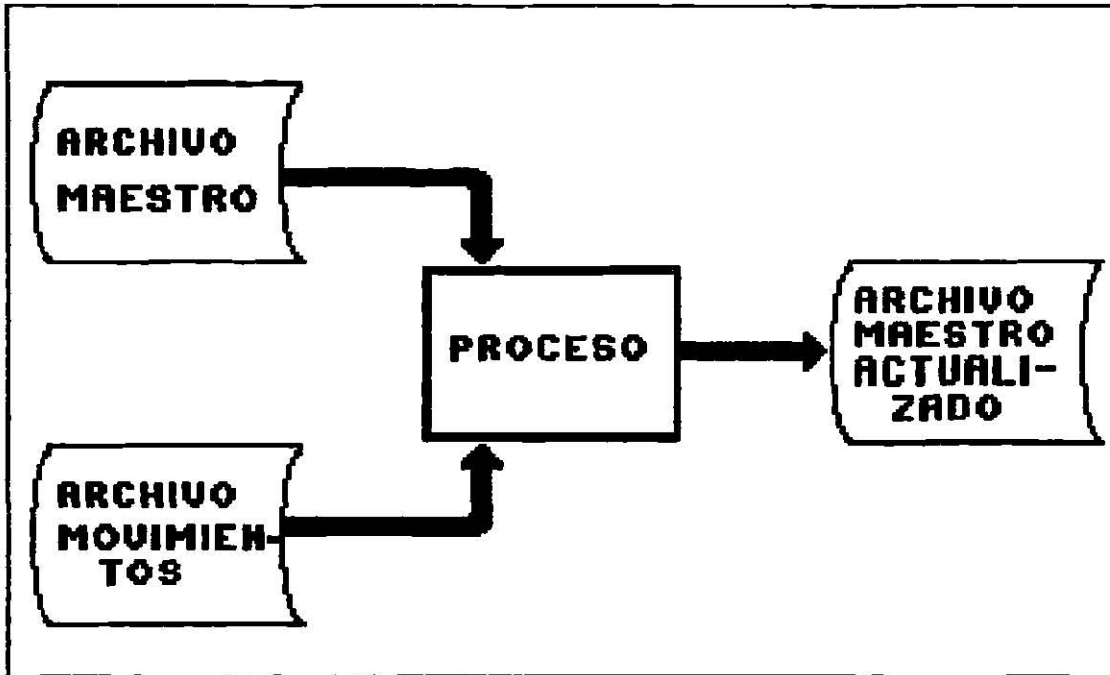
Las redes locales nos ofrecen muchas ventajas para el manejo de información, pero como todas las cosas, tienen también restricciones por lo que es importante ver qué tipo de aplicaciones son las idóneas para que funcionen adecuadamente en una LAN. En el presente capítulo trataremos sobre este tipo de aplicaciones, sus diferencias con las aplicaciones uni-usuario; trataremos de establecer metodologías o esquemas de diseño de aplicaciones multi-usuario y herramientas para desarrollarlas, con algunos ejemplos.

### **3.1 Aplicaciones típicas de una LAN.**

Podemos clasificar las aplicaciones computacionales en dos importantes tipos : interactivas y batch.

Las aplicaciones tipo batch son aquellas que procesan grandes cantidades de información usando generalmente archivos que contienen los movimientos (o actualizaciones) y archivos que van a ser afectados (modificados o actualizados) según los movimientos que se encuentren en los primeros. Para ilustrar este

tipo de aplicaciones veamos, por ejemplo, el proceso de un sistema en el que la información a procesar, primero se captura en forma masiva y luego se procesa.



Este tipo de aplicaciones (batch) son de uso común en los equipos grandes debido, no solo a su velocidad de proceso sino a la capacidad que tienen de atender un gran número de aplicaciones simultáneamente; cosa que en los equipos pequeños (microcomputadoras) no se da en forma eficiente, esto es cuando se ejecuta una aplicación batch en una microcomputadora, ésta queda bloqueada hasta que termine de ejecutar la aplicación.

Si la microcomputadora está incorporada a una red local, es decir que es un nodo de la red, el costo de ejecutar una

aplicación batch en ella sería tal que este nodo permanecería ocupado por un buen tiempo y en este caso tendríamos que considerar el tiempo de transmisión de datos, que exigiría el proceso, desde donde se encuentra el nodo hasta donde se encuentre el server. Por lo que las aplicaciones batch no son eficientes cuando se ejecutan en LAN's.

Las aplicaciones interactivas, como su nombre lo indica, son aquellas que interactúan con el usuario. Este tipo de aplicaciones son las que al ser ejecutadas, "esperan" a que el usuario introduzca la selección del tipo de movimiento que desea hacer, estas pueden ser.-

- Teclar una clave de acceso.
- Teclar una clave de tipo de actualización.
- Teclar selección de algún proceso especial y pequeño.

Citemos como ejemplo un sistema de inventario, en el que de diferentes localizaciones físicas, es operado por varias personas que piden o que hacen bajas, altas y cambios a un archivo que contiene las descripciones y cantidades de artículos que conforman el inventario. Ellos (los usuarios) actualizan simultáneamente y en tiempo real, el archivo de inventario a través de transacciones (llamaremos transacción a toda agrupación lógica de movimientos a archivos que el usuario ejecuta desde su nodo o terminal), por lo que a este tipo de aplicaciones se les ha llamado transaccionales. Dadas las características del modo de mantener actualizado un cierto conjunto de información, al tipo



de transmisión de datos que requiere y a la forma de proceso; este tipo de aplicaciones son ideales para funcionar en redes locales.

El hecho de que las aplicaciones tipo transaccionales sean idóneas para operar en redes locales, no significa que otro tipo de aplicación no pueda funcionar. Siempre, dentro de sistemas completos, existirá la necesidad de procesos batch (en cortes mensuales, reportes resumidos, opciones de recuperación o respaldo, etc.).

Existe también el procesamiento distribuido. En las máquinas computadoras grandes toda necesidad de proceso se satisface en una unidad central de proceso y sus terminales son simples dispositivos de entrada/salida, en cambio, en las redes locales sus terminales son inteligentes y tienen la capacidad de procesar información (microcomputadoras); desde este punto de vista podemos decir que esto es procesamiento distribuido, pero no solo es esto lo que encierra el concepto. Se han dado importantes avances en el software de red: como el poder direccionar trabajos (Job's), que requieren o que consumen mucho tiempo de CPU, a otros nodos que no están siendo usados o están siendo usados pero no a toda su capacidad, sin que el usuario de este último nodo se entere. Esta forma de procesamiento, aliviaría la necesidad de ejecutar trabajos que "amarrarían" un nodo sin que este pueda volver a ser usado durante el tiempo en el que el trabajo sea terminado.

## Diferencias básicas entre aplicaciones Uni-usuario y aplicaciones Multiusuario.

Las diferencias que hay entre las aplicaciones uni-usuario y multi-usuario radican básicamente en la forma de acceder y actualizar los archivos.

Para referirnos de forma precisa a cada una de las diferencias, a continuación listamos los diferentes tipos de accesos y actualizaciones que se le puede hacer a un archivo durante la ejecución de una aplicación en una microcomputadora :

- Abrir un archivo.
- Juntar dos archivos.
- Hacer un browse a un archivo. (poder verlo y actualizarlo directamente ).
- Borrar un archivo.
- Indizar un archivo.
- Mezclar dos archivos.
- Reemplazar un campo en todos los registros de un archivo.
- Borrar una cierta cantidad de registros de un archivo.
- Borrar un registro.
- Reemplazar un registro ( reemplazar parte de él ).
- Insertar un nuevo registro.

Todas estas formas de actualización requieren atención especial al momento de diseñar las aplicaciones. Esto es, en el ambiente multi-usuario es de primordial importancia desarrollar

rutinas de protección contra colisiones de datos, haciendo bloqueos de archivos, bloqueos de registros y rutinas sofisticadas de atrape de errores.

Una aplicación uni-usuario no contempla la posibilidad de que dos o más usuarios deseen actualizar un mismo archivo simultáneamente. Veamos la siguiente secuencia de pasos que pueden suceder al ejecutarse una aplicación en ambiente multi-usuario y que causan conflictos en el manejo de los datos.-

1.- Un usuario ejecuta una aplicación desde un nodo A :

Lo que sucede es que en la memoria RAM del nodo A se carga una copia de la aplicación y comienza a correr.

2.- Otro usuario ejecuta la misma aplicación, pero desde un nodo B :

De nuevo, una copia de la aplicación es cargada, pero esta vez a la memoria RAM del nodo B.

3.- El primer usuario pide hacer un cambio a un registro de un archivo determinado :

Internamente la aplicación localiza el registro mencionado y carga a la memoria una copia de él.

4.- El segundo usuario pide hacer un cambio al mismo registro que está siendo visto por el primer usuario :

Internamente la aplicación ( corriendo en el nodo B ) busca el registro y carga una copia de él a la memoria

del nodo B.

5.- El primer usuario teclea algunas modificaciones al registro y lo manda grabar :

La aplicación toma la copia ya modificada y reemplaza lo que había en el disco, quedando así actualizado el archivo.

6.- El segundo usuario teclea algunas modificaciones al registro y lo manda grabar :

Aquí es donde, si la aplicación no está debidamente preparada tomará la copia del registro modificada por el segundo usuario y reemplazará el registro que está grabado en el disco, perdiendo inevitablemente la actualización hecha por el primer usuario.

Antes de continuar, es necesario aclarar porqué las búsquedas que se mencionan en los pasos anteriores llegan exactamente al mismo registro. Es simple, el área de almacenamiento de archivos ( de trabajo, de programas, de software de paquetes, etc. ) es común para todos los nodos de la red y se encuentra en la misma situación física del servidor de red.

La secuencia de pasos de arriba solo representa la operación de uno de los tipos de actualización que se presentaron al inicio de este tema.

Las rutinas de error, en un sistema multi-usuario, deben ser capaces de reconocer y manipular las posibles colisiones en los accesos de lectura y actualización de datos. Esto es, por ejemplo, que al darse estas colisiones la rutina de error debe enviar un mensaje al segundo usuario diciendo que su actualización no se realizara hasta que se consulte de nuevo el contenido del registro.

En el ambiente multi-usuario, en consecuencia de sus diferencias con el ambiente uni-usuario, se manejan conceptos que se relacionan con la compartición de archivos o más precisamente, con el control de transacciones. Tales conceptos son : uso exclusivo, colisiones, locking de archivos, locking de registros y el "deadly embrace".

#### Uso exclusivo.-

Cuando un usuario abre un archivo para usarlo en forma exclusiva, los otros usuarios deberán estar completamente prvenidos para que no puedan accesar este archivo; no se les permitirán accesos ni de lectura, ni de escritura. Algunas operaciones para archivos multi-usuario requieren, necesariamente, que dichos archivos sean abiertos en forma exclusiva, en general, se deberan tomar precauciones extremas para determinar la necesidad del uso exclusivo dado que esto provoca fuertes restrincciones sobre los demás usuarios.

### Colisiones.-

La situación que resulta cuando un usuario intenta acceder un registro o un archivo el cual esta siendo accedido por otro usuario. Cuando esto ocurre, datos dañados o datos corrompidos se pueden suscitar si más de un usuario intenta actualizar el registro. Hay maneras para protegerse contra este tipo de daños a los datos a través de bloqueos a los registros y a los archivos (locking's). En algunos lenguajes o paquetes ya existen rutinas que hacen locks a archivos y registros.

### Locking de un archivo.

Es el mecanismo mediante el cual un usuario evita que otros usuarios hagan accesos de escritura ( pero no de lectura, como sucede en el uso exclusivo ) a un archivo. El locking de archivos evita la corrupción de datos que puede ocurrir cuando un usuario modifica ( o abre ) uno o más registros en un archivo cuando otro usuario requiere necesariamente acceso de actualización del mismo archivo. Hay que estar concientes de que prohibir que otros usuarios puedan hacer actualizaciones a varios registros dentro de un archivo es generalmente ineficiente y no productivo.

### Locking de un registro.-

Es el mecanismo mediante el cual a un usuario se le previene que no puede hacer una escritura sobre un registro que esta siendo accedido por otro usuario. El locking de registros evita la corrupción de datos el cual puede ocurrir cuando un usuario modifica un registro mientras otro usuario está, simultáneamente



actualizando el mismo registro.

Deadly embrace ( abrazo mortal ).

Es la situación que ocurre cuando un usuario hace el locking de un registro y quiere acceder otro registro el cual ya un segundo usuario le hizo un locking; quien a su vez, está intentando acceder el registro bloqueado ( tiene un locking ) por el primer usuario. Este evento es muy raro, pero la probabilidad de que suceda aumenta cuando el tiempo que permanece bloqueado un registro aumenta.

Por último podemos describir el ambiente multi-usuario como un ambiente operativo basado en la combinación de hardware y software multi-usuario que hace posible que dos o más usuarios tengan acceso concurrentemente a los archivos y periféricos.

A continuación se tratará del modo de utilizar los anteriores conceptos para que se dé el ambiente multi-usuario.

### 3.3.- Metodologías típicas de diseño.

En este tema trataremos dos metodologías de diseño de aplicaciones multi-usuario. La primera de ellas, Locking Permanente, es más simple de desarrollar y nos proporciona todas las protecciones que necesitamos para evitar la corrupción de datos, pero es poco eficiente como se verá más adelante. La segunda metodología, aquí tratada y que llamaremos Locking Dinámico es más eficiente, pero el costo para desarrollarla es mayor.

Como ya se mencionó, las diferencias críticas que existen entre las aplicaciones uni-usuario y multi-usuario radican básicamente en la forma de acceder los archivos, razón por la cual detallaremos las metodologías, Locking Permanente y Locking Dinámico, basándonos en los diferentes tipos de actualización que se le pueden hacer a los archivos durante la ejecución de una aplicación en una microcomputadora. Un subconjunto de tales tipos de actualización no varía su forma de ser tratadas dentro de cualquiera de las metodologías, de modo que serán tratadas en un tema aparte.

Antes de entrar en detalle con las metodologías de diseño es necesario tratar las operaciones básicas del ambiente multi-usuario que son : Los Locking's o Bloqueos.

### 3.3.1.- Como hacer los llamados Locking's o Bloqueos.

Los locking's o bloqueos, son mecanismos a través de los cuales se previenen colisiones; esto es, que las colisiones que se den no causen daño a los datos.

Existen lenguajes que ya poseen rutinas o funciones especiales que efectúan locking's o bloqueos, sin embargo es necesario ver la manera de emular estas funciones.

Como ya se sabe, un archivo es un conjunto de registros y estos a su vez contienen la totalidad de la información para lo cual fué creado el archivo. Para adaptar la organización de los archivos que son accesados por un sistema determinado, y con ellos los registros que contienen, al ambiente multi-usuario; es necesario manejar una tabla (arreglo) que contenga la siguiente información sobre cada archivo :

- 1.- Nombre del archivo.
- 2.- Estatus del archivo.
- 3.- Usuarios usando el archivo.
- 4.- Registros bloqueados dentro del archivo.

Donde el campo de *estatus del archivo* podrá contener los siguientes valores.-

#### ESTATUS ARCHIVO

0	Archivo libre.
1	Archivo en uso.

- 2 Archivo bloqueado.
- 3 Archivo en uso exclusivo.

El campo de *usuarios usando el archivo* contendrá el número de usuarios que actualmente están usando el archivo.

Y el campo de *registros bloqueados dentro del archivo* contendrá la cantidad actual de registros con estatus de bloqueados que están contenidos dentro del archivo.

La adaptación de los registros consiste solo de agregar un campo de *estatus del registro* que será transparente para el usuario y contendrá los valores siguientes.-

ESTATUS REGISTRO

- 0 Registro libre.
- 1 Registro bloqueado.

El manejo de esta tabla y el campo de *estatus* agregado a los registros se detallará más adelante.

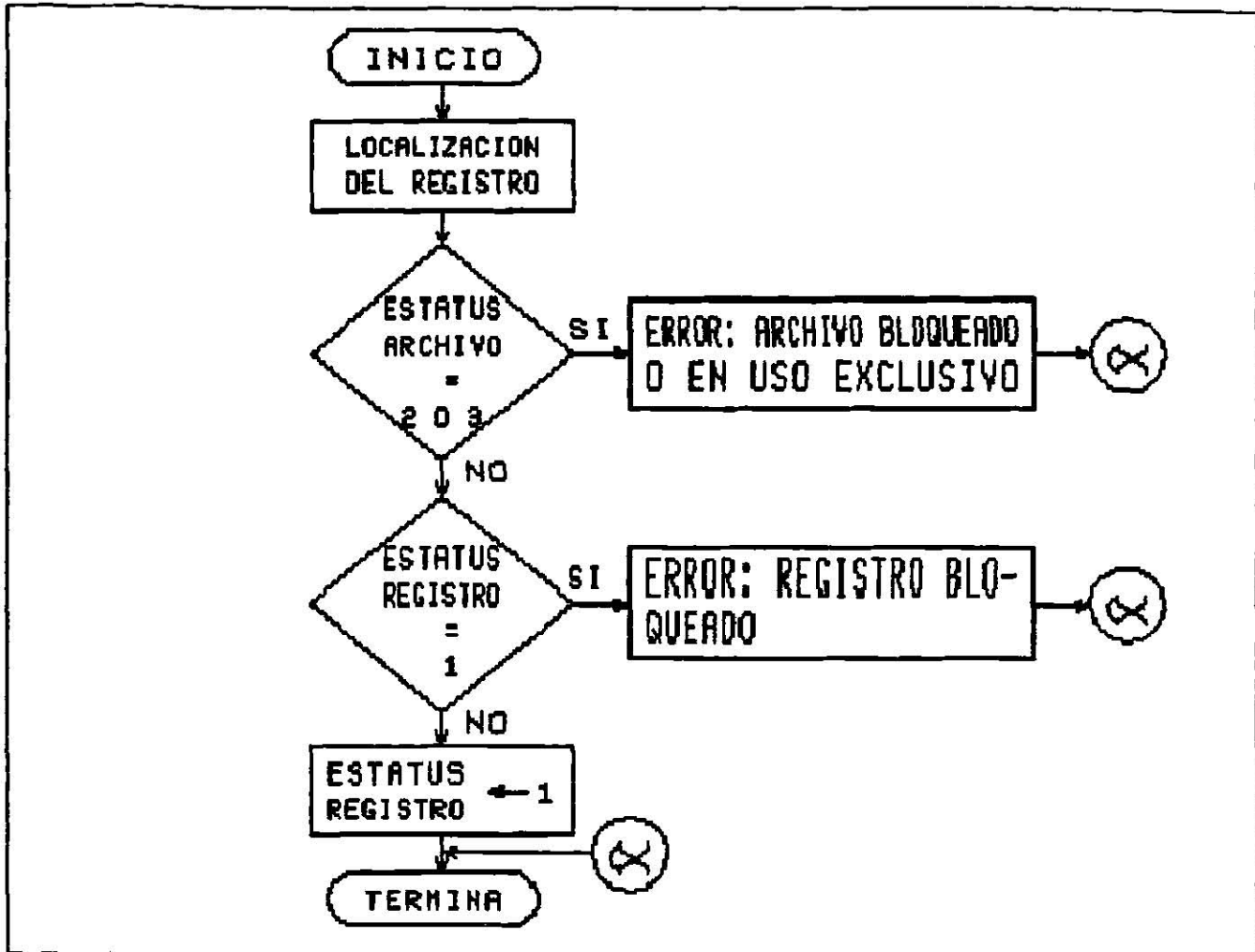
Se deberá tener especial atención al abrir un archivo, ya que desde ese momento el estatus del archivo cambiará, y al actualizar registros se validarán las posibilidades que hay de hacerlo, en base al *estatus del registro y del archivo* al que pertenece.

## Locking o Bloqueo y Desbloqueo de un Registro.-

Cuando se desea hacer el locking o bloqueo de un registro se requiere primero que el archivo haya sido abierto sin ningún problema y el campo de *estatus del archivo* (que está en nuestra tabla) ya se haya actualizado, si fué necesario; también es necesario que el registro ya haya sido localizado. Para llevar a cabo el bloqueo se realizarán los siguientes pasos.-

- 1.- Preguntar por el campo de *estatus del archivo* (tabla).
- 2.- Si el *estatus del archivo* es 2 ó 3 entonces no se puede bloquear el registro dado que el archivo ha sido bloqueado por otro usuario o bien está siendo usado en forma exclusiva, marcar error enviando su código.
- 3.- Si el *estatus del archivo* es 1 entonces
  - 3.1.- Preguntar por el *estatus del registro*.
  - 3.2.- Si el *estatus del registro* es 1 entonces el registro no puede ser bloqueado dado que otro usuario ya lo tiene bloqueado, marcar error enviando su código.
  - 3.3.- Si el *estatus del registro* es 0 entonces se reemplaza por 1, y termina sin error.

Se pueden tomar diferentes alternativas cuando un registro se encuentra ya bloqueado, pero si vemos la acción de bloqueo como solo una función, simplemente marcaremos el error y el programa que ejecutó la función tomará esas alternativas como se verá más adelante.



Para desbloquear un registro, se reemplazará el campo de *estatus del registro* por un 0, de esta forma el registro volverá a estar libre.

Locking o Bloqueo, Desbloqueo y Uso Exclusivo de un Archivo.-

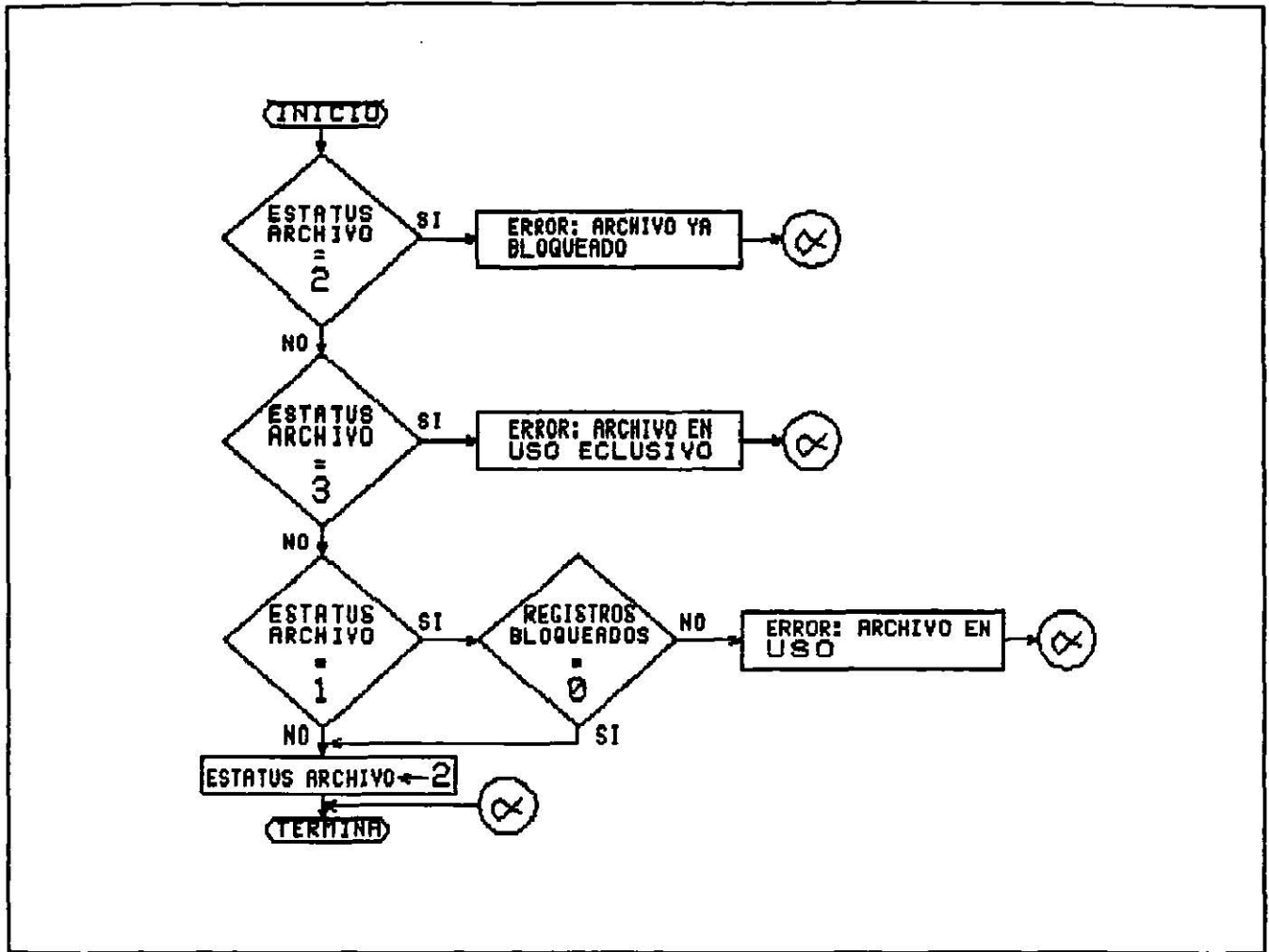
Cuando se hacen actualizaciones a todos los registros, o a la mayoría de ellos, que forman un archivo o bien cuando se hace

algún tipo de actualización del archivo, es conveniente bloquear el archivo completo.

Para bloquear un archivo es necesario realizar los siguientes pasos.-

- 1.- Preguntar por el *estatus de archivo* (en la tabla).
- 2.- Si el *estatus del archivo* es 2, entonces marcar error enviando su código, dado que no se puede bloquear porque otro usuario ya lo tiene bloqueado.
- 3.- Si el *estatus del archivo* es 3, entonces marcar error enviando su código, ya que no se puede bloquear el archivo porque otro usuario lo ha abierto de uso exclusivo.
- 4.- Si el *estatus del archivo* es 1, significa que otro u otros usuarios están usando el archivo, entonces.-
  - 4.1.- Preguntar por el campo de *registros bloqueados dentro del archivo* que está en la tabla.
  - 4.2.- Si *registros bloqueados dentro del archivo* es mayor que 0, entonces marcar error enviando su código ya que otro(s) usuario(s) tienen bloqueados registros que pertenecen al archivo.
  - 4.3.- Si el campo de *registros bloqueados dentro del archivo* es igual a 0, entonces asignar un 2 al campo de *estatus del archivo*, y termina sin error.
- 5.- Si el *estatus del archivo* es 0, entonces reemplazarlo por un 2 y termina sin error.





Para desbloquear un archivo, se reemplazará el campo de estatus del archivo por un 1 si el campo de usuarios usando el archivo es mayor que 0. En caso de que el campo de usuarios usando el archivo sea 0 entonces el estatus del archivo se reemplazará también por un 0.

Incluiremos aquí mismo la forma de hacer una apertura de archivo para uso exclusivo, esto es bloquear un archivo pero con la restricción de que no se podrán hacer accesos a él ni de lectura, ni de escritura. Los pasos que se realizan o condiciones

que se necesitan para abrir en forma exclusiva un archivo son:

- 1.- Preguntar por el campo de *estatus del archivo* en la tabla.
- 2.- Si es diferente de 0, entonces marcar error enviando código ya que el archivo está siendo usado, está bloqueado o abierto ya en forma exclusiva por otro usuario.
- 3.- Si es 0, entonces se le asignará un 3 y termina sin error.

Para cerrar un archivo que fué abierto de uso exclusivo se reemplazará el *estatus del archivo* por un 0 para dejarlo libre de nuevo.

Como se podrá notar estas rutinas de bloqueos (de registros, de archivos) y aperturas de archivos, protegen a los datos que contienen los archivos más sin embargo pueden darse colisiones en los accesos a nuestra tabla o al actualizar el campo de *estatus* que se les agregó a los registros por lo que corresponderá, en estos casos, entrar en acción a las rutinas de error, que en su oportunidad serán tratadas.

3.3.2.- Operaciones o tipos de Actualización Comunes en Ambas Metodologías.

La variante que existe entre el Locking Permanente y el Locking Dinámico es básicamente la forma de tratar un registro en particular de un archivo, y como existen operaciones que se realizan con archivos completos, estas se ejecutan de la misma

manera para cualquiera de ellas.

- Abrir un archivo.

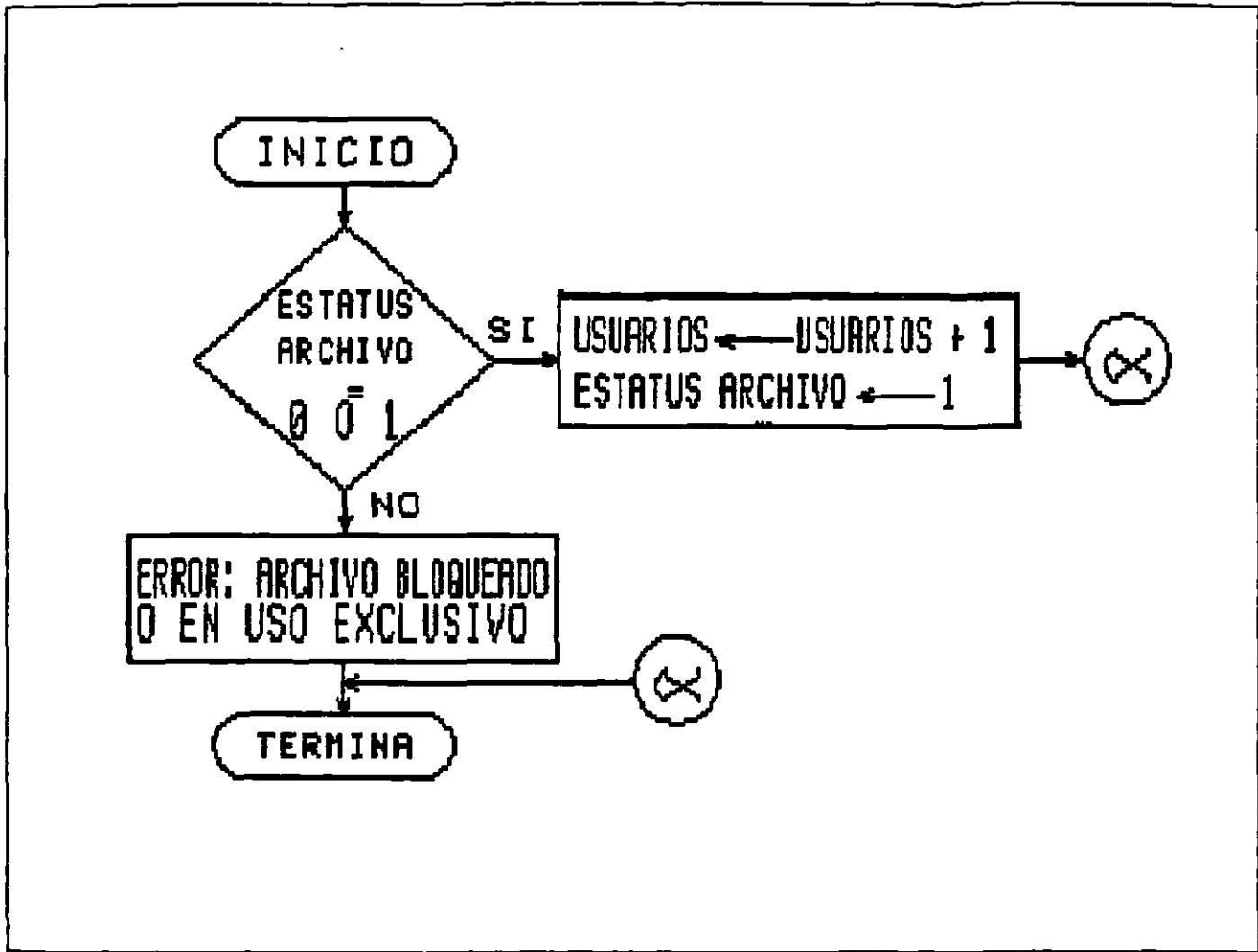
En las aplicaciones que requieren acceder archivos, es importante considerar que tipo de necesidad se tiene de ese archivo que se abrirá.

- Si es solo para consultarlo o bien para actualizar interactivamente algunos de sus registros, bastará validar que el *estatus del archivo* en la tabla de archivos sea 0 ó 1, y sumar uno al campo de *usuarios usando el archivo*.

En caso de que el *estatus del archivo* haya sido 0 entonces se reemplazará por un 1 y cuando se termine de usar será cerrado como se indica en la operación de cerrar archivo (detallada más adelante).

En el caso de que el *estatus* haya resultado 2 ó 3 marcar error enviando su código o se podrán hacer más intentos ya sea en base a un periodo de tiempo establecido o a una cantidad de intentos dada antes de marcar el error, el control del tiempo o los intentos lo puede llevar el programa mismo o la rutina de error.

- Si es para actualizar todos o una gran parte de sus registros del archivo en forma batch, entonces se ejecutará el bloqueo de archivo y una vez terminadas las actualizaciones se desbloqueará para dejarlo libre y que otros usuarios puedan usarlo.



- "Pegar" o juntar dos archivos.

Para ejecutar esta operación es necesario bloquear cada uno de los archivos. En este tipo de actualización de archivos, que son de uso no muy frecuente dentro de una aplicación, generalmente es necesario reorganizar los índices (si los hay) del archivo donde quedarán los registros de los dos primeros archivos; por lo que sería recomendable que este tercer archivo (o cualquiera de los dos primeros, en el caso de que en uno de ellos se deje el total de registros) se abra como de uso exclusivo. Y al finalizar la operación, desbloquear los archivos y

en su caso cerrar el archivo de forma adecuada.

- Browse de un archivo.

El browse de un archivo se refiere a actualizar directamente cualquier registro del archivo, sin validación alguna ni restricciones, y se suele usar en aquellas aplicaciones que son de mantenimiento de archivos, esto es que cuando se detecta algún registro invalido y nuestro sistema no contempla la forma de accesarlo, entonces podemos incluir, en nuestro sistema, una aplicación de este índole para proteger la integridad de datos en nuestros archivos. Este tipo de actualización requiere que el archivo sea abierto de uso exclusivo. Al finalizar el browse cerrarlo de uso exclusivo como se explicó en el tema 3.3.1.

- Borrar un archivo.

Obviamente para borrar un archivo es necesario abrirlo en forma exclusiva, pero también actualizar nuestra tabla de archivos dandolo de baja.

- Indexar un archivo.

Esto es, crear una llave de acceso a un archivo, para esto se requiere que el archivo sea bloqueado. Pero si la llave ya habia sido creada esto se verá como una reindexación y en este caso requerirá que el archivo sea abierto de uso exclusivo. Terminando la operación, el archivo se deberá desbloquear o en su defecto cerrarlo adecuadamente.

- Mezclar dos archivos.

De nuevo, como cuando se juntan dos archivos, se requiere bloquear ambos y abrir de uso exclusivo el archivo donde se dejarán lo registros mezclados, que puede ser cualquiera de los dos archivos originales y al terminar el mezclado desbloquearlos y cerrar el archivo que haya sido abierto de uso exclusivo.

- Reemplazar un campo en todo el archivo.

Como se indicó en la operación de abrir un archivo, para actualizar todos los registros de un archivo se requiere que este sea bloqueado y al finalizar la actualización desbloquearlo.

- Borrar una cierta cantidad de registros de un archivo.

Esta operación también queda dentro del caso anterior, de modo que el archivo al que se le desea borrar una cierta cantidad de registros deberá ser bloqueado y terminando la actualización se deberá desbloquear.

- Modificar la estructura de un archivo.

Esto es, cambiar la estructura de todos los registros que componen al archivo. Esto requiere que el archivo sea abierto en forma de uso exclusivo y al termino de la operación cerrarlo adecuadamente.

- Cerrar un archivo (que fué abierto para consulta normal).

Al cerrar un archivo, es necesario actualizar la tabla de archivos restandole un 1 al campo de *usuarios usando el archivo*,

y preguntando por el valor resultante; si el campo de *usuarios usando el archivo* fué 0, entonces asignar un 0 al *estatus del archivo*; si el campo de *usuarios usando el archivo* fué mayor que 0 no se deberá mover el campo de *estatus del archivo* (que deberá contener un 1), sino simplemente terminar la operación sin error.

### 3.3.3.- El Locking o Bloqueo Permanente.

El locking o bloqueo permanente es una forma sencilla de actualizar un registro y se requiere cuando se hacen cambios interactivos a registros. La forma de realizarlo se explica en los siguientes pasos.

- 1.- Se localiza el registro que se desea actualizar.
- 2.- Se bloquea el registro.
- 3.- El usuario tecleará sus cambios y mandará grabarlos.
- 4.- Una vez que los cambios han sido grabados el registro se desbloqueará.

Como se vé, es una forma muy segura de actualizar un registro, pero poco eficiente. Imaginemos un usuario que está haciendo actualizaciones a diferentes registros en forma interactiva e interrumpe su trabajo por alguna causa; muy probablemente, este usuario pudo haber dejado un registro con estatus de bloqueado, y el resultado de ello es que el resto de los usuarios no podrán bloquear ese mismo registro o en su defecto el archivo completo.



Existen variantes dentro del bloqueo permanente que minimizarían el riesgo de que un usuario deje bloqueado un registro. Una es que el programa conseda solo un cierto tiempo de bloqueo dependiendo posiblemente de la cantidad de información que contiene el registro.

Otra forma que se puede seguir para minimizar el riesgo de que un usuario deje bloqueado un registro se presenta en los siguientes pasos:

- 1.- Localizar el registro.
- 2.- El usuario teclea sus cambios al registro y lo manda grabar.
- 3.- Antes de grabarlo se bloquea el registro.
- 4.- Grabar los cambios.
- 5.- Desbloquear el registro.

Como se puede observar, el tiempo que permanecen bloqueados los registros es considerablemente menor, sin embargo existe la posibilidad de que mientras un usuario esté tecleando sus cambios (paso 2), un segundo usuario accese, modifique o haga otros cambios y los grabe (ejecute los 5 pasos), y al grabar sus cambios el primer usuario se perderán las actualizaciones del segundo usuario como ya se vió en el tema 3.2.

#### 3.3.4.- El Locking o Bloqueo Dinámico.

El bloqueo dinámico es una técnica de actualización de registros muy eficiente, pero existen aplicaciones multi-usuario

que debido a su naturaleza o forma de actualizar registros, no ameritan el empleo de esta técnica.

El bloqueo dinámico minimiza el tiempo de bloqueo de un registro y además evita la posible pérdida de movimientos o actualizaciones hechas por otros usuarios. Los pasos a seguir, en el bloqueo dinámico, para actualizar un registro son:

1.- Localizar el registro.

2.- Presentarlo en pantalla y mantener en memoria una copia del registro tal y como se encuentra en el disco.

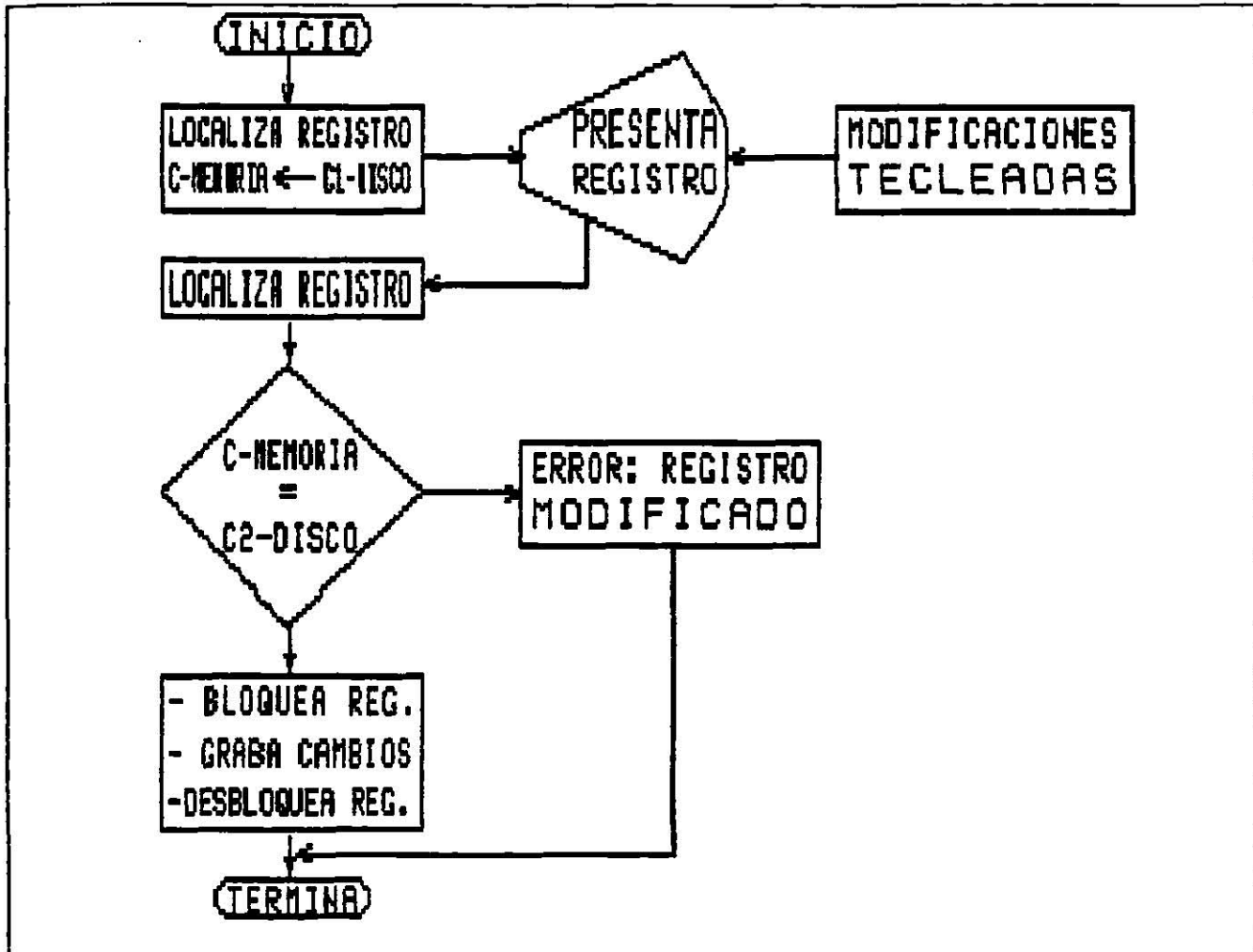
3.- El usuario teclea sus cambios y manda grabar el registro.

4.- Antes de grabar el registro, este se localiza y se lee de nuevo desde el disco.

5.- Se compara la nueva lectura con la copia que se mantuvo en memoria.

6.- Si la nueva lectura y la copia son iguales entonces bloquear el registro, grabar los cambios y desbloquearlo.

7.- Si la nueva lectura y la copia son diferentes entonces mandar un mensaje de error que diga que algún usuario ya modificó el mismo registro y por lo tanto deberá consultarlo de nuevo antes de modificarlo.



### 3.4 Rutinas de Error.

En los sistemas uni-usuario las rutinas de atrape y manejo de errores no son críticas, es decir pueden no llevarlas dado que las ocurrencias de errores inesperados son poco probables, salvo en aquellos casos en los que el usuario debe insertar un diskette para algún respaldo o recuperación, cuando la impresora debe estar en línea o con papel, etc.

En los sistemas multi-usuario, en cambio las rutinas de atrape y manejo de errores son primordiales. Como se vió en el tema anterior, muchas operaciones o tipos de actualización de uso común en los sistemas, requieren de la ejecución de funciones especiales de locking's o bloqueos y aperturas especiales de archivos, las cuales al ser llamadas, la ocurrencia de error es muy probable y no debemos estar expuestos a ellos.

En algunos lenguajes de programación se pueden hacer interrupciones al sistema operativo cuando ha sucedido un error, o también existen instrucciones que son capaces de monitorear la ejecución de las aplicaciones para que al cometerse un error, se ejecute automáticamente una rutina o programa que haga un maanejo eficiente del error sucedido; tales rutinas o programas son las que aquí llamamos rutinas de error.

Continuando con la emulación de nuestro ambiente multi-usuario que fué iniciada al abordar el tema de los locking's o bloqueos, podemos diseñar nuestras aplicaciones o programas, que forman el sistema, de manera que ellas mismas sean capaces de saber que al ejecutar una de las funciones especiales (bloqueos o aperturas) hubo un error, y entonces cedan el control o controlen la ejecución de la rutina de error. Esto se puede hacer si la función nos envia un código de error como parámetro. Algunos ejemplos de códigos de error y lo que significan, que podrían usarse en un sistema, se presentan en seguida:

CODIGO DE ERROR	Contenido
00	La operación o función se ejecutó sin error.
01	Archivo en uso exclusivo por otro usuario.
02	Archivo bloqueado por otro usuario.
03	Registro en uso.
04	Registro bloqueado.

Esta organización de código de errores puede ser usada para nuestra emulación del ambiente multi-usuario, sin embargo si el lenguaje que usamos para desarrollar nuestras aplicaciones tiene el poder o las herramientas necesarias para hacer una interface con el sistema operativo de la red, la organización de códigos de error dependería del sistema operativo o del lenguaje o paquete que estemos usando.

Ahora bien, si nuestra aplicación ejecuta una función de locking o bloqueo y esta nos regresa un código de error diferente de 00, esto es que hubo algún error en el intento de bloqueo, no resultaría eficiente si nuestra aplicación desistiera en ese momento de la operación que quiere hacer, por lo que es conveniente seguir haciendo intentos de bloqueo hasta conseguirlo ya que en cualquier momento ese registro o archivo lo pueden liberar. Una forma de conseguir hacer más intentos sería la siguiente:

1.- La aplicación ejecuta, en una de sus instrucciones, una

función de locking o bloqueo.

2.- Si la función regresa con código de error diferente de 00, entonces.-

2.1.- Se ejecuta la rutina de error; ya sea automáticamente (en el caso del software de desarrollo que si contempla el atrape de errores) o a través de alguna validación hecha por la aplicación.

2.2.- La rutina de error valida el tipo de error o bien determina que error se produjo en base al código de error que previamente le fué pasado como parámetro.

2.3.- Como la rutina detectará que fué un intento de bloqueo, esta posiblemente mandará un mensaje al usuario donde especifique que debe esperar un momento ya que otro usuario tiene bloqueado el registro o archivo según sea el caso, y regrese el control a la aplicación para que esta realice un nuevo intento. Puede también incluirse en la misma rutina de error la ejecución de los intentos.

3.- Cuando la aplicación logre hacer su bloqueo, esta continuará con las siguientes operaciones.

Como recordaremos, hay conceptos o situaciones que se dan en el ambiente multi-usuario, y que de hecho ya hemos detallado la mayoría de ellos, pero nos falta contemplar el llamado "deadly embrace" o abrazo mortal. Pues bien, con la técnica de intentos múltiples de bloqueos presentada arriba, la probabilidad de que suceda un abrazo mortal es muy alta por lo que sería recomendable

agregar una restricción al número de intentos que se hagan.

Como se ha visto a lo largo de este capítulo, existen muchas alternativas de diseño de aplicaciones que funcionen en un ambiente multi-usuario y que podrán ser tan sofisticadas como se desee o se requiera. En el siguiente capítulo presentaremos lo que se refiere ya a la programación , citando algunas rutinas que se necesitan agregar a las operaciones de actualización de datos y manejo de errores.

**4. PROGRAMANDO  
APLICACIONES MULTI-USUARIO EN  
FOXBASE+/LAN.**



En este capítulo presentamos algunas rutinas ideadas para adaptar los programas o aplicaciones hechos en FoxBase-LAN+ al ambiente multi-usuario y que de hecho deberán ser incluidas en los diseños de aplicaciones multi-usuario.

Antes que nada, es indispensable configurar en forma adecuada el FoxBase+/LAN indicando, entre otras cosas, que todos los accesos a los archivos se harán de forma compartida, es decir hay que incluir la siguiente instrucción :

```
SET EXCLUSIVE OFF
```

para que el estatus default de los archivos al ser abiertos sea libre y no exclusivo.

Un ejemplo de una configuración adecuada que se hizo en el inicio del menú principal de un sistema es el siguiente:

```
*
*   Menu de Entrada
*   =====
*
DO WHILE .T.
SET PATH TO X:.  && \mFOX21\UTIL
SET BELL OFF
SET DATE AMER
SET DELETE ON
SET EXCLUSIVE OFF Off
SET TALK OFF
SET SCORE OFF
SET STATUS OFF
SET SAFETY ON
ON ERROR Do Error With Error()
ON ESCAPE Return
SET COLOR TO W/b, BG/ ,
CLEAR ALL
```

Continuación...

```
CLEAR
Public AcInt, AcPan
Store 0 To AcInt
Store .F. To Accie
:
:
```

Para reemplazar el contenido de un registro se usa la instrucción REPLACE pero hay que bloquear el registro antes de hacerlo, una rutina que nos podría ayudar a ejecutar un REPLACE dando la oportunidad de hacer varios intentos sería :

```
Do While .T.
If RLock()
REPLACE ALTA WITH .T.
Unlock
Exit
EndIf
EndDo
```

donde ALTA es un campo de un registro de una base de datos. Podría agregarse un contador en el ciclo para no dar lugar a un "deadly embrace", esto es que se puede quedar ciclado el programa un buen tiempo esperando que el registro sea liberado, y si no se pudo ejecutar el REPLACE después de cierta cantidad de intentos mandar un mensaje de error.

En los sistemas es muy común tener opciones en las que se ofrecen reportes ordenados por diferentes campos y el FoxBase+ ofrece la capacidad de crear, en ese momento, el índice que se requiere para obtener el reporte deseado; pero si el nombre del archivo de índices es fijo en el sistema, muy probablemente

cuando dos usuarios pidan la opción de reporte pero con diferente orden, a uno de ellos (al primero que pidió el reporte) se le enviará un reporte que no concuerda con el orden que pidió. Por lo que se hace necesario recurrir a otras funciones que nos generen un nombre de archivo en forma aleatoria, para que cada usuario que pida un reporte, trabaje con archivos de índices distintos:

```
RgSys3 = SYS(3)
INDEX ON PROJ_NO TO &RgSys3
UNLOCK
SET SAFETY ON
SET INDEX TO &RgSys3                                && Selecciona Indice
                                                       Recien Formado
-
-
RgSys3 = RgSys3 + '.IDX'                            && Completar nombre de
                                                       archivo para borrarlo
```

Lo que hace la rutina anterior es generar un nombre de archivo aleatorio e indexa la base de datos en uso dejando los índices bajo este nombre de archivo, Ejecuta la presentación del reporte (no está incluida en la rutina anterior) y al final completa el nombre de archivo con su extensión para borrarlo, ya que si no lo hace la cantidad de archivos generados pronto llenarían el disco donde se encuentra alojado el sistema.

En algunos sistemas la llave de los registros que se manejan requiere de ser controlada y se crean rutina que generan la llave que debe seguir para dar de alta un nuevo registro. Por ejemplo, un número de artículo que se desea continuo y no se repita, para

poder usarlo como la llave de acceso de ese artículo se crea una rutina de generación del número siguiente disponible de artículo, pero no resulta sencillo si existen varios usuarios dando altas por lo que se podría controlar el proceso poniendo un candado de manera que impida a más de un usuario efectuar altas simultáneamente:

```

* Checa que no este puesto el candado
If Alta && Candado de altas a cartera de artículos
  Store .F. To AltPar
  @ 10,20 Say '
  @ 11,20 Say '
  @ 12,20 Say '
  @ 13,20 Say '
  @ 14,20 Say '
  Read
Else
  Store .T. To TAltPar
  Do While .T.
    If RLock()
      REPLACE ALTA WITH .T.
      UnLock
      Exit
    EndIf
  EndDo
  .
  .
  .

SELECT 4
If TAltPar
  Do While .T.
    If RLock()
      REPLACE ALTA WITH .F.
      UnLock
    Exit
  EndIf
EndDo
EndIf

```

Si en el sistema son muy frecuentes las altas este manejo Presentado no resulta eficiente porque solo un usuario podrá

hacer altas a la vez.

En las rutinas que hasta ahora hemos presentado los bloqueos se realizan casi automáticamente a través de una validación o pregunta, sin embargo estas funciones de bloqueo también se pueden aplicar directamente, pero están muy propensas a causar un error. Los errores causados por las funciones de bloqueo, aperturas de archivos para uso exclusivo y otros, se pueden controlar a través de una rutina de atrape de errores como la que sigue:

\* Ultima Actualización : 14:54:35 3/12/1989

\*

\* Rutina de Atrape de Error

\* Autor : Sergio M. Aguirre Taméz

\*

\*=====

\*

Parameters NError

Save Screen To AcPan            && Salvar pantalla en el momento del error

Set Color To w/b, Gb/n        && Color = Azul sobre Blanco  
Pos = 1                        && Posición de la barra sombreada de Tiempo

AcInt = AcInt + 1

Do Case

  Case AcTCG

    \* Valida que no se esté efectuando el cierre de mes ya que  
    \* los archivos son exclusivos del operador durante el  
    \* mismo.

    @ 10,28 Say '||

    @ 11,28 Say '||

    @ 12,28 Say '||

    @ 13,28 Say '||

    @ 14,28 Say '||

                  Efectuándose Cierre de Mes  
                  Será Regresado al Menú Principal  
                  Intente más Tarde. Dé Alguna Tecla

Store Space(1) To Regresa

Set Color To W/B, b/b,

@ 11,30 Get Regresa

Read

&& Hacer Un Get Invisible

Continuación...

```
Set Color To W/b, Gb/n,  
Restore Screen From AcPan  
Return To Master
```

Case NError = 108

```
Cursor = Sys(2002) && apaga el cursor  
* Se ha producido un error al abrir un archivo de uso  
* exclusivo o al tratar de bloquearlo.
```

```
If AcInt = 1
```

```
@ 10,28 Say '┌──────────────────────────┐'  
@ 11,28 Say '│ Archivo en Uso │'  
@ 12,28 Say '│ Por Favor Espere │'  
@ 13,28 Say '│ Tiempo : │'  
@ 14,28 Say '└──────────────────────────┘'
```

```
Set Color To W/n && Color Negro  
@ 13,39 Say '█' && Ventana Negra en la zona  
del reloj
```

```
Set Color To w/b, Gb/n && Restablecer el color  
EndIf
```

```
If (AcInt/10) >= Pos
```

```
Pos = Pos + 1  
@ 13,Pos+38 Say '█' && Avance del reloj cada 10  
intentos
```

```
EndIf
```

```
Cursor = Sys(2002,1)
```

Case NError = 109

```
* Se ha producido un error al tratar de  
* bloquear un registro.
```

```
Cursor = Sys(2002)
```

```
@ 10,28 Say '┌──────────────────────────┐'  
@ 11,28 Say '│ Registro en Uso │'  
@ 12,28 Say '│ Por Favor Espere │'  
@ 13,28 Say '│ Tiempo : │'  
@ 14,28 Say '└──────────────────────────┘'
```

```
Set Color To W/n && Color Negro  
@ 13,39 Say '█' && Ventana Negra en la zona del  
reloj
```

```
Set Color To w/b, Gb/n && Restablecer el color
```

```
If (AcInt/10) >= Pos
```

```
Pos = Pos + 1  
@ 13,Pos+39 Say '█' && Avance del reloj cada 10  
intentos
```

```
EndIf
```

```
Cursor = Sys(2002,1)
```

Continuación...

Case NError = 125

@ 10,23 Say ' |  
@ 11,23 Say ' |  
@ 12,23 Say ' |  
@ 13,23 Say ' |  
@ 14,23 Say ' |

```
Impresora no Lista.  
Enciendala y Prepare Papeleria  
<R>eintentar ó <A>bortar
```

Set Color To W/b, B/B  
Store Space(1) To Alterna  
Do While .Not. Alterna \$'AaRr'  
@ 13,55 Get Alterna Pict '!'  
Read

EndDo

Restore Screen From AcPan

If Alterna = 'R'  
Set Consol Off  
Set Devi To Prin  
Return

Else

If AcCie  
Close All  
Use RgCtrl Exclusive  
Repl CanGrl With .F.  
Use

EndIf

Return To Master

EndIf

Otherwise

\* no fué un error que se pueda controlar.

LonMen = 5 + Len(Message())  
Col = 40 - (LonMen + 3)/2  
@ 11,Col To 13,Col+LonMen Double  
@ 12,Col+1 Say SubStr(Str(Error()),8,3)+' '+Message()  
Wait ''

If AcCie  
Use RgCtrl exclusive  
Repl CanGrl With .F.  
Use

EndIf

Return To Master

EndCase

Restore Screen From AcPan

If AcInt >= 100

Return

Else

Retry

EndIf

\*eof

Los errores producidos por las funciones directas de bloqueo de registros y archivos caen en los casos segundo y tercero (errores 108 y 109), en estos casos la rutina de error presenta un "reloj" que avanza según el número de intentos y si se rebaza el número de intentos permitidos la aplicación deberá mandar el mensaje de error al usuario diciendo que intente más tarde.

Se podrían seguir presentando rutinas para más casos especiales, pero no cubriríamos más que un mínimo porcentaje de las ocurrencias posibles de sistemas particulares, así pues creemos haber cumplido el objetivo de este tema que es dar una idea general de algún desarrollo multi-usuario en un lenguaje dado.



## **5. CONCLUSIONES**

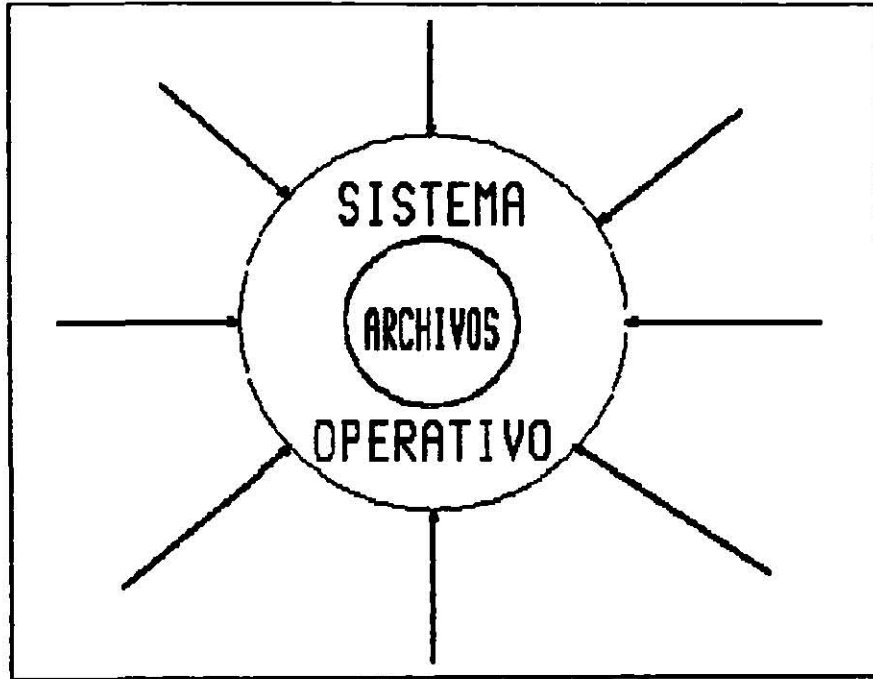
Las técnicas de actualización presentadas en este material están enfocadas a la protección de la información de los archivos, pero la elección de la técnica o combinación de técnicas a usar en un diseño estará sujeta a un cuidadoso análisis del sistema particular, en sus aspectos de formas de actualización, necesidades de consultas, necesidades de reorganización, corte periódicos, etc.

Resulta un esfuerzo adicional cada vez que se desarrolla un sistema en ambiente multi-usuario en una red local por lo que resultaría muy conveniente establecer una interface con los archivos que ya nos proporcione, las protecciones que hemos tratado a lo largo de este material.

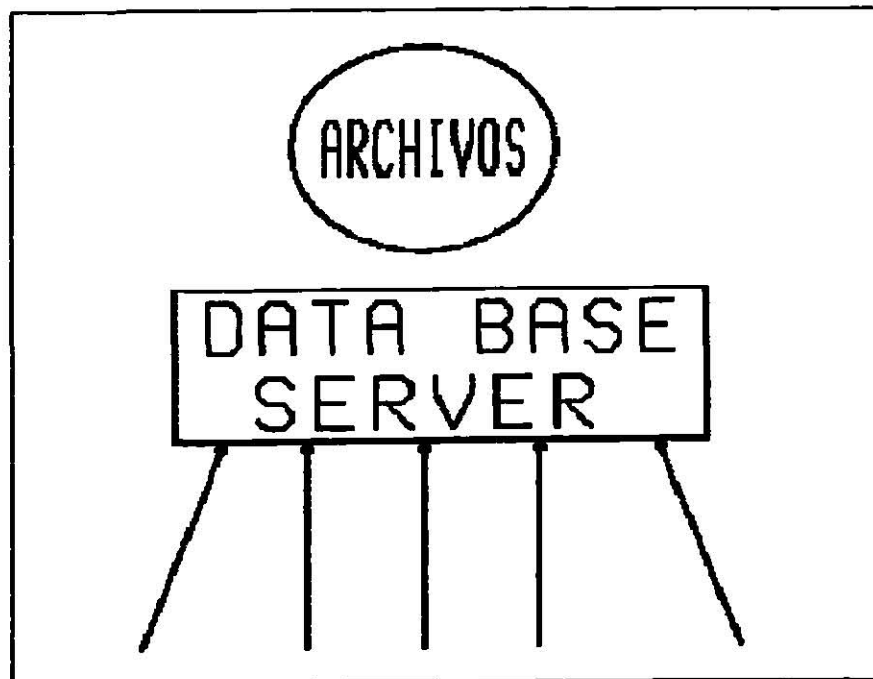
Se dice que la tendencia de las redes locales es establecer un mecanismo que administre los accesos a los archivos desde cualquier tipo de aplicación y desde cualquier tipo de máquina; de esta forma, al desarrollar aplicaciones multi-usuario, los programadores no tendrían que estar haciendo las veces de emular el ambiente multi-usuario. Tal vez lo único que se tendría que hacer sería pedir el recurso (registro o archivo) que se desea actualizar y ponerlo disponible, hacer el reemplazo y dejarlo libre, que también podría ser automático.

El esquema general de acceso a archivos actual en las redes

locales se puede ver como sigue:



El esquema de accesos a los que tienden las redes locales:



Donde las flechas representan solicitudes de acceso de actualización o consulta que hacen aplicaciones, en el primer esquema la interface la hace el sistema operativo de la red y en el segundo esquema la interface es a través de un software que bien podría ser un servidor de base de datos.

## Bibliografía

- Introducción a las Redes Locales (Folleto de IBM).
- Apuntes Sobre Redes Locales (Documento elaborado para un curso en HYLSA, sobre introducción a las Redes Locales).
- Revista Byte del mes de Junio (Tabla de S.O.'s en el Mercado)
- Manual de FoxBase+/LAN Ver 2.1.



