

**Escuela Superior Politécnica del Litoral
Facultad de Ingeniería Eléctrica**

TITULO:

Tutorial de Redes Lan's

Autores:

Johnny Avalos Herrera

Alex Miranda Andrade

Raúl Hernández

Informe final del Tópico Especial de Graduación
"REDES DE AREA LOCAL" presentada a la
Facultad de Ingeniería Eléctrica como requisito
parcial para la Obtención del Título de Ingeniero
en Electricidad Especialización Electrónica.

Director: Ing. Jaime Puente Pérez

Año Lectivo
1993 - 1994

Guayaquil

Ecuador

I N D I C E

INTRODUCCION	1
PARTE I : INSTALACION DE NOVELL NETWARE	
1.- Antecedentes	3
Generalidades	4
Preparar el servidor	4
Cargar NetWare 386	5
Cargar los disk drivers	6
Cargar el programa Install	7
Particionando el Servidor	7
Creando los volúmenes en el Servidor	8
Instalando los archivos del sistema operativo ..	8
Instalación del Print Server	9
PARTE II : PROGRAMA SOBRE LAN's	
1.- Programa sobre LAN's	13
Antecedentes	13
2. Programa Principal	14
Descripción y Desarrollo	14
Descripción de las principales funciones	14
3. Tutorial de LAN's	15
Propósito del Tutorial	15
Contenido del Tutorial	15
Estrategias para la creación del programa	16
Descripción de las principales fuciones	17
4. Sistema de Diseño de Redes Token Ring	21
Propósito del Programa	21
Estrategias para la creación del programa	21
Descripción de las principales funciones	22
5. Diagramas de Flujo	27
CONCLUSIONES Y RECOMENDACIONES	30
BIBLIOGRAFIA	31
APENDICES	

INTRODUCCION

El presente informe detalla los pormenores de los trabajos realizados a lo largo de la segunda parte del Tópico Especial de Graduación "Redes de Area Local".

El informe esta dividido en dos grandes capítulos, cada uno de los cuales corresponde a un trabajo realizado por nuestro grupo.

El primer capítulo corresponde a la instalación de Novell Netware versión 3.11 en la Red Token Ring del laboratorio de microcomputadores, se da un informe detallado de los pasos que se siguieron para realizar dicha instalación y alguna información adicional.

La Segunda parte corresponde al programa desarrollado por nuestro grupo que incluye dos puntos importantes:

- Un tutorial de LAN'S : donde se detalla los puntos más importantes que debe saber toda persona que trabaja con computadoras y planea introducirse al mundo de las Redes de Area Local. En esta parte se da una explicación corta pero concisa acompañada de gráficos de temas relacionados con el Hardware de las LAN'S y de los criterios de diseño que se deben seguir para generar una red acorde con las especificaciones impuestas por los respectivos Estándares de Redes de Area Local. La parte más compleja en lo que se relaciona con el diseño de redes corresponde a las redes Token Ring, debido a la cantidad de cálculos que hay que hacer, por esta razón se creó la segunda parte del programa.

- Un sistema que haga cálculos numéricos para diseñar una red Token Ring, tanto STP como UTP. El sistema tiene opciones para realizar los cálculos con cables tipo 1,2 y 9 para STP Y Redes TRN UTP de 4 Mbps y 16 Mbps, dándonos como resultado un mensaje, si los datos cumplen con las especificaciones de diseño, el programa lo indica, pero si se necesita usar repetidores, también lo dice dándonos la oportunidad de hacer los cálculos incluyendo los repetidores.

En general es un programa que nos brinda una interfase amigable con el usuario, ya que la parte del tutorial está realizada en modo gráfico, mostrándonos gráficos ilustrativos relacionados con cada tema incluido en el programa.

Los métodos seguidos para desarrollar este programa son mencionados posteriormente.

PARTE I

INSTALACION DE NOVELL NETWARE

1.- INSTALACION DE NETWARE

ANTCEDENTES

El Laboratorio de Microcomputadores de la Facultad de Ingeniería Eléctrica cuenta con una Red de Area Local con Topología Token Ring, cuyo servidor es un microcomputador IBM modelo PS2/80 con 10 estaciones de trabajo IBM PS/25.

El servidor cuenta con un disco duro ESDI de 110 MB y con 8 MB de RAM. Inicialmente tenía cargado Novell Netware 286.

Al iniciar la segunda parte de nuestro Tópico de Graduación esta red se encontraba inoperativa, ya que el diskette de booteo del servidor se había extraviado. Conversando con el Profesor sobre el tema y conociendo que en el disco duro se hallaban instalado solamente el sistema operativo de Netware y varios utilitarios muy comunes en nuestro medio (procesador de palabras, hoja electrónica y Turbo "C"), sin contener archivos de datos de importancia, se decidió realizar la reinstalación del Software de red, sin intentar salvar los programas instalados; es decir, no se eligió hacer un upgrade al software de red, sino que se reinstaló completamente el sistema operativo, cargando esta vez Novell Netware versión 3.11. El procedimiento de instalación del software de red es explicado posteriormente

Una vez instalado el software de red, se pensó en los utilitarios a cargar, como procesador de palabras se instaló el WordPerfect 5.1 versión para red, como compilador de Turbo "C" se instaló el Turbo "C" ++ versión 2.0 y como hoja electrónica de cálculo se instaló el QPRO 3.0.

Los usuarios a trabajar en la red no debían tener acceso a escribir en el disco de la red, debido a que se trata de un disco de pequeña capacidad, y evitar así que se llene de archivos innecesarios y pueda causar problemas en el funcionamiento de la red, además de salvaguardar el disco del servidor de posibles contaminaciones de virus y asignar la capacidad de disco solamente a utilitarios que puedan ser aprovechados por todos los usuarios de la red.

Un hecho muy importante de destacar es, que al correr el Turbo "C" 2.0, éste crea un archivo de swapeo sobre el disco, si un usuario no tiene atributos de escritura sobre el disco, no podrá cargarlo, viéndonos en la necesidad de asignar derechos de escritura a los usuarios creados, pero limitando éste derecho a un espacio muy reducido, que sea suficiente para poder ejecutar el Turbo "C", ya que no existe manera de poder redireccionar el archivo de swapeo a diskette. Algo parecido sucede con el WordPerfect, pero en él sí existe manera de direccionar la creación de éstos archivos a diskette, una vez que se cierran éstos

programas, los archivos son borrados del disco. de
pruebas se llegó a la conclusión de que un espacio El
suficientes para éstos efectos, resignando algo más n
disco a los usuarios creados (1 por estación más ayuda

GENERALIDADES

Este capítulo está dedicado a quienes van a instalar Netware por primera vez o para quienes van a hacer el upgrade de Netware 286 a Netware 386.

En algunos casos es beneficioso experimentar con el proceso de instalación antes de hacer la instalación final, ya que una preinstalación puede ayudarle a evitar conflictos de hardware y anomalías en la instalación.

Los pasos y procedimientos envueltos en la instalación de Netware 386 son listados aquí:

- 1.- Preparar el servidor y hardware adicional.
- 2.- Crear una partición booteable de DOS en el disco duro o crear un diskette booteable.
- 3.- Ejecutar el comando SERVER para comenzar Netware sobre el servidor.
- 4.- Cargar el driver para el disco requerido.
- 5.- Comenzar el programa INSTALL.
- 6.- Particionar el (los) disco(s) para NetWare.
- 7.- Crear discos imagen, si es necesario.
- 8.- Crear y montar volúmenes
- 9.- Instalar los archivos del sistema operativo y utilitarios.
- 10.- Crear los archivos de inicio.
- 11.- Cargar los módulos adicionales requeridos.

PREPARAR EL SERVIDOR

Instalar todo el equipo y periféricos en el servidor como se describe en el Tutorial de LAN's y luego arranque el servidor con DOS.

Los archivos de arranque del servidor pueden estar almacenados en 2 lugares:

- En un diskette booteable.
- En una pequeña partición de DOS sobre el disco duro del servidor.

Creando un diskette booteable de Netware

El diskette debe ser de alta densidad y que esté formateado con sistema, luego copie los archivos del diskette de Netware 386 Operating System-1 sobre el nuevo diskette. Debe crear un archivo AUTOEXEC.BAT sobre el diskette, incluyendo el comando SERVER en él. Debe tener creado un archivo de configuración del sistema (CONFIG.SYS) con el comando "FILES=40" sobre él.

Creando una partición de DOS

Crear una pequeña partición de DOS de 2 a 3 MB con el comando FDISK del DOS, lo cual da el espacio suficiente para almacenar los archivos de arranque de NetWare. El manual recomienda una partición de 1 MB, pero es recomendable crearla ligeramente más grande para guardar archivos que pueda necesitar en el futuro.

Formateando la partición de DOS

Una vez que la partición de DOS ha sido creada, usted puede formatearla con el comando FORMAT del DOS, use la opción "/s" para hacer que los archivos de arranque del DOS sean copiados sobre la partición, lo cual la hace booteable. Asegúrese de que la partición se encuentre activa.

Copiando los archivos de NetWare

Cuando el proceso de formateo haya sido completado sobre el drive "C", copie el contenido del diskette de NetWare 386 Operating System-1 sobre el drive "C". Copiando éstos archivos sobre el disco duro se agiliza el proceso de instalación y el proceso de booteo normal del servidor.

CARGAR NETWARE 386

La siguiente serie de pasos son realizados mientras el sistema operativo de NetWare está levantado y corriendo. Durante esta fase el sistema operativo no está listo para soportar una red hasta que usted instale los módulos cargables de NetWare, la partición de NetWare y cree los volúmenes. Tipee SERVER para comenzar NetWare, el sistema le preguntará:

.File Server Name: ingrese un único nombre para el servidor de 3 a 47 caracteres de longitud. No use espacios en blanco. Si usted está haciendo un upgrade, asegúrese de que el nombre sea diferente del servidor existente.

.IPX Internal network number: ingrese un único número que sea diferente de otros servidores de NetWare atachados a la red. El número puede ser de 1 a 8 caracteres de longitud en formato hexadecimal. Asegúrese de que no cree conflictos con otras redes o

direcciones de nodo sobre el cableado, incluyendo el número de proceso de DOS de un servidor corriendo NetWare 286 no dedicado. El número de IPX permite a futuras versiones de NetWare soportar un sistema tolerante a fallas en el cual un servidor entero es duplicado por un segundo servidor.

CARGAR LOS DISK DRIVERS

Después de ingresar el IPX internal number, el sistema operativo le muestra el colon prompt (:). Usted está listo para cargar los drivers que soporten los discos internos o subsistemas de discos externos.

Cuando Usted carga los disk drivers, los I/O porte e interrupciones por default son sugeridas. En algunos casos estas sugerencias son leídas de la tarjeta, pero usted puede cambiar la opción recomendada.

Si usted tiene más de un controlador de disco, cargue un driver para cada uno. Cuando 2 o más discos existen, los seteos para cada uno deben ser diferentes para evitar conflictos. Cuando un driver es cargado más de una vez, sólo una copia de éste código es colocado en memoria. Un pequeño bloque es usado para diferenciar entre cada tarjeta.

Los drivers disponibles que se encuentran en el diskette de NetWare Operating System-1 son:

- DCB.DSK : driver para discos DCB.
- ISADISK.DSK : driver para sistemas ISA
- PS2ESDI.DSK : driver para IBM PS/2 ESDI
- PS2MFM.DSK : driver para IBM PS/2 MFM
- PS2SCSI.DSK : driver para IBM PS/2 SCSI

Para otros sistemas los drivers deben ser obtenidos del fabricante.

Los siguientes parámetros deben ser especificados cuando instala los disk drivers. Una vez que son leídos de la tarjeta, los parámetros pueden ser aceptados directamente por usted. Sin embargo, las opciones pueden ser cargadas automáticamente al cargar el archivo AUTOEXEC.NCF, estas opciones son:

- DMA = # Reemplace # con el canal para DMA reservado para el driver.
- INT = # Reemplace # con la interrupción seteada en el hardware sobre la tarjeta.
- MEM = # Reemplace # con la dirección de memoria reservada para el driver.
- PORT= # Reemplace # con el I/O port reservado para el driver.
- SLOT= # Sobre máquinas MCA y EISA con bus master reemplace # con el número de slot en el que está instalada la

tarjeta.

CARGAR EL PROGRAMA INSTALL

Los siguientes pasos son realizados usando el programa **INSTALL**, el cual es manejado por menús y es usado para preparar las particiones de los discos, los volúmenes y cargar los archivos del sistema operativo y utilitarios.

Para comenzar tipee **LOAD INSTALL**.

PARTICIONANDO EL SERVIDOR.

El disco duro del servidor debe ser particionado para Netware, de la misma manera que usted instaló la partición del DOS. Escoja **DISK OPTIONS** del menú de instalación.

Las siguientes opciones son dadas para **DISK OPTIONS**:

-**FORMAT(Optional)**: Escoja **Format** si el disco duro no ha sido formateado en bajo nivel por el fabricante.

-**PARTITIONS TABLES**: Crea las particiones para Netware. Si más de un disco es instalado, seleccione el disco a particionar. Del menú de **Partitions Options** escoja "Create Netware Partition", **INSTALL** selecciona el siguiente cilindro evaluable después de la partición del DOS como punto de inicio por default y usa el resto del drive para la partición del Netware.

INSTALL automáticamente ubica el 2% del espacio de la partición como una área de redirección, si necesita cambiarlo escoja "Change Hot Fix".

-**SURFACE TEST(Optional)**: Hace un test de la superficie del disco.

-**MIRRORING**: Es una característica opcional que puede usar para asegurar la integridad de sus datos en el sistema de archivos de Netware. Un segundo drive espejo del contenido del 1er drive y entra en funcionamiento si el primer drive falla. El 2do drive debe tener una capacidad mayor o igual a la del 1er drive. El espacio extra del 2do drive no es usado.

Para establecer el "Mirroring" primero particione los dos discos como ya se indicó. Seleccione "Mirroring" del menú para "Disk Options", luego seleccione el 1er drive, luego presione **INS** cuando aparezca el menú de "Mirrored Netware Partitions" y seleccione el drive secundario de las particiones evaluables en el menú.

CREANDO LOS VOLUMENES EN EL SERVIDOR.

Los volúmenes mantienen colecciones de directorios y archivos. Los volúmenes en Netware pueden alcanzar varios disk drives, lo cual significa que usted puede crear volúmenes extremadamente grandes o incrementar el tamaño de un volumen existente adicionando discos duros. Para mejor funcionamiento una partición debe incluir justo un volumen, pero si un volumen abarca varios drives, Novell recomienda crear volúmenes imagen.

Para crear los volúmenes elija "Volumen Options", del menú de instalación, luego presione INS. Si más de un drive es evaluable, seleccione el drive que almacenará el volumen que está siendo creado.

El primer volumen creado le es dado el nombre de SYS por default y no deberá ser cambiado. Los volúmenes en Netware pueden ser seteados para usar bloques de tamaños de 4,8,16,32 ó 64KB. El bloque por default es de 4KB, lo cual significa que el área más pequeña del disco a ser usada para almacenar un archivo es de 4KB.

Los segmentos de volumen son usados para ubicar espacio del disco para múltiples volúmenes en un simple disco o para que un simple volumen alcance varios discos. Si usted en solo volumen, usted puede usar el segmento por default mostrado en el "tamaño inicial del segmento".

Antes de que usted continúe con la instalación de Netware, debe montar los volúmenes creados, ya que al crearlos el sistema operativo no los tiene abiertos para su uso. Use el comando MOUNT para esto. Regrese al prompt de la consola presionando Alt-Esc y luego tipee MOUNT-ALL presione Alt-Esc nuevamente para regresar al menú de instalación.

INSTALANDO LOS ARCHIVOS DEL SISTEMA OPERATIVO.

Usted está listo para copiar el sistema de Netware y archivos utilitarios al volumen SYS. Del menú de instalación escoja "System Option", luego elija "Copy System and Public Files", Install le pide que ingrese los diskettes correspondientes.

Un número de módulos de LAN drives son suministrados con Netware para soportar varias tarjetas de red populares. Asegúrese de copiar el LAN driver de su tarjeta al disco C o al diskette de arranque de la red. Usted debe ingresar # de interrupción a ser usado, memoria, I/O ports para evitar conflictos con otros dispositivos.

Cuando usted carga los LAN drivers, las interrupciones e I/O ports por default son sugeridos, en algunos casos estos seteos son

leídos directamente de la tarjeta. Si usted tiene más de una tarjeta de red, cargue un driver para cada, asegúrese de que los seteos no crean conflictos. Cuando los drivers son cargados más de una vez, solo una copia del código es colocada en memoria, un pequeño bloque es usado para diferenciar entre las tarjetas.

Los siguientes drivers son suministrados por Netware 386 y se encuentran en diskette System-1

TRXNET.LAN	Novell RX-Net, RX-NetII y RX-Net/2 driver.
3C503.LAN	3COM EtherlinkII-ASSY 2227 driver.
3C505.LAN	3COM Etherlink Plus-ASSY 2012 driver.
3C523.LAN	3COM Etherlink/MC driver.
NE2.LAN	Novell NE/2 Ethernet driver.
NE232.LAN	Novell NE/2-32 Bits Ethernet driver.
NE1000.LAN	Novell NE1000 Ethernet driver.
NE2000.LAN	Novell NE2000 Ethernet driver.
NE3200.LAN	Novell NE3200 Ethernet driver.
TOKEN.LAN	Driver para adaptadores IBM Token Ring.

En nuestro caso se cargó el driver TOKEN.LAN de la siguiente manera:

```
:LOAD TOKEN.LAN
```

Luego debe atarse a un protocolo, para nuestro caso es el protocolo IPX.

```
:BIND IPX TO TOKEN.
```

Si usted está instalando más de una tarjeta a la vez, debe decirle al sistema, que tarjeta liga al protocolo de comunicaciones. BIND le pregunta por un # de red usado para el driver, el cual debe ser ingresado en formato hexagesimal y puede tener de 1 a 8 dígitos de longitud.

Los archivos AUTOEXEC.NCF y STARTUP.NCF son preparados por INSTALL. Seleccione las opciones para crear estos archivos en el menú de "System Options". Las opciones seteadas en los pasos anteriores son automáticamente insertadas en estos archivos antes de grabarlos si es necesario.

Cuando usted crea el archivo STARTUP.NCF o lo edita, el sistema le pregunta por una letra del drive donde irá, si tiene partición de DOS tipee C o si bootea de diskette tipee A. Para completar el proceso de instalación usted puede cargar desde la consola el monitor y el printserver e incluirlos en el archivo STARTUP.NCF.

INSTALACION DEL PRINTSERVER

Los siguientes pasos muestran como instalar un PrintServer que

puede servir hasta 8 File Servers.

Cargar el Módulo PSERVER.NLM

PSERVER.NLM fue copiado en SYS:SYSTEM durante la instalación.

1.- Para cargar el Print Server sobre un File Server con Netware v3.11, tipee en la consola del servidor:

```
LOAD PSERVER printserver
```

2.- Modifique el archivo AUTOEXEC.NCF, inserte la línea anterior en él. El File Server automáticamente cargará el Print Server cada vez que arranca si modifica el AUTOEXEC.NCF.

Crear las colas de impresión

- 1.- Ingrese a la red como Supervisor.
- 2.- Entre PCONSOLE y seleccione "Print Queue Information".
- 3.- Presione <Insert> y tipee el nombre de la cola de impresión.
- 4.- Asigne operadores de la cola de impresión y usuarios (opcional). El sistema automáticamente asigna al SUPERVISOR como operador y a EVERYONE como usuario de la cola.
- 5.- Presione <Escape> para regresar al menú de "Print Queue Information".
- 6.- Presione <Escape> si desea retornar al menú de "Available Options".

Crear el Print Server

- 1.- Seleccione "Print Server Information" del menú "Available Options".
- 2.- Presione <Insert> y tipee el nombre del Print Server, luego presione <Enter>.
- 3.- Seleccione el Print Server apropiado. Aparecerá el menú "Print Server Information".
- 4.- Asigne el password para el Print Server y operadores y usuarios del mismo (opcional). El sistema automáticamente asigna al SUPERVISOR como operador y a EVERYONE como usuarios.

Definir las Impresoras

- 1.- Seleccione "Print Server Configuration" del menú "Print Server Information".
- 2.- Seleccione "Printer Configuration".
- 3.- Seleccione el número de la Impresora que Usted quiere definir. Luego complete el "Printer # Configuration" :

Nombre: ingrese un nombre para la impresora.

Tipo : seleccione el tipo apropiado de impresora de acuerdo a las siguientes opciones:

- Paralela, LPT o Serial, COM: Use esta opción si usted conecta la impresora directamente al servidor o a una estación dedicada como Print Server. Especifique el puerto LPT o COM que usará.
- Paralela Remota, LPT o Serial Remota, COM: use esta opción si usted está conectando la impresora a una estación de la red.
- Remote Other/Unknown: esta opción suministra información acerca de la impresora cuando usted corra RPRINTER.
- Defined Elsewhere: esta opción se usá cuando actualiza las impresoras para file servers adicionales.

- 4.- Presione <Escape> y grabe los cambios.
- 5.- Presione <Escape> y regresa al "Print Server Configuration Menu".

Asigne Colas de Impresión a las Impresoras

Cuando usted asigna colas de impresión a las impresoras ya creadas, usted autoriza al Print Server a que sirva esas colas.

- 1.- Seleccione "Queues Serviced by Printer" del "Print Server Configuration Menu".
 - 2.- Seleccione la impresora adecuada.
 - 3.- Presione <Insert>. Aparecerá la ventana "Available Queues."
 - 4.- Seleccione la cola que servirá la impresora e ingrese un número de 1 a 10 que indica el nivel de prioridad de la cola, presione <ENTER>.
- Repita los pasos 3 y 4 para asignar colas adicionales a esta impresora.
- 5.- Presione <Escape> para salir de PCONSOLE.

PARTE II

PROGRAMA SOBRE LAN's

1.- PROGRAMA SOBRE LAN'S

ANTECEDENTES

El programa desarrollado tiene dos objetivos :

El primero es brindar a los usuarios un Tutorial de Redes de Area Local, donde se indique los pormenores que debe saber toda persona que desea ingresar al mundo de las redes, donde se le indique las opciones que tiene para preparar el hardware de la red, así como también las especificaciones para las diferentes topologías que existen y las maneras que existen de interconectar las Redes de Area Local.

El segundo objetivo es crear un Sistema de Diseño para Redes Token Ring, ya que éstas son las que llevan un cálculo matemático más complejo que las otras topologías existentes, debido a lo cual existen tablas pre-establecidas para determinar el alcance que puede tener la red y en base a las cuales se creó ésta parte del programa.

Como podemos observar, el programa está dividido en dos subprogramas, razón por la cual se crearon tres módulos independientes pero enlazados entre sí.

El primer módulo equivale a un programa principal, le brinda al usuario una pantalla de presentación del programa y que le a elegir si quiere ver la parte de Tutorial o la parte de Diseño, razón por la cual se la llama PROGRAMA PRINCIPAL. Es el encargado de invocar a los otros módulos según la opción que se elija, y es donde retorna la ejecución del programa cuando el usuario sale de la opción en la que se encuentra.

El segundo módulo se relaciona con la parte del Tutorial de Redes, en él se encuentran todas las funciones que realizan las tareas necesarias para que se puedan ejecutar el Tutorial, razón por la cual se le llamó TUTORIAL DE LAN'S.

El tercer módulo tiene que ver con la parte de Diseño de Redes Token Ring, en él se encuentran todas las funciones necesarias para ejecutar el Sistema de Diseño de Redes Token Ring, razón por la cual se le llamó SISTEMA DE DISEÑO TOKEN RING.

3.- TUTORIAL DE NETWARE

PROPOSITO DEL TUTORIAL

El tutorial de Redes de Area de Local (LAN'S) tiene como propósito introducir a las personas que usan este tutorial en el mundo de las redes. El texto incluido dentro del tutorial corresponde al material visto a lo largo de la segunda parte del tópico además de información adicional recopilada por nuestro grupo de trabajo.

El propósito fundamental es instalarlo en el laboratorio de microcomputadores para que puedan consultarlo todos los estudiantes que deseen y tengan conocimientos generales de lo que son las redes y de los materiales y equipos que se necesitan para instalarlas, además de que por encontrarse en español será fácilmente entendible, a diferencia de los libros, manuales u otros textos de ayuda que se encuentran en Inglés.

Una vez finalizado el estudio del tutorial, el estudiante estará en capacidad de discernir sobre que tipo de red es más conveniente para una aplicación dada, en base a necesidades, número de estaciones, costo, equipo disponible, etc. Esperamos que el tutorial sea de mucha utilidad para personas que se dediquen o tengan el deseo de trabajar en una red de computadoras.

CONTENIDO DEL TUTORIAL

El tutorial nos brinda temas tanto del Hardware requerido para la instalación de una red, así como de las reglas que se deben seguir para generar una red, en fin los tópicos que nos brinda el tutorial son mencionados aquí.

- Por que establecer una LAN.
- Evaluando el Hardware del Server:
 - Procesador 386
 - Procesador 486
 - Bus ISA
 - Bus EISA
 - Bus MCA
 - Consideraciones de memoria
 - El disco duro del server
 - Interfases de Disco
 - La tarjeta de red.
- Interconectando Redes :
 - Repetidoras
 - Bridges
 - Routers
 - Backbones
 - Gateways.
- Topologías de Red
- Ethernet y la IEEE 802.3

- 10 Base - 5 - Especificaciones
 - Componentes
- 10 Base - 2 - Especificaciones
 - Componentes
- IEEE 802.5 Token Ring
- Token Ring UTP: - Especificaciones
 - Componentes
- Token Ring STP: - Especificaciones
 - Componentes
- Reglas de diseño Token Ring
- Requerimientos de transmisión
- Multimedia

ESTRATEGIAS PARA LA CREACION DEL TUTORIAL

Como ya se dijo anteriormente, el tutorial esta desarrollado en modo gráfico, por lo cual se necesitó de un utilitario para realizar los gráficos, y a la vez conocer la manera de mostrarlos desde el "C". Se estudió la manera de hacer esto y la decision recayó sobre un utilitario del Windows llamado "PAINTBRUSH".

Por medio de este utilitario se desarrollaron los gráficos correspondientes a casa tópico del tutorial, los archivos gráficos son de extension BMP, y su graficación se realiza pixel por pixel, razón por la cual la graficación se realiza un poco lenta en máquinas antiguas, pero en máquinas actuales, como por ejemplo un 486 a 33MHZ la graficación es muy rápida.

El formato de los gráficos se desarrolló para una pantalla VGA, la cual contiene 640 x 480 pixeles, además se realizó el programa para que acepte 16 colores, por lo cual se necesita de un driver para operar en modo grafico, el cual es el archivo EGAVGA.BGI.

Como todos sabemos, un tutorial consta de texto que explica ciertos aspectos de un tema, entonces necesitamos crear archivos de texto, de extensión HLP en donde se encuentre la siguiente información:

- La lista de los tópicos que contiene el tutorial, así como el nombre de los archivos gráficos asociada cada tópico, se encuentra en el archivo DESC.HLP
- Un archivo HELP.HLP, donde esta la información de ayuda.

Estos archivos de texto pueden crearse tanto en el editor del DOS como en el editor del turbo "C", usamos los caracteres extendidos para señalar las tildes, la ñ y otros símbolos especiales que se encuentran en el tutorial.

Debido a que estos programas se vuelven un poco largos, se trabajó con la modalidad de proyecto, el cual guarda todos los archivos .C (Fuentes) y genera un solo ejecutable. Los archivos fuentes que corresponden a la parte del tutorial son HELP.C y START.C.

DESCRIPCION DE LAS PRINCIPALES FUNCIONES

La función main1 ()

Es la función principal de la parte del tutorial, es en esta función donde se llaman a los demás bloques o funciones para realizar tareas específicas.

La función present1 ()

Esta función genera la pantalla de presentación del tutorial, así como también se crean los cuadros para la descripción del tópico, para el menú de tópicos disponibles y para el gráfico de ayuda. Observando la función present1 () podemos ver como nos movemos en la pantalla, lo cual se hace en píxeles y no en filas y columnas, además podemos ver como se manejan los 16 colores y como se manejan además los tipos de fants para la escritura que será mostrada.

La función opengraph ()

Por medio de esta función se inicia la programación gráfica, en esta función se le indica al turbo "C" que se abre una sesión gráfica para un sistema de video VGA.

La función sombras1 ()

Genera una sombra para la ventana del menú de tópicos disponibles y de esta manera indicamos que estamos en esa ventana.

La función sombras2 ()

Genera una sombra para la ventana de la descripción del tópico y de esta manera indicamos que estamos en esa ventana.

La función limp_sombras1 ()

Limpia la sombra generada por sombras1 ()

La función limp_sombras2 ()

Limpia la sombra generada por sombras2 (). Estas 4 funciones trabajan una con la ayuda de la otra, para generar el efecto del cambio de ventana.

La función loadbmp (char fbmp[18])

Esta función toma el archivo gráfico que le pasan como argumento, lo abre para lectura binaria y luego lo muestra en la ventana para el gráfico de ayuda, además realiza el algoritmo adecuado para graficar los archivos BMP del Paint-Brush y reubica los colores, ya que no son iguales los números de los colores para el Paint-brush y para el turbo "C".

La función loadbmp640 (char fbmp[18])

Esta función hace exactamente lo mismo que la anterior, con la excepción de que el gráfico que muestra no es el de la ventana de ayuda, sino que es un gráfico de pantalla completa, es decir de 640 x 480 pixeles.

La función menuh ()

La función menuh genera el menú de tópicos disponibles, se utiliza un algoritmo sencillo para generar una lista de tópicos que pueda ser desplazada por las flechas de desplazamiento hacia arriba y hacia abajo, moviéndose de esta manera desde el primer tópico hasta el último tópico y viceversa. Además es la función que nos da la interfase con el usuario cuando nos encontramos en el menú de tópicos disponibles.

La función limp_menu ()

Es una función que ayuda a menuh () a poder generar el menú de tópicos disponibles, lo que hace es limpiar la ventana antes de volver o mostrar el menú luego de algún cambio por desplazamiento de los tópicos.

La función print_desc (char tema [70])

Esta función lo que hace es imprimir en la pantalla, la descripción de un tópico elegido, claro está que muestra solamente 25 líneas, pero se muestra el resto de la descripción presionando la tecla adecuada, moviéndose no como en el menú de tópicos disponibles línea por línea, sino que nos movemos página por página, tanto hacia arriba como hacia abajo, para poder observar toda la descripción.

La función refresh_desc ()

Esta es la función encargada de imprimir en la pantalla la descripción del tópico cuando nos desplazamos de página, tanto hacia arriba como hacia abajo.

La función ventana ()

Esta es la función que nos da la interfase con el usuario

cuando nos ubicamos en la ventana de descripción del tópico, en esta función es invocada la función `refresh_desc ()` para cambiar de página.

La función `loadtopics ()`

Esta función toma el archivo `TOPICS-HLP` y lo abre para lectura, el archivo mencionado contiene información sobre:

- Tópicos disponibles
- Gráficos pequeños asociados con este tópico
- Gráfico grande asociado con este tópico.

Esta función carga el contenido del archivo en los siguientes arreglos.

- `Topics [100][40]`; lo cual nos indica que el programa puede disponer de hasta 100 tópicos; actualmente están en uso solamente 35.
- `Topbmp [100][18]`; guardará los nombres de los archivos gráficos pequeños relacionados con cada tópico.
- `Top[100][18]`; guardará los nombres de los archivos gráficos grandes (de pantalla completa) relacionados con cada tópico.

La función `limp_desc ()`

Toma el arreglo `desc [200][90]` y lo encera asignándoles caracteres nulos `'\0'` al primer elemento de cada fila del arreglo. Este arreglo se carga con la descripción del tópico elegido, por lo cual podemos ver que cada descripción de un tópico podrá ser hasta 200 líneas.

La función `limp_top ()`

Toma los arreglos `topics`, `topbmp`, `top` y `fhlp` y les asigna caracteres nulos a los primeros elementos de cada fila del arreglo, de esta manera quedan inicializados los arreglos.

La función `load_desc (pos)`

Esta función toma al archivo `desc.hlp` y lo abre para lectura, luego busca la descripción correspondiente al tópico elegido, lo cual le es dicho por medio de la variable `pos`.

El archivo `DESC.HLP` contiene las descripciones de todos los tópicos disponibles, los cuales son diferenciados por medio del carácter extendido `alt + 219 (■)` ubicando un carácter tanto al inicio como al final de cada descripción, podemos desarrollar un algoritmo sencillo para la diferenciación de una descripción de otra.

La función `menu_top (int cur)`

Esta función genera un menú de opciones, desde donde se puede

seleccionar entre: - Salir del programa
- Ver gráfico grande
- Ver la ayuda

La función help ()

Cuando presiona la tecla F1 se activa el menu de ayuda, donde nos indica la función de ciertas teclas que son especiales en el programa. Esto puede hacerlo en conjunto con las funciones load_help (), print_help ()

La función load_help()

Toma el archivo HELP.HLP y lo abre para lectura, carga su contenido en el arreglo fhhelp [x][z] y cierra el archivo.

La función print_help ()

Toma el contenido del arreglo fhhelp [x][z] y lo imprime en la pantalla.

4.- SISTEMA DE DISEÑO DE REDES TOKEN RING

PROPOSITO DEL PROGRAMA

Esta parte del programa está orientada a realizar un sistema que ayude a realizar los cálculos para las distancias que son permitidas para las redes Token Ring, tomando en cuenta parámetros importantes como son: número de MAU'S, máximas distancias del cable de lobe, distancias entre wiring closets, etc., los cuales son los que producen atenuación dentro de las Redes Token Ring.

Para la parte STP las atenuaciones producidas por el número de Wiring Closets y el número de MAU'S se encuentran dentro de tablas ya establecidas, en base a las cuales hemos basado nuestro criterio de diseño. Las tablas se encuentran cargadas en el programa de tal manera que el programa primero calcula la distancia máxima de transmisión sin regeneración (DMTR) del caso tratado y luego lo compara con el de la tabla, para decir luego si la red tratada está dentro de las especificaciones permitidas para Redes Token Ring.

En la parte de redes UTP el criterio de diseño es distinto, debido a que las redes UTP son redes pequeñas, las atenuaciones se basan en el número de bloques de conexión, número de paneles de conexión, número de MAU'S, número de salidas en pared, máxima longitud del cable de lobe, etc, pero los valores permitidos no se encuentran en tablas, sino que son fijos teniendo las distancias drive para TRN 4 Mbps : 160 mts y para TRN 16 Mbps : 100 mts. Cada elemento posee una longitud equivalente en mts por la atenuación que causa, de esta manera se calcula las distancias máximas para redes UTP y se las compara con las distancias drive para tomar luego una decisión.

En sí, el programa es de mucha ayuda, ya que nos indica cuando necesitamos usar repetidores, y nos permite además realizar los cálculos para segmentos de anillo que contienen repetidores y luego nos informa si nuestros cálculos son permitidos o no.

ESTRATEGIAS PARA REALIZAR EL PROGRAMA DE DISEÑO

A diferencia de la parte del tutorial, esta parte del programa no está desarrollada en ambiente gráfico, está desarrollada simplemente en ambiente DOS, debido a lo cual se necesitó crear varias funciones para desarrollar las diferentes pantallas de presentación.

Esta parte del programa se desarrolló en 4 archivos fuentes, que son:

- TABLA.C
- TOKEN.C
- STP.C
- UTP.C

El archivo TABLA.C contiene almacenado en arreglos los valores de las distancias máximas permitidas por cada caso de Redes STP tanto las que usan cable tipo 1,2 ó 9., cuando se necesita conocer un valor, el programa llama esta función, la cual le retorna el valor que el programa necesita.

En el archivo TOKEN.C se encuentra la función principal sobre la cual gira esta parte del programa, se encuentran además las funciones utilizadas para crear las pantallas de presentación y de mensajes que se utilizan a lo largo del programa.

El archivo STP.C contiene todas las funciones necesarias para realizar cálculos para el diseño de redes STP, así como también todas las funciones necesarias para el ingreso de los datos de la red a ser tratada.

El archivo UTP.C contiene todas las funciones necesarias para realizar los cálculos para el diseño de redes UTP, así como también todas las funciones necesarias para el ingreso de los datos de la red a ser tratada.

DESCRIPCION DE LAS PRINCIPALES FUNCIONES

La función main2 ()

En esta función se comienza a crear la pantalla de presentación, la cual invoca a la función Menu-Stp () para terminarla.

La función Menu-Stp ()

Esta función termina de crear la pantalla de presentación, mostrándonos el menú principal de opciones, teniendo tres posibilidades:

- Redes tipo STP
- Redes tipo UTP
- Salir

Al mismo tiempo espera por la opción que uno elija, si elije Redes STP invoca a la función STP (), si elije Redes UTP invoca a la función UTP () y si elije salir, le retorna un valor a la función main2 () que le obliga a terminar la ejecución de esta parte del programa.

La función MensajeOK ()

Como su propio nombre lo indica nos muestra un mensaje, que para nuestro caso se lo usa para indicar que se necesita usar repetidoras dentro del diseño de las redes, así como también para otros mensajes adicionales, una vez que uno termine de leer el mensaje presiona ENTER y el programa sigue su ejecución. Para su total ejecución utiliza la ayuda de la función OK.

La función Mensaje OKCancel ()

Muestra un mensaje, pero nos da dos opciones:

- Una es seguir con la ejecución del programa.
- La otra opción es cancelar la ejecución del programa. Para hacer esto utiliza la ayuda de las funciones OK y Cancel

Las funciones que describimos a continuación se relacionan con la parte del diseño de Redes Token Ring STP, se hallan en el archivo STP.C.

La función STP ()

Esta es la función principal de la parte de diseño de redes Token Ring STP, es de aquí desde donde se llaman a las funciones necesarias para pedir el ingreso de los datos, donde se crea el menú de redes STP y donde se realizan los cálculos de las distancias para obtener la distancia máxima de transmisión sin regeneración, realizar la comparación con los valores permitidos para DMTR que se hallan tabulados en el archivo TABLA.C y tomar una decisión si la red tiene los parámetros adecuados para operar en forma eficiente o si se necesita del uso de repetidores, y de ser así llamar a la función encargada de realizar los cálculos para segmentos de anillo que usen repetidoras.

A continuación explicaremos en forma breve pero concisa las diferentes funciones que se elaboraron para desarrollar el programa.

La función TipoCable ()

Le pregunta al usuario que tipo de cable va a utilizar para el diseño, para las redes STP tenemos tres opciones:

- Cable tipo 1
- Cable tipo 2
- Cable tipo 8

Esta función se encarga de abrir una ventana en la cual le pide al usuario que escoja el tipo de cable, cuando el usuario hace la elección, ésta es almacenada para luego junto con el número de MAU'S y número de WC'S buscar el valor de la DMTR en la tabla correcta.

La función GetDevice ()

Esta función abre una ventana para pedirle al usuario que ingrese el número de WC'S y el número de MAU'S, ofreciendo una interfase amigable con el usuario, una vez abierta la ventana, se llama a la función encargada de receptor los valores pedidos:

La función getmau () es la encargada de receptor el número de MAU'S y almacenarlo para futura utilización, tanto en cálculos de distancias, como para buscar en las tablas el valor del DMTR permitido.

La función getwc () es la encargada de receptor el número de WC'S y almacenarlo para futura utilización.

La función Getdist_wc ()

Esta es la función que se encarga de abrir una interfase amigable con el usuario para pedirle que ingrese los valores de las distancias que existen entre Wiring Closets, para luego invocar a la función encargada de receptor los valores y almacenarlos.

La función getdistwc es la se encarga de receptor estos valores y almacenarlos para uso futuro.

La función getdist_ll ()

se encarga de abrir una ventana para solicitarle al usuario que ingrese el valor de la longitud del cable de lobe más largo del anillo, para luego llamar a la función getdistlll () que es la que se encarga de receptor éste valor y almacenarlo.

La función repetidora ()

Cuando el valor del DMTR calculado supera al valor del DMTR de la tabla, la función STP () invoca a la función que se encarga de pedir los nuevos valores de los parámetros a usar y hacer los cálculos correspondientes para luego comparar con los valores tabulados y tomar una decisión para informarle luego al usuario.

La función Getdist_wc_rep ()

Es usada por la función repetidora () para pedirle al usuario que ingrese las distancias entre wiring closets y repetidoras, luego la función getdistwcrep () es la encargada de receptorlas y almacenarlas.

Las funciones que describimos a continuación se relacionan con el diseño de redes UTP y se hallan en el archivo UTP.C ()

La función UTP ()

Es la función principal de esta parte del programa, genera varios menús, invoca a las funciones encargadas de recibir los datos, se encarga de hacer los cálculos necesarios para hallar los valores del DMTR de la red analizada y realizar las comparaciones con valores teóricos para luego decidir si la red a ser diseñada es correcta o no.

La función GetDevice1 ()

Es muy parecida a la función GetDevice () de la parte de diseño STP, con la diferencia de que solicita el ingreso del número de MAU's y de paneles de conexión a usar.

La función getmauUtp() es la encargada de recibir y almacenar el número de MAU's UTP a usar.

La función getpc () es la encargada de recibir y almacenar el número de paneles de conexión a ser usados.

La función GetDevice2 ()

Esta función nos pide que ingresemos el número de bloques de conexión y de salidas en pared a ser usados.

La función getbc () se encarga de recibir y almacenar el valor del número de bloque de conexión que serán usados.

La función getsp () se encarga de recibir y almacenar el valor de número de salidas en pared que serán usadas.

La función Getdist_mau ()

Se encarga de decirle al usuario que ingrese las distancias que existen entre los MAU's UTP.

La función getdistmau () recibe y almacena estos valores.

La función Getdist_l112 ()

Se encarga de decirle al usuario que ingrese el valor de la longitud del cable de lobe más largo.

La función getdistl112 () se encarga de recibir y almacenar estos valores.

Cabe señalar que para esta parte del programa; es decir, el diseño de redes Token Ring UTP, no se implementó una función que haga los cálculos usando repetidoras, ya que las redes UTP generalmente son redes pequeñas y en el cálculo de la DMTR pesa más

la longitud del cable de lobe más largo que cualquier otra distancia, por esta razón cuando se necesite el uso de repetidoras el programa nos dará un mensaje indicandonos que intentemos de nuevo disminuyendo la longitud del cable de lobe más largo. En la práctica lo que se utiliza son Drivers que permiten extendernos un poco en la longitud del cable de lobe y no tener que usar repetidoras.

5.- DIAGRAMAS DE FLUJO

DIAGRAMA DE FLUJO DEL PROGRAMA PRINCIPAL

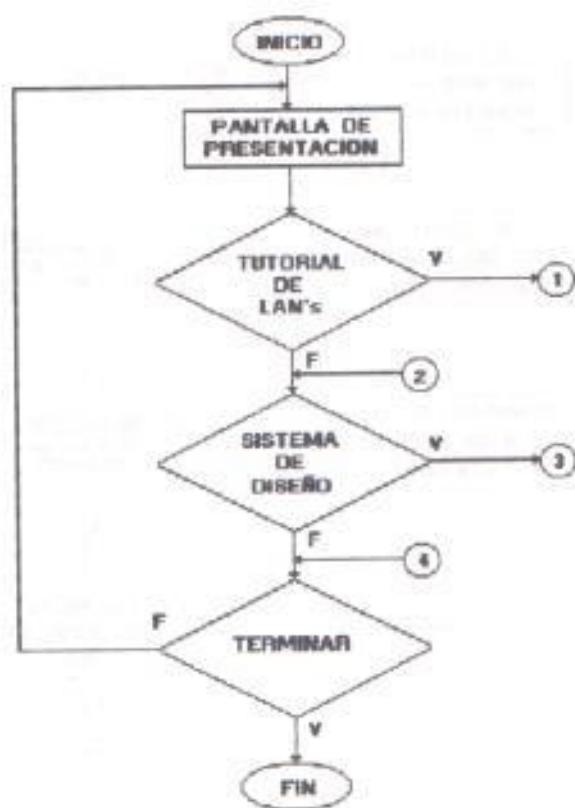


DIAGRAMA DE FLUJO DEL TUTORIAL DE LAN'8

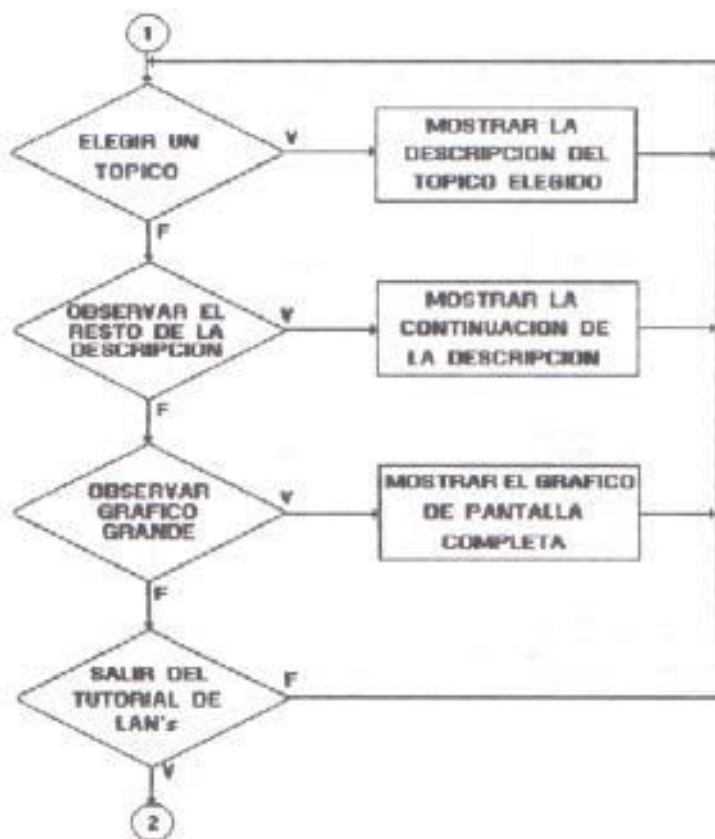
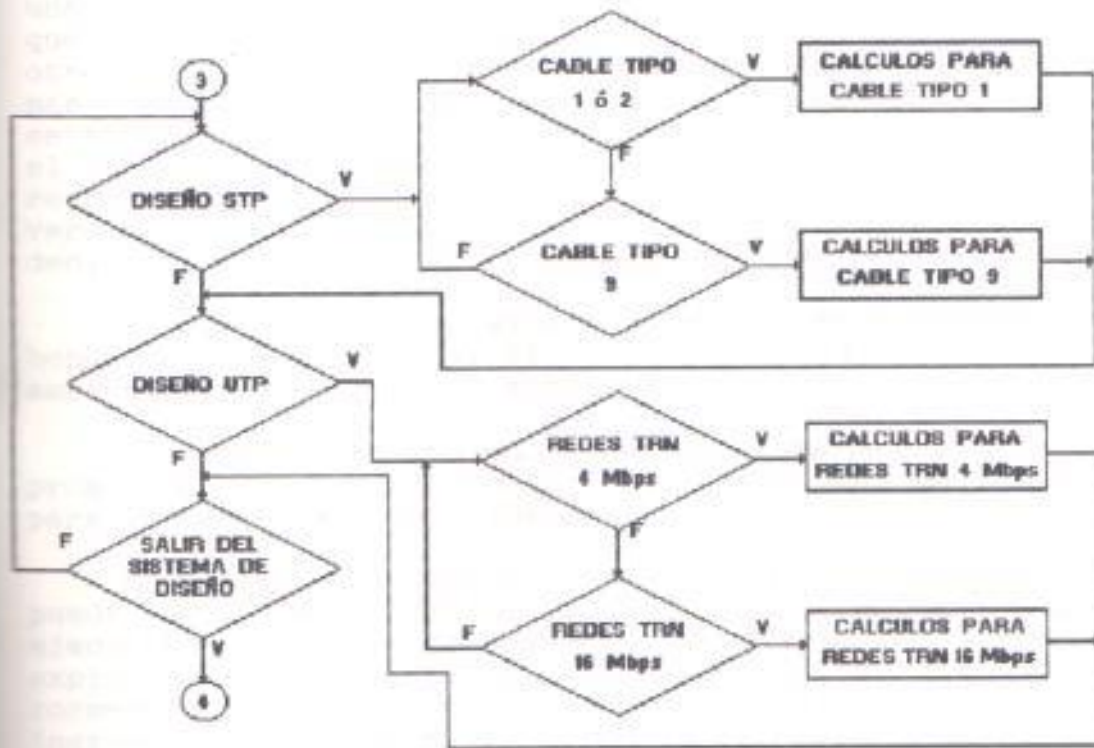


DIAGRAMA DE FLUJO DEL SISTEMA DE DISEÑO



CONCLUSIONES

Como se pudo observar en la parte de la instalación de Novell Netware versión 3.11 es muy sencilla, a diferencia de versiones anteriores como por ejemplo la versión 2.15 y 2.20 que tienen un procedimiento mucho más complejo de instalación. La versión 3.11 es una versión que genera servidores dedicados; es decir, un servidor que solo se dedica a funcionar como servidor de la red, sin atender otros requerimientos como en los servidores no dedicados, donde a más de funcionar como servidores de la red funcionan como estaciones de trabajo, lo cual provoca lentitud en la red, ya que el microprocesador del servidor debe dividir su atención a los requerimientos de la red y a las demandas de la estación. Las versiones 2.15 y 2.20 pueden generar servidores dedicados y no dedicados.

En la parte de programación se ha podido comprobar de las bondades que nos brinda el lenguaje "C" para programar tanto en ambiente gráfico como en ambiente DOS.

El programa es en realidad un programa pequeño que llama a dos programas separados, el uno para el tutorial de redes y el otro para la parte de diseño Token Ring.

El Tutorial de Lan's fue desarrollado de tal manera que uno pueda adicionarle Tópicos sin necesidad de cambiar el archivo ejecutable, simplemente editando los archivos de texto tal como se explicó anteriormente e incluyendo los archivos gráficos correspondientes al tópico adicionado. Hasta 100 Tópicos pueden ser ingresados en el Tutorial, ya que la estructura del mismo se diseñó para soportarlos. Como se pudo ver; el turbo "C" es capaz de mostrar gráficos en el PaintBrush, el cual es un utilitario del Windows, el programa está diseñado para leer el archivo gráfico y mostrarlo en pantalla, los archivos generados son de extensión BMP y pueden ser editados o generados en el CorelDraw siempre y cuando sean grabados en formato BMP.

La parte de diseño Token-Ring tiene almacenados todas las especificaciones requeridas para diseñar una red Token-Ring, de tal manera que puede ser de gran ayuda para generar una red Token-Ring, fue creado en ambiente DOS y no en ambiente gráfico, como la parte del TUTORIAL, por lo cual se puede ver la diferencia en las presentaciones de ambas partes, ya que el modo gráfico nos brinda la oportunidad de crear pantallas de presentación mucho más elegantes que las creadas por ambiente DOS, por lo cual una modificación futura para el programa podría ser modificar la parte de DISEÑO DE REDES TOKEN RING de ambiente DOS a ambiente gráfico, con lo cual se resaltaría aún más la presentación del programa y no se tendría el cambio brusco que existe al cambiar de ambiente gráfico a DOS.

BIBLIOGRAFIA

- DATACOMM SOLUTIONS
South Hills, Edición 1993.
Toda la Revista.
- Learning C ++
SAMS 1st Edition 1991. Tom Swan
Todo el Libro.
- NOVELL NETWARE version 3.11: System Administration
Novell, Inc. 1991. Edición Marzo 1991.
Todo el Libro.
- NOVELL NETWARE version 3.11: Installation
Novell, Inc. 1991. Edición Marzo 1991.
Todo el Manual.
- NOVELL NETWARE version 3.11: Print Server
Novell, Inc. 1991. Edición Marzo 1991.
Todo el Manual.
- NOVELL NETWARE 386: The Complete Reference
Osborne Mc Graw-Hill 1990. Tom Sheldon.
Todo el Libro.
- TURBO C ++: Library Reference.
Borland International 1990.
Todo el Manual.
- TURBO C ++: Library Reference: Runtime Library, Global
Variables, Cross Reference.
Borland International, Inc. 1990.
Todo el Manual.


```
// PRINCIPAL.C //
```

```
#include <alloc.h>
#include <stdio.h>
#include <stdlib.h>
#include <graphics.h>
```

```
int op=1;
```

```
////////////////////////////////////
//////////////////////// FUNCION PRINCIPAL DEL PROGRAMA //////////////////////
////////////////////////////////////
```

```
void main()
{
    opengraph();
    present();
    closegraph();
}
```

```
// FUNCIONES USADAS EN LA PARTE DE DISEÑO DE LAS PANTALLAS DE PRESENTACION
```

```
// Para mayor información sobre las funciones empleadas en los programas refiérase al manual TURBO "C" LIBRARY REFERENCE.
// bar(int left, int top, int right, int bottom) : dibuja una línea de dos dimensiones de acuerdo con la última especificación de
// estilo de llenado y de color. La esquina superior izquierda viene dada por (left,top) y la esquina inferior derecha viene dada
// por (right, bottom).
// line (int x1, int y1, int x2, int y2) : dibuja una línea entre 2 puntos especificados.
// la línea es dibujada con el color actual, usando el el estilo de línea y grueso actual (o default) entre los puntos
// especificados (x1,y1) y (x2,y2).
// outtextxy(int x, int y, char far #textstring) : muestra un string de texto en una posición especificada.
// (x,y) es la posición en la cual será mostrado el texto, usando la última especificación para tipo de letra, dirección y tamaño.
// textstring es el texto que será mostrado, o la variable que contiene el texto a ser mostrado.
// rectangle(int left, int top, int right, int bottom) : dibuja un rectángulo según la última especificación de estilo de línea,
// grueso y color. (left, top) es la esquina superior izquierda del rectángulo y (right, bottom) es la esquina inferior derecha.
// setcolor(int color) : setea el color con el que se va a mostrar los gráficos o textos en modo gráfico.
// setfillstyle(int pattern, int color) : setea la manera de llenado y el color de un bloque.
// pattern indica el estilo en que será llenado el bloque, puede tener valores entre 0 y 11. Con valor de 1 hace un llenado sólido
// color indica el color con el que será llenado el bloque, puede tener valores entre 0 y 15.
// setlinestyle(int linestyle, unsigned upattern,int thickness) :setea el ancho y estilo de una línea.
// linestyle especifica el estilo de la línea que va a ser dibujada( sólida, centrada, cortada, etc).
// thickness especifica si el ancho de la línea es normal (1 pixel) o grueso (3 pixeles).
// upattern es usado sólo si linestyle es 4.
// settextstyle(int font, int direction, int charsize) : setea las características del texto que será graficado
// font indica la calidad de letra que será mostrada:
// 0 DEFAULT_FONT, 1 TRIPLEX_FONT, 2 SMALL_FONT, 3 SANS_SERIF_FONT, 4 GOTHIC_FONT.
// direction indica la dirección en que se mostrará el texto:
// 0 HORIZ_DIR (de izquierda a derecha), 1 VERT_DIR (de abajo hacia arriba).
// charsize indica el tamaño de letra, 0 muestra un font en un rectangulo de 8x8 pixeles, 1 el doble, 2 el triple, etc.
```

```

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX//
//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX//
//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX//

// Esta función se encarga de mostrarnos la pantalla de presentación y de invocar a las funciones principales de la parte
// tanto del tutorial como del diseño de redes Token Ring.

present()
{
char ca;
int x;
setfillstyle(1,15); bar(0,0,640,400);
setfillstyle(1,1); bar(0,0,640,20);
setfillstyle(1,7); bar(121,51,519,174);
setcolor(0); settextstyle(0,0,1);
outtextxy(175,70,"TOPICO : REDES DE AREA LOCAL");
outtextxy(175,90,"PROFESOR : ING. JAIME PUENTE");
outtextxy(175,110,"ALUMNOS : JOHNNY AVALOS HERRERA");
outtextxy(175,130," ALEX MIRANDA ANDRADE");
outtextxy(175,150," RAUL HERNANDEZ");
setcolor(0);
rectangle(118,48,22,177);
setcolor(8);
line(119,176,521,176);
line(120,175,520,175);
line(521,176,521,49);
line(520,176,520,50);
rectangle(165,60,475,165);
setcolor(15);
line(165,165,475,165);
line(475,165,475,60);
setcolor(15); settextstyle(0,0,1);
outtextxy(145,0,"PROYECTO FINAL DE TOPICO DE GRADUACION");
setfillstyle(1,7);
bar(121,250,240,280);
bar(121,281,519,406);
setcolor(0);
line(118,247,243,247);
line(118,247,118,409);
line(243,247,243,278);
line(243,278,522,278);
line(118,409,522,409);
line(522,409,522,279);
setcolor(8);
line(119,408,518,408);
line(120,407,518,407);
line(121,406,518,406);
line(519,408,519,281);
line(520,408,520,280);
line(521,408,521,279);
setcolor(0); settextstyle(0,0,1);
outtextxy(148,262,"OPCIONES");

```



```

setcolor(8);
  rectangle(136,255,225,275);
setcolor(15);
  line(136,275,225,275);
  line(225,275,225,255);
setfillstyle(1,7);
  bar(0,480-39,640,480);
setcolor(0);
  rectangle(0,480-39,640,480);
setcolor(15);
  line(1,480-38,640-1,480-38);
  line(2,480-37,640-2,480-37);
  line(1,480-38,1,480-1);
  line(2,480-37,2,480-2);
setcolor(8);
  line(1,480-1,640-1,480-1);
  line(2,480-2,640-2,480-2);
  line(3,480-3,640-3,480-3);
  line(640-1,480-38,640-1,480-1);
  line(640-2,480-37,640-2,480-2);
  line(640-3,480-36,640-3,480-3);
setcolor(8);
  line(11,480-30,135,480-30);
  line(11,480-30,11,480-10);
  line(150,480-30,150,480-10);
  line(150,480-30,640-11,480-30);
setcolor(15);
  line(12,480-9,135,480-9);
  line(640-11,480-30,640-11,480-10);
  line(135,480-30,135,480-10);
  line(151,480-10,640-11,480-10);
setcolor(0); settextstyle(11,0,3);
  outtextxy(25,480-22,"J. Avalos H.");
settextstyle(11,0,3);
  outtextxy(170,480-22,"Escuela Superior Politecnica del Litoral (ESPOL) 1993. ");
for (;;)
{
  if (op==1) // cuando op = 1 el programa nos permite elegir la parte del TUTORIAL DE LANS
  {
    opcion1();
    opcion5();
    opcion6();
  }
  if (op==2) // cuando op = 2 el programa nos permite elegir la parte del DISEÑO DE REDES TOKEN RING.
  {
    opcion2();
    opcion4();
    opcion6();
  }
  if (op==3) // cuando op = 3 el programa nos permite SALIR AL SISTEMA.
  {
    opcion4();
  }
}

```

```

    opcion5();
    opcion3();
}
ca=getch(); // el programa se encuentra receplando constantemente el valor de la tecla que presione el usuario.
if (ca==80 && op==1)
{
    // cuando las teclas presionadas son las flechas hacia arriba o hacia abajo, el programa
    // la opción que le permite elegir al usuario
    op=2;
    continue;
}
if (ca==80 && op==2)
{
    op=3;
    continue;
}
if (ca==72 && op==3)
{
    op=2;
    continue;
}
if (ca==72 && op==2)
{
    op=1;
    continue;
}
if (ca==13 && op==1) // cuando op = 1 y la tecla que presiona el usuario es ENTER, se ha seleccionado la parte del
{ // TUTORIAL DE LANS.
    bajo1();
    sound(1000); // emite un sonido de 1000 Hertz
    delay(100); // durante un periodo de tiempo de 100 milisegundos.
    nosound(); // elimina el sonido
    delay(200); // ofrece un retardo de 200 ms para permitir observar el aspecto tridimensional de la pantalla
    main1(); // de presentación, luego llama a la función principal de la parte del Tutorial.
}
if (ca==13 && op==2) // cuando op = 2 y la tecla que presiona el usuario es ENTER, se ha seleccionado la parte del
{ // DISEÑO DE REDES TOKEN RING.
    bajo2();
    sound(1000);

    nosound();
    delay(200);
    main2 (); // llama a la función principal de la parte de DISEÑO.
}
if (ca==13 && op==3) // cuando op = 3 y la tecla que presiona el usuario es ENTER, se ha seleccionado SALIR DEL
{ // SISTEMA.
    bajo3();
    sound(1000);
    delay(100);
    nosound();
    delay(200);
    closegraph();
    exit(0); // termina la ejecución del programa.
}

```

```

}
return(0);
}

//*****
//***** LA FUNCION opengraph () *****
//*****

// Esta función abre la función gráfica de programación.

opengraph()
{
    int driver,mode;
    driver=9;           // los valores para video VGA son 9 y 2.
    mode=2;
    initgraph (&driver,&mode,""); // inicia la sesión gráfica de programación para video VGA.
    return (0);
}

//*****
//***** FUNCIONES PARA ELEGIR LA OPCION *****
//*****

// Mediante una combinación de las siguientes funciones se crea el ambiente tridimensional para la elección de la opción
// que el usuario vaya a elegir; es decir, puede elegir: correr el tutorial, la parte de diseño de redes Token Ring o
// puede elegir salir del sistema.
// Cuando op=1, se muestra el mensaje "TUTORIAL DE REDES DE AREA LOCAL" de manera que parezca que esa ventana se ha levantado.
opcion1()
{
    setfillstyle(1,8);
    bar(138,290,504,310);
    setfillstyle(1,0);
    bar(142,311,508,314);
    bar(505,294,508,314);
    setcolor(0);
    outtextxy(175,298,"TUTORIAL DE REDES DE AREA LOCAL");
    outtextxy(176,299,"TUTORIAL DE REDES DE AREA LOCAL");
    outtextxy(177,300,"TUTORIAL DE REDES DE AREA LOCAL");
    outtextxy(178,301,"TUTORIAL DE REDES DE AREA LOCAL");
    setcolor(15);
    outtextxy(174,297,"TUTORIAL DE REDES DE AREA LOCAL");
    return(0);
}

// Cuando op=2, se muestra el mensaje "DISEÑO DE REDES TOKEN RING" de manera que parezca que esa ventana se ha levantado.
opcion2()
{
    setfillstyle(1,8);
    bar(138,330,504,350);
    setfillstyle(1,0);
    bar(142,351,508,354);
    bar(505,334,508,354);

```



```

setcolor(0);
outtextxy(197,338,"DISEÑO DE REDES TOKEN RING");
outtextxy(198,339,"DISEÑO DE REDES TOKEN RING");
outtextxy(199,340,"DISEÑO DE REDES TOKEN RING");
outtextxy(200,341,"DISEÑO DE REDES TOKEN RING");
setcolor(15);
outtextxy(196,337,"DISEÑO DE REDES TOKEN RING");
return (0);
}

```

```

// Cuando op=3, se muestra el mensaje "SALIR AL SISTEMA" de manera que parezca que esa ventana se ha levantado.
opcion3()

```

```

{
setfillstyle(1,8);
bar(138,370,504,390);
setfillstyle(1,0);
bar(142,391,508,394);
bar(505,374,508,394);
setcolor(0);
outtextxy(245,378,"SALIR AL SISTEMA");
outtextxy(246,379,"SALIR AL SISTEMA");
outtextxy(247,380,"SALIR AL SISTEMA");
outtextxy(248,381,"SALIR AL SISTEMA");
setcolor(15);
outtextxy(244,377,"SALIR AL SISTEMA");
return (0);
}

```

```

// Cuando op<>1, se muestra el mensaje "TUTORIAL DE REDES DE AREA LOCAL" de manera normal.
opcion4()

```

```

{
setfillstyle(1,7);
bar(136,290,514,314);
setcolor(0);
outtextxy(179,302,"TUTORIAL DE REDES DE AREA LOCAL");
setcolor(8);
rectangle(142,294,502,314);
setcolor(15);
line(142,314,502,314);
line(502,314,502,294);
return(0);
}

```

```

// Cuando op<>2, se muestra el mensaje "DISEÑO DE REDES TOKEN RING" de manera normal.
opcion5()

```

```

{
setfillstyle(1,7);
bar(136,330,514,354);
setcolor(0);
outtextxy(201,342,"DISEÑO DE REDES TOKEN RING");
setcolor(8);
rectangle(142,334,502,354);
}

```

```

setcolor(15);
line(142,354,502,354);
line(502,334,502,354);
return (0);
}

```

// Cuando op<>3, se muestra el mensaje "SALIR AL SISTEMA" de manera normal.

opcion6()

```

{
setfillstyle(1,7);
bar(136,370,514,394);
setcolor(0);
outtextxy(249,382,"SALIR AL SISTEMA");
setcolor(8);
rectangle(142,374,502,394);
setcolor(15);
line(142,394,502,394);
line(502,394,502,374);
return (0);
}

```

// Cuando op=1 y la tecla presionada por el usuario fue ENTER, se muestra el mensaje "TUTORIAL DE REDES DE AREA LOCAL" de manera // que parezca que esa ventana se ha presionado, dándole el aspecto tridimensional a la pantalla.

bajol()

```

{
setfillstyle(1,7);
bar(138,290,504,293);
bar(138,290,141,314);
setfillstyle(1,8);
bar(142,294,508,314);
setcolor(8);
outtextxy(174,297,"TUTORIAL DE REDES DE AREA LOCAL");
outtextxy(175,298,"TUTORIAL DE REDES DE AREA LOCAL");
outtextxy(176,299,"TUTORIAL DE REDES DE AREA LOCAL");
outtextxy(177,300,"TUTORIAL DE REDES DE AREA LOCAL");
setcolor(15);
outtextxy(178,301,"TUTORIAL DE REDES DE AREA LOCAL");
return(0);
}

```

// Cuando op=2 y la tecla presionada por el usuario fue ENTER, se muestra el mensaje "DISEÑO DE REDES TOKEN RING" de manera // que parezca que esa ventana se ha presionado, dándole el aspecto tridimensional a la pantalla.

bajo2()

```

{
setfillstyle(1,7);
bar(138,330,504,333);
bar(138,330,141,354);
setfillstyle(1,8);
bar(142,334,508,354);
setcolor(8);
outtextxy(196,337,"DISEÑO DE REDES TOKEN RING");
outtextxy(197,338,"DISEÑO DE REDES TOKEN RING");
}

```

```

outtextxy(198,339,"DISEÑO DE REDES TOKEN RING");
outtextxy(199,340,"DISEÑO DE REDES TOKEN RING");
setcolor(15);
outtextxy(200,341,"DISEÑO DE REDES TOKEN RING");
return (0);
}

```

```

// Cuando op=3 y la tecla presionada por el usuario fue ENTER, se muestra el mensaje "SALIR AL SISTEMA" de manera
// que parezca que esa ventana se ha presionado, dándole el aspecto tridimensional a la pantalla.
bajo3()

```

```

{
setfillstyle(1,7);
bar(138,370,504,373);
bar(138,370,141,394);
setfillstyle(1,8);
bar(142,374,508,394);
setcolor(8);
outtextxy(244,377,"SALIR AL SISTEMA");
outtextxy(245,378,"SALIR AL SISTEMA");
outtextxy(246,379,"SALIR AL SISTEMA");
outtextxy(247,380,"SALIR AL SISTEMA");
setcolor(15);
outtextxy(248,381,"SALIR AL SISTEMA");
return (0);
}

```


HELP.C

```
#include <malloc.h>
#include <stdio.h>
#include <stdlib.h>
#include <graphics.h>

// declaración de las variables globales del programa.
int new=1;
int flag=0;
int posc;
int posm;
int posd;
char topics [100][40];
char topbop [100][18];
char top [100][18];
char desc [200][90];
char fhel [ 50][40];

//*****
//***** FUNCION PRINCIPAL DEL PROGRAMA *****
//*****

// Esta es la función principal del programa en lo que corresponde al tutorial de LAN's

void main1()
{
    int rep=1,x;
    posc=0; posm=0;
    posd=0;
    limptop();
    loadtopics();
    present1();
    for (x=0; x!=2;)
    {
        if (rep==1) rep = menuh(rep); // se ejecuta la ventana de menú de tópicos disponibles.
        if (rep==2) rep = ventana(rep); // se ejecuta la ventana de descripción del tópico.
        if (rep==-1) break; // se sale del lazo y termina la ejecución del tutorial.
    }
    new=1; flag=0; // antes de salir del tutorial, se vuelven a dejar inicializadas estas variables.
    present();
}

//*****
//***** LA FUNCION present1 () *****
//*****

// Esta función es la encargada de mostrar la pantalla de presentación y de marginar la pantalla para mostrar tanto el texto
// del tutorial como de los gráficos

present1()
{
```

```

setfillstyle (1,15); bar(0,0,640,480);
setfillstyle (1,1); bar(0,0,640,19);
setfillstyle (1,7); bar(0,0,19,19);
setcolor(0); setlinestyle(0,1,0);
    line (0,39,640,39);
    line (0,19,640,19);
    line (19,0,19,19);
    line (4,8,4,11);
    line (15,8,15,11);
    line (16,9,16,12);
    line (4,8,15,8);
    line (4,11,15,11);
    line (5,12,16,12);
setcolor(0); settextrstyle(0,0,1);
    outtextxy (10,26,"Opciones");
    outtextxy (100,26,"Ayuda-F1");
    outtextxy (10,27,"_");
    outtextxy (100,27,"_");
setcolor(15); settextrstyle(0,0,1);
    outtextxy(260,8,"TUTORIAL DE LANs");
setfillstyle(1,7);
    bar(0,480-39,640,480);
setcolor(0);
    rectangle(0,480-39,640,480);
setcolor(15);
    line(1,480-38,640-1,480-38);
    line(2,480-37,640-2,480-37);
    line(1,480-38,1,480-1);
    line(2,480-37,2,480-2);
setcolor(8);
    line(1,480-1,640-1,480-1);
    line(2,480-2,640-2,480-2);
    line(3,480-3,640-3,480-3);
    line(640-1,480-38,640-1,480-1);
    line(640-2,480-37,640-2,480-2);
    line(640-3,480-36,640-3,480-3);
setcolor(8);
    line(11,480-30,135,480-30);
    line(11,480-30,11,480-10);
    line(150,480-30,150,480-10);
    line(150,480-30,640-11,480-30);
setcolor(15);
    line(12,480-9,135,480-9);
    line(640-11,480-30,640-11,480-10);
    line(135,480-30,135,480-10);
    line(151,480-10,640-11,480-10);
setcolor(0);
settextstyle(11,0,3);
    outtextxy(25,480-22,"J. Avalos H.");
settextstyle(11,0,3);
    outtextxy(170,480-22,"Escuela Superior Politecnica del Litoral (ESPOL) 1993. ");
setcolor(0);

```

```

        rectangle(349,251,630,50);
        rectangle(346,254,633,47);
setfillstyle(1,15);
        bar(7,47,327,424);
setcolor(1);
        rectangle(7,47,327,424);
        rectangle(10,50,324,421);
        rectangle(8,48,326,423);
        rectangle(9,49,325,422);
setfillstyle(1,1);
        bar(7,47,327,67);
setfillstyle(1,1);
        bar(346,261,627,424);
setfillstyle(1,15);
        bar(349,264,624,421);
setfillstyle(1,1);
        bar(349,264,624,284);
settextstyle(0,0,1); setcolor(15);
        outtextxy(380,270,"MENU DE TOPICOS DISPONIBLES");
        // cuando se ingresa al TUTORIAL DE LANs, flag está inicializada en 0, por lo tanto se grafica el mensaje de
        // bienvenida.
        if (flag==0)
        {
            setcolor(15);
            outtextxy(57,55,"DESCRIPCION DEL TOPICO");
            setcolor(0);
            outtextxy(17,100,"BIENVENIDO AL TUTORIAL DE LANs");
            outtextxy(17,115,"Para elegir el tema a consultar, des-");
            outtextxy(17,125,"plácese en la ventana de tópicos dis-");
            outtextxy(17,135,"ponibles, con las flechas de desplaza-");
            outtextxy(17,145,"miento arriba-abajo.");
            outtextxy(17,165,"Para consultar el tema de su interés,");
            outtextxy(17,175,"ubíquese con la barra en el tópico que");
            outtextxy(17,185,"desea consultar, luego presione ENTER.");
            outtextxy(17,205,"Para temas cuyo texto abarque más de");
            outtextxy(17,215,"una página, desplácese a la ventana");
            outtextxy(17,225,"izquierda con la flecha de desplaza-");
            outtextxy(17,235,"miento izquierda, luego muévase en la");
            outtextxy(17,245,"ventana de arriba hacia abajo (o vice-");
            outtextxy(17,255,"versa) con las flechas de desplazamien-");
            outtextxy(17,265,"to arriba-abajo.");
            outtextxy(17,285,"Para retornar a la ventana de tópicos,");
            outtextxy(17,295,"presionar la flecha de desplazamiento");
            outtextxy(17,305,"hacia la derecha.");
            outtextxy(17,325,"Para más ayuda presione F1.");
            loadbap("portada.bap");
            flag = 1; // se cambia el valor de flag a 1 para no volver a graficar esto mientras se este en el
            // tutorial.
        }
return(0);
}

```



```

//*****//
//***  GENERAN Y LIMPIA LAS SOMBRAS DE LAS VENTANAS  ***//
//*****//

// Mediante combinaciones de éstas funciones se generan los efectos de los cambios en las ventanas de menú de opciones
// y de descripción del tópico.
sombas1() // genera la sombra en la ventana de tópicos disponibles, indicándonos que nos
          // encontramos en esa ventana.
{
    setfillstyle(1,7);
    bar(628,269,636,434);
    bar(352,425,636,434);
    return(0);
}

limp_sombas1() // limpia la sombra en la ventana de tópicos disponibles.
{
    setfillstyle(1,15);
    bar(628,269,636,434);
    bar(352,425,636,434);
    return(0);
}

sombas2() // genera la sombra en la ventana de descripción del tópico, indicándonos que nos
          // encontramos en esa ventana.
{
    setfillstyle(1,7);
    bar(328,55,335,434);
    bar(17,425,335,434);
    return(0);
}

limp_sombas2() // limpia la sombra en la ventana de descripción del tópico.
{
    setfillstyle(1,15);
    bar(328,55,335,434);
    bar(17,425,335,434);
    return(0);
}

//*****//
//***** LA FUNCION loadbmp () *****//
//*****//

// Esta función toma el archivo gráfico que le es pasado como argumento y lo gráfica en la ventana que existe para mostrar
// los gráfico pequeños.

loadbmp (char fbmp[18])
{
    long int r;
    int z,x;
    int bmp,posy;
    FILE *file;
    file=fopen(fbmp,"rb"); // abre el archivo gráfico que le es pasado como argumento para lectura binaria.
    x=350;
    posy=250;

```

```

for (r=0; ; ++r)
{
    z=0;
    bmp=fgetc(file);
    if (bmp==EOF) break;
    if (z==0 && bmp==251) { bmp=14; z=1; } // se redireccionan los colores del PAINT_BRUSH al TURBO "C".
    if (z==0 && bmp==3) { bmp=2; z=1; }
    if (z==0 && bmp==1) { bmp=12; z=1; }
    if (z==0 && bmp==249) { bmp=12; z=1; }
    if (z==0 && bmp==252) { bmp=9; z=1; }
    if (z==0 && bmp==4) { bmp=1; z=1; }
    if (z==0 && bmp==254) { bmp=11; z=1; }
    if (z==0 && bmp==248) { bmp=8; z=1; }
    if (z==0 && bmp==6) { bmp=3; z=1; }
    if (r>1077)
    {
        putpixel (x, posy, bmp); // se grafica pixel a pixel en la posición (x, posy) los bmp del archivo.
        x++;
        if (x==630) // se grafica de izquierda a derecha y de abajo hacia arriba, linea por linea,
        { // ya que así son grabados los archivos BMP.
            x=350;
            posy--posy;
        }
    }
}
fclose(file); // se cierra el archivo que se abrió para lectura binaria.
return(0);
}

```

```

//*****
//***** LA FUNCION loadbmp640x480 () *****
//*****

```

// Esta función se encarga de leer el archivo gráfico grande y de mostrarlo en pantalla, cuando el gráfico existe, cuando no hay gráfico grande llama a la función no_graph(), la cual le avisa al usuario que no hay gráfico grande para ese // tópico.

```

loadbmp640(char bmp[18], int cur)
{
    long int r;
    int z, x;
    int bmp, posy;
    char c;
    unsigned size;
    void far *image;
    FILE *file;
    file=fopen(bmp, "rb"); // se abre el archivo gráfico que le es pasado como argumento para lectura binaria.
    if (file==NULL)
    {
        no_graph (cur); // si no hay gráfico grande se llama a esta función que le indica esto al usuario.
        return (0);
    }
}

```

```

size=imagesize(0,0,280,200); // calcula el tamaño de la ventana en la que está graficado el gráfico pequeño.
if ((image = farmalloc(size))!= NULL)
// si no hay suficiente memoria para almacenar esta ventana muestra un mensaje indicándolo y termina la ejecución del programa.
{
    closegraph();
    printf("Error: not enough heap space in save_screen().\n");
    exit(1);
}
getimage(350,50,630,250,image); // guarda el gráfico pequeño en memoria.
x=0;
posy=480;
for (r=0;;++r)
{
    z=0;
    bmp=fgetc(file);
    if (bmp==EOF) break;
    if (z==0 && bmp==251) { bmp=14; z=1; }
    if (z==0 && bmp==3) { bmp=2; z=1; }
    if (z==0 && bmp==1) { bmp=12; z=1; }
    if (z==0 && bmp==249) { bmp=12; z=1; }
    if (z==0 && bmp==252) { bmp=9; z=1; }
    if (z==0 && bmp==4) { bmp=1; z=1; }
    if (z==0 && bmp==254) { bmp=11; z=1; }
    if (z==0 && bmp==248) { bmp=8; z=1; }
    if (z==0 && bmp==6) { bmp=3; z=1; }
    if (r>1077)
    {
        putpixel (x,posy,bmp);
        x++;
        if (x==640)
            { x=0; posy--posy; }
    }
}
fclose(file); // cierra el archivo que abrió para lectura binaria.
for (r=0;;++r)
{
    c=getch(); // el gráfico permanecerá en pantalla hasta que no se presione la tecla ESC.
    if (c==27) break;
    continue;
}
present1();
refresh_desc();
if (cur==1)
{
    limp_sombras2(); // si la posición del usuario antes de graficar el bmp grande era la ventana de tópicos disponibles
    sombras1(); // se sombrea ésta ventana y se limpia la sombra de la ventana de descripción del tópico.
}
if (cur==2)
{
    limp_sombras1(); // si la posición del usuario antes de graficar el bmp grande era la ventana de descripción del
    sombras1(); // tópico se sombrea ésta ventana y se limpia la sombra de la ventana de tópicos disponibles.
    sombras2();
}

```



```

    limp_menu();
    setfillstyle(1,7);
    if (posc==0) bar(350,290,623,305);
    if (posc==1) bar(350,305,623,320); // se vuelve a crear la ventanade tópicos disponibles.
    if (posc==2) bar(350,320,623,335);
    if (posc==3) bar(350,335,623,350);
    if (posc==4) bar(350,350,623,365);
    if (posc==5) bar(350,365,623,380);
    if (posc==6) bar(350,380,623,395);
    if (posc==7) bar(350,395,623,410);
    settextstyle (0,0,1);setcolor(0);
    outtextxy (360,295,topics[posa]);
    outtextxy (360,310,topics[posa+1]);
    outtextxy (360,325,topics[posa+2]);
    outtextxy (360,340,topics[posa+3]);
    outtextxy (360,355,topics[posa+4]);
    outtextxy (360,370,topics[posa+5]);
    outtextxy (360,385,topics[posa+6]);
    outtextxy (360,400,topics[posa+7]);
}
putimage(350,50,image,COPY_PUT); // se coloca la imagen guardada en memoria en la ventana de gráfico pequeño.
farfree(image); // se libera la memoria ocupada con el gráfico pequeño.
return(0);
}

//*****
//
//          BUCLE PARA CORRER EL MENU
//
//*****

// Esta función es la que genera la interacción entre el programa y el usuario, cuando éste se encuentra en la
// ventana de tópicos disponibles

menuh(int cur)
{
    int x,y,z,new1=0;
    limp_sombras2();  sombras1();
    for (x=0;x!=2;)
    {
        if (new==1)
        {
            limp_menu();
            setfillstyle(1,7);
            if (posc==0) bar(350,290,623,305);
            if (posc==1) bar(350,305,623,320);
            if (posc==2) bar(350,320,623,335);
            if (posc==3) bar(350,335,623,350);
            if (posc==4) bar(350,350,623,365);
            if (posc==5) bar(350,365,623,380); // se generan la lista de tópicos disponibles de acuerdo a la posición
            if (posc==6) bar(350,380,623,395); // del cursor (posc) y a la posición del tópico (posa).
            if (posc==7) bar(350,395,623,410);
        }
    }
}

```

```

settextstyle(0,0,1); setcolor(0);
outtextxy(360,295, topics[posm+0]);
outtextxy(360,310, topics[posm+1]);
outtextxy(360,325, topics[posm+2]);
outtextxy(360,340, topics[posm+3]);
outtextxy(360,355, topics[posm+4]);
outtextxy(360,370, topics[posm+5]);
outtextxy(360,385, topics[posm+6]);
outtextxy(360,400, topics[posm+7]);
new=0;
}
y=getch();
if (y==68) // si la tecla presionada es F10 se activa el menú.
{
menu_top(cur);
continue;
}
if (y==62) // si la tecla presionada es F4 se muestra el gráfico grande.
{
new=1;
loadbmp640(top[posm+posc],cur);
continue;
}
if (y==13 && new1==0) // si la tecla presionada es ENTER, se muestra la descripción del tópico elegido.
{
load_desc (posm+posc);
new1=1;
continue;
}
if (y==72) // si la tecla presionada es la FLECHA UP, se cambia el tópico disponible a elegir una posición
// hacia arriba.
{
if (posc==0)
{
if (posm==0) continue;
posm--posm;new=1;new1=0; continue;
}
posc--posc;new=1;new1=0;
}
if (y==80) // si la tecla presionada es la FLECHA DOWN, se cambia el tópico disponible a elegir una
// posición hacia arriba.
{
if (posc==7)
{
if (topics[posm+8][0]!='\0') continue;
posm++posm;
new=1;
new1=0;
continue;
}
posc++posc;
new=1;
new1=0;
}
}

```

```

if (y==75)          // si la tecla presionada es FLECHA-IZQ, nos cambiamos a la ventana de descripción del tópico.
{
    cur=2;
    return (cur);
}
if (y==61)         // si la tecla presionada es F3, salimos del TUTORIAL DE LANs y regresamos a la pantalla de
{                 // presentación principal.
    cur=-1;
    new=1;
    return (cur);
}
if (y==59) help(cur); // si la tecla presionada es F1, activamos la ventana de ayuda.
}
return(cur);
}

```

```

//*****
//***** LA FUNCION limp_menu () *****
//*****

```

// Esta función pinta de blanco la ventana del menú de tópicos disponibles

```

limp_menu()
{
    setfillstyle(1,15); bar(349,285,624,421);
    return(0);
}

```

```

//*****
//***** LA FUNCION print_desc () *****
//*****

```

// Esta función imprime el contenido del tópico en la ventana de descripción del tópico, imprime 25 líneas por ventana

```

print_desc (char bmp[18],char tema[70])
{
    int x,y,z;
    setfillstyle(1,1); bar(7,47,327,67);
    setcolor(15); settextstyle(0,0,1);
        outtextxy(20,55,tema);
    setfillstyle(1,15); bar(17,75,323,420);
    setcolor(0); settextstyle(2,0,4);
    for (x=0; x!=25; ++x)
        {
            outtextxy (17,(80+(x*13)),desc[posd+x]);
        }
    loadbmp(bmp); // invoca a la función encargada de graficar el BMP pequeño asociado con el tópico elegido.
    return(0);
}

```

```

//*****
//***** LA FUNCION refresh_desc () *****
//*****

```



```
// Cuando el contenido de la descripción del tópico lleva más de 1 pag, se usa esta función para mostrar la página anterior
```

```
refresh_desc()
{
    int x;
    setfillstyle(1,1); bar(7,47,327,67);
    setcolor(15); settextstyle(0,0,1);
        outtextxy(20,55,topic[posc+posc]);
    setfillstyle(1,15); bar(17,75,323,420);
    setcolor(0); settextstyle(2,0,4);
    for (x=0;x!=25;++x)
        {
            outtextxy (17,(80+(x*13)),desc[posd+x]);
        }
    return(0);
}
```

```
////////////////////////////////////////////////////////////////////////
//***** LA FUNCION ventana () *****/
////////////////////////////////////////////////////////////////////////
```

```
// Esta función nos da la interacción entre el programa y el usuario, cuando éste se encuentra en la ventana de
// descripción del tópico
```

```
ventana(int cur)
{
    int x,y,z;
    limp_sombras1(); sombras2();
    for (x=0;x!=2;) // se crea un bucle infinito.
        {
            y=getch(); // el programa espera por el ingreso de una opción.
            if (y==68) // si la tecla presionada es F10 se activa el menú
                {
                    menu_top(cur);
                    continue;
                }
            if (y==62) // si la tecla presionada es F4 se muestra el gráfico grande.
                {
                    new=1;
                    loadbmp640(top[posc+posc],cur);
                    continue;
                }
            if (y==61) return(-1); // si la tecla presionada es F3, regresamos al programa principal.
            if (y==72 && posd!=0) // si la tecla presionada es FLECHA-DOWN, se muestra la siguiente página relacionada con el
                { // tópico actual.
                    posd=posd-24;
                    refresh_desc();
                    continue;
                }
            if (y==80) // si la tecla presionada es FLECHA-UP, se muestra la página anterior relacionada con el
                { // tópico actual.
                    if (desc[posd+24][0]!='\0') continue;
                }
        }
}
```

```
    posd=posd+24;
    refresh_desc();
    continue;
}
if (y==59) // si la tecla presionada es F1, se muestra la ventana de ayuda.
{
    help (cur);
    continue;
}
if (y==77) // si la tecla presionada es FLECA-DER, se cambia a la ventana de tópicos disponibles.
// ya que se retorna a la función main1 con el valor de 1.
{
    cur=1;
    return(cur);
}
}
return(0);
}
```

START.C

```
#include <stdio.h>
#include <alloc.h>
#include <graphics.h>

// declaración de las variables globales del programa, se las declara como extern ya que también existen en un archivo externo
// como lo es HELP.C.
extern int posd,flag;
extern posm, posc;
extern int new;
extern char topics[100][40];
extern char topbmp[100][18];
extern char top [100][18];
extern char desc [200][90];
extern char fhel[ 50][40];

//*****
//***** LA FUNCION loadtopics () *****
//*****

// ESTA FUNCION LEE EL ARCHIVO TOPICS.HELP Y LO CARGA EN LOS SIGUIENTES ARREGLOS :
// TOPICS[], EN EL CUAL CARGA LOS TITULOS DE LOS TEMAS A TRATAR EN EL TUTORIAL, Y QUE SON MOSTRADOS EN LA VENTANA DE
// MENU DE TOPICOS DISPONIBLES;
// TOPBMP[], EN EL CUAL CARGA LOS NOMBRES DE LOS GRAFICOS PEQUEÑOS A MOSTRAR EN LA VENTANA PARA MOSTRAR DICHOS GRAFICOS;
// TOP[], EN EL CUAL GUARDA LOS NOMBRES DE LOS GRAFICOS GRANDES A MOSTRAR, QUE SON DE PANTALLA COMPLETA.
// PARA DIFERENCIAR QUE GUARDAR EN CADA ARREGLO, HACE USO DEL CARACTER '\n'.

loadtopics()
{
    int x,y,z,r;
    char ca;
    FILE *file;
    file = fopen ("topics.hlp","r");
    if (file==NULL)
    {
        clrscr();
        printf("\nError abriendo archivos..!\n");
        exit(1);
    }
    z=0; r=1;
    for (z=0; z!=100; ++z)
    {
        ca=fgetc(file);
        if (ca==EOF)
        {
            if (r==1) topics[x][z]='\0';
            if (r==2) topbmp[x][z]='\0';
            if (r==3) top [x][z]='\0';
            break;
        }
    }
}
```



```

        if (ca=='\n')
        {
            if (r==3)
            {
                r=1; x=++x; z=-1;
                continue;
            }
            r=++r;
            z=-1;
            continue;
        }
        if (r==1) topics[x][z]=ca;
        if (r==2) topbmp[x][z]=ca;
        if (r==3) top [x][z]=ca;
    }
    fclose(file);
    return(0);
}

```

```

//*****//
//***** LA FUNCION limpdesc () *****//
//*****//

```

// Limpia el arreglo desc[][] asignándole caracteres nulos ('\0') al primer elemento de cada arreglo para inicializarlo.

```

limpdesc()
{
    int x;
    for (x=0;x!=200;++x)
    {
        desc[x][0]='\0';
    }
    return(0);
}

```

```

//*****//
//***** LA FUNCION limptop () *****//
//*****//

```

// Limpia los arreglos topics[][], topbmp[][], top[][], fhelpp[][] asignándoles caracteres nulos ('\0') al primer elemento de // cada arreglo para inicializarlos.

```

limptop()
{
    int x;
    for (x=0;x!=100;++x)
    {
        topics[x][0]='\0';
        topbmp[x][0]='\0';
        top [x][0]='\0';
        fhelpp[x][0]='\0';
    }
}

```

```

return(0);
}

```

```

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX//
//XXXXXXXXXXXX LA FUNCION load_desc () XXXXXXXXXXXXXXX//
//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX//

```

```

// Toma el archivo DESC.HLP y carga el contenido del t3pico a tratar (de acuerdo a la variable pos) en el arreglo
// topics[][]. La b3squeda del t3pico la realiza con la ayuda del caracter ALT+219, ya que cada t3pico tiene este caracter
// tanto al inicio como al final del texto correspondiente. A medida que pasa de t3pico, la variable pos se incrementa en
// uno, hasta el momento en que pas y pos sean iguales, en este caso estamos en presencia del t3pico deseado.

```

```

load_desc (int pos)
{
int x,y,z,pass;
char ca;
FILE #file;
posd=0;
limpdsc ();
file = fopen ("desc.hlp","r");
if (file==NULL)
{
closegraph();
printf("Error abriendo archivo de descripcion ..!\n");
exit(0);
}
pass=0;
for (x=0;x!=2;)
{
ca=fgetc(file); // en la variable ca guarda el caracter del archivo que est3 siendo le3do.
if (ca=='&& pass==pos) // realiza la b3squeda del t3pico escogido por medio del caracter ALT-219 (3)
{ // cuando pass==pos, estamos en presencia del t3pico elegido.
y=0; // inicializa el contador de las filas del arreglo que guardar3 la descripci3n.
for (z=0;z!=300;++z)
{
ca = fgetc(file);
if (ca=='\n') // cuando ca es un ENTER, incrementa el contador de las filas del arreglo
{ // que va a guardar la descripci3n.
desc[y][z]='\0';
y++;
z=0;
}
if (ca==' ' || ca==EOF) // cuando la descripci3n termina, se inicializan los contadores de las
{ // filas y las columnas, para luego invocar a la funci3n encargada de
desc[y][z]='\0'; // mostrarlos en pantalla.
y=0;
z=0;
print_desc (topbmp[pos],topics[pos]);
fclose(file);
return(0);
}
desc[y][z]=ca; // cuando ca es un caracter de datos, se almacena en el arreglo que contendr3 la

```

```

    }
    } // descripción del tópico elegido.
}
if (ca=='\n') // analiza los caracteres del archivo desc.hlp.
{
    // cuando ca es '\n', nos encontramos en el inicio de un nuevo tópico.
    for (z=0;z!=3;)
    {
        ca=fgetc(file);
        if (ca==EOF) // si termina la lectura del archivo DESC.HLP y no se encuentra el tópico
        {
            // regresamos sin mostrar nada en la ventana de descripción.
            fclose(file); // se cierra el archivo.
            return(0);
        }
        if (ca=='\n') // se ha encontrado el final de un tópico, por lo cual se incrementa el contador
        {
            // pass, que es el que lleva el control del tópico en el que nos encontramos.
            pass++;
            break;
        }
    }
}
}
if (ca==EOF)
{
    print_desc(topbmp[pos],topics[pos]);
    break;
}
}
return(0);
}

```

```

//*****//
//***** LA FUNCION menu_top () *****//
//*****//

```

// Esta función se encarga de mostrarnos el menú de opciones receptando a la vez la opción que elija el usuario.
// La variable cur que le es pasada le indica a la función en que ventana se encontraba el usuario al momento de llamar
// al menú de opciones; es decir, en la ventana de tópicos disponibles o en la ventana de la descripción.

```

menu_top(int cur)
{
    int x,y,z=1,nw=1;
    void far *image2;
    unsigned size;
    limp_sombras1 (); limp_sombras2 (); // cuando se encuentra en el menú de opciones, se borran las sombras de las ventanas.
    size=imagesize(0,0,250,120);
    if ((image2 = farmalloc(size)) == NULL) // analiza si hay memoria para almacenar la porción de pantalla sobre la que se va
    {
        // a graficar el menú de opciones.
        closegraph();
        printf ("Error: not enough heap space in save_screen().\n");
        exit(1);
    }
    getimage(0,0,250,120,image2); // guarda la imagen en la variable image2.
    settxtstyle(0,0,1); setcolor(0);
    for (x=0;x!=3;)

```



```

{
  if (nw==1)
  {
    nw=0;
    putimage(0,0,image2,COPY_PUT);
    if (z==1; ;z==2)
    {
      setfillstyle(1,7); bar(15,51,170,110);
      setfillstyle(1,1); bar(5,41,160,100);
      setfillstyle(1,15); bar(8,44,157,97);
      if (z==1) // grafica una barra de color plomo claro si nos encontramos en esta opción.
        { // que es "Gráfico # 2 - F4".
          setfillstyle(1,7); bar(10,50,155,67);
        }
      if (z==2) // grafica una barra de color plomo claro si nos encontramos en esta opción.
        { // que es "Salir - F3".
          setfillstyle(1,7); bar(10,74,155,90);
        }
      outtextxy(18,55,"Gráfico # 2 - F4");
      outtextxy(18,80,"Salir - F3");
      setfillstyle(1,7); bar(5,21,80,37);
    }
    if (z==3) // grafica una barra de color plomo claro si nos encontramos en esta opción.
    { // que es "Ayuda-F1".
      setfillstyle(1,7); bar(91,21,170,37);
    }
    outtextxy(10,26,"Opciones");
    outtextxy(100,26,"Ayuda-F1");
  }
  // comienza la interacción con el usuario, ya que recepta la opción que éste elija.
  y=getch(); // y será la tecla que el usuario presione.
  if (y==75 && z==3) // si y es FLECHA-IZQ y z=3 se hace que z=1 para cambiarse a las opciones de la izquierda.
  { // además se hace nw=1 para indicar que se debe graficar los cambios.
    nw=1;
    z=1;
    continue;
  }
  if (y==77 && z==1) // si y es FLECHA-DER y z=1 se hace que z=3 para cambiarse a las opciones de la derecha.
  { // además se hace nw=1 para indicar que se debe graficar los cambios.
    nw=1;
    z=3;
    continue;
  }
  if (y==77 && z==2) // si y es FLECHA-DER y z=2 se hace que z=3 para cambiarse a las opciones de la derecha.
  { // además se hace nw=1 para indicar que se debe graficar los cambios.
    nw=1; z=3;
    continue;
  }
  if (y==72 && z==2) // si y es FLECHA-UP y z=2 se hace que z=1 para cambiarse a la opción superior de las opciones
  { // de la izquierda, además se hace nw=1 para indicar que se debe graficar los cambios.
    nw=1; z=1;
    continue;
  }
}

```

```

    }
    if (y==80 && z==1) // si y es FLECHA-DOWN y z=1 se hace que z=2 para cambiarse a la opcion inferior de las opciones
    {
        // de la izquierda, además se hace nw=1 para indicar que se debe graficar los cambios.
        nw=1; z=2;
        continue;
    }
    if ((y==13 || y==62) && z==1) // se selecciona la opción "Gráfico # 2"
    {
        new=1;
        loadbmp640(top[posc+posm],cur);
        break;
    }
    if ((y==59 || y==13) && z==3) // se selecciona la opción "Ayuda"
    {
        help(cur);
        break;
    }
    if (y==27) // se sale del menú de ayuda.
        break;
    if (y==13 && z==2) // se selecciona la opción salir del TUTORIAL.
    {
        new=1; flag=0;
        present();
    }
    if (y==61 && z==2) // se selecciona la opción salir del TUTORIAL
    {
        new=1; flag=0;
        present();
    }
}
putimage (0,0,image2,COPY_PUT); // al salir del menú de ayuda, colocamos nuevamente la imagen borrada por éste.
farfree (image2); se libera la memoria que fue ocupada.
return(cur);
}

```

```

//*****
//***** LA FUNCION help () *****
//*****

```

// Esta función se encarga de mostrarnos la ventana de ayuda en la cual se indica al usuario el uso de las teclas calientes // a usarse en la parte del tutorial.

```

help (int cur)
{
    char c;
    int y=0,z;
    unsigned size;
    void far *image3;
    limp_sombras1 (); // se limpia las sombras de las ventanas.
    limp_sombras2 ();
    size=imagesize(0,0,280,200);
    // se graba en memoria la imagen sobre la que se va a graficar la ventana de ayuda.
}

```

```

if ((image3 = farmalloc(size)) == NULL)
{
    closegraph();
    printf("Error: not enough heap space in save_screen().\n");
    exit(1);
}
getimage(350,50,630,250,image3);
setfillstyle (1,7);
bar (351,52,628,249);
setfillstyle (1,1);
bar (351,52,628,72);
setcolor (15);
settextstyle (0,0,1);
outtextxy (425,60,"MENU DE AYUDA");
load_help(); // se invoca a la función encargada de cargar el archivo HELP.HLP en un arreglo.
print_help(); // se invoca a la función encargada de mostrar el contenido del arreglo en pantalla.
for (z=0; z!=2;)
{
    y = getch (); // el programa espera que el usuario presione la tecla ESC para salir de esta opción.
    if (y==27) break;
    continue;
}
putimage (350,50,image3,COPY_PUT); // se grafica el gráfico guardado en memoria.
free (image3); // se libera la memoria.
if (cur==1) // si la posición de la sombra era en la ventana de tópicos disponibles.
    sombras1();
else
    sombras2(); // si la posición de la sombra era en la ventana de descripción del tópico.
return(0);
}

```

```

//#####
//##### LA FUNCION load_help () #####
//#####

```

// Esta función toma el contenido del archivo HELP.HLP y lo carga en el arreglo fhlp [][].

```

load_help()
{
    char ca; int z=0, x;
    FILE #file;
    file = fopen ("help.hlp","r"); // abre el archivo HELP.HLP para lectura.
    if (file==NULL)
    {
        clrscr();
        printf("\nError abriendo archivos..\n");
        exit(1);
    }
    x=0;
    for (z=0; z!=100; ++z)
    {
        ca=fgetc(file);

```

```

        if (ca==EOF)
        {
            fhhelp [x][z]='\0';
            break;
        }
        if (ca=='\n')
        {
            x++; z=-1;
            continue;
        }
        fhhelp [x][z]=ca; // almacena el contenido de HELP.HLP en el arreglo fhhelp.
    }
fclose(file);
return(0);
}

```

```

//*****//
//***** LA FUNCION print_help () *****//
//*****//

```

// Esta función toma el contenido del arreglo fhhelp [][] y lo imprime en la ventana del menú de ayuda.

```

print_help ()
{
    int x;
    setcolor(0);
    settxtstyle(0,0,1);
    for (x=0; x!=15; ++x)
    {
        if (fhhelp[x][0] == EOF)
            break;
        outtextxy (360,(90+(x*10)),fhhelp [x]);
    }
    return (0);
}

```

```

//*****//
//***** LA FUNCION no_graph () *****//
//*****//

```

// Esta función nos muestra un mensaje en la pantalla cuando un tópico no tiene gráfico grande

```

no_graph(int cur)
{
    int y,z;
    unsigned size;
    void far *image4;
    limp_sombras1 ();
    limp_sombras2 ();
    size=imagesize(0,0,200,40);
    /* get byte size of image */
}

```



```

if ((image4 = farmalloc(size)) == NULL)
{
    closegraph();
    printf("Error: not enough heap space in save_screen().\n");
    exit(1);
}
getimage(220,200,420,240,image4);
// comienza a graficar la ventana donde muestra el mensaje.
setfillstyle (1,3);
bar (220,200,420,240);
setcolor (1);
rectangle(220,200,420,240);
rectangle(221,201,419,239);
rectangle(222,202,418,238);
setcolor (1);
settextstyle (0,0,1);
outtextxy (254,208,"OPCION NO VALIDA");
outtextxy (254,225,"PARA ESTE TOPICO");
for (z=0; z!=2;)
{
    y = getch (); // espera a que el usuario presione la tecla ESC, para salir de esta opción.
    if (y==27) break;
    continue;
}
putimage (220,200,image4,COPY_PUT);
free (image4);
if (cur==1)
    sombras1();
else
    sombras2();
return(0);
}

```

GUÍA DE INSTALACION DEL PROGRAMA

El programa consta de 3 directorios de 3.5" HD, en cada directorio se encuentran almacenados todos los archivos correspondientes al programa; es decir, se encuentran el archivo de configuración de los archivos de texto, los archivos gráficos, etc. Para poder instalarlo, deberá crear un directorio en cada unidad de la equiva en la que desea instalarlo. Luego deberá copiar el contenido de estos directorios al directorio creado. La ejecución de la instalación se va a realizar el programa, debe tener un directorio de datos VGA, ya que el programa fue desarrollado para ser ejecutado en un PC.

En la instalación no se copia todos los archivos del programa, el programa no funciona correctamente, o para la parte de la ejecución al momento de ejecutarlo según fuera el caso.

Se va a instalar sobre el disco duro de la computadora. El directorio REDEI para conservar la configuración del programa, el archivo de configuración es REDEI. Este directorio debe ser creado en algún directorio, el nombre de este directorio puede ser cualquier cosa, ya que el programa no ejecuta ninguna operación de lecturas sobre los archivos de configuración. El programa de instalación se ejecutará sobre el directorio creado.

A P E N D I C E B

GUÍA DE INSTALACION DEL PROGRAMA

GUIA DE INSTALACION DEL PROGRAMA

El programa consta de 3 diskettes de 3.5" HD, en cuyo interior se encuentran almacenados todos los archivos necesarios para correr el programa; es decir, se encuentran el archivo ejecutable, los archivos de texto, los archivos gráficos, etc. Para mejor funcionamiento Ud. deberá crear un directorio sobre el disco duro de la máquina en la que desea instalarlo, luego deberá copiar el contenido de éstos diskettes al directorio creado. La máquina en la cual se va a instalar el programa, debe tener una interfase de video VGA, ya que el programa fue desarrollado para este tipo de video.

Si en la instalación no se copian todos los archivos mencionados, el programa no funcionará adecuadamente, o peor no podrá correrse indicando el mensaje apropiado según fuere el caso.

Una vez instalado sobre el disco duro de la máquina, deberá tipearse REDES para comenzar la ejecución del mismo (ya que el archivo ejecutable es REDES.EXE), si desea cambiarle el nombre no hay ningún problema, el problema sería cambiarle el nombre a los demás archivos, ya que durante la ejecución del mismo se realiza operaciones de lecturas sobre los archivos de texto y archivos gráficos, ocasionándose problemas al no encontrar el archivo con el nombre original.

Una vez dentro del programa, el mismo le indica lo que debe realizar, ya que consta de menús de ayuda que le indican las funciones de las teclas a emplear haciendo fácil el manejo del programa.