



UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

7  
2ij

## FACULTAD DE INGENIERIA

Desarrollo de un sistema  
tutorial para apoyo a la  
materia de redes de  
computadoras.

# T E S I S

Que para obtener el título de:  
Ingeniero en Computación

PRESENTA:

Alejandro Alvarado Azpeitia

Director de Tesis:

Ing. Gabriel Castillo Hernández



México D. F.  
TESIS CON  
FALLA DE ORIGEN

1996

TESIS CON  
FALLA DE ORIGEN



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

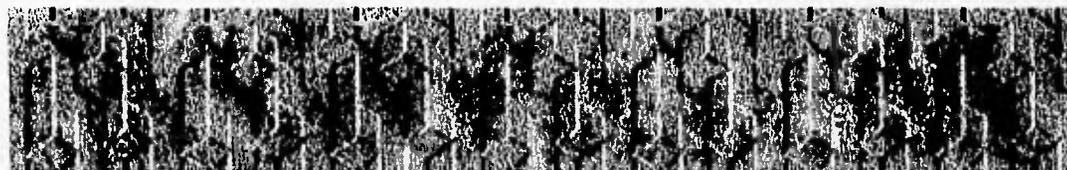
**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

**TESIS**

**COMPLETA**



**DESARROLLO DE UN SISTEMA  
TUTORIAL PARA APOYO A LA  
MATERIA DE REDES DE  
COMPUTADORAS.**



# INDICE

## AGRADECIMIENTOS INTRODUCCION

1

## ANTECEDENTES

### CAPITULO 1: COMPUTACION EN LA EDUCACION.

1.1	Software educativo.	1
1.2	Interfaces para el usuario.	2
1.3	Hipertexto y animación para el aprendizaje.	7

### CAPITULO 2: AMBIENTE DE WINDOWS.

2.1	Microsoft Windows.	11
2.1.1	¿ Qué es Windows ?	11
2.1.2	Windows vs. MS-DOS.	12
2.1.3	Elementos de una aplicación.	14
2.1.4	Modos de operación.	21
2.1.5	Requerimientos de hardware y software.	22
2.2	Coceptos de programación en Windows.	23
2.2.1	Programación por eventos.	23
2.2.2	Objetos y programación de objetos.	24
2.2.3	Manipuladores.	24
2.2.4	Instancias.	24
2.2.5	Ciclos del mensaje.	25
2.2.6	Multitarea.	26
2.2.7	Administración de memoria.	26
2.2.8	Recursos.	27
2.2.9	Componentes de una aplicación.	27

## ANALISIS

### CAPITULO 3: ANALISIS DE HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES EN WINDOWS.

3.1	Herramientas para el desarrollo de aplicaciones en Windows.	31
3.1.1	Desarrollo de aplicaciones para usuarios.	31
3.1.2	Constructores de interfases.	33
3.1.3	¿ Porqué usar herramientas de ayuda para REDTUT ?	36
3.2	Manejo de imágenes en Windows.	37
3.2.1	Formato de imágenes.	37
3.2.2	Imágenes con herramientas de ayuda.	40
3.2.3	¿ Porqué usar formato .BMP para las imágenes de REDTUT ?	42
3.3	Técnicas de animación en ambiente Windows.	43
3.3.1	Implementación y control de animación.	43
3.3.2	Algoritmos para animación.	45
3.3.3	Animación en tiempo real.	47
3.3.4	Animación por cuadros de imágenes.	47
3.3.5	¿ Qué técnica debe utilizarse en la animación de REDTUT.	49

**CAPITULO 4: DISEÑO DE REDTUT.**

<b>4.1</b>	<b>Diseño de REDTUT con herramientas de ayuda en Windows.</b>	<b>51</b>
4.1.1	Recursos para el manejo de información.	51
4.1.2	Tipos de acceso a la información.	52
4.1.3	Tópicos de información.	55
4.1.4	Distribución de la pantalla.	59
<b>4.2</b>	<b>Creación de texto, imágenes y animación.</b>	<b>60</b>
4.2.1	Creación de texto.	60
4.2.2	Creación de imágenes.	67
4.2.3	Creación de animación.	70
<b>4.3</b>	<b>Diseño de la navegación.</b>	<b>83</b>
4.3.1	Navegación por temas.	83
4.3.2	Navegación de la evaluación.	84
4.3.3	Navegación por índice.	85
4.3.4	Navegación por búsqueda.	86
<b>4.4</b>	<b>Creación de ayuda en línea y glosario.</b>	<b>86</b>
4.4.1	Ayuda en línea.	86
4.4.2	Glosario.	87

**CAPITULO 5: INTEGRACION DE REDTUT.**

<b>5.1</b>	<b>Integración de texto, imágenes y animación.</b>	<b>89</b>
5.1.1	Integración de imágenes al texto.	89
5.1.2	Integración de animación al texto.	92
<b>5.2</b>	<b>Compilación de REDTUT.</b>	<b>93</b>
5.2.1	Generación de archivos (.HLP).	93
5.2.2	Generación del archivo ejecutable (.EXE).	96

**RESULTADOS****CAPITULO 6: EVALUACION DE FUNCIONALIDAD Y MECANICA DE ACTUALIZACION DE REDTUT.**

<b>6.1</b>	<b>Evaluación de funcionalidad de REDTUT.</b>	<b>103</b>
6.1.1	Navegación por temas.	103
6.1.2	Navegación por índice.	104
6.1.3	Navegación por búsqueda.	105
6.1.4	Barra de menú.	106
6.1.5	Barra de botones.	107
6.1.6	Ayuda en línea.	107
6.1.7	Glosario.	108
6.1.8	Animación.	109
6.1.9	Flexibilidad, transportabilidad y efectividad.	110
<b>6.2</b>	<b>Mecánica de actualización de REDTUT.</b>	<b>111</b>
6.2.1	Actualización de archivos .HLP.	111
6.2.2	Actualización de archivos .EXE.	112

**COMENTARIOS FINALES****113**

## APENDICES

<b>APENDICE A:</b>	<b>EQUEMAS DE LA NAVEGACIÓN POR TEMAS EN REDTUT.</b>	I
<b>APENDICE B:</b>	<b>MACRO-FUNCIONES EN SISTEMAS DE AYUDA EN WINDOWS.</b>	VI
<b>APENDICE C:</b>	<b>FUNCION WINHELP.</b>	IX
<b>APENDICE C:</b>	<b>REDTUT</b>	
<b>Capítulo 1:</b>	<b>Comunicación de datos.</b>	XI
<b>Capítulo 2:</b>	<b>Redes de comunicación de datos.</b>	XXVIII
<b>Capítulo 3:</b>	<b>Arquitecturas de redes.</b>	XLVII
<b>Capítulo 4:</b>	<b>Diseño de redes.</b>	LXV
<b>BIBLIOGRAFIA</b>		LXXIX

# AGRADECIMIENTOS

*A Dios que siempre me fortaleció en los momentos difíciles de de mi carrera y me motiva ha ofrecer mis conocimientos como un servicio a los demás.*

*A mis padres que me guiaron por el camino que me diera una formación profesional y por su apoyo para que pudiera lograr una de mis metas.*

*A Irma, Martín, Rocío, Israel y Luis, que me han apoyado de diversas maneras para terminar esta tesis.*

*A la UNAM y la Facultad de Ingeniería por haber contribuido en gran parte de mi formación como persona, al brindarme conocimientos y experiencias que me han permitido valorar el papel que desempeñan dentro del país.*

*Al Ing. Gabriel Castillo Hernández que nunca me negó su tiempo y dedicación para que esta tesis cumpliera sus objetivos.*

*Al D.G Martín Alvarado por haberme dedicado tiempo y sus conocimientos en el diseño gráfico del sistema tutorial.*

*A mis amigos del grupo "Genezalet" que me animan a superarme en todos los aspectos de mi vida.*

*A mis compañeros del Laboratorio de Sistemas de Información que han aportado sus opiniones para que este tesis sea más funcional.*

# INTRODUCCION

Actualmente las computadoras se han convertido en las más nuevas y versátiles herramientas para la enseñanza en todos los niveles educativos. Sin embargo, su empleo se ha concretado específicamente como un apoyo en la educación, dado que todavía no han podido sustituir el factor humano en el proceso enseñanza-aprendizaje.

Generalmente el uso de las computadoras en la enseñanza, caen en el concepto de "Instrucción Asistida por Computadora" (CAI, *Computer Assisted Instruction*). CAI es un método de enseñanza o instrucción, individual o de grupo, donde un programa de computadora instruye a los estudiantes a través de un diálogo interactivo. Estos diálogos, cuya principal tarea es enseñar se clasifican generalmente como:

1. Programas del tipo "ejercita-y-practica" (*drill-and-practice*)
2. Simulaciones
3. Tutoriales

Los programas del tipo "ejercita-y-practica" se usan para reforzar lecciones pasadas en vez de enseñar lecciones nuevas. Estos programas permiten que el estudiante aprenda a su propio ritmo, al repetir una idea o concepto en orden para que el estudiante lo memorice. El programa hace preguntas y examina al estudiante sobre un tema particular hasta que el estudiante memoriza el material.

Las simulaciones son programas que imitan los principios, causas y efectos encontrados en la realidad. Las simulaciones proporcionan aprendizaje en situaciones donde se desea desafiar la habilidad del pensamiento del estudiante; son usadas para enseñar a los estudiantes como funcionan las cosas. Dado que las simulaciones crean un modelo de como algunas cosas funcionan, los estudiantes pueden cambiar las variables de un programa para ver como reacciona.

Los tutoriales son programas que intentan impartir y enseñar activamente principios y teorías. Un tutorial es un método de enseñanza por el cual un concepto es explicado al estudiante; y luego, son elaboradas preguntas acerca del tema para examinar su aprovechamiento. El progreso del estudiante puede ser automáticamente monitoreado conforme las respuestas son dadas y revisadas con exactitud. Los tutoriales computarizados son usualmente empleados para la educación de adultos.

Independientemente del tipo de programa que se elija para el proceso de enseñanza-aprendizaje por computadora, es necesario disponer de una interface de usuario que cumpla con las características principales que permitan que el usuario sólo preste atención al programa de enseñanza, y no pierda tiempo en tratar de entender como interactuar con él.

El futuro de las computadoras en la educación depende del desarrollo de los modelos de aprendizaje. Sin embargo, la aplicación de las computadoras en el campo educacional seguirá desarrollándose en la medida en que vayan surgiendo necesidades que requieran de soluciones atractivas proporcionadas por computadoras.

La idea de desarrollar un sistema tutorial de redes de computadoras surge debido a que las limitaciones de presupuesto y espacio que sufre la UNAM, en la Facultad de Ingeniería, limitan a que la materia de redes de computadoras, siendo una materia eminentemente práctica, se imparta básicamente en forma conceptual. Por lo que un sistema tutorial que ayude al alumno a entender mejor los conceptos y a visualizar los fenómenos asociados a la materia es recomendable. Dada esta necesidad se determina el siguiente objetivo:

*Diseñar y desarrollar un sistema que permita a los alumnos de la materia de redes de computadoras reafirmar y actualizar sus conocimientos. El sistema deberá ser una interface gráfica que permita al usuario interactuar con él y ejemplificar los conceptos estudiados. El sistema se enfocará específicamente a la materia de "Redes de computadoras".*

Esta tesis está dividida en cinco partes principales que contienen los capítulos que muestran el desarrollo gradual del sistema tutorial de redes de computadoras (REDTUT) e información adicional que lo complementan.

La **parte I** corresponde a la parte de "**Antecedentes**" y esta integrada por el **capítulo 1: "Computación en la educación"** y el **capítulo 2: "Ambiente de Windows"**, los cuales son necesarios para conocer las características que debe adoptar el sistema dentro del ambiente en el que se desarrollará.

La **parte II** corresponde a la parte de "**Análisis**" que contiene el **capítulo 3: "Análisis de herramientas para el desarrollo de aplicaciones en Windows"** y presenta un análisis de las herramientas que más se acercan a la solución del desarrollo del sistema tutorial.

La **parte III** corresponde a la parte de "**Desarrollo**" y esta formada por el **capítulo 4: "Diseño de REDTUT"** y el **capítulo 5: "Integración de REDTUT"**, en los cuales se explica el desarrollo del sistema tutorial, una vez elegidas las herramientas más apropiadas para sus fines.

La **parte IV** trata los "**Resultados**" obtenidos y contiene el **capítulo 6: "Evaluación de funcionalidad y mecánica de actualización de REDTUT"**, en donde se muestran las características, ventajas y desventajas del sistema terminado y la forma como es posible mantener el sistema actualizado.

La **parte V** es la parte de "**Apéndices**" y contiene información adicional que es importante para el desarrollo de sistemas de ayuda en Windows, así como una copia del texto contenido en el sistema tutorial (REDTUT) relacionado con el curso de "Redes de computadoras", el cual puede servirle al lector como texto de consulta, en caso de que no se disponga de la copia del sistema en discos.

**P A R T E I**  
**A N T E C E D E N T E S**

# **CAPITULO 1**

## **COMPUTACION EN LA EDUCACION**

**1.1 Software educativo.**

**1.2 Interfaces para el usuario.**

**1.3 Hipertexto y animación en el aprendizaje.**

## 1.1 Software educativo

El software de computadoras personales se presenta en muy diversas formas, al igual que el número de aplicaciones. Es por esto, que es necesario conocer como son categorizados los programas, según la función que desempeñan. Una de las clasificaciones es la siguiente:

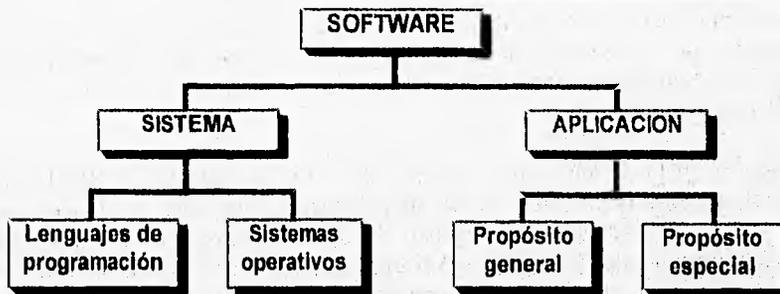


Figura 1-1 Clasificación del software para computadoras personales.

Como se muestra en la figura 1-1, un software de aplicación desempeña funciones no relacionadas con el sistema de la computadora en sí. La mayoría de las aplicaciones que existen pueden clasificarse dentro de las dos categorías:

1. De propósito general
2. De propósito especial

Los *programas de propósito general* son adaptados a una amplia variedad de tareas, como procesamiento de palabras, operaciones en hojas de cálculo, elaboración de gráficas, comunicaciones, etc. En éste caso el usuario define el problema y el procedimiento y usa el software que pueda solucionarlo.

Los *programas de propósito especial* se dedican a desempeñar tareas específicas como de educación, de contaduría, de análisis estadístico, etc. En éste caso los programas contienen características, para resolver problemas, especialmente diseñadas para esas tareas particulares.

Dentro de las aplicaciones de propósito especial se encuentra el *software educativo*, que se refiere a los programas cuya tarea principal es educar a los usuarios, ya sea aportándoles nuevos conocimientos, o bien entrenándolos en materias de cualquier disciplina que comúnmente son estudiadas por procedimientos tradicionales.

Dentro de los programas educativos encontramos una clasificación de tres tipos; principalmente:

1. Programas de autoestudio
2. Programas de apoyo
3. Programas de educación informática

Los *programas de autoestudio* son aquellos que permiten al usuario adquirir conocimientos de una determinada materia de forma autónoma. Para conseguir que un paquete educativo pueda ser catalogado como de autoestudio, la exposición debe ser transparente, ya que el usuario no contará con el apoyo de un profesor. Los programas de autoestudio deberán estar acompañados con documentación apropiada para cumplir el objetivo que se persigue.

En un programa de autoestudio básicamente deben contemplarse 3 puntos (Fig. 1-2):



Figura 1-2 Componentes básicos de un programa de autoestudio.

1. **Introducción.-** Se deben especificar los objetivos, un índice, el tiempo necesario aproximado para cubrir el capítulo y cualquier aclaración que pueda ser interesante para el usuario.
2. **Exposición.-** Después de la introducción el programa debe ofrecer al usuario de forma interactiva, la posibilidad que elija uno de los capítulos. La exposición debe estar basada, fundamentalmente en textos resumidos con abundantes imágenes gráficas.
3. **Pruebas de control.-** Una vez terminada la exposición de un capítulo, el programa interrogará al usuario para que él mismo llegue a decidir si sus conocimientos son suficientes como para pasar a otro capítulo. Las preguntas de control deben ser elaboradas utilizando procedimientos pseudoaleatorios para no repetir siempre las mismas cuestiones.

Los *programas de apoyo* sirven como complemento a las exposiciones realizadas por un profesor humano. Por lo tanto no es imprescindible que su documentación y sencillez de manejo sean tan rigurosas como en el caso de los programas de autoestudio. El objetivo principal de estos programas radica en complementar la labor del profesor. Generalmente suelen hacer especial hincapié en ejercicios prácticos que sirven para aclarar las dudas que puedan presentarse sobre las lecciones teóricas.

Los *programas de educación informática* pueden ser de tipo autoestudio o de apoyo, pero siempre enfocados al estudio de la informática. Una de sus principales características es que el ordenador se utiliza a sí mismo, tanto para realizar exposiciones teóricas como para ejercicios prácticos.

Generalmente el software educativo contribuye a racionalizar la adquisición de conocimientos, forzando a que el estudiante razone y no memorice. Adicionalmente, aportan al usuario cierta formación en informática.

## **1.2 Interfaces para el usuario**

La tecnología de la computación cada día está más disponible a más gente en una gran variedad de dispositivos y contextos, es por esto que se ha incrementado la necesidad de hacer interfaces más accesibles y fáciles de manejar.

En términos generales una *interface* es el hardware y software a través del cual un humano y una computadora pueden comunicarse. Sin embargo, en la actualidad este término ha sido relacionado, frecuentemente, con la presentación de la pantalla de un programa, que es responsable de permitir al usuario la comunicación y control de la computadora.

Una *interface* para el usuario tiene sus propias formas de comunicación con el usuario, así como diversos caminos para ejecutar los comandos. Las interfaces para usuario pueden clasificarse como sigue:

1. Interfaces manejadas por comandos (Command-driven Interfaces)
2. Interfaces manejadas por menús (Menu-driven Interfaces)
3. Interfaces gráficas de usuario (GUI's, Graphical User Interfaces)
4. Interfaces de lenguaje natural
5. Interfaces de manipulación directa
6. Interfaces de llenado de forma (form-filling)

### ***Interfaces manejadas por comandos***

Solicitan o incitan al usuario a teclear un comando o una cadena de comandos de palabras sintácticamente correctas, que posteriormente son trasladadas como una instrucción con el fin de ejecutar

alguna función del programa. Los comandos típicos empiezan con un verbo que es seguido por una o más palabras llamadas argumentos, que tienen la función de completar el comando.

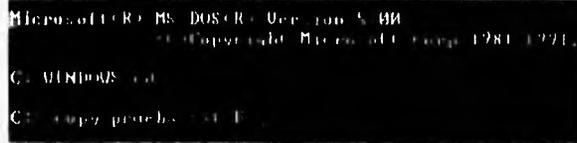


Figura 1-3 Ejemplo de una interface manejada por comandos.

Los programas que más frecuentemente usan este tipo de interfaces son los sistemas operativos (Fig. 1-3 MS-DOS, UNIX, etc.). Una característica distintiva de este tipo de interfaces es que la mayoría de las veces no proporcionan un soporte explícito para que el usuario conozca el conjunto de comandos disponibles. Esto obliga a que el usuario conozca o aprenda de memoria los comandos ofrecidos por la interface.

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>◆ Rápida</li> <li>◆ Eficiente</li> <li>◆ Precisa</li> <li>◆ Concisa</li> <li>◆ Flexible</li> <li>◆ Iniciada por el usuario</li> <li>◆ Atractiva</li> </ul>	<ul style="list-style-type: none"> <li>◆ Requiere Instrucción larga</li> <li>◆ Necesita de un uso regular</li> <li>◆ Se carga en memoria alta</li> <li>◆ Manejo de errores deficiente</li> </ul>

Tabla 1-2 Ventajas y desventajas de las interfaces manejadas por comandos.

### Interfaces manejadas por menús

Ofrecen al usuario una lista de opciones, llamada menú. Por este camino, el usuario puede escoger diferentes comandos o funciones, para la ejecución de una tarea (Fig.1-4).

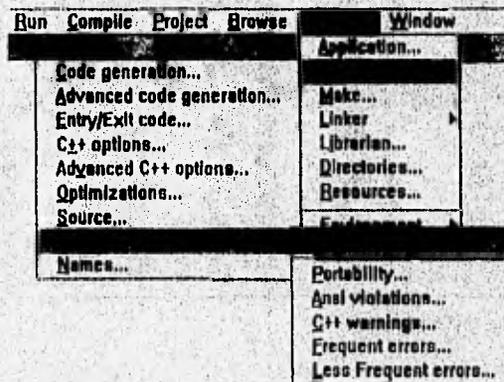


Figura 1-4 Ejemplo de una interface manejada por menús.

Para elegir la estructura de esta interface se requiere conocer las diversas técnicas de menús. La técnica más usada es la de estructura de árbol, que permite la selección de un gran número de opciones que

están distribuidas en los diferentes menús proporcionados en un flujo de niveles jerárquicos. Existen otras técnicas como:

- ◆ Menús Pull-down (despliegue hacia abajo)
- ◆ Menús Pop-up (instantáneos)
- ◆ Menús scrollables (desplazables)

Las interfaces de manejo de menús, son generalmente más fácil de aprender que las interfaces de manejo de comandos.

<b>Ventajas</b>	<b>Desventajas</b>
<ul style="list-style-type: none"> <li>◆ Mínima necesidad de teclear</li> <li>◆ Se carga en memoria baja</li> <li>◆ Estructura bien definida</li> <li>◆ Diseño de software sencillo</li> <li>◆ Disponible para herramientas CAD</li> </ul>	<ul style="list-style-type: none"> <li>◆ Algunas veces lenta</li> <li>◆ Consume espacio en la pantalla</li> <li>◆ No apropiada para entrada de datos</li> <li>◆ No apropiada para diálogos iniciados por el usuario</li> </ul>

Tabla 1-2 Ventajas y desventajas de las interfaces manejadas por menús.

**Interfaces gráficas de usuario**

Estas interfaces usan dibujos y símbolos gráficos para representar comandos, opciones o acciones. Frecuentemente se presentan al usuario con la representación gráfica de un escritorio: Iconos, mapas de bits para manejo de archivos, discos y programas, etc. Además, estas interfaces proporcionan el método de ventanas (método para dividir la pantalla dentro de varios rectángulos de tamaño variable para ver dos o más aplicaciones al mismo tiempo Fig.1-5).

Los dos ejemplos más conocidos de interfaces gráficas de usuario son: la de Macintosh y la de Microsoft Windows para computadoras basadas en MS-DOS.

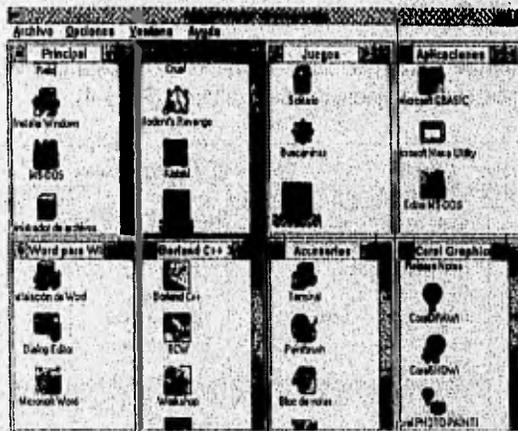


Figura 1-5 Ejemplo de una interface gráfica de usuario

**Interfaces de lenguaje natural**

Permiten al usuario introducir frases simples de algún idioma usual (ingles, por ejemplo), que son trasladadas dentro de comandos o acciones por el programa de aplicación. Estas interfaces se usan

principalmente donde no es práctico entrenar gente en un lenguaje de comandos más formal o como interfaces antes de que la gente empiece a usar una computadora.

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>◆ No requiere de una sintaxis especial</li> <li>◆ Flexible</li> <li>◆ Potente</li> <li>◆ Natural</li> <li>◆ Iniciativa mezclada</li> </ul>	<ul style="list-style-type: none"> <li>◆ Ambiguo</li> <li>◆ Impreciso</li> <li>◆ Verboso</li> <li>◆ Opaco</li> <li>◆ Diseño de software complejo</li> <li>◆ Ineficiente</li> </ul>

Tabla 1-3 Ventajas y desventajas de las interfaces de lenguaje natural.

**Interfaces de manipulación directa**

La característica esencial de este tipo de interface, es que ciertas clases de representación directa son presentadas al usuario por el sistema, cuando la operación o comando deseado se activa por manipulación directa, personificando en la pantalla una realidad virtual.

Estos tipos de interfaces se usan para el control de procesos, control de tráfico aéreo, simuladores, video-juegos, diseño de sistemas asistido por computadora (CAD), etc.

En la manipulación directa, los elementos sobre la pantalla se comportan como si ellos fueran los objetos que ellos representan. Las características que distinguen la manipulación directa están obligadas a ser:

1. Representaciones continuas de objetos de interés.
2. Acciones físicas en lugar de sintaxis complicadas.
3. Operaciones rápidas, incrementables, reversibles con retroalimentación visual inmediata.

La manipulación directa es fácil de aprender para usuario nuevos, fácil de operar para usuarios intermedios y fácil de extender para expertos.

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>◆ Analogía de tareas</li> <li>◆ Reduce el proceso de familiarización y aprendizaje</li> <li>◆ Fácil de retener</li> <li>◆ Fomenta la exploración</li> <li>◆ Visualmente atractiva</li> <li>◆ Potente</li> <li>◆ Concisa</li> <li>◆ Herramientas de diseño disponibles</li> </ul>	<ul style="list-style-type: none"> <li>◆ Puede requerir software complejo o largo.</li> <li>◆ Puede requerir el despliegue de gráficas de alta resolución</li> <li>◆ Puede requerir de dispositivos de entrada auxiliares</li> <li>◆ Requiere del diseño de gráficas especializadas</li> </ul>

Tabla 1-4 Ventajas y desventajas de las interfaces de manipulación directa.

**Interfaces de llenado de forma**

Estas interfaces son comúnmente usadas debido a que el humano esta inherentemente familiarizado con el concepto de llenado de formas. En estas interfaces toda la información en el registro es simultáneamente visible, dando al usuario la impresión de control sobre el diálogo (Fig.1-6).

Figura 1-6 Ejemplo de una interface de llenado de forma.

Estas interfaces usan: protección de pantalla, campos de datos variables en la pantalla, campos opcionales, entradas por omisión, ayuda, navegación del cursor en la pantalla, capacidad de corregir errores, etc.

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>◆ Es un concepto familiar</li> <li>◆ Entrada de datos simplificada</li> <li>◆ Requiere de entrenamiento limitado</li> <li>◆ Se carga en memoria baja</li> <li>◆ Estructura bien definida</li> <li>◆ Diseño de software sencillo</li> <li>◆ Herramientas de diseño de diálogo ampliamente usadas</li> </ul>	<ul style="list-style-type: none"> <li>◆ Algunas veces lenta</li> <li>◆ Consume espacio en la pantalla</li> <li>◆ No ideal para selección de comandos</li> <li>◆ Requiere de control del cursor en la pantalla</li> <li>◆ Mecanismo de navegación no explícita</li> </ul>

Tabla 1-5 Ventajas y desventajas de las interfaces de llenado de forma.

Dentro de todas las categorías de interfaces posibles, la interface de usuario gráfica es considerada la más amigable para el usuario, por la facilidad que proporciona para aprender y usar.

Independientemente de la interface que se utilice, generalmente se debe cumplir con las siguientes metas:

**Metas de la Ingeniería del sistema:**

- ◆ Funcionalidad apropiada.
- ◆ Rentabilidad, disponibilidad, seguridad e integridad de los datos.
- ◆ Estandarización, Integración, consistencia y portabilidad.

**Metas del diseño del factor humano:**

- ◆ Tiempo de aprendizaje.
- ◆ Velocidad de ejecución.
- ◆ Rango de errores por el usuario.
- ◆ Satisfacción del tema.

### 1.3 Hipertexto y animación en el aprendizaje

#### HIPERTEXTO

El término *hipertexto* es usado para definir una técnica para organizar información textual de una forma no secuencial y no lineal, con el fin de facilitar el acceso rápido de grandes cantidades de información. En otras palabras, puede decirse que hipertexto es una red de nodos (documentos, archivos, artículos, tarjetas, páginas, etc) que contienen información (usualmente texto) que está conectada por ligas (puntos de referencias, menciones). En el caso de que la información consista de gráficos, sonido y video, la técnica recibe el nombre de "Hipermedia". En el caso particular de que la información sean gráficos recibe el nombre de "Hipergráficos".

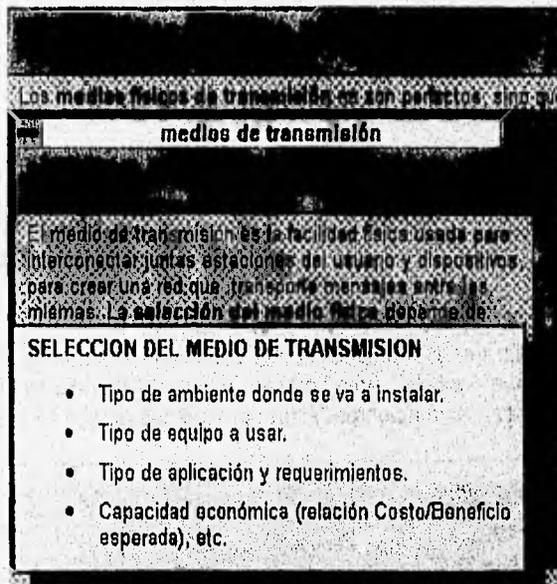


Figura 1-7 Uso de hipertexto en una aplicación de Windows.

Lo que hace diferente al hipertexto con respecto a otras formas de organización de información tradicional (en papel por ejemplo), es que las ligas entre nodos son soportadas por una máquina. Cuando el lector selecciona una liga de hipertexto, el "movimiento" entre los dos nodos toma lugar automáticamente. La importancia del hipertexto reside en el hecho de que puede alterar el camino en el cual leemos, escribimos y organizamos información (Fig. 1-7).

El hipertexto se puede presentar de diversas formas para diferentes aplicaciones (Tabla 1-6). En el caso de los sistemas cuya tarea principal es la presentación de información, el hipertexto es usado con el fin de proporcionar una interface uniforme. Los documentos de hipertexto soportados por computadoras son extensibles y editables, es decir, el usuario puede agregar nuevas ligas, agregar nuevos nodos, o cambiar el contenido de los nodos existentes, sin alterar la integridad del documento.

APLICACIONES DE HIPERTEXTO E HIPERMEDIA		
Oficinas	Recursos informativos	Aprendizaje personal
<ul style="list-style-type: none"> <li>◆ Catálogos de productos</li> <li>◆ Reportes anuales</li> <li>◆ Resumen y biografías</li> <li>◆ Nuevas revistas</li> <li>◆ Documentación Software</li> <li>◆ Manuales de políticas</li> </ul>	<ul style="list-style-type: none"> <li>◆ Enciclopedias, glosarios y diccionarios</li> <li>◆ Libros de referencia legal y médico.</li> <li>◆ Anotaciones literarias</li> <li>◆ Gulas de viaje y restaurantes</li> </ul>	<ul style="list-style-type: none"> <li>◆ Instrucción y exploración</li> <li>◆ Reparación y mantenimiento de manuales</li> <li>◆ Mapas geográficos</li> <li>◆ Documentación técnica y ayudas en línea</li> <li>◆ Libros de cocina y reparación del hogar</li> <li>◆ Hipernovelas, Hiperpoemas</li> </ul>

Tabla 1-6 Algunas aplicaciones que se le dan al hipertexto y a la hipermedia.

Dadas las características funcionales del hipertexto, su impacto en el campo de la educación se ha venido incrementando, dado a que no solo ofrece diferentes técnicas de presentación de información, sino también un estilo que fomenta la cualidad de pensar. Los sistemas con hipertexto han sido considerados como la base de los sistemas educacionales.

El hipertexto ofrece un ambiente de información basado en computadora el cual permite el soporte de todas las actividades relacionadas con la manipulación de la información.

### ANIMACION

La animación por computadora es la técnica de crear ilusión de movimiento al rotar, trasladar o cambiar de posición dibujos estáticos en la pantalla. Animación no es el arte de dibujos que se mueven, sino el arte de movimientos que son dibujados. Animación es "el arte gráfico que ocurre en el tiempo" (Fig. 1-8).



Figura 1-8 La animación puede producirse por movimiento de un dibujo.

La animación por computadora debe ser usada para comunicar y no para decorar. Debe usarse para transmitir cambios, movimiento y progresiones. También puede ser usada para entretenimiento, para relato de historias, como diversión, como identificación, para instrucción y enseñanza, etc. Específicamente, la animación es usada para:

- ◆ Explicar un dispositivo mecánico complejo con muchas partes en movimiento
- ◆ Mostrar incrementos, decrementos o cambios en algunas variables físicas
- ◆ Ayudar al usuario a reconocer objetos sólidos por un poco de analogía alrededor o detrás de ellos
- ◆ Simular condiciones de variación
- ◆ Mostrar procesos y procedimientos
- ◆ Ilustrar software dinámico, como algoritmos, protocolos de comunicación de datos y flujos a través de las estructuras y ciclos del programa
- ◆ Analizar procesos
- ◆ Mostrar, mas que informar que hace un comando
- ◆ Demostrar como opera un dispositivo

La animación puede ser especificada en diferentes caminos, y las computadoras pueden generar animación usando una variedad de técnicas. Cada una de esas técnicas tiene una aplicación particular. La combinación exacta del soporte de hardware y la habilidad del software, determinan un indicador significativo de la calidad y el volumen de animación por computadora, que un ambiente particular puede soportar.

El efecto que tiene la animación por computadora en el campo de la educación depende de la función que desempeñe. La animación por computadora es un medio de comunicación muy interesante, comparado con los métodos estáticos más comúnmente empleados que contienen texto y dibujos.

El uso de la animación en programas diseñados para el aprendizaje, como en el caso particular de tutoriales, tienen las siguientes ventajas:

- ◆ Retran secuencias de pasos complejas e involucran situaciones más reales y convincentes que una explicación meramente textual.
- ◆ Mantienen el interés del usuario por ser animado y divertido.

# **CAPITULO 2**

## **AMBIENTE DE MICROSOFT WINDOWS**

**2.1 Microsoft Windows.**

**2.2 Conceptos de programación en Windows**

## 2.1 Microsoft Windows.

### 2.1.1 ¿ Qué es Windows ?

Microsoft Windows es un entorno operativo que funciona con MS-DOS. Es un entorno de ventanas multitarea basado en gráficos.

Windows reúne el control del tipo apuntar y seleccionar menús instantáneos y la habilidad de ejecutar aplicaciones escritas especialmente para Windows, así como las aplicaciones estándar que funcionan con MS-DOS.

Para el desarrollo de aplicaciones, Windows ofrece abundantes subrutinas incorporadas que permiten la fácil implementación de menús instantáneos, barras de desplazamiento, cuadros de diálogo, íconos y otras muchas características de un interfaz gráfico de fácil manejo para el usuario. También ofrece al programador el tratamiento de la salida de video, el teclado, el "mouse", la impresora, el puerto serie y los temporizadores del sistema de forma independiente de los dispositivos. Esto hace posible que un mismo programa de aplicación se ejecute de forma idéntica en diversas configuraciones de hardware.

Con la aparición de Windows NT (*Nueva tecnología*), sistema operativo de 32 bits, los usuarios pueden beneficiarse de la multitarea con requisa y multinivel. Windows NT es diseñado para ofrecer un entorno de computación avanzada abierto, basado en especificaciones estándar, para los sistemas basados en microprocesadores.

### **Historia de Windows**

Microsoft inicia trabajando sobre la primera versión de Windows en 1983 y libera la versión 1.1 de Windows en 1985. Esta versión fue diseñada para un sistema IBM PC 8088 con 256 K en RAM. Windows 1.1 automáticamente organiza en mosaico la ventana de aplicación y soporta ventanas instantáneas (pop-up).

En 1987, Microsoft libera la próxima versión mejorada de Windows y es numerada como 2.0. Esta versión contiene una nueva interface de usuarios que soporta las ventanas sobrepuestas. Una mejor ventaja de Windows 2.0 fue el uso de memoria expandida. Sin embargo, Windows 2.0 sólo podía operar en modo real, la cantidad de memoria que podía ser direccionada estaba limitada a 1 Mbyte.

Microsoft libera la versión 3.0 de Windows en 1990. Esta versión agrega el soporte de menús, cuadros de listas y botones creados por el usuario. Adicionalmente Windows 3.0 corre en máquinas 80386 ó 80486, con capacidad de direccionar 16 Mbytes de memoria.

Más tarde surge la versión 3.1 de Windows que proporciona mejores servicios que la versión 3.0, como son:

- ◆ Intercambio dinámico de datos
- ◆ Características de ligado y pegado de objetos
- ◆ Soporta multimedia
- ◆ Proporciona accesorios adicionales
- ◆ Administración de memoria

Posteriormente se libera la versión 3.11 para el trabajo en grupo; Windows NT, para trabajo en red y finalmente Windows 95 que es la última versión en el mercado.

**Popularidad de Windows**

Las interfaces gráficas de usuarios usan un display de mapa de bits para expresar información. Debido a que el display no esta limitado a un conjunto de caracteres ASCII, como en el caso de los programas tradicionales basados en texto para MS-DOS, la imagen sobre el display refleja precisamente la imagen producida sobre un dispositivo de salida tal como una impresora o un plotter. Esta correlación entre la imagen de la pantalla y la imagen de la salida es llamada: " WYSISYG " (What-You-See-Is-What-You-Get), y es esta una razón de que las interfaces gráficas de usuarios sean populares.

Otros de los motivos de la popularidad de Windows, se debe a las ventajas que proporciona tanto al usuario como al programador (Tabla 2-1).

<b>Ventajas de Windows para el usuario</b>	<b>Ventajas de Windows para el programador</b>
<ul style="list-style-type: none"> <li>◆ Todas las aplicaciones de Windows contienen las mismas características básicas.</li> <li>◆ El uso de imágenes basadas en gráficos para representar aplicaciones y datos.</li> <li>◆ El uso de WYSIWYG en la interface de usuario (lo que se ve en pantalla se imprime).</li> <li>◆ Capacidad de multitarea.</li> </ul>	<ul style="list-style-type: none"> <li>◆ El diseño básico de la Interface y las herramientas ya están establecidas.</li> <li>◆ Permite al programador diseñar representaciones gráficas de estructuras físicas como archivos y directorios, menús pop-up y cuadros de diálogo.</li> <li>◆ Es posible acceder más memoria de la que se dispone con las aplicaciones tradicionales de MS-DOS.</li> <li>◆ Proporciona la capacidad de desarrollar gráficos como dispositivos independientes, es decir, que el código no depende de una cierta configuración del sistema.</li> </ul>

Tabla 2-1 Tabla que muestra las ventajas del uso del ambiente Windows.

**2.1.2 Windows vs. MS-DOS.**

En comparación con el entorno MS-DOS convencional, el entorno operativo de Windows ofrece ventajas considerables tanto a los usuarios como a los programadores (Tabla 2-2).

<b>Operación</b>	<b>MS-DOS</b>	<b>Windows</b>
Entrada	La entrada es recibida via las llamadas de funciones.	Es recibida por una cola de mensajes apropiada
Gráficos	Los gráficos son independientes del programa.	Proporciona operaciones de gráficas como dispositivos independientes (líneas, círculos, rectángulos, etc.)
Tareas	Ejecuta una sola tarea a la vez	Ejecuta múltiples tareas a la vez.

Tabla 1.2-2 Diferencias entre MS-DOS y Windows.

Algunos de los servicios adicionales que ofrece Windows en comparación con los servicios ofrecidos por MS-DOS son los siguientes:

- ◆ **Una interface gráfica para el usuario estándar** Esta interface gráfica de Windows se sirve de dibujos o iconos para representar unidades de disco, archivos, subdirectorios, aplicaciones y la mayoría de las ordenes y acciones del sistema operativo (Fig. 2-1). Se dice que la interface gráfica es estándar dado que todas las aplicaciones para Windows tienen la misma apariencia, tanto en la interface de teclado como en la del "mouse".

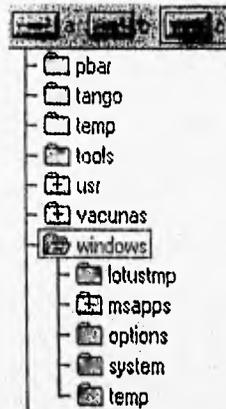


Figura 2-1 Elementos gráficos que permiten mejor visualización en la Interface gráfica de Windows.

- ♦ **La entrada de datos es controlada por un sistema de colas.** Debido a la filosofía de recursos compartidos que se dispone en el ambiente Windows, se requiere de un sistema de cola donde Windows reciba la entrada del teclado, del "mouse" y del temporizador. La cola es responsable de redireccionar la entrada recibida a la aplicación apropiada. Estas entradas tienen un formato uniforme denominado mensaje de entrada, el cual contiene la hora del sistema, el estado del teclado, el código de exploración de cualquier tecla pulsada, la posición del "mouse", información acerca del dispositivo que la generó, etc.
- ♦ **Capacidad de multitarea.** En un entorno operativo con capacidad de multitarea se permite al usuario tener varias aplicaciones, o varias instancias de una aplicación, ejecutándose en forma concurrente (Fig. 2-2). En Windows se habla de que varios procesos pueden ejecutarse concurrentemente, sin embargo, la realidad es que sólo uno de ellos puede usar el procesador en un instante dado. Por lo tanto, es responsabilidad de Windows el reparto de tiempo de uso del microprocesador.

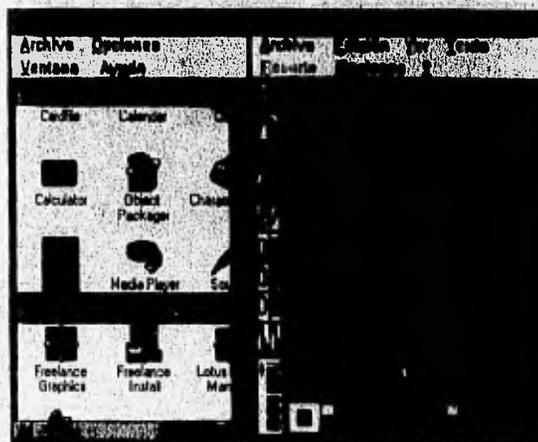


Figura 2-2 Aplicaciones de Windows ejecutándose concurrentemente.

- ♦ **Independencia de los dispositivos.** Windows libera al programador de tener en cuenta todas las variedades posibles de monitores, impresoras y dispositivos de entrada disponibles (Fig. 2-3). Es por esto que todas las aplicaciones escritas para Windows deben ser diseñadas de modo que incluyan controladores para todos los dispositivos posibles. Windows consigue la independencia de dispositivos, especificando las características que debe tener el hardware. De esta manera se asegura que la entrada de alguna aplicación sea válida y este predefinida.

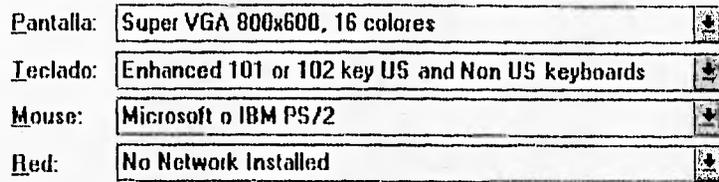


Figura 2-3 Caja de diálogo que permite cambiar el hardware con el que opera Windows.

- ◆ **Ofrece bibliotecas de enlace dinámico.** Gran parte de la funcionalidad de Windows la ofrecen las bibliotecas de enlace dinámico, llamadas DLL ( *Dynamic Link Libraries* ), que mejoran el sistema operativo base ofreciendo una potente y flexible interface gráfica de usuario. Las bibliotecas de enlace dinámico contienen funciones predefinidas que se unen a un programa de aplicación cuando éste se carga (dinámicamente), en lugar de hacerlo cuando se genera el archivo .EXE (estáticamente). Esta operación permite el ahorro de memoria, debido a que independientemente del número de aplicaciones que se estén ejecutando, sólo existe una copia de la biblioteca en la memoria RAM.
- ◆ **Capacidad de ejecutar aplicaciones de MS-DOS.** Windows puede ejecutar fácilmente algunas aplicaciones existentes para MS-DOS (Fig. 2-4). Estas aplicaciones deben hacer uso de las interrupciones software del MS-DOS y de la ROM BIOS de la PC, para leer del teclado y escribir en la pantalla. Generalmente estas aplicaciones pueden ejecutarse en una ventana. Las aplicaciones que no están posibilitadas para ejecutarse en Windows, son las que escriben directamente en la pantalla, utilizan gráficos o toman control directo de las interrupciones hardware del teclado. Estas aplicaciones no pueden ser ejecutadas en una ventana.



Figura 2-4 Ventana en el ambiente Windows que ejecuta comandos del sistema operativo MS-DOS

### 2.1.3 Elementos de una aplicación:

La mayoría de las aplicaciones en Windows usan los siguientes elementos para interactuar con el usuario:

#### ◆ Ventanas

Una ventana es el dispositivo primario de entrada y salida de cualquier aplicación de Windows. Por otro lado, una ventana en Windows, para el usuario, es una porción rectangular de la pantalla que tiene una apariencia idéntica en todas las aplicaciones. Una ventana es creada y controlada por una aplicación, cada vez que es seleccionada alguna opción de la ventana el programa responde. Asimismo, si se cierra una ventana, el programa asociado finaliza su ejecución.

La pantalla en el ambiente de Windows es dividida en diferentes ventanas, por lo que el usuario puede elegir alguna de las aplicaciones que se ejecutan utilizando el teclado o el "mouse" para la selección.

Todas las ventanas de Windows poseen ciertas características y comportamientos en común. Los componentes de una ventana se muestran en la Fig. 2-5.

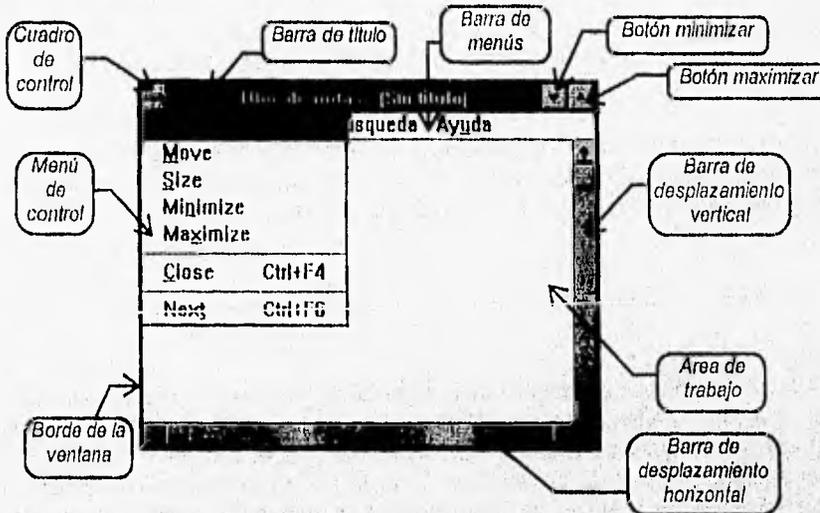


Figura 2-5 Componentes fundamentales de una ventana de Windows.

- ◆ **Borde.-** Es constituido por las líneas que sirven de marco en la ventana. Su función es indicarnos cual es la ventana activa y modificar el tamaño de una ventana cuando el usuario desplaza el puntero del mouse sobre él.
- ◆ **Barra de título.-** Muestra el nombre del programa de aplicación, centrado en la parte superior de la ventana asociada.
- ◆ **Cuadro de control.-** Todas las aplicaciones de Windows tienen un cuadro de control. El cuadro de control es situado en la esquina superior izquierda de la ventana, es un pequeño rectángulo blanco rodeado por un cuadro sombreado. Si se hace click en el cuadro de control, Windows mostrará el menú del sistema.
- ◆ **Menú de control.-** El menú de control también llamado menú del sistema, se activa haciendo click sobre el cuadro de control. Este menú ofrece opciones estándar de la aplicación, como Restaurar, Mover, Tamaño, Minimizar, Maximizar, Cerrar y Cambiar a.
- ◆ **Botón minimizar.-** En la esquina superior derecha de todas las ventanas aparece el botón minimizar que contiene una flecha que apunta hacia abajo y que hace que la ventana se reduzca a un icono.
- ◆ **Botón maximizar.-** Este botón está situado en la esquina superior derecha de todas las ventanas y muestra una flecha que apunta hacia arriba. El cuadro maximizar se usa para que la ventana de una aplicación ocupe toda la pantalla.
- ◆ **Barra de desplazamiento vertical.-** Esta situada a la derecha de las ventanas, bajo el cuadro maximizar. La barra de desplazamiento vertical tiene una flecha en ambos extremos, apuntando en sentidos opuestos; una banda sombreada y un cuadro transparente. Este cuadro denominado cuadro de desplazamiento, indica la posición del contenido actualmente mostrado en relación al documento completo, que está representado por la banda sombreada. La barra de desplazamiento vertical se usa para seleccionar una de entre las múltiples páginas de salida que se desea observar.

- ◆ **Barra de desplazamiento horizontal.**- Esta situada en la parte inferior de todas las ventanas. Tiene en sus extremos flechas que apuntan en sentidos opuestos, una banda sombreada y un cuadro transparente. El cuadro transparente tiene la misma función que en la barra de desplazamiento vertical. La barra de desplazamiento horizontal se usa para seleccionar una de entre las múltiples columnas de información que se desea observar.
- ◆ **Barra de menú.**- La barra de menú está colocada abajo de la barra de título. La barra de menú se utiliza para realizar selecciones de menús y submenús. Estas selecciones se pueden realizar apuntando y haciendo "click " sobre la orden en el menú en cuestión, o bien, usando una combinación de "teclas rápidas".
- ◆ **Area de trabajo.**- El área de trabajo ocupa la parte más grande de todas las ventanas. El área de trabajo es el área de salida principal de la aplicación. La administración del área de trabajo es responsabilidad de la aplicación y sólo esta puede generar salida en el área de trabajo.

◆ **Menús**

Los menús son los medios principales de la entrada de un usuario a alguna aplicación de Windows. Un menú es una lista de elementos o comandos que representan las acciones que puede realizar la aplicación (Fig.2-6). La selección de un menú se realiza por el uso del "mouse" o desde el teclado, desplazándose hasta el elemento considerado o por la pulsación de una combinación de teclas. La respuesta de Windows es el envío de un mensaje a la aplicación, indicándole que comando se ha seleccionado.

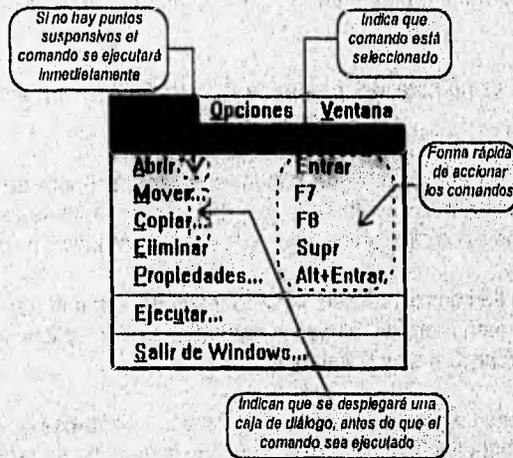


Figura 2-6 Lista de comandos del menú archivo de una ventana de Windows.

◆ **Cuadros de diálogo**

Un cuadro de diálogo es una ventana temporal que una aplicación despliega para que el usuario pueda aportar más información para un comando (Fig. 2-7). Un cuadro de diálogo contiene uno o más controles. Un control es una ventana más pequeña que tiene una función de entrada o salida muy simple.

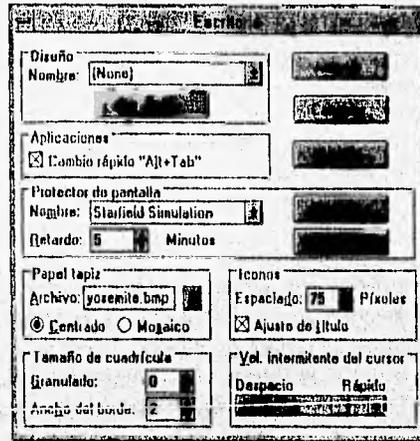


Figura 2-7 Ejemplo de un cuadro de diálogo en Windows.

Los cuadros de diálogo permiten al **usuario** seleccionar elementos de una lista que se presentan en una ventana, activar botones de opción, introducir cadenas o enteros directamente desde el teclado, etc.

Los cuadros de diálogo se llaman seleccionando una opción de un menú y se presentan al usuario como ventanas instantáneas. Para distinguir una opción que abre una caja de diálogo de otras opciones del menú, se añaden puntos suspensivos (...) a continuación del nombre de la opción correspondiente al cuadro de diálogo (Fig 2-6).

#### ◆ Mensajes

Un mensaje es una notificación de que se ha producido un suceso de interés, que puede o no requerir una acción específica. Estos sucesos pueden ser iniciados por el usuario o ser generados por alguna aplicación.

El sistema de mensajes de Windows es una estructura utilizada para distribuir la información en un entorno multitarea y hace posible que Windows pueda compartir el procesador entre distintas aplicaciones. Una aplicación sólo puede tener acceso al microprocesador cuando recibe un mensaje.

Otra función de los mensajes en Windows es posibilitar que una aplicación responda a los sucesos producidos en el entorno, por la propia aplicación, otras aplicaciones que se ejecutan concurrentemente, por el usuario o por Windows.

Por lo tanto, se podría decir que la tarea principal de una aplicación en Windows consiste en procesar mensajes.

#### ◆ Iconos

Un ícono es un pequeño símbolo que se utiliza para mostrar al usuario una operación, una idea o una aplicación. El ícono de cada aplicación se distingue por su forma y por su etiqueta (Fig. 2-8).

Usando las herramientas de Windows los usuarios pueden diseñar sus propios iconos, independientes de los dispositivos.



Figura 2-8 Iconos que representan aplicaciones en Windows.

◆ **Botones**

Los botones son símbolos gráficos con efectos de relieve que corresponden a una determinada función dentro de un cuadro de diálogo (principalmente). Los botones tienen la función de confirmar los datos suministrados por el usuario en un cuadro de diálogo, ejecutar comandos, cancelar la ejecución de algún comando, abrir un nuevo cuadro de diálogo, etc. Independientemente de la función que tengan, todos los botones van rotulados con el nombre de la función que ejecutan (Fig. 2-9).

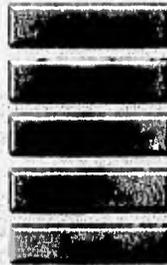


Figura 2-9 Botones proporcionados por una aplicación de Windows

◆ **Cursores**

Un cursor de Windows es un símbolo gráfico, que sustituye al carácter de subrayado parpadeante de MS-DOS. Los cursores cambian en función de la situación del dispositivo apuntador; esto sirve como ayuda para mostrar las operaciones que están disponibles (Fig. 2-10).

Usando las herramientas de Windows, es posible la creación de propios cursores.

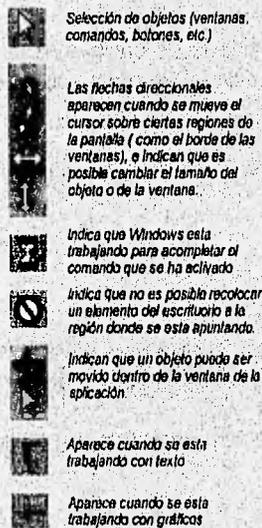


Figura 2-10 Algunos cursores proporcionados por Windows.

### ◆ Puntos de inserción

Los puntos de inserción son símbolos que se sitúan en alguna parte de una ventana para indicar al usuario donde se recibirá la entrada de datos. Los puntos de inserción se distinguen fácilmente de los demás señalizadores de la pantalla porque parpadean (Fig. 2-11).

Al contrario de lo que sucede con los iconos y los cursores, una aplicación puede crear sus propios puntos de inserción.



Entrada de texto.

Figura 2-11 Indica un punto de inserción en una ventana.

### ◆ Cuadros de mensaje

Los cuadros de mensaje son ventanas instantáneas que contienen un título, un icono y un mensaje (Fig. 2-12). La aplicación necesita suministrar el título del mensaje; el mensaje en sí mismo; instrucciones que indican que icono le corresponde usar; y, si es el caso, la respuesta correspondiente que se debe utilizar (Sí, No, Ok, Ayuda, Aceptar, Cancelar, etc.).



Figura 2-12 Cuadro de mensaje proporcionado por Windows.

### ◆ Fuentes

Una fuente es un recurso que define un conjunto completo de caracteres a partir de un tipo de letra, con cierto tamaño y estilo. Las fuentes se pueden manipular para darle al texto diversas apariencias (Fig. 2-13). Un tipo de letra es un diseño básico de caracteres que posee ciertos *serifs* (remates de las letras) y anchos de impresión.

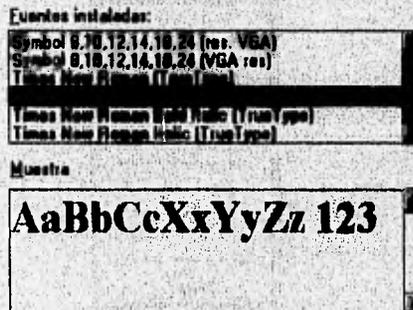


Figura 2-13 Fuentes proporcionadas por Windows.

Las aplicaciones pueden usar diversas fuentes dentro de las que ofrece Windows ( System, Courier, Times New Roman, etc.) o fuentes definidas por el usuario.

Windows permite la modificación dinámica de un tipo de letra, la cual puede consistir en cambiar en negrita, cursiva, subrayado o modificar el tamaño de la fuente.

◆ **Matrices de puntos**

Una matriz de puntos puede considerarse como una fotografía instantánea, en pixeles, de una porción de la pantalla y que se encuentra almacenada en memoria (fig. 2-14). Las matrices de puntos se utilizan siempre que una aplicación desea mostrar rápidamente una imagen gráfica, dado que el objeto se transfiere directamente de memoria. Las matrices de puntos se utilizan para mostrar dibujos en pantalla y para crear pinceles.

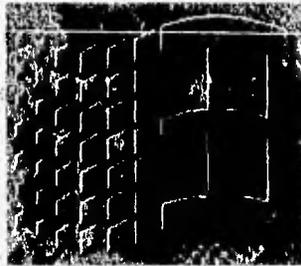


Figura 2-14 Matriz de puntos usada por Windows para tapiz del escritorio.

Una matriz de puntos tiene dos inconvenientes principales:

1. Requiere cantidad de memoria impredeciblemente grande. Por cada pixel mostrado, debe haber una representación equivalente en memoria. La cantidad de memoria que se requiere depende de si el monitor es a color (4 bits por pixel; 16 colores) o monocromático (1 bit por pixel).
2. Solo contienen un dibujo.

◆ **Lápices**

Los lápices se usan para dibujar líneas y contornos (Fig. 2-15). Poseen 3 características básicas:

1. Ancho de línea.
2. Estilo (punteado, a rayas o sólido).
3. Color.

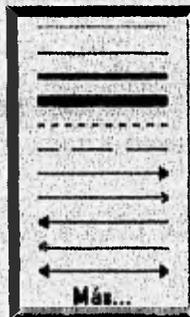


Figura 2-15 Lápices proporcionados por una aplicación de Windows.

Las aplicaciones de Windows definen un lápiz para dibujar líneas negras y otro para líneas blancas. Por otro lado, el usuario puede crear sus propios lápices.

#### ◆ Pinceles

Los pinceles se usan para colorear y rellenar zonas de la pantalla con patrones predefinidos (Fig. 2-16). Los pinceles tienen un tamaño predefinido de 8 X 8 píxeles, y al igual que los lápices poseen 3 características básicas:

1. Tamaño.
2. Patrón de relleno.
3. Color.



Figura 2-16 Algunos pinceles proporcionados por una aplicación de Windows.

El patrón de relleno puede ser desde un color sólido hasta una trama de líneas en diagonal, o cualquier otra combinación definida por el usuario.

#### 2.1.4 Modos de operación.

Microsoft Windows corre sobre microprocesadores de la familia 8086 de Intel. Esta familia incluye los microprocesadores 8086, 8088, 80186, 80188, 80286, 80386 y 80486. A pesar que las capacidades de hardware de estos procesadores varían ampliamente, Windows no está limitado a estas capacidades ya que cuenta con tres modos de operación básicos que pueden adaptarse a la configuración del sistema dado (Tabla 2-3). Los tres modos básicos de operación son:

1. Modo real
2. Modo estándar
3. Modo mejorado 386

Opciones	Función
/R	Inicia Windows en modo real
/S	Inicia Windows en modo estándar
/3	Inicia Windows en modo mejorado 386

Tabla 2-3 Modificadores ofrecidos por Windows para determinar su modo de operación

## CAPITULO 2: AMBIENTE DE MICROSOFT WINDOWS

Windows opera en modo real cuando la computadora tiene un procesador 8086, 8088, 80186 ó 80188, con capacidad de memoria menor a 1 Mbyte. La desventaja de usar este modo de operación es que limita el aprovechamiento de las características de manejo de memoria de Windows 3.0 y 3.1.

El modo de operación estándar es usado cuando la computadora tiene un procesador 80286 ó 80386 con menos de 1 Mbyte de memoria. El modo estándar se define como el modo de operación normal para Windows. Cuando se opera en éste modo se tiene acceso a la memoria extendida (hasta 16 Mb), se cargan múltiples aplicaciones y hay intercambio entre ellas.

Windows opera en el modo mejorado 386 cuando la computadora tiene un procesador 80386 ó 80486 con menos de 2 Mbytes de memoria. Una de las características principales de este modo es que tiene la capacidad de apoyar el uso de una memoria virtual. Con un sistema de memoria virtual una computadora puede tener acceso a más memoria que la que realmente tiene, lo cual permite ejecutar programas más grandes o múltiples en forma simultánea.

### 2.1.5 Requerimientos de Software y Hardware.

Al igual que todos los programas que manejan texto y gráficos, Windows requiere que tanto el software como el hardware cumplan con algunas características para que su desempeño sea eficiente. Independientemente de la plataforma que el usuario o programador utilice, Windows requiere del software y hardware que se muestra en la tabla 2-4.

Software	Hardware
<ul style="list-style-type: none"><li>◆ Sistema operativo MS-DOS 3.1 o posterior.</li><li>◆ Para programación:<ul style="list-style-type: none"><li>◆ Compilador Microsoft C/C++ (versión 7.0 o posterior).</li><li>◆ Kit de desarrollo de software (SDK), de Microsoft</li><li>o</li><li>◆ Compilador de Borland C/C++ (Versión 3.1 o posterior).</li><li>◆ Herramientas para la estructura de las aplicaciones de Borland.</li></ul></li></ul>	<ul style="list-style-type: none"><li>◆ IBM PC XT, AT, PS/2 o compatible con microprocesador 80286, 80386 ó 80486. (para programación 80386 o posteriores)</li><li>◆ Adaptador gráfico (VGA, SVGA o mejor).</li><li>◆ Ratón.</li><li>◆ Disco duro (de mas de 80 Mbytes con 10 Mbytes libres).</li><li>◆ 2 Mbytes en RAM (para programación 4 Mbytes como mínimo).</li><li>◆ Unidad de disco de 1.2Mbytes (5 ¼ ") o de 1.44 Mbytes (3 ½ ").</li></ul>

Tabla 2-4 Requerimientos de software y hardware para trabajar eficientemente con Windows.

## 2.2 CONCEPTOS DE PROGRAMACION EN WINDOWS

### 2.2.1 Programación por eventos

La mayoría de los programas de MS-DOS se escriben usando una metodología de programación secuencial basada en el manejo de procedimientos, la cual tiene distintos inicios y distintos finales. El programa directamente controla la secuencia de los eventos del programa o procedimientos.

Los programas en Windows, a diferencia de los de MS-DOS, son manejados por eventos. Los programas manejados por eventos son controlados por la ocurrencia de eventos, en vez que por secuencias de eventos (Fig. 2-17).

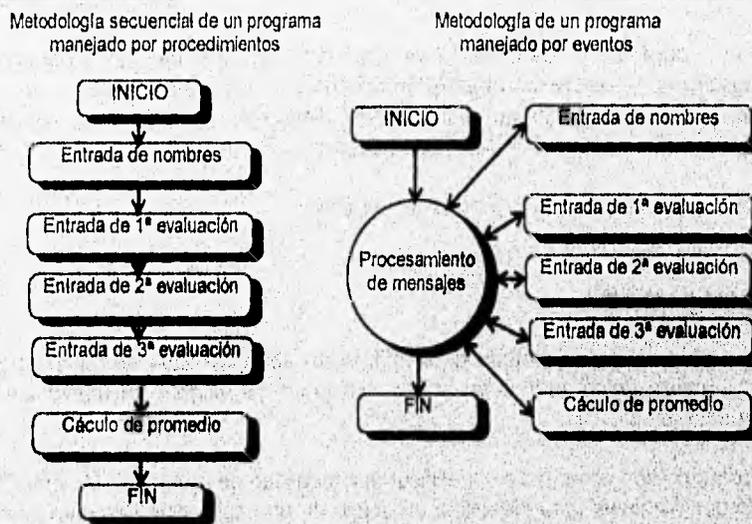


Figura 2-17 Cálculo de promedios de una clase por 2 metodologías: manejada por procedimientos y manejada por eventos.

Los programas manejados por eventos son diseñados para cubrir las limitaciones de las metodologías secuenciales manejadas por procedimientos, por medio del procesamiento de eventos en una forma no secuencial.

La programación manejada por eventos gira en torno a la generación y procesamiento de mensajes. Un mensaje tiene información acerca de un evento que ha ocurrido. Por lo tanto, el trabajo de un programador de Windows consiste en manejar y ordenar los mensajes de Windows, enviados a y desde la aplicación que se está desarrollando. Debido a que los mensajes de Windows son manejados por eventos no aparecen en algún orden predefinido.

Las metodologías de programación manejada por eventos proporcionan muchos beneficios y son muy usadas por aplicaciones que requieren intensa interacción con el usuario. De esta forma el usuario no es forzado a adaptarse al molde del procedimiento de un programa.

Con la programación manejada por eventos, el usuario controla los procedimientos, mientras que las aplicaciones manejadas por procedimientos tienden a controlar al usuario.

### 2.2.2 Objetos y programación de objetos.

Un objeto en Windows es un grupo de elementos que cuando son asociados componen un todo. En la programación en Windows es posible trabajar con objetos que son módulos de código y luego combinarlas para crear un objeto más grande que se denomina aplicación.

En términos de Windows un objeto consiste de todos los elementos de una ventana o de una aplicación, usados para desplegar información al usuario.

Para hacer una simple llamada a Windows, un usuario puede tomar un objeto (la ventana principal de una aplicación, por ejemplo) y moverlo como un todo a una nueva posición sobre la pantalla. Cuando el usuario mueve un objeto todos los elementos asociados con este también se mueven.

Cuando trabajamos con un objeto en Windows también se hace referencia a las partes del objeto, relativas al objeto mismo. Si se quiere desplegar un elemento sobre la pantalla se debe colocar el elemento en la ventana relativa a la aplicación (en la ventana misma) y no relativa a la pantalla física. Se debe pensar que cualquier elemento asociado con una ventana es soportado en la ventana misma y no en la pantalla.

Cuando se usa programación para Windows realmente se esta usando programación orientada a objetos. En la programación orientada a objetos, un objeto es un tipo de datos abstracto que consta de una estructura de datos y varias funciones que actúan sobre dicha estructura de datos. Además en Windows, los objetos reciben mensajes que pueden hacer que aquellos se modifiquen.

En Windows los manipuladores dan al objeto su identidad.

### 2.2.3 Manipuladores

Un manipulador (*Handle*) es un método de identificación del objeto que se quiere usar o modificar. El manipulador es un nombre único dado a un objeto. Después de asignar un manipulador a un objeto es posible hacer referencia de él.

Los manipuladores no son aptos para especificar una localidad de memoria. Un manipulador es un valor entero único usado por Windows para identificar un objeto en una aplicación tal como ventanas, instancias, menús, localidades de memoria, dispositivos de salida, controles, archivos, etc.

La mayoría de los manipuladores están listados en tablas internas. Windows usa manipuladores índices para acceder la información almacenada en esas tablas. Una aplicación de Windows usualmente puede acceder sólo manipuladores y no el dato actual que el manipulador representa. Cuando una aplicación debe examinar o cambiar la información, esta proporciona el manipulador y Windows hace el resto.

### 2.2.4 Instancias

Una instancia (*Instance*) es el manipulador de una aplicación. Una instancia se crea la primera vez que se inicia una aplicación. Windows identifica una aplicación por instancias, dado su ambiente de programación orientada a objetos.

Debido a que Windows opera en un ambiente multitarea, permite que múltiples copias de una misma aplicación puedan correr simultáneamente y les sigue la pista, a cada una de estas instancias, asignándoles su propio manipulador.

Windows ahorra recursos del sistema utilizando el mismo código para todas las instancias de una aplicación. Sólo se gestiona de forma separada el segmento de datos de cada instancia. Además, la primera instancia de una aplicación tiene un papel muy importante; crea todos los objetos necesarios para el funcionamiento de la aplicación (cuadros de diálogos, menús, clases de ventanas, etc.) permitiendo, en la mayoría de los casos, que estos recursos estén disponibles para todas las demás aplicaciones.

2.2.5 Ciclo del mensaje

Las aplicaciones de Windows reciben la entrada en forma de mensajes. Los mensajes contienen información acerca del dispositivo que generó la entrada, el estado actual del teclado, la posición del cursor, el estado del "mouse" y el tiempo del sistema.

Una aplicación puede recibir mensajes de cuatro fuentes diferentes:

1. Del usuario
2. De Windows
3. Del propio programa de aplicación
4. De otra aplicación.

Un componente fundamental de todas las aplicaciones de Windows es el bucle de procesamiento de mensajes, el cual es responsable de procesar un mensaje enviado por Windows al cuerpo principal del programa.

El ciclo del mensaje (Fig. 2-18) puede mencionarse como sigue:

1. Windows monitorea todos los dispositivos de entrada y coloca los mensajes de entrada dentro del sistema de colas. Windows sólo tiene una cola de mensajes del sistema, pero cada programa que se ejecuta concurrentemente con Windows dispone de su propia cola de mensajes.
2. Luego, copia los mensajes de entrada desde el sistema de colas a la cola de la aplicación apropiada.
3. El bucle de mensajes de la aplicación, recibe el mensaje desde la cola de la aplicación y envía cada mensaje al procedimiento de ventana apropiado para su procesamiento. Hecho esto, el procedimiento de ventana ejecutará la acción deseada.

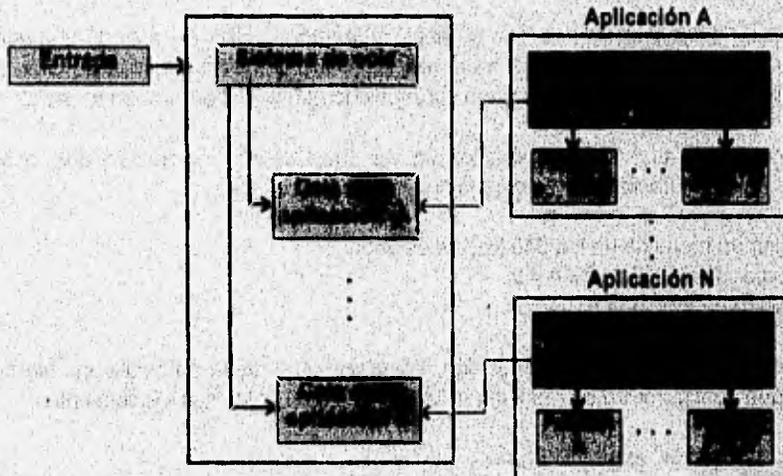


Figura 2-18 Ciclo del mensaje en Windows.

### **2.2.6 Multitarea**

El termino multitarea se refiere a la capacidad de ejecutar más de una tarea a la vez. En Windows, esto implica que todas las aplicaciones que se están ejecutando concurrentemente deben compartir los recursos de la computadora ( CPU, memoria, dispositivos de entrada y salida, etc.).

Debido a que las computadoras personales (PC's) están basadas a sólo un microprocesador, manejan las tareas una a la vez. Por lo tanto el programador sólo puede simular la multitarea. Existen dos métodos principales para simular la multitarea:

1. Método no preventivo
2. Método preventivo

El método no preventivo depende de las aplicaciones que comparten los recursos de la computadora. Cada aplicación debe estar diseñada para reconocer que otras aplicaciones existen y contar con algún medio para liberar el control a otra aplicación. Este es el método usado por Windows para proporcionar capacidad de multitarea.

El método preventivo requiere del sistema operativo para manejar múltiples aplicaciones. Con éste método el sistema operativo no depende de la aplicación para liberar el control del CPU. El sistema operativo se encarga de suspender la ejecución de una aplicación para darle oportunidad a otra.

### **2.2.7 Administración de memoria**

La memoria es uno de los recursos compartidos más importante en Windows. Con más de una aplicación ejecutándose al mismo tiempo, las aplicaciones deben compartir la memoria a fin de no agotar este recurso.

Windows es capaz de consolidar el espacio de memoria libre no utilizada moviendo bloques de código y datos en memoria. El espacio de memoria libre se genera conforme empiezan a ejecutarse unos programas a la vez que otros finalizan su ejecución, es decir, la memoria puede quedar fragmentada.

Windows permite a las aplicaciones "sobrecargar" la memoria. Esto es, cuando una aplicación contiene más código del que realmente cabe en memoria en un determinado momento, Windows retira bloques de código de la memoria y posteriormente vuelve a cargar dicho código del archivo en cuestión.

Cuando se desarrolla una nueva aplicación en Windows debe considerarse que la aplicación puede ejecutarse bajo tres configuraciones de memoria distintas, principalmente:

- ◆ La configuración de memoria de los 640 Kbytes estándar.
- ◆ La memoria expandida EMS versión 4.0
- ◆ Una configuración de memoria protegida.

Cuando el usuario selecciona una aplicación, Windows comprueba automáticamente cuatro elementos para determinar que configuración de memoria utilizará. Los elementos son los siguientes:

1. El tipo de procesador que se esta utilizando.
2. La cantidad de memoria disponible de los 640 Kbytes base
3. La cantidad de memoria expandida EMS versión 4.0 disponible
4. La cantidad de memoria extendida disponible.

Debido a que Windows no puede saber cuánta memoria necesitará una aplicación en un determinado momento, la gestión de este recurso limitado corresponde a la aplicación. Es decir, cada aplicación controla

cuanta memoria necesita, especificando al mismo tiempo cuanta memoria deja de estar disponible para el resto de las aplicaciones. Para que cada aplicación gestione eficientemente la memoria debe considerar los siguientes principios básicos:

- ◆ Limitar el número de objetos usados.- Es decir, compartir los objetos siempre que se pueda, en lugar de forzar a cada tarea a crear su propia copia del objeto y no dejar accidentalmente en memoria objetos no utilizados.
- ◆ Minimizar el tamaño de cada objeto.- Un objeto puede minimizarse: segmentando su código, asignando la mínima cantidad de memoria necesaria para representar un objeto o por medio del uso conservador de las matrices de puntos.
- ◆ Definir los objetos como reubicables.- Cuando se crean objetos en Windows existe la oportunidad de declararlos como reubicables o no reubicables. Si se declaran reubicables se autoriza a Windows para mover objetos de memoria con el fin de maximizar la memoria.

### 2.2.8 Recursos

Los recursos representan uno de los principales aspectos de la programación en Windows. La principal característica de los recursos es que se pueden agregar a un programa sin tener que agrandar notablemente su código. No existe algún número límite de recursos que pueda ser agregado a un programa. Los recursos a los que nos referimos son: menús, cuadros de diálogo, mapas de bits, iconos, fuentes, cursores, aceleradores, lápices, pinceles, etc.

Un recurso representa los datos que se incluyen en el archivo ejecutable (.EXE) de un programa, aunque no residan en el segmento de datos normal de un programa. Cuando Windows carga un programa en memoria para ejecutarlo, muy a menudo deja los recursos en el disco, y los carga en memoria sólo en caso de que se requieran.

Habitualmente, una aplicación define sus recursos como de sólo lectura y descartables. Esto permite que Windows pueda desechar un recurso cuando se necesite más memoria. Si se necesita nuevamente el recurso, Windows simplemente leerá el disco y volverá a cargar los datos en memoria. En caso de que se tengan varias instancias de una misma aplicación ejecutándose concurrentemente, Windows no sólo comparte el código del programa, sino también compartirá las definiciones de los recursos.

Los recursos en Windows son diseñados para ser herramientas expansibles, es decir, sólo están limitados por la imaginación del programador.

### 2.2.9 Componentes de una aplicación.

La construcción de una aplicación de Windows puede suponer pasar por alguna o todas las etapas siguientes:

1. Crear la función WinMain (función que contiene el cuerpo principal de un programa en Windows) y las funciones de ventana asociadas en "C" o "C++".
2. (Opcional). Usar las herramientas de Windows para crear cursores, matrices de puntos, iconos, cuadros de diálogo, etc. propios de la aplicación.
3. Crear un archivo de definición de recursos (.RC) que defina los recursos de la aplicación. Este archivo lista y nombra los recursos creados en el punto anterior. También puede definir otros recursos.

## CAPITULO 2: AMBIENTE DE MICROSOFT WINDOWS

4. Crear el archivo de definición de módulos (.DEF), el cual define los atributos de los módulos de la aplicación como: atributos de segmento, tamaño de la pila, etc.
5. Compilar y enlazar todo el código fuente en lenguaje C/C++; ensamblar todos los archivos fuente en lenguaje ensamblador (si los hay)
6. Compilar el archivo de definición de recursos (.RC) y el archivo resultante (.RES) agregarlo al archivo ejecutable.

La figura 2-19 muestra los pasos necesarios para construir una aplicación de Windows.

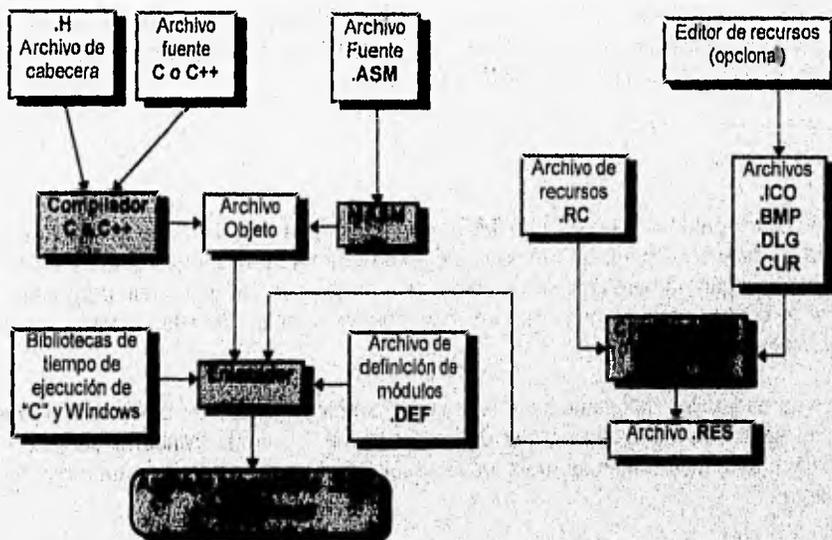


Figura 2-19 Componentes de una aplicación en Windows.

En general cuando se escribe una aplicación en Windows se debe considerar lo siguiente:

- ◆ No tomar control exclusivo del CPU; dado que es un recurso compartido. A pesar de Windows es un sistema multitarea, esta es no preventiva.
- ◆ No intentar acceder directamente la memoria o los dispositivos de hardware como: el teclado, el "mouse", la pantalla, el temporizador y los puertos seriales y paralelos. Windows requiere absoluto control de esos recursos para garantizar equidad y accesos interrumpidos para todas las aplicaciones que se están ejecutando.
- ◆ Dentro de las aplicaciones, todas las funciones que Windows puede llamar deben ser definidas con la palabra reservada **PASCAL**, esto garantiza que las funciones acceden los argumentos correctamente.
- ◆ Todas las aplicaciones deben tener una función WinMain. Esta función es el punto de inicio de la aplicación, ya que contiene las sentencias y funciones que crean ventanas y que reciben y despachan los mensajes los mensajes asociados con la aplicación.
- ◆ Cuando se usan funciones de ventanas, verificar siempre el valor que regresan.
- ◆ No usar funciones de tiempo de corrida de C, para entrada y salida de la consola (getchar, putchar, scanf y printf).

- ◆ No usar funciones de tiempo de corrida de C, para entrada y salida de archivos por el puerto serial y paralelo.
- ◆ Es posible usar funciones de tiempo de corrida de C, para entrada y salida de archivos de disco (OpenFile, por ejemplo).
- ◆ Es posible usar funciones de tiempo de corrida de C, para administración de memoria. (malloc, calloc, realloc, etc.).

**P A R T E II**  
**A N A L I S I S**

## **CAPITULO 3**

# **ANALISIS DE HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES EN WINDOWS**

3.1 Herramientas para el desarrollo de aplicaciones en Windows.

3.2 Manejo de imágenes en Windows.

3.3 Técnicas de animación en ambiente Windows.

## 3.1 Herramientas para el desarrollo de aplicaciones en Windows.

### 3.1.1 Desarrollo de aplicaciones para usuarios.

En la actualidad las computadoras personales son ampliamente usadas por los usuarios para el desarrollo de aplicaciones.

Hasta ahora, la programación es la técnica apropiada para el desarrollo de software a gran escala. Sin embargo, la programación puede ser muy difícil dado que requiere de muy alto nivel de conocimientos técnicos y entrenamiento especializado. Debido a esto, surge la necesidad de usar ciertas alternativas a la programación tradicional, entre ellas están:

- ◆ Macro lenguajes.
- ◆ Generadores de programas.
- ◆ Lenguajes de programación de base de datos.
- ◆ Constructores de interfaces.
- ◆ Aplicaciones híbridas.
- ◆ Ambientes objeto.

#### Macrolenguajes

Una *macro* es un comando sencillo que ejecuta una secuencia de otros comandos. Las macros pueden ser usadas para automatizar tareas repetitivas o complejas que consumen demasiado tiempo.

Un *macrolenguaje* es un lenguaje de programación de propósito especial, implantado dentro de una aplicación. Los macrolenguajes son generalmente más fáciles de usar que los lenguajes de programación tradicional, debido a que están limitados a manejar un flujo de clases de datos que son parte de la aplicación y con los cuales el usuario ya está familiarizado (Fig. 3-1).

```
Sub MAIN
Begin Dialog DiálogoUsuario 496, 180, "Ejemplo"
  OKButton 347, 12, 88, 21
  CancelButton 348, 42, 88, 21
  GroupBox 12, 12, 308, 147, "Salto"
  OptionGroup .salto
    OptionButton 22, 23, 207, 16, "Estudio Audiológico"
    OptionButton 22, 41, 173, 16, "Estudio «Sonido"
    OptionButton 22, 58, 120, 16, "Audiometría", .BotónDeOpción1
    OptionButton 22, 75, 244, 16, "Análisis clínico experimen
  .BotónDeOpción2
    OptionButton 22, 92, 220, 16, "Operación oído izquier
  .BotónDeOpción3
    OptionButton 22, 109, 213, 16, "Operación oído dere
  .BotónDeOpción4
End Dialog
```

Fig 3-1 Ejemplo de un segmento de código del macrolenguaje proporcionado por Excel para Windows.

Los macrolenguajes no eliminan el análisis de requerimientos de tareas, la definición del problema, y la planeación y el desarrollo de la solución.

La ventaja de los macrolenguajes consiste en la rapidez con la cual las aplicaciones pueden ser desarrolladas por no programadores. Su principal desventaja, es que frecuentemente fallan con el criterio de ser un buen lenguaje de programación estructurada.

### **Generadores de programas**

Los *generadores de programas*, algunas veces llamados generadores de aplicación o lenguajes de 4ª generación, ofrecen un camino para generar programas basados en especificaciones escritas acerca de los problemas que se van a resolver. Los generadores de programas pueden ser encontrados en los paquetes de bases de datos o paquetes específicos para sistemas muy largos o complejos.

Los generadores de programas deben proporcionar los medios para crear documentación para la aplicación, como bases de datos, la estructura de la aplicación, los programas, definición de pantallas, etc.

Los generadores de programas son llamados "lenguajes no procedurales", debido a que definen problemas en términos de los resultados deseados, en vez de en términos de programación procedural. Con los generadores de programas, se puede evitar la programación por pasos que debe ser especificada en un lenguaje de programación tradicional ( lenguaje C, Pascal, etc.).

### **Lenguajes de programación de bases de datos**

Los *lenguajes de programación de bases de datos* son similares a los lenguajes de programación tradicional, pero son generalmente más fáciles de aprender, dado que limitan los tipos de problemas, enfocándose sólo en resolver lo relacionado con entrada, salida, almacenamiento, ordenación y reporte de los datos. El desarrollar aplicaciones con lenguajes de programación de bases de datos puede ser fácil, flexible y barato debido a que están orientados a un determinado tipo de problemas.

En la programación con estos tipos de lenguajes se debe considerar los siguiente:

- a) El estilo y los límites del programa de base de datos en cuestión.
- b) Las capacidades de los diversos lenguajes proporcionados por las diferentes herramientas de bases de datos.

### **Constructores de interfaces**

Un *constructor de interface* permite ensamblar una interface de usuario, capaz de definir objetos gráficos como ventanas, botones, barras de desplazamiento, etc. Si a un constructor de interface se le agrega un lenguaje de programación sencillo, es capaz de desarrollar una aplicación completa.

Los ejemplos más comunes de constructores de Interfaces son: "Hypercard" para Macintosh y "Toolbook" para MS-DOS basado en computadoras con Microsoft Windows.

Los constructores de interfaces son herramientas que pueden desarrollar un amplio rango de aplicaciones que pueden incluir texto, gráficos, video, sonido y animación. Los tutoriales interactivos son aplicaciones comúnmente desarrolladas por los constructores de interfaces.

A pesar de lo que ofrecen, los constructores de interfaces no son herramientas que puedan desarrollar las diversas aplicaciones que pueda necesitar el usuario; inclusive, para algunas aplicaciones es mejor usar la programación tradicional o lenguajes de bases de datos. Sin embargo, los constructores de interfaces ofrecen al programador ciertas ventajas que permiten el desarrollo de aplicaciones muy atractivas, entre ellas están:

- ◆ **Estilo.**- Las aplicaciones pueden ser creadas con diferentes combinaciones de tipos de datos, ya sean texto, gráficos, voz, sonido, animación, etc.
- ◆ **Creatividad.**- Pueden proporcionar un rico ambiente gráfico interactivo, usando dibujos y gráficos, en lugar de texto.

- ◆ **Flexibilidad.**- Se pueden construir prototipos de una aplicación, con cierta facilidad, para hacer un modelo inicial más fácil de crear y modificar.

### **Aplicaciones híbridas**

Las *aplicaciones híbridas* proporcionan al programador la capacidad de ensamblar sistemas generados por otras herramientas, para desarrollar aplicaciones que satisfagan las necesidades específicas de un usuario. Es decir, implican el uso de una combinación de herramientas como constructores de interfaces, lenguajes de bases de datos, lenguajes de programación tradicional, etc.

Dada la combinación de herramientas, permitida por las aplicaciones híbridas, es posible crear un ambiente atractivo, que combine varias posibilidades de multimedia.

### **Ambientes de objeto**

Las aplicaciones en un ambiente orientado a objeto, son derivados de la programación orientada a objetos. Los *ambientes objeto* explotan la habilidad de prefabricar componentes llamados objetos. En esos ambientes prediseñados, los objetos pueden ser fácilmente combinados y modificados de muy diferentes formas para producir una nueva aplicación.

Los ambientes objeto son usados con interfaces gráficas de usuario, como es el caso de Microsoft Windows. Los ambientes objeto generan objetos para formar un sistema software que construye bloques con los cuales se pueden desarrollar aplicaciones.

Independientemente de la herramienta que se elija para el desarrollo de aplicaciones, su uso resulta ser rápido, flexible y barato, tanto en tiempo como en el factor económico, comparado con el uso de técnicas de programación tradicional. Esto se debe a que las aplicaciones desarrolladas con este tipo de herramientas, están orientadas a resolver problemas específicos de los usuarios.

Sin embargo, las herramientas para el desarrollo de aplicaciones de usuario, no siempre satisfacen todas las necesidades de una aplicación. Hay casos en que los lenguajes de programación tradicional son más adecuados para el desarrollo de algunas aplicaciones.

### **3.1.2 Constructores de interfaces.**

El uso de computadoras personales se ha incrementado en gran medida, por lo que surge la necesidad de adoptar software que sea fácil de aprender y usar, como es el caso de Microsoft Windows. No obstante, los programadores en ambiente Windows deben contar con gran habilidad, experiencia y mucho tiempo para el desarrollo de aplicaciones. Dado este problema surgen los constructores de interfaces en ambiente Windows.

### **ToolBook para Windows**

*ToolBook para Windows* es un constructor de interfaces en un ambiente orientado a objetos, es decir, no es posible crear aplicaciones en ToolBook al escribir una lista de sentencias en un programa, más bien se pueden crear aplicaciones al crear y/o modificar los objetos que incluyen dichas aplicaciones. Después de crear objetos para una aplicación, se deben crear guiones (*scripts*) que definan el rol de cada objeto que operará en la aplicación. Los *scripts* son programas que se escriben usando el lenguaje de programación proporcionado por ToolBook, *OpenScript*.

ToolBook puede crear aplicaciones que manejen bases de datos tradicionales, con ventajas adicionales sobre los tipos de programas que sólo desarrollan bases de datos. Estas ventajas son:

- ◆ Proporciona un sistema de ayuda para aprender como se programa el uso de ToolBook.
- ◆ Es posible crear interfaces de usuarios que incorporen y exploten las ventajas de Microsoft Windows.

Con ToolBook también es posible crear aplicaciones que incluyan facetas de hipermedia. Una forma es proporcionar todas las herramientas que se requieren para crear una base de datos con hipermedia, permitiendo almacenar y establecer enlaces entre textos, gráficos y animación.

ToolBook también ofrece una solución al problema de acumulación de desarrollo de aplicaciones en las diversas corporaciones. Esta solución se presenta por dos razones; primero, al permitir al programador obtener más trabajo terminado en menos tiempo; segundo, dado que la habilidad requerida para crear una aplicación en ToolBook, es significativamente menor a la que se requiere en el desarrollo de aplicaciones con lenguajes de programación tradicional.

Una de las principales desventajas de ToolBook para Windows, es que no genera archivos ejecutables en Windows (.EXE), por lo que es imprescindible el uso del software de ToolBook para poder ejecutar alguna aplicación desarrollada en él. Esto significa que se debe contar, adicionalmente, con recursos disponibles en la computadora (disco duro, memoria, etc).

### **Visual Basic**

Visual Basic es un sistema de programación gráfico que permite crear aplicaciones de Windows reales con código Basic. Visual Basic proporciona a los programadores que tienen habilidades pero están desfasados con la programación en Windows, las herramientas necesarias para la creación de elementos gráficos que son comúnmente empleados en la mayoría de las aplicaciones en Windows.

El sistema de programación Visual Basic, permite crear objetos, asignar y cambiar sus propiedades y luego agregarlos al código funcional de Basic. La filosofía de programación en Visual Basic consiste, primero, en crear objetos como ventanas, íconos, menús, etc. y luego escribir los procedimientos que invoquen cada objeto.

No se puede decir que el sistema de programación Visual Basic sea un sistema de programación orientado a objetos, dado que tales objetos carecen de las propiedades de herencia y polimorfismo, las cuales deben estar presentes en cualquier ambiente orientado a objetos real. Además, las aplicaciones desarrolladas correrán en Windows que es un ambiente visualmente orientado, que proporciona librerías de herramientas y objetos para programación de aplicaciones.

Una de las ventajas del uso de Visual Basic, es que es posible desarrollar aplicaciones potentes y complejas que los usuarios pueden correr como programas individuales en Windows. Para correr cualquier aplicación en Windows fuera de Visual Basic, se debe crear un archivo ejecutable. Si se distribuye el archivo, se debe instalar la librería de enlace dinámico VBRUN100.DLL, en el directorio de Windows.

Visual Basic reduce drásticamente la dificultad y cantidad de tiempo empleado en la programación en Windows.

### **Herramientas de ayuda en Windows**

Una ayuda es información en línea optimizada para acceso y uso rápido. Generalmente, los usuarios de computadoras consultan este tipo de ayuda, cuando se encuentran con alguna dificultad en el desarrollo de su trabajo.

Debido a que los sistemas de ayuda deben ofrecer al usuario un apoyo rápido y efectivo, actualmente proporcionan numerosas formas de acceso a la información de ayuda y diferentes medios para navegar

desde un t3pico de ayuda a otro. Los sistemas de ayuda son un medio capaz de entregar informaci3n al usuario(Fig. 3-2).

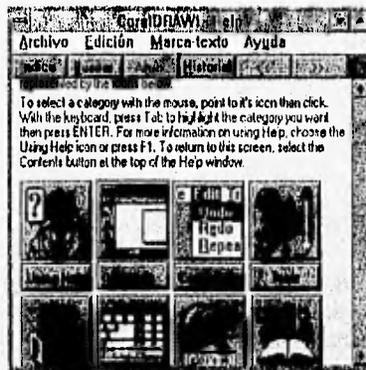


Figura 3-2 Sistema de ayuda proporcionado por Windows.

Los sistemas generados por las herramientas de ayuda, requieren de un programa (WINHELP.EXE) proporcionado por Windows, para poder ser ejecutados. El WINHELP tiene la propiedad de ejecutar cualquier archivo de ayuda (.HLP) de cualquier aplicaci3n de Windows.

Algunas ventajas y desventajas del uso de las herramientas de ayuda en Windows se muestran en la tabla 3-1.

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>◆ Reducen costos de beneficios, de actualizaci3n y de distribuci3n.</li> <li>◆ Agregan flexibilidad al ciclo de desarrollo del producto, al eliminar el retardo requerido para impresi3n.</li> <li>◆ Eliminan potencialmente los manuales multivolumen, proporcionados para productos de software muy grandes.</li> <li>◆ Est3n siempre disponibles.</li> <li>◆ Ofrecen potencial de manejar gr3ficos y potencial de capacidades multimedia como animaci3n, video y voz.</li> <li>◆ Proporcionan medios potentes de acceso a la informaci3n, como el uso de hipertexto y acceso sensitivo de contexto a las librerias de los t3picos de ayuda.</li> <li>◆ Permiten al usuario cortar y pegar c3digos sencillos y otra informaci3n de ayuda dentro de otra aplicaci3n de Windows.</li> <li>◆ Reducen la diferencia psicol3gica entre el uso del producto y el aprendizaje del producto.</li> </ul>	<ul style="list-style-type: none"> <li>◆ El programador requiere habilidad para el desarrollo del sistema de ayuda.</li> <li>◆ Frecuentemente toman la forma de archivos grandes, principalmente cuando se usan ampliamente gr3ficos a color.</li> <li>◆ Deben ser usados en computadora.</li> <li>◆ Hacen el copiado ilegal de software m3s factible.</li> <li>◆ Carecen de mecanismos de lectura familiares, como n3meros de p3gina, encabezados, etc.</li> <li>◆ Compiten por espacio en la pantalla con la interface del producto del software.</li> <li>◆ Deben estar dise1ados para monitores con diferentes dimensiones y capacidades.</li> <li>◆ Hacen problem3tica la inclusi3n detallada de gr3ficos, dado que el 3rea de despliegue es frecuentemente peque1a y la resoluci3n de la pantalla limitada.</li> <li>◆ No son adecuados para desplegar una lectura extendida dada la resoluci3n limitada de la pantalla.</li> <li>◆ Debido a que el texto de la ayuda es poco cohesiva, dada la necesidad de modularidad estricta (la informaci3n de ayuda est3 dividida en t3picos 3nicos ), el flujo y continuidad de la impresi3n de documentos se pierde.</li> </ul>

Tabla 3-1 Ventajas y desventajas del uso de sistemas de ayuda en Windows.

La principal ventaja de los sistemas de ayuda es el acceso de la informaci3n (uso de hipertexto), dado que la computadora puede acceder r3pidamente la informaci3n requerida.

Para que los sistemas generados por herramientas de ayuda incrementen su potencial, pueden servirse del uso de hipergr3ficos, los cuales deben ser generados con el editor hotspot (proporcionado por el compilador de Borland C++ para Windows).

El diseño de sistemas de ayuda o de un tutorial en línea, con el uso de herramientas de ayuda en Windows, puede ser altamente creativo y completamente una tarea atractiva.

Los tutoriales desarrollados con la herramientas de ayuda en línea, cuentan con la ventaja que pueden ser ilustrados por gráficos y animación.

**3.1.3 ¿ Porqué usar herramientas de ayuda para REDTUT ?**

Después de haber conocido las diversas formas posibles de construir una aplicación, es evidente que para el desarrollo de una aplicación en ambiente Windows debe elegirse una técnica soportada para tal tarea. Además, debe considerarse que la técnica más apropiada deba contener las características necesarias para construir una interface adecuada para el aprendizaje (hipertexto, gráficos, animación, etc).

En Windows es posible desarrollar aplicaciones por dos vías principales:

1. Programación tradicional.
2. Constructores de interfaces

Comparando estas dos técnicas, se puede observar que la técnica de programación tradicional tiene algunas desventajas al lado de los constructores de interfaces. Dos desventajas principales de la programación tradicional se refieren a:

- ◆ Al tiempo requerido para el desarrollo de la aplicación.
- ◆ A la gran cantidad de código requerido para desarrollar los medios adecuados para una interface para el aprendizaje (hipertexto, animación, manejo de gráficos, etc).

Por lo tanto, los constructores de interfaces soportados por Windows se convierten en los mejores candidatos para el desarrollo del tutorial de redes. De acuerdo con las características mostradas de los constructores de interfaces en Windows más apropiados (Toolbooks, Visual Basic y herramientas de ayuda), se creó una tabla que muestra las principales ventajas y desventajas de cada constructor (Tabla 3-2).

Constructor de interfaces	Ventajas	Desventajas
<b>ToolBooks</b>	<ul style="list-style-type: none"> <li>◆ Fácil de aprender.</li> <li>◆ Soporta manejo de bases de datos.</li> <li>◆ Soporta hipermedia.</li> <li>◆ Reduce el tiempo de programación.</li> </ul>	<ul style="list-style-type: none"> <li>◆ Se requiere aprender un lenguaje de programación (OpenScript).</li> <li>◆ No genera archivos ejecutables (.EXE), por lo que requiere del ToolBook para su ejecución.</li> </ul>
<b>Visual Basic</b>	<ul style="list-style-type: none"> <li>◆ Es flexible la creación de objetos.</li> <li>◆ Generan archivos ejecutables (.EXE) que pueden ejecutarse fuera de Visual Basic.</li> <li>◆ Reducen el tiempo de programación para el desarrollo de aplicaciones.</li> <li>◆ Soportan el uso de hipermedia.</li> </ul>	<ul style="list-style-type: none"> <li>◆ Se requieren conocimientos de programación en Basic.</li> <li>◆ Para correr los programas fuera de Visual Basic, debe instalarse una librería de enlace dinámico (VBRUN100.DLL).</li> <li>◆ La ejecución de una aplicación dentro de Visual Basic es un poco lenta, dado es interpretada y no compilada.</li> <li>◆ El archivo del programa final es generalmente largo.</li> </ul>

<p><b>Herramientas de Ayuda</b></p>	<ul style="list-style-type: none"> <li>◆ No requieren del conocimiento de algún lenguaje de programación.</li> <li>◆ Es de muy fácil manejo</li> <li>◆ Ofrecen capacidad de manejar multimedia.</li> <li>◆ Proporcionan hipertexto e hipergráficos para acceder la información.</li> <li>◆ El software generado es fácil de copiar y distribuir.</li> <li>◆ Genera archivos ejecutables en Windows (.HLP).</li> <li>◆ Permiten el diálogo con el usuario.</li> <li>◆ Permiten copiar información contenida en el sistema de ayuda a otras aplicaciones.</li> <li>◆ Proporcionan la capacidad de imprimir tópicos de información.</li> </ul>	<ul style="list-style-type: none"> <li>◆ Requiere el uso de WINHELP.EXE.</li> <li>◆ Requiere del uso de un procesador de textos (Word para Windows).</li> <li>◆ Requiere del uso del editor hotspot.</li> <li>◆ El tamaño del archivo final está limitado por el compilador.</li> <li>◆ El uso de gráficos a color incrementa el tamaño del archivo final.</li> <li>◆ Está limitado a ciertas capacidades de monitores.</li> <li>◆ No es posible el uso de gráficos muy grandes.</li> </ul>
-------------------------------------	---	---

Tabla 3-2 Ventajas y desventajas de las técnicas posibles para desarrollar el tutorial de redes.

Comparando las ventajas y desventajas obtenidas con el uso de los constructores de Interfaces mostrados y considerando que el tutorial de redes debe ser un software educativo de fácil manejo, fácil acceso y fácil adquisición, se determina que el empleo de herramientas de ayuda en Windows es la técnica más adecuada para tales fines.

A pesar de que las herramientas de ayuda, usualmente son empleadas para generar ayuda en línea de aplicaciones de Windows, proporcionan todos los requisitos para desarrollar una interface adecuada para el aprendizaje. Además, los sistemas que generan son familiares para el usuario de Windows, lo cual permite al usuario dedicar un tiempo mínimo para aprender el funcionamiento del tutorial.

Los archivos desarrollados con herramientas de ayuda (.HLP) solo requieren de un programa ejecutable (WINHELP.EXE) proporcionado por Windows para ser ejecutados. Pero, como el tutorial de redes a desarrollar requiere del ambiente propio de una aplicación de Windows, es imprescindible el uso de la programación tradicional. Además, la parte de animación, es soportada pero no proporcionada por las herramientas de ayuda, este es otro motivo para el empleo de la técnica de programación tradicional.

Debido a que el desarrollo del sistema tutorial de redes de computadoras, utilizará técnicas de programación tradicional y el apoyo de un constructor de interfaces, tal sistema entra dentro de la clasificación de aplicaciones híbridas.

### 3.2 Manejo de imágenes en Windows.

#### 3.2.1 Formato de Imágenes.

Una imagen es un arreglo de píxeles de dos dimensiones, en ocasiones llamado "raster". En la computadora cada píxel es representado en una de las 3 siguientes formas:

##### 1. Imágenes monocromáticas.

Si la imagen es monocromática, el color de cada píxel es expresado con el valor de un bit (un 1 ó un 0).

##### 2. Imágenes a color.

Si la imagen tiene color, el color de cada píxel es expresado en términos de intensidad de Rojo (R), verde (G) y azul (B) que forman el color. Típicamente, cada componente del color es representado por un byte, es

decir, proporciona 256 niveles para cada componente de color. Esta aproximación requiere 3 bytes para cada pixel y permite más de  $256 \times 256 \times 256 = 16,777,216$  de distintas combinaciones de valores RGB (colores).

### **3. Imágenes basadas en paleta de colores.**

Para imágenes basadas en paletas, el valor de cada pixel es interpretado como un índice dentro de una tabla de valores RGB conocida como paleta de colores o mapa de colores (*colormap*). Para cada pixel, se debe desplegar el valor RGB correspondiente al contenido del pixel. El número de bits requeridos para almacenar el valor de cada pixel depende del número de colores en la paleta. Los tamaños comunes de paletas de colores son 16 (requieren 4 bits por pixel) y 256 (requieren 8 bits por pixel).

#### **Características comunes de los archivos de imágenes.**

Para garantizar en Windows que una imagen almacenada pueda ser leída, interpretada y desplegada, el archivo de imagen debe contener como mínimo la siguiente información:

- ◆ Dimensiones de la imagen.
- ◆ Número de bits por píxeles.
- ◆ Tipo de imagen ( si los valores de píxeles deben ser interpretados como colores RGB o como índices a una paleta de colores)
- ◆ Paleta de colores (si la imagen usa una)
- ◆ Dato de la imagen (se refiere al arreglo de valores de píxeles)

La mayoría de las imágenes contienen este conjunto de información, pero cada formato de archivo específico organiza la información de diferente forma.



Figura 3-3 Formato típico de un archivo de imagen.

El formato de un archivo típico (Fig. 3-3) debe contener:

- ◆ Un encabezado que contiene información acerca de los datos de la imagen y la paleta de colores. Su tamaño puede estar en un rango de 128 bytes.
- ◆ La paleta de colores es agregada en el caso de que los valores de los píxeles en la imagen requieran una paleta.
- ◆ Los datos de la imagen. Usualmente el arreglo de píxeles es almacenado línea por línea.

Sin embargo, el orden de la información en el encabezado, la colocación de los datos de la imagen, el orden RGB, etc. pueden cambiar según el formato de imagen empleado.

### *Formatos de archivo de imagen más comunes*

Existen diversas formas de almacenar una imagen en un archivo. Esto es debido a que existen diferentes tipos de formatos de archivos de imagen, entre los más populares se encuentran:

#### **Formato PCX**

El formato **PCX** fue diseñado antes de la generación actual de tecnología de monitores de PC's. Originalmente es usado por PC PaintBrush de ZSoft, y es un formato de archivo de imagen popular que usan diversas aplicaciones y son soportadas por scanners. Los archivos **PCX** usan una forma de codificación en longitud de corrida (RLE) para almacenar la imagen en forma comprimida.

El formato **PCX** no es eficiente para comprimir imágenes complejas y grandes, dado que puede presentarse una compresión negativa (se requiere de más memoria de la que ocupa la imagen).

#### **Formato BMP**

Windows usa el formato **BMP** para almacenar una imagen como un mapa de bits del tipo DIB (*Device-Independent Bitmap*, Mapa de bits como dispositivo-independiente). El formato DIB permite a Windows desplegar el mapa de bits sobre cualquier tipo de dispositivo despegable. El término "dispositivo independiente" significa que el mapa de bits especifica el color del pixel en una forma independiente al método usado por una pantalla para desplegar color. Además el formato DIB incluye una paleta de colores y almacena los datos de la imagen en una forma estándar, dada la característica de dispositivo independiente.

El formato **BMP** de Windows puede almacenar imágenes con 1 (monocromático), 4 (16 colores), 8 (256 colores) ó 24 (16 millones de colores) bits por pixel.

Los archivos **BMP** son almacenados sin compresión. Esto significa que es posible cargarlos dentro de aplicaciones de Windows y desplegarlos con un mínimo de tiempo de consumo para la manipulación de los datos.

Usualmente los archivos **BMP** son considerablemente grandes. En caso de que el archivo **BMP** tenga dimensiones modestas, pueden ser desplegados rápidamente y con código corto relativamente, con el uso de la función BitBlt.

#### **Formato TGA**

El formato de archivo **TGA** de TARGA "Truevision" de 24 bits, es originado por los adaptadores de pantalla para PC's de alta resolución de "Truevision". Hay varios tipos diferentes de archivos **TGA**, el más popular es un formato de 24 bits por pixel que usa 8 bits para cada componente de RGB. Su formato puede almacenar archivos de imágenes arriba de 16 millones de colores. Sin embargo, el tamaño del archivo de una imagen de 24 bits puede ser muy grande.

#### **Formato TIFF**

El formato **TIFF** (*Tagged Image File Format*) fue desarrollado por la corporación Aldus y Microsoft como un formato de archivo flexible, con independencia del sistema, para almacenar imágenes monocromáticas a través de imágenes de color de 24 bits. La mayoría de los procesadores de texto o paquetes de publicidad, pueden leer y usar imágenes con formato **TIFF**. Adicionalmente, todos los scanners proporcionan software de control que pueden almacenar imágenes en formato **TIFF**.

#### **Formato GIF**

El formato **GIF** (*Graphics Interchange Format*) fue desarrollado por CompuServe para el almacenamiento compacto de imágenes con más de 256 colores. Los archivos **GIF** almacenan imágenes usando la forma de compresión LZW (*Lempel-Ziv and Wels*).

El formato GIF es ideal para almacenar y distribuir imágenes. Una de sus principales desventajas es que en su mayoría, están sujetas a la noción de imágenes de dominio público.

**Formato WMF**

Este formato de archivo WMF se asigna a los meta-archivos de Windows, que consisten de una colección de funciones GDI (*Graphics Device Interface*, Interface de dispositivos gráficos) que describen una imagen. Debido a que los archivos WMF ocupan menos espacio y son dispositivos más independientes que los mapas de bits, proporcionan almacenamiento conveniente para imágenes que aparecen repetidamente en alguna aplicación o requieren ser movidos desde una aplicación a otra.

Para generar archivos WMF, una aplicación de Windows crea un dispositivo de contexto que envía comandos GDI a un archivo o alguna localidad de memoria. La aplicación puede más tarde cargar el meta-archivo y desplegar la imagen.

**Formato MRB**

Un archivo MRB (*Multiple-Resolution Bitmap*, Mapa de bits de resolución múltiple) contiene una imagen en mapa de bits con la capacidad de ser desplegado sobre monitores con diferente resolución (CGA, EGA, VGA y 8514) sin deformar la imagen (forma, tamaño, color, etc.).

Para crear un archivo MRB se deben crear los archivos de mapas de bits para las diferentes resoluciones de pantalla (CGA, EGA, VGA ó 8514) y luego usar el compilador de mapas de bits de resolución múltiple (MRBC) para combinarlos dentro de una imagen de mapa de bits.

**3.2.2 Imágenes con herramientas de ayuda.**

Las imágenes gráficas tienen una función muy importante en los sistemas de ayuda. Es por esto que permiten el manejo de algunas clases especiales de gráficos, entre ellas están:

◆ **Gráficos a color.**

Los sistemas de ayuda en Windows soportan mapas de bits gráficos de 16 colores. Los colores permiten al usuario distinguir e identificar muchos de los objetos de la interface que se muestran en el gráfico.

◆ **Gráficos "hotspot".**

Los gráficos "hotspot", oficialmente llamados como "hipergráficos", son mapas de bits especiales, que cuando son seleccionados despliegan ventanas pop-up, saltan a tópicos relacionados o ejecutan macrofunciones del sistema de ayuda.

◆ **Gráficos con múltiples "hotspot".**

Los gráficos con múltiples "hotspot", oficialmente llamados "hipergráficos segmentados", permiten la creación de múltiples hotspots sobre un sólo mapa de bits (Fig. 3-4).

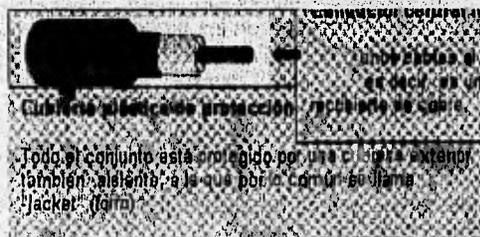


Figura 3-4 Gráfico hotspot.

**Limitaciones en el uso de gráficos en los sistemas de ayuda.**

- ◆ Se requiere de tiempo extra para la creación e incorporación de gráficos del sistema. El programador debe decidir, cuantos y que tipos de gráficos incluir y como usarlos en el sistema.
- ◆ Los gráficos en el sistema de ayuda no pueden ser altamente detallados. Si se cambia el tamaño de la imagen (si son mapas de bits), puede significativamente degradar la calidad de la imagen.
- ◆ El tamaño de los gráficos permitidos es limitado, debido a que el tamaño de los archivos de ayuda está restringido por el compilador de archivos de ayuda.

**Consideraciones:**

- ◆ El compilador de ayuda soporta varios tipos de mapas de bits:
  - .BMP (mapa de bits estándar de Windows)
  - .DIB (mapa de bits independiente de Windows)
  - .WMF (meta-archivo de Windows)
  - .SHG (mapa de bits que contiene los hotspots creados con el editor hotspot)
  - .MRB (mapa de bits de resolución múltiple)
- ◆ El compilador de ayuda de Windows puede leer mapas de bits monocromáticos o mapas de bits de 16 colores. No soporta imágenes con 16 colores.
- ◆ Las imágenes gráficas, especialmente mapas de bits a color de gran tamaño, pueden significativamente incrementar el tamaño del archivo de ayuda. En la tabla 3-3 se muestra como el tamaño y tipo de los mapas de bits afectan el tamaño del archivo de ayuda.

	Tipo de mapa de bits	Tamaño del mapa de bits	Tamaño del archivo de ayuda
<b>Prueba 1</b>	Color	4,330 K	10,894 K
	Monocromático	1,142 K	9,981 K (se reduce 9%)
<b>Prueba 2</b>	Color	128,118 K	82,956 K
	Monocromático	32,062 K	38,317 K (se reduce 54%)

Tabla 3-3 Muestra como el tipo y tamaño de los mapas de bits afecta el tamaño del archivo de ayuda.

**Editor Hotspot (Editor de "zonas-calientes")**

El editor "hotspot" es usado para crear y editar hipergráficos. Un *hipergráfico* es un mapa de bits que incluye uno o más hotspots. Un *hotspot* es una región de un gráfico que está asociada con una acción. Un hotspot tiene una liga asociada ( una liga a salto, a una ventana pop-up, a una ventana secundaria o a una macro). Cuando el usuario selecciona el hotspot, la ayuda ejecuta la acción de acuerdo con el tipo de liga asociado.

Un hotspot puede cubrir cualquier porción de un mapa de bits. Un mapa de bits puede incluir múltiples hotspots que ligan a tópicos de ayuda o ejecutan macros de ayuda, cuando son seleccionados.

Con el editor hotspot, se pueden asignar hotspots a mapas de bits y asignar atributos a cada hotspot (Fig. 3-5). Cada hotspot incluye los siguiente atributos:

- ◆ Cadena de contexto
- ◆ Tipo de liga

- ◆ Bordes de hotspot visible o invisible
- ◆ Identificador del hotspot

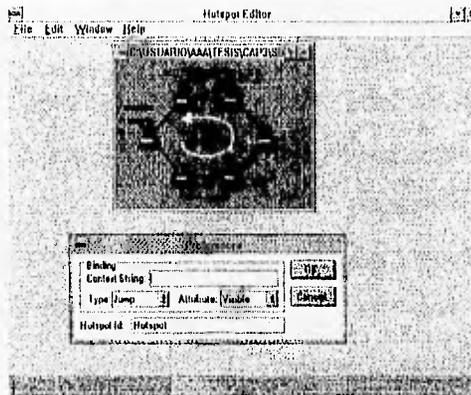


Figura 3-5 Editor Hotspot proporcionado en Windows.

El editor hotspot guarda los hipergráficos en un archivo con formato comprimido especial. Los archivos de hipergráficos generalmente tienen una extensión .SHG.

### 3.2.3 ¿Porqué usar formato .BMP para las imágenes de REDTUT?

A pesar, de que en Windows es posible manejar diversos formatos de imágenes, los sistemas generados con herramientas de ayuda sólo pueden soportar ciertos tipos de formatos de archivos (.BMP, .DIB, .WMF, .SHG, .MRB). En la tabla 3-4 se muestran las ventajas y desventajas de los diferentes formatos de archivos de imágenes, soportados por las herramientas de ayuda en Windows.

Formato	Ventajas	Desventajas
<b>.BMP</b>	<ul style="list-style-type: none"> <li>◆ Es soportado por diversas aplicaciones de dibujo en Windows.</li> <li>◆ Es almacenado como una imagen del tipo dispositivo independiente (DIB).</li> <li>◆ Es cargado y desplegado rápidamente.</li> <li>◆ Es aceptado por las funciones de programación en Windows.</li> <li>◆ Es editado por PaintBrush de Windows.</li> <li>◆ Puede ser usado por el editor hotspot.</li> </ul>	<ul style="list-style-type: none"> <li>◆ Es almacenado sin compresión.</li> <li>◆ Generalmente son archivos grandes.</li> <li>◆ No es posible hacer imágenes demasiado complejas y alta calidad.</li> </ul>
<b>.WMF</b>	<ul style="list-style-type: none"> <li>◆ Es soportado por diferentes aplicaciones de dibujo en Windows.</li> <li>◆ Ocupa menos espacio que los mapas de bits.</li> <li>◆ Es más independiente que los mapas de bits.</li> <li>◆ Es conveniente para imágenes que aparecen repetidamente.</li> <li>◆ Es aceptado por las funciones de programación.</li> <li>◆ Puede ser usado por el editor hotspot.</li> </ul>	<ul style="list-style-type: none"> <li>◆ La imagen esta limitada por las funciones de programación GDI.</li> <li>◆ Requiere de la elaboración de un programa para generar los archivos con este formato.</li> <li>◆ Si es generado por alguna aplicación de dibujo, el tamaño del archivo se incrementa.</li> </ul>
<b>.MRB</b>	<ul style="list-style-type: none"> <li>◆ Es desplegable sobre monitores con diferente resolución sin modificarse su tamaño, forma o color.</li> <li>◆ Puede ser usado por el editor hotspot</li> </ul>	<ul style="list-style-type: none"> <li>◆ Generalmente son archivos grandes.</li> <li>◆ Es formado por varios mapas de bits.</li> <li>◆ Requieren del uso de un compilador especial.</li> </ul>

Tabla 3-4 Ventajas y desventajas de formatos de imágenes soportados por herramientas de ayuda.

**NOTA:**

- ◆ El formato SHG de una imagen es creado por el editor hotspot, después de que a una imagen con formato BMP, WMF o MRB se le asigna uno o varios hotspots.
- ◆ El formato DIB tiene las características del formato BMP, sólo que es soportado por menos aplicaciones de dibujo en Windows.

Podría pensarse que el formato MRB es el más indicado para desplegar imágenes, dado que son independientes del tipo de monitor. Sin embargo, las imágenes que el tutorial usará no son muy detalladas o complejas por lo que el uso del formato MRB no sería muy necesario. Además, la mayoría de los monitores que actualmente se utilizan son VGA o mejores (Super VGA, Ultra VGA, etc.), descartando de esta manera el uso de monitores más antiguos (CGA y EGA), que sería la razón principal de usar un formato MRB.

Por otro lado, dadas las desventajas del formato WMF, no sería adecuado utilizarlo por lo complejo y tardado que resultaría.

En el caso del formato BMP, se tiene como principal desventaja el gran tamaño de los archivos, sin embargo, es posible controlar este problema de tal forma de que las imágenes sean pequeñas dentro del tutorial. Adicionalmente se cuenta con la herramienta necesaria para editar imágenes BMP (PaintBrush) y también es soportado por la técnica de programación tradicional (requerido para el uso de animación). Es por esto que se determina que el uso de imágenes con formato BMP es el más adecuado para el tutorial de redes generado por herramientas de ayuda en Windows.

### 3.3 Técnicas de animación en ambiente Windows.

#### 3.3.1 Implementación y control de la animación.

La plataforma de programación en el ambiente gráfico de Windows es enriquecida con el potencial de la animación. La animación en computadoras personales se organiza en dos categorías fundamentales:

1. **Implementación.** Se relaciona con los mecanismos para crear ilusión de movimiento sobre la pantalla.
2. **Control.** Se relaciona con el manejo de objetos y eventos que serán animados.

#### **Implementación**

Windows proporciona varios métodos diferentes para implementación de animación. Los 4 métodos más prácticos son:

1. Animación de ciclo fijo (*Fixed-loop animation*).
2. Animación de ciclo de espera (*Idle-loop animation*).
3. Animación con manipulador de mensajes basada en el reloj (*Timer-based message-handler animation*).
4. Animación de llamada directa basada en el reloj (*Timer-based direct-call animation*).

Para cada uno de estos cuatro métodos, existe un algoritmo particular que puede ser usado para animación de tiempo real o para animación por cuadros de imágenes, y pueden ser controlados por cualquiera de los tres métodos proporcionados.

Independientemente del método que se utilice, Windows proporciona algunas herramientas de bajo nivel para la implementación de animación. Tales herramientas tienen que ver con la manipulación y almacenamiento de arreglos gráficos (mapas de bits) y básicamente son tres:

1. **Bitbltting** : Se refiere al proceso de pegar una mapa de bits pequeño (llamado "bitblt") sobre la pantalla, ya sea desde una localidad de memoria a pantalla, de la pantalla a una localidad de memoria, desde una localidad de memoria a otra o desde una posición de pantalla a otra, mediante el uso de la función BitBlt().

2. **Copiando páginas** : Se refiere al proceso de copiar un mapa de bits del tamaño de una ventana completa. Una página es un buffer de memoria o también llamada cuadro de imagen.
3. **Matices y cambios** : El GDI proporciona operadores de rastreo que producen diferentes efectos visuales, cuando se escribe un mapa de bits en una nueva posición. Los operadores de rastreo usan lógica booleana.

### **Control**

Los tres métodos principales de control de animación, proporcionados por Windows son:

1. Animación por guión (*Scripted animation*)
2. Animación por procedimiento (*Procedural animation*)
3. Animación basada físicamente (*Physically based animation*)

#### **Animación por guión**

Es la versión asistida por computadora de la animación tradicional por película. Como su nombre lo dice es gobernada y dirigida por un guión. Este tipo de control de animación es también llamado control explícito, animación por celdas, animación convencional, animación tradicional o animación por película.

El control de animación por guión, debe considerar tres reglas importantes:

1. *Deformación*: Se refiere a la acción de aplastar o estirar un objeto durante la animación.
2. *Cámara de mecanismos*: Se refiere al punto de vista del movimiento.
3. *Puesta en escena*: Se refiere al ritmo, desarrollo del argumento, accesorios, escenario, etc. de la animación.

#### **Animación por procedimiento**

Se refiere a una animación orientada a objetos. Durante la reproducción de animación, el software prepara el próximo cuadro de imagen con el cálculo de la próxima posición para cada objeto en escena. Los cálculos están frecuentemente basados en reglas del ambiente definido por el programador. Estas reglas pueden describir no solo el entorno de un objeto, sino también el entorno del objeto en relación con otros en escena.

#### **Animación basada físicamente**

La animación basada físicamente, emplea leyes físicas para gobernar el movimiento de objetos durante la reproducción de animación. El movimiento en un sistema de animación basado físicamente es modelado usando restricciones. Una restricción es una condición o fuerza limitante. Este tipo de animación esta relacionada con posiciones, velocidades, fuerzas, masas, etc. y consiste de 4 categorías:

1. *Avances cinemáticos*: Proceso de calcular el resultado de la aplicación de velocidad o aceleración a un objeto.
2. *Avances dinámicos*: Proceso de calcular el resultado de la aplicación de fuerzas, cargas o restricciones a un objeto.
3. *Retrocesos cinemáticos*: Proceso de calcular la velocidad o aceleración requerida para mover un objeto.
4. *Retrocesos dinámicos*: Proceso de calcular las fuerzas o restricciones requeridas para mover un objeto.

### 3.3.2 Algoritmos para animación.

Los algoritmos que proporciona Windows para implementar animación se muestran la siguiente figura (Fig. 3-6):

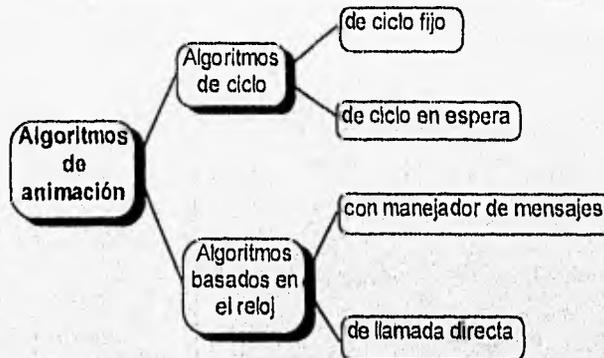


Figura 3-6 Principales algoritmos de animación proporcionados por Windows.

#### **Algoritmos de ciclo**

##### *Algoritmo de ciclo fijo*

Un ciclo fijo es un bloque de código en "C" que se ejecuta repetidamente por un número predeterminado de iteraciones. El código está usualmente contenido en el núcleo de las funciones de la aplicación. Mientras que el ciclo se está ejecutando, otras aplicaciones que pueden estar ejecutándose en el ambiente de Windows, están deshabilitadas para compartir el CPU. Un código de ciclo fijo es usado para exponer cortos de animación, donde la brevedad de la secuencia de animación no afectará a otras aplicaciones que quizás estén ejecutándose.

##### *Algoritmo de ciclo en espera*

Este algoritmo consiste en llamar a las funciones "GetMessage( )" (recibe mensajes) y "DispatchMessage( )" (envía mensajes), en ese orden, para sondear la cola de la aplicación y verificar si hay entradas por parte del usuario durante el ciclo. El siguiente código agregado al ciclo fijo crea un ciclo en espera:

```

if (GetMessage(&msg,0,0,0))
{
    if (TranslateAccelerator (MainhWnd, hAccel, &msg))
        continue;
    TrasslateMessage (&msg);
    DispatchMessage (&msg);
}
  
```

**Algoritmos basados en el reloj**

Los algoritmos basados en el reloj ofrecen el medio más efectivo de controlar la reproducción de una secuencia de animación (Fig. 3-7 ). Para esto, se hace una llamada a la función SetTimer( ) en donde se especifica un intervalo de tiempo. Windows envía una entrada de reloj a la aplicación actual, cada vez que el intervalo concluye. La entrada del reloj puede tomar dos formas:

1. Con manipulador de mensajes.
2. De llamada directa.

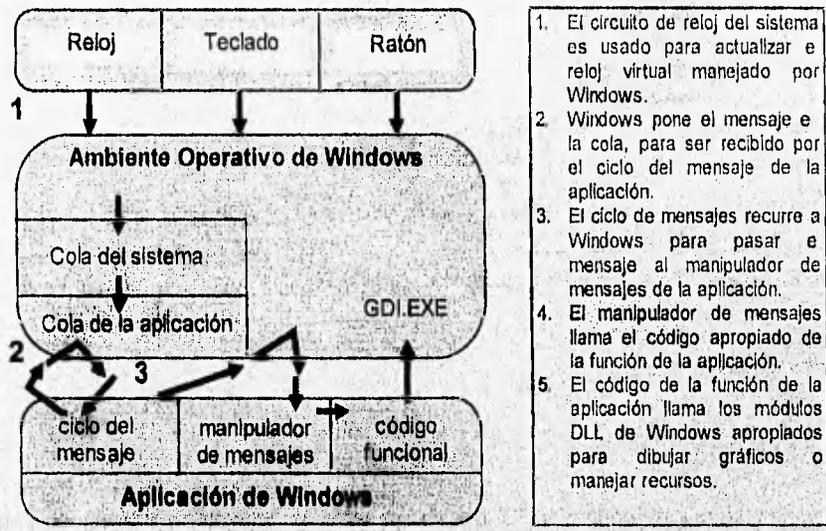


Figura 3-7 Animación basada en el reloj del sistema.

**Animación con manipulador de mensajes**

Este método proporciona buen desempeño en tiempo de corrida y permite el control exacto de la velocidad de reproducción, mediante el ajuste de los intervalos de tiempo en la función SetTimer( ) de API. El método de manipulador de mensajes es versátil, dado que proporciona dos caminos diferentes para desplegar el próximo cuadro de imagen:

1. El manipulador de mensajes puede llamar la función apropiada, cuando un mensaje del reloj (WM\_TIMER) es recibido desde Windows.
2. Cualquier función de alguna aplicación puede llamar la función que despliega cuadros de imágenes cuando esta lo requiera.

**Animación de llamada directa**

Al declarar una función como EXPORTS en el archivo de definición (.DEF) se obliga a Windows a llamar la función directamente, en cada entrada del reloj. Por lo tanto, los pasos a seguir en este algoritmo son:

1. En el archivo de definición .DEF, se declara la función como EXPORTS (la función recibe el nombre de "función de llamada"):

```
EXPORTS zTimed
```

2. Declarar la función de llamada en el encabezado del archivo:

```
FAR PROC lpfn zTimed
```

3. Cuando la aplicación es iniciada, se debe hacer una llamada a `MakeProcInstance()`, para permitir a Windows inicializar una liga con la dirección de la función de llamada:

```
lpfn zTimed=MakeProcInstance(zTimed, hInstance);
```

4. En tiempo de corrida el código de la aplicación hace una llamada a `SetTimer()`, para especificar el intervalo de tiempo e indicar la función que se desea que Windows llame directamente:

```
SetTimer (hWnd, 1, wFrameRate, lpfnzTimed)
```

### 3.3.3 Animación en tiempo real.

La animación en tiempo real, también llamada "animación basada en reparto" (*cast-based animation*), usa imágenes dinámicas. Durante la reproducción de la animación, cada elemento o imagen es dibujado por el software al mismo tiempo que la animación es desplegada sobre la pantalla. La animación en tiempo real puede ser implementada por tres formas:

1. Bitblt de pantalla hacia la pantalla
2. Bitblt de un cuadro de imagen oculto hacia la pantalla
3. Dibujo de un cuadro de imagen oculto hacia la pantalla

#### **Bitblt sobre la pantalla hacia la pantalla**

Si un mapa de bits pequeño (bitblt), es copiado o movido desde una posición a otra en la pantalla, los mecanismos de prevención de corrupción de la pantalla, frecuentemente hacen la técnica impráctica. Es decir, si un bitblt es movido, el fondo de la posición previa debe ser restaurado. El parpadeo es inevitable. Si el bitblt es copiado o movido, una función `PUT` transparente debe usualmente ser invocada, para garantizar que el bitblt es pegado limpiamente en la nueva posición.

#### **Bitblt de un cuadro de imagen oculto hacia la pantalla**

Si los bitblts ya han sido creados y almacenados en memoria, pueden ser usados para construir una imagen completa sobre un cuadro de imagen oculto. El cuadro de imagen oculto, es copiado para ser desplegado en la ventana como el próximo cuadro de imagen en la secuencia de la animación.

Este método puede producir buen desempeño, pero algunas veces limitan al usuario al interactuar con la animación debido a que todas las imágenes han sido creadas con anticipación.

#### **Dibujo de un cuadro de imagen oculto hacia la pantalla**

El método más versátil de implementación en hardware, se relaciona con el dibujo de cada imagen sobre un cuadro de imagen oculto en tiempo de corrida. Cada imagen completada, es copiada para ser desplegada en una ventana durante la reproducción de la animación. Este método proporciona un rico potencial para interacción con el usuario con la animación en tiempo de corrida, pero el proceso de dibujo de cada nuevo cuadro de imagen puede ser tiempo consumido.

### 3.3.4 Animación por cuadros de imágenes.

La animación por cuadros de imagen usa imágenes estáticas. Durante la reproducción de la animación el software trae y despliega la imagen que ha sido previamente completada y almacenada en disco. Las rutinas de la animación por cuadros de imagen, son frecuentemente muy pequeñas, de tal manera que es posible cargar una secuencia de animación completa dentro de la memoria, siempre y cuando haya suficiente

espacio en memoria física. Por otro lado, las rutinas cargarán cada cuadro de imagen directamente desde el disco, según cada cuadro de imagen sea requerido durante la reproducción.

La creación y reproducción de animación por cuadros de imágenes, puede ser implementada por tres formas:

1. Bitblt de pantalla hacia un cuadro de imagen almacenado
2. Dibujar de pantalla hacia un cuadro de imagen almacenado
3. Reproducir cuadros de imagen para desplegar un pantalla

***Bitblt de pantalla hacia un cuadro de imagen almacenado***

Las imágenes bitblt (mapas de bits pequeños), son usados para construir cada cuadro de imagen completo, sobre la pantalla en una ventana. El software luego guarda la imagen desde la pantalla al disco duro. Si la aplicación requiere crear nuevos cuadros de imagen, en tiempo de corrida, el proceso puede ser invisible para el usuario al crear cada nueva imagen sobre una página oculta y guardar cada nuevo cuadro de imagen completo en un disco duro o en memoria.

***Dibujar de pantalla hacia un cuadro de imagen almacenado***

Cuando crear imágenes bitblt previamente resulta impráctico, frecuentemente se usa código que dibuje cada cuadro de imagen de la nada sobre la pantalla. El software luego guarda cada cuadro de imagen completo, desde la pantalla al disco duro. Si los nuevos cuadros de imágenes son creados en tiempo de corrida, el proceso puede ser ocultado al usuario, al construir cada nuevo cuadro de imagen sobre una página oculta y guardarlo en disco duro o en memoria.

***Reproducir cuadros de imagen para desplegar en pantalla***

La reproducción de una secuencia de animación por cuadro de imagen, es usualmente desde disco a pantalla o desde memoria a pantalla. Los cuadros de imagen que componen la secuencia de animación, son usualmente almacenados en disco duro. Si la memoria lo permite, las rutinas de reproducción usualmente intentarán cargar todos los cuadros de imagen dentro de memoria, desde donde pueden ser rápidamente desplegados, al ser copiados en secuencia hacia la ventana de la pantalla. Si la memoria no permite una precarga, las rutinas de animación estarán obligadas a cargar cada cuadro de imagen desde disco, según sea requerido durante la reproducción. El buen desempeño de la animación está directamente relacionado con el desempeño del disco duro.

Con este método de animación la cantidad de memoria física y el modo de memoria con el que opera Windows, determinan si la secuencia de una animación se ejecutará desde disco (basada en disco) o desde RAM (basada en RAM) (Tabla 3-5).

	<b>Modo real</b>	<b>Modo estándar</b>	<b>Modo aumentado</b>
80386SX 16 Mhz 2 MB en RAM	basada en disco	basada en disco	basada en disco
80386DX 33 Mhz. 4 MB en RAM	basada en disco	basada en RAM	basada en RAM

Tabla 3-5 Ejecución de animación según capacidad de la máquina y el modo de operación.

3.3.5 ¿Qué técnica debe utilizarse en la animación de REDTUT?

El papel que tendrá la animación en el tutorial de redes, es el de ejemplificar procesos o funciones que tengan relación con el uso de las redes de computadoras (flujo de datos, corrección de errores, etc.) Por lo que no se requiere de una técnica de animación muy sofisticada para tales efectos.

Cualquiera de las técnicas de animación, mostradas anteriormente, es capaz de soportar el grado de animación que se desea proyectar, sin embargo, no cualquiera es adecuada para ser empleada en el tutorial. Por tal motivo, es necesario hacer una comparación de las diferentes técnicas de animación, algoritmos y formas de control, para elegir el camino más adecuado. Las tablas de comparación se muestran a continuación:

Técnica de animación	Ventajas	Desventajas *
<b>En tiempo real</b>	<ul style="list-style-type: none"> <li>◆ No requiere de espacio en disco duro para almacenar secuencias de animación.</li> <li>◆ Usa imágenes dinámicamente.</li> <li>◆ Es posible la animación de bitblts.</li> <li>◆ Su eficiencia no depende de almacenamiento secundario.</li> <li>◆ Hace posible la interacción del usuario con la animación.</li> </ul>	<ul style="list-style-type: none"> <li>◆ Si se usan bitblt, el parpadeo en el movimiento es frecuentemente inevitable.</li> <li>◆ Si se utilizan cuadros de imagen ocultos, la animación puede ser lenta.</li> <li>◆ Si se usa movimiento de bitblts, es necesario restaurar el fondo de la posición previa.</li> </ul>
<b>Por cuadros de imagen</b>	<ul style="list-style-type: none"> <li>◆ Las rutinas que producen la animación son pequeñas.</li> <li>◆ Existe la posibilidad de reproducir la secuencia de animación rápidamente.</li> <li>◆ La creación de cuadros de imagen puede darse en tiempo de corrida.</li> </ul>	<ul style="list-style-type: none"> <li>◆ Dado que se usan imágenes estáticas, se requiere de almacenamiento previo de las imágenes.</li> <li>◆ Se requiere de suficiente espacio en memoria (RAM), para almacenar una secuencia de animación completa.</li> <li>◆ Requiere de espacio en disco duro.</li> <li>◆ Su eficiencia depende de la eficiencia del disco duro.</li> <li>◆ No es fácil la interacción del usuario con la animación.</li> </ul>

Tabla 3-6 Comparación de las técnicas de animación en Windows.

Algoritmos	Ventajas	Desventajas
<b>Ciclo fijo</b>	<ul style="list-style-type: none"> <li>◆ Solo se requiere de un bloque de código pequeño.</li> <li>◆ Es posible determinar el tiempo que dure la animación.</li> </ul>	<ul style="list-style-type: none"> <li>◆ No comparte el uso del CPU con otras aplicaciones, mientras se está ejecutando el ciclo.</li> <li>◆ No es posible variar la velocidad de reproducción.</li> <li>◆ Solo es adecuada para secuencias de animación breves.</li> </ul>
<b>Ciclo en espera</b>	<ul style="list-style-type: none"> <li>◆ Solo se requiere de un bloque de código pequeño.</li> <li>◆ La animación puede prolongarse hasta que el usuario decida.</li> </ul>	<ul style="list-style-type: none"> <li>◆ No comparte el uso de CPU con otras aplicaciones, mientras se está ejecutando el ciclo.</li> <li>◆ No es posible variar la velocidad de reproducción.</li> </ul>
<b>Basada en reloj con manipuladores de mensajes</b>	<ul style="list-style-type: none"> <li>◆ Controla exactamente la velocidad de reproducción de la animación</li> <li>◆ Su desempeño en tiempo de corrida</li> <li>◆ El despliegue del próximo cuadro de imagen, puede ser activado por Windows o por la llamada de una función.</li> </ul>	<ul style="list-style-type: none"> <li>◆ Requiere de mas control en las rutinas para generar animación.</li> </ul>

**CAPITULO 3: ANALISIS DE HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES EN WINDOWS**

<b>Basada en reloj con llamada directa</b>	<ul style="list-style-type: none"> <li>◆ El despliegue del próximo cuadro de imagen, es automáticamente controlado por Windows.</li> <li>◆ Controla exactamente la velocidad de reproducción de la animación</li> </ul>	<ul style="list-style-type: none"> <li>◆ Requiere de más control en las rutinas para generar animación.</li> </ul>
--	---	--

Tabla 3-7 Comparación de los algoritmos proporcionados por Windows para generar animación.

<b>Tipo de control</b>	<b>Ventajas</b>	<b>Desventajas</b>
<b>Por guión</b>	<ul style="list-style-type: none"> <li>◆ Es muy familiar, debido a que es gobernada como una animación tradicional.</li> <li>◆ No se requieren de cálculos.</li> <li>◆ Puede controlar un grado de animación muy complejo.</li> </ul>	<ul style="list-style-type: none"> <li>◆ Deben considerarse diversos parámetros, tanto del objeto como del ambiente del objeto que se va a animar.</li> </ul>
<b>Por procedimiento</b>	<ul style="list-style-type: none"> <li>◆ Esta basada en reglas definidas por el programador.</li> <li>◆ Puede manipular los objetos por separado.</li> </ul>	<ul style="list-style-type: none"> <li>◆ Se requieren de cálculos para el despliegue del próximo cuadro de imagen.</li> </ul>
<b>Basada físicamente</b>	<ul style="list-style-type: none"> <li>◆ La animación se acerca más a movimientos reales.</li> </ul>	<ul style="list-style-type: none"> <li>◆ Se requiere del uso de leyes físicas.</li> <li>◆ Se requiere de cálculos.</li> </ul>

Tabla 3-8 Comparación de los tipos de control de animación proporcionados por Windows.

Después de haber analizado las ventajas y desventajas de los diferentes algoritmos, técnicas y formas de controlar la animación en Windows, se determina que el elegir una u otra forma, depende inherentemente del proceso o función que se desea animar. Por lo tanto, la técnica de animación a emplear, será distinta según el grado de complejidad del proceso o función en cuestión. Pero sí deben considerarse algunos puntos de importancia, que son los que determinarán el uso o no, de alguna técnica de animación:

- ◆ Evitar al máximo el uso de almacenamiento secundario (disco duro).
- ◆ Que sea posible que el usuario interactue con la animación.
- ◆ No es tan relevante que la animación sea físicamente real.
- ◆ Generalmente, es necesario controlar la velocidad de la reproducción de la animación.

**P A R T E    I I I**  
**D E S A R R O L L O**

# **CAPITULO 4**

## **DISEÑO DE REDTUT**

**4.1** Diseño de REDTUT con herramientas de ayuda en Windows.

**4.2** Creación de texto, imágenes y animación.

**4.3** Diseño de la navegación.

**4.4** Creación de ayuda en línea y glosario.

## 4.1 Diseño del tutorial de redes con herramientas de ayuda.

### 4.1.1 Recursos para manejo de información.

Algunas de las características más importantes a considerar, en las interfaces para el aprendizaje, son la presentación y las diferentes formas de manejar la información. Para tal fin, las herramientas de ayuda proporcionan los siguientes recursos:

- ◆ **Barra de menú:** Incluye comandos que se relacionan con el manejo de la información (abrir, cerrar, imprimir, marcar, etc.).
- ◆ **Tópico de contenido:** El tópico de contenido es una tabla del contenido de toda la información proporcionada por el sistema. Puede ser accesado por el botón de contenido o por la función de programación WINHELP( ).
- ◆ **Cuadro de diálogo de búsqueda:** Despliega una lista de palabras claves, la cual es registrada cuando se crean los tópicos de información. Al elegir una palabra clave accedemos el tópico de información con el que se relaciona.
- ◆ **Secuencia de páginas (Browse Sequence):** Permite al usuario navegar hacia adelante o hacia atrás, a través de una secuencia predefinida en los tópicos de información. Puede ser activada con los botones marcados con << (pág. previa) o >> (pág. siguiente).
- ◆ **Regreso (Back):** Regresa al usuario al tópico que ha sido más recientemente mostrado. Puede ser activado con el botón de regreso y en caso de ser presionado repetidamente, permite al usuario retroceder sus pasos a través del sistema.
- ◆ **Historia:** Despliega una lista de los últimos 50 tópicos de información que el usuario ha accesado y permite al usuario regresar a cualquiera de ellos.
- ◆ **Marcador de páginas (Bookmark):** Permite al usuario establecer una lista de tópicos, que el usuario necesite acceder nuevamente. Este servicio puede ser accesado en la barra de menú.
- ◆ **Acceso por contexto-sensitivo:** Se refiere a la capacidad de acceder un tópico de información relacionado con algún objeto sobre la interface que esté actualmente seleccionada. Es una forma muy rápida de acceder información relevante. Este servicio es activado por medio de las teclas F1 y shift+F1, principalmente.
- ◆ **Salto:** Este servicio es activado desde los tópicos de información, ya sea por algún texto "resaltado" o algún "hotspot" de alguna imagen. Su función es saltar de un tópico de información a otro.
- ◆ **Ventanas pop-up:** Son ventanas pequeñas que se despliegan instantáneamente, sobre la ventana que la genera, con el fin de proporcionar información breve. Este servicio es proporcionado desde los tópicos de información al seleccionar texto "resaltado" o "hotspot" de imágenes. Son cerrados si se da un click en cualquier parte de la ventana.
- ◆ **Ventanas secundarias:** Estas ventanas, a diferencia de las pop-up, permanecen desplegadas, hasta que son cerradas por el usuario. Las ventanas secundarias son activadas con un click en algún texto "resaltado" o algún hotspot de alguna imagen en algún tópico de información.

Algunos beneficios obtenidos con los recursos de acceso proporcionados por las herramientas de ayuda son:

- ◆ El usuario puede hacer anotaciones en los tópicos de información.
- ◆ El usuario puede copiar y pegar desde el sistema generado por herramientas de ayuda a cualquier aplicación de Windows.
- ◆ El usuario puede imprimir tópicos de información. Desafortunadamente, la impresión facilitada es primitiva y sólo imprime tópicos uno a la vez.
- ◆ El usuario puede tener varios sistemas generados con herramientas de ayuda al mismo tiempo.
- ◆ El programador tiene la posibilidad de agregar una variedad de servicios especiales, como es el crear un botón que active alguna tarea particular en el sistema (uso de macros).

Adicionalmente a los recursos de acceso, las herramientas de ayuda proporcionan 50 comandos o macros que permiten al programador extender la potencia del sistema a desarrollar. Estas macros se relacionan con las siguientes funciones:

- ◆ Modificar la barra de botones y la barra de menú
- ◆ Acceso a los botones estándares y elementos de menú
- ◆ Salto o despliegue de un tópico en una ventana pop-up
- ◆ Manejo de la ventana del sistema
- ◆ Agregar acceso del teclado para macros
- ◆ Crear y manipular marcadores de texto
- ◆ Ejecutar aplicaciones de Windows
- ◆ Registrar una función en una librería de enlace dinámico (DLL) y usarla en una macro

#### **4.1.2 Tipos de acceso a la información**

Una de las principales ventajas del uso del hipertexto, es el acceso rápido a la Información. Las herramientas de ayuda proporcionan diversas formas de acceder los tópicos de información (módulos de información independientes) creados por el programador, las cuales están orientadas a hipertexto. Las formas principales de acceso a la información son:

- a) Acceso por tópico de contenido
- b) Saltos
- c) Secuencia de página
- d) Regreso, historia y marcador de páginas
- e) Anotación
- f) Búsqueda por lista de claves
- g) Contexto sensitivo

##### **Acceso por tópico de contenido**

El tópico de contenido es el primero en desplegarse cuando se ejecuta un archivo .HLP. Por tal motivo, es muy común que contenga una visión de la estructura de la información de todo el sistema (semejante a un Índice).

El tópico de contenido puede ser representado como el tope de niveles, de un diagrama jerárquico, donde todos los tópicos de información están en el fondo del diagrama (Fig. 4-1).

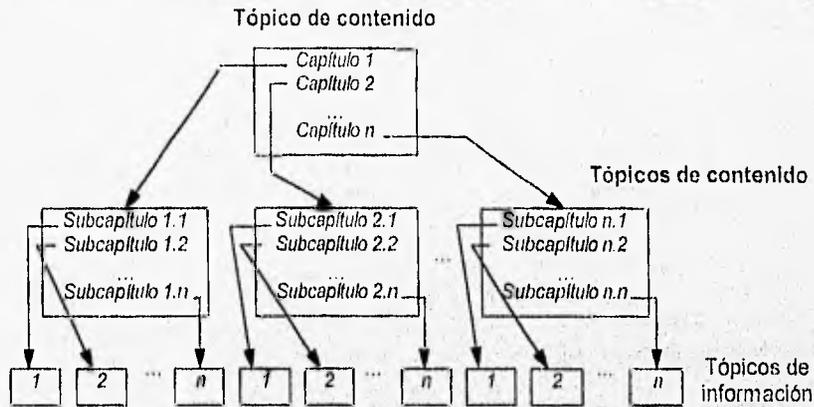


Figura 4-1 Acceso por tópicos de contenido.

**Salto**

Los saltos son enlaces electrónicos usados principalmente para ligar tópicos de información. Debido a la naturaleza modular de la información, es posible saltar de un tópico de información a otro, obteniendo de esta forma una lectura no-secuencial, que puede resultar atractiva para el programador. Es por esto, que a diferencia del acceso de tópicos de contenido, los saltos no son estructurados, dado que sólo son usados si el programador decide conectar un tópico con otro (Fig. 4-2).

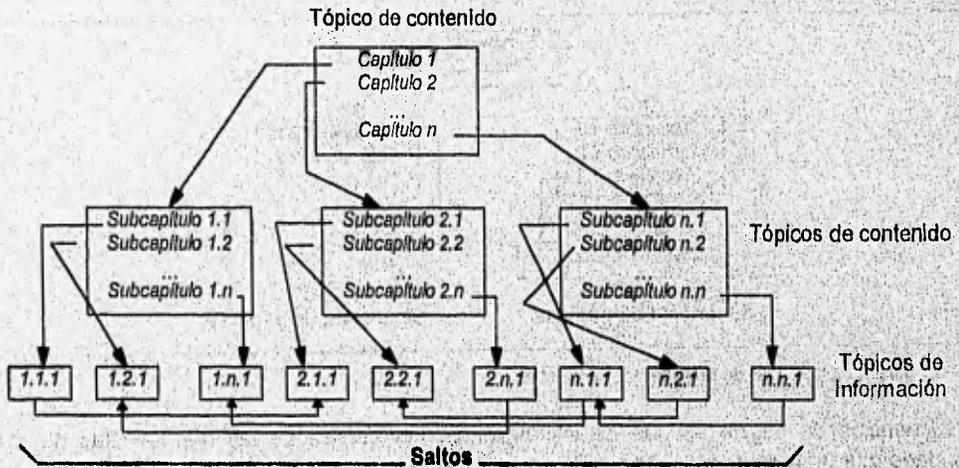


Figura 4-2 Saltos agregados a una estructura jerárquica proporcionada por tópicos de contenido.

En caso de que el acceso de la información sea únicamente por saltos, este toma la forma de un hipertexto sin estructura "enredado" (Web), en la cual se suprime la estructura jerárquica proporcionada por los tópicos de contenido (Fig. 4-3).

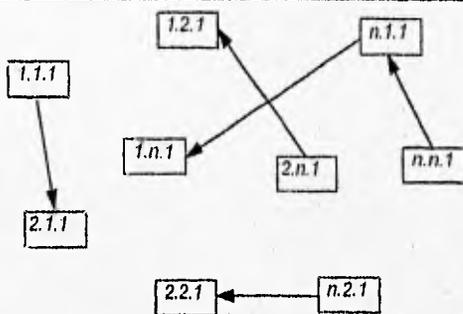


Figura 4-3 Muestra la estructura de un hipertexto sin estructura ("enredada").

Los saltos pueden ser colocados en cualquier parte de un tópico de información o listados en ventanas pop-up, que son desplegadas desde los tópicos de información. Las ventanas pop-up están muy relacionadas con los saltos, con la diferencia de que las ventanas pop-up representan una navegación restringida; del tópico de información a la ventana pop-up y viceversa. Es posible modificar esta navegación insertando saltos o ventanas pop-up, dentro de las ventanas pop-up.

Otro tipo de ventanas que están relacionadas con los saltos, son las secundarias, cuya diferencia con respecto a las ventanas pop-up, es que permanecen visibles aún si la ventana "padre" es cerrada.

### Secuencia de páginas

La secuencia de páginas es una forma de simular la lectura secuencial de un libro pero con los tópicos de información. Las herramientas de ayuda permiten el manejo de múltiples secuencias de páginas independientemente (Fig. 4-4).

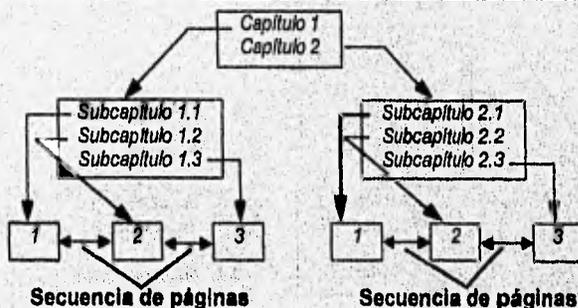


Figura 4-4 Muestra secuencias de páginas independientes.

Las secuencias de página es uno de los aspectos más débiles de las herramientas de ayuda en Windows, dado que un mismo tópico no puede aparecer en dos lugares en una misma secuencia o en más de una secuencia. Otra de las limitaciones de peso, es que no hay algún indicador de cual es el tópico próximo o previo y que no hay forma de ver todos los tópicos construyendo una sola secuencia de páginas.

### Regreso, Historia y Marcador de páginas

Con el botón de "regreso" se permite al usuario retroceder sus pasos a través del sistema. Es decir, permite regresar a los tópicos de información en secuencia inversa.

Con el botón de "historia" se despliega una lista de los últimos 50 tópicos de información que se han desplegado, con el fin de que el usuario pueda regresar a cualquier tópico de información sin necesidad de regresar uno a la vez.

Con el marcador de páginas se le permite al usuario "marcar" uno o más tópicos de información en una lista, para tener la posibilidad, en un futuro, de desplegarlos rápidamente.

**Anotación**

Dentro de la teoría de hipertexto es muy atractivo el concepto de hipertexto como un documento "abierto", que puede ser modificado por el usuario. Con este recurso proporcionado por herramientas de ayuda, es posible hacer anotaciones que se consideren importante, en cada tópico de información, de forma muy semejante a los garabatos hechos en los manuales impresos.

**Búsqueda por lista de claves**

La búsqueda por lista de claves es muy semejante a los índices "alfabéticos" de los libros, dado que permiten al usuario acceder el tópico de información de un tema específico (Fig. 4-5). Desafortunadamente, el crear la lista de claves, es una tarea que requiere de alto control.

Cuando se despliega la lista de claves, es posible la búsqueda "por cadena", donde se debe seleccionar el tema específico que se requiera y en el acto se mostrarán todos los tópicos relacionados, dando la posibilidad de elegir alguno. Otra posibilidad es la búsqueda "booleana", donde se puede preguntar por algún tópico de información tecleando sólo una palabra relacionada con el tema a buscar. En la búsqueda booleana se requiere que el usuario tenga cierto conocimiento del contenido del sistema.

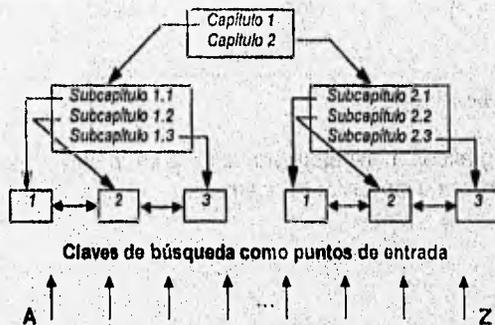


Figura 4-5 Búsqueda por lista de claves;

**Contexto sensible**

El acceso por contexto sensible, consiste en el uso de las teclas F1 y la combinación Shift + F1, para evitar el tópico de contenido y el cuadro de diálogo de búsqueda, con el fin de proporcionar rapidez y acceso directo al tópico de información deseado. En esta forma de acceso, el usuario no busca activamente la información de ayuda, sino que una ventana particular, una caja de diálogo o algún otro estado visible del sistema, determina cual tópico de información debe ser desplegado cuando el usuario oprime las teclas mencionadas. Debido a este enlace con el estado del sistema, el contexto sensible puede ser el principio de una forma rudimentaria de ayuda "inteligente".

**4.1.3 Tópicos de información**

Las herramientas de ayuda manejan la información en tópicos de información, donde un tópico de información es un módulo independiente de información. Tal información, generalmente, debe estar rigurosamente controlada, tanto en terminología como en formato. Sin embargo, los tópicos que se emplearán en REDTUT, tienen que ser diseñados según la función que vayan a desempeñar y el tipo de información que contendrán. Los tópicos de información que se requieren para el tutorial de redes son:

1. Tópico de Índice
2. Tópico de contenido
3. Tópico de introducción
4. Tópico de tema
5. Tópico de temas múltiples
6. Tópico de tema final
7. Tópico de evaluación

**Tópico de Índice**

El tópico de Índice tendrá la misma función que desempeña el tópico de contenido en los sistemas de ayuda. Como su nombre lo indica, contendrá un índice general desglosado del contenido del tutorial, con el fin de acceder cualquier tema con sólo seleccionarlo (Fig. 4-6).

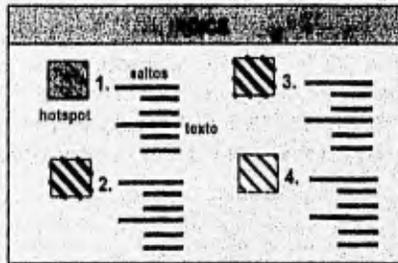


Figura 4-6 Tópico de Índice del tutorial de redes.

Este tópico utilizará texto, gráficos, hotspots y ventanas pop-up para desplegar la información. Para el acceso de la información se servirá de los saltos básicamente.

**Tópico de contenido**

La información que debe abarcar el tópico de contenido, se concretará a un índice que permita iniciar la navegación de un capítulo o un subcapítulo del tutorial.

El índice será representado como un arreglo de mapas de bits y texto, que indicarán gráfica y textualmente, el título del capítulo o subcapítulo en cuestión, los cuales serán accedados con saltos (Fig. 4-7)

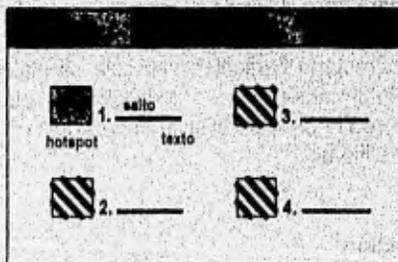


Figura 4-7 Tópico de contenido del tutorial de redes.

### ***Tópico de introducción***

El tópico de introducción, como su nombre lo dice, dará una introducción del capítulo seleccionado, con el fin de tener una idea del contenido de los temas que se desarrollarán en el capítulo, y de esta manera decidir si se continúa con el siguiente tópico o se regresa a elegir otro capítulo.

El contenido de este tópico puede ser desarrollado con texto e imágenes formateados de la manera más conveniente según el tema y con el uso de ventanas pop-up y hotspots (Fig. 4-8).

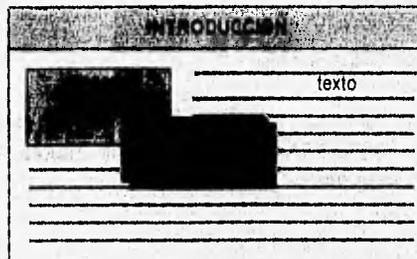


Figura 4-8 Tópico de introducción del tutorial de redes.

### ***Tópico de tema***

En los tópicos de tema se desarrollará, particularmente, un tema relacionado con el subcapítulo del tutorial. Para el desarrollo de dicho tema, podrán utilizarse texto, imágenes, hotspots, ventanas pop-up y, si se requiere, un programa de animación que ejemplifique de forma visual el tema en cuestión (Fig. 4-9).

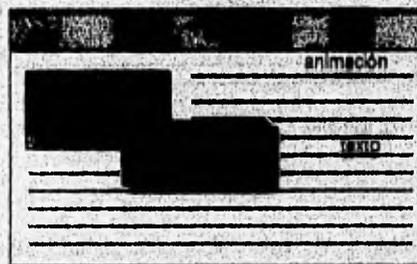


Figura 4-9 Tópico de tema del tutorial de redes.

### ***Tópico de temas múltiples***

Los tópicos de temas múltiples tienen las mismas características de los tópicos de tema, con la variante de que tienen la capacidad de desplegar múltiples tópicos de tema en ventanas secundarias (Fig. 4-10).

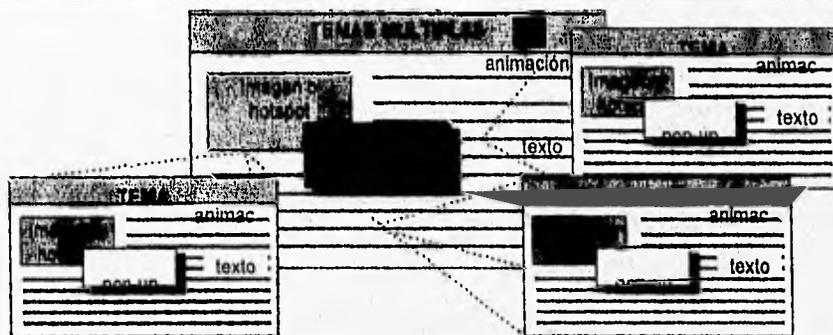


Figura 4-10 Tópico de temas múltiples del tutorial de redes.

Los tópicos de tema de las ventanas secundarias sólo pueden ser desplegados uno a la vez. Además, dadas las propiedades de las ventanas secundarias, es muy importante cerrarlas cuando ya se haya consultado el tema por cuestiones de memoria.

**Tópico de tema final**

El tópico de tema final puede ser un tópico de temas múltiples, con la diferencia de que es el último tópico del subcapítulo, el cual debe permitir que el usuario decida si continua con la evaluación o si comienza con otro subcapítulo (Fig. 4-11).



Figura 4-11 Tópico de tema final del tutorial de redes.

**Tópico de evaluación**

En el tópico de evaluación se examina el aprendizaje del usuario sobre el subcapítulo correspondiente. Para éste fin, se presenta un cuestionario de opciones múltiples, de tal forma que el continuar con la navegación normal del tutorial o volver a repasar algún tema del subcapítulo, depende de la opción que sea elegida por el usuario.

En los tópicos de evaluación se podrán utilizar texto, imágenes, hotspots, ventanas pop-up y saltos (Fig. 4-12).

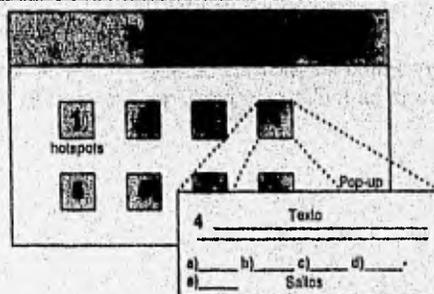


Figura 4-12 Tópico de evaluación del tutorial de redes.

Existen otros servicios que podrán ser prestados por el tutorial como son:

- a) Glosario.- Contendrá definiciones de términos empleados en los temas desarrollados en el tutorial.
- b) Referencias bibliográficas.- Contendrá referencias a bibliografía que sea de utilidad para obtener mayor información relacionada con el tema.
- c) Mapa de localización.- Su función será indicar la localización jerárquica del tópico que está desplegado en pantalla.

Estos servicios podrán ser accedidos por los tópicos de introducción, de tema, de temas múltiples y/o tema final. Además, también se mantendrán los recursos estándares proporcionados por la herramientas de ayuda como: regreso, historia, Índice, anotaciones, etc.

#### 4.1.4 Distribución de la pantalla.

Como ya se ha mencionado anteriormente, la distribución de la pantalla es uno de los aspectos más importantes a considerar en las interfaces para el aprendizaje. Por esta razón, es necesario conocer las limitaciones y capacidades con las que se cuentan al distribuir la pantalla con herramientas de ayuda.

##### **Limitaciones:**

- ◆ Tiene menor legibilidad comparada con la de los textos impresos.
- ◆ La información desplegada en la pantalla es generalmente vista en un área más pequeña que el tamaño de las páginas de los textos impresos.
- ◆ Debido a que existen diversos tamaños, resoluciones y capacidades de color en las pantallas, es necesario realizar el diseño del sistema con base en las máquinas de menor capacidad disponible, de modo que un significativo número de usuarios pueda utilizarlo (en éste caso VGA 640X480).
- ◆ El tipo de fuentes a usar es limitado dado que se debe utilizar alguno que esté contenido en la generalidad de las máquinas de los usuarios del tutorial (Arial, MS Sans Serif, etc.).
- ◆ Muchos recursos proporcionados por el procesador de texto (tablas complejas, bullets, etc.), no pueden ser utilizados, dado que no son soportados por el compilador de herramientas de ayuda.
- ◆ En el caso particular del tutorial, la cantidad de información por pantalla debe ser controlada, dado que por la resolución del monitor es posible que aparezcan barras de desplazamiento que no serían apropiadas para una buena lectura del usuario.

##### **Capacidades:**

- ◆ Es posible el uso de espaciado e indentación para la organización de la información en la pantalla.
- ◆ Se permite el uso de listas y tablas para información estructurada.
- ◆ Se permite el uso de color en texto y fondo sobre la pantalla. En el tutorial se podrá hacer uso del color de la forma que se crea más conveniente.
- ◆ Se proporciona una región en la parte superior de la ventana denominada "región de no desplazamiento", la cual permanece estática aún si se usan barras de desplazamiento. En esta área es recomendable poner los títulos y botones de uso general.

## 4.2 Creación de texto, imágenes y animación.

### 4.2.1 Creación de texto.

El texto creado está contenido en los archivos de tópico, los cuales contienen toda la información que se mostrará al usuario, organizada en los diferentes tópicos de información diseñados para el tutorial.

Un archivo de tópico es un archivo de texto con formato RTF (Rich Text Format, Formato de texto enriquecido), que es un formato que soporta notas al pie y el formato de las instrucciones que se usan para crear saltos (subrayado, doble subrayado y texto oculto). Este tipo de formato puede ser generado por diversos procesadores de texto como:

- ◆ Word para Windows (Ver. 1.1 ó superiores)
- ◆ Word Perfect para Windows (Ver. 5.1 ó superior)
- ◆ Lotus Ami Pro
- ◆ Word 5.0 (o superior) para MS-DOS.

En éste caso se usará Word para Windows Ver. 2.0c dado que existen plantillas con los estilos básicos definidos y además proporciona los recursos indispensables para la presentación en la pantalla.

Para codificar un archivo de tópico se deben crear códigos de control (caracteres de formato especial, véase tabla 4-1) que determinen como serán estructurados los tópicos en el archivo.

<b>Código de control</b>	<b>Que crea</b>	<b>Descripción</b>
Separador de páginas	Delimitador de tópico	Separa un tópico de otro.
<b>Notas al pie</b>		
Signo de libra (#)	Cadena de contexto	Identifica un tópico y son esenciales si se desea crear hipertexto.
Signo de dólar (\$)	Título	Determina el título del tópico
Signo de suma (+)	Secuencia de páginas	Determina el orden del tópico dentro de la secuencia de páginas. (opcional)
Letra "K" (K)	Palabra clave	Determina la lista de palabras que aparece en el cuadro de diálogo de búsqueda. (opcional)
Asterisco (*)	Indicador que establece etiquetas (Build-Tag)	Determina cuales tópicos son incluidos, cuando el compilador construye el archivo de ayuda. (opcional)
Signo de exclamación (!)	Referencia a una macro	Se ejecuta una macro cuando el usuario despliega el tópico. (opcional)
<b>Formatos de texto</b>		
Texto oculto	Referencia a una cadena de contexto	Especifica la cadena de contexto para el tópico que será desplegado, cuando el usuario encuentre el texto que inmediatamente le precede.
Texto subrayado	Texto pop-up	Especifica que un texto puede ser desplegado temporalmente.
Texto subrayado con doble línea	Texto de salto	Determina el texto que puede elegir el usuario para saltar a otro tópico.

Tabla 4-1 Descripción de los códigos de control.

Un archivo de tópico puede contener uno o más tópicos de información, los cuales deben estar delimitados por un separador de página, deben contener un nombre único (una cadena de contexto) y además, la posibilidad de contener otros 8 códigos de control.

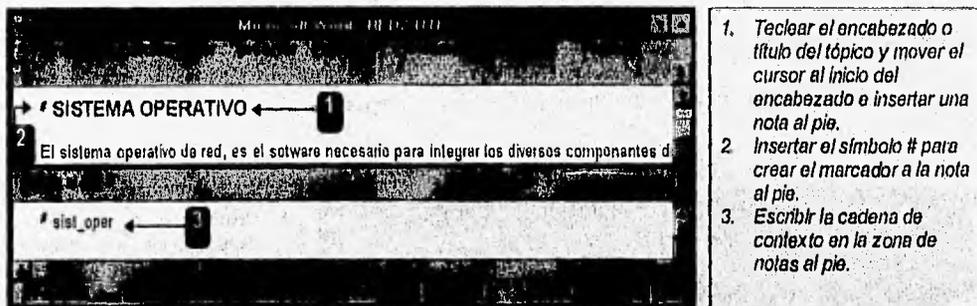
Independientemente del tipo de tópico de información que se vaya a crear, se deben agregar los códigos de control correspondientes según el tipo de tópico, de la siguiente manera:

### Agregar cadenas de contexto

Las características de las cadenas de contexto son:

- ◆ Se utilizan sólo en el caso de crear hipertexto.
- ◆ Los tópicos que no tienen cadena de contexto son accedidos por secuencias de páginas o por búsqueda de palabras clave.
- ◆ Cada tópico accesible, vía liga de hipertexto, debe tener un único identificador o cadena de contexto, aún si existe un archivo "project" ( archivo que reúne varios programas para ser compilados juntos), con varios archivos de tópico.
- ◆ Pueden contener letras (a-z o A-Z), números (0-9), puntos (.) y carácter de subrayado inferior (\_).
- ◆ No son casos sensitivos.

Los pasos para agregar una cadena de contexto se muestran en la figura 4-13

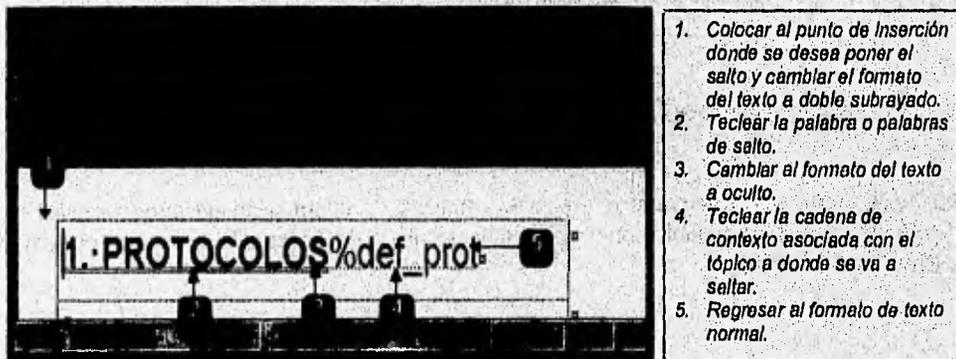


1. Teclar el encabezado o título del tópico y mover el cursor al inicio del encabezado e insertar una nota al pie.
2. Insertar el símbolo # para crear el marcador a la nota al pie.
3. Escribir la cadena de contexto en la zona de notas al pie.

Figura 4-13 Muestra como agregar una cadena de contexto usando Word.

### Agregar saltos

Los pasos a seguir para crear un salto se muestran en la figura 4-14.



1. Colocar al punto de inserción donde se desea poner el salto y cambiar el formato del texto a doble subrayado.
2. Teclar la palabra o palabras de salto.
3. Cambiar al formato del texto a oculto.
4. Teclar la cadena de contexto asociada con el tópico a donde se va a saltar.
5. Regresar al formato de texto normal.

Figura 4-14 Muestra como crear saltos de hipertexto usando Word.

Adicionalmente a los saltos normales, existen algunos tipos de saltos especiales:

- 1) **Saltos "Interarchivos".**- Son referencias a tópicos de información de otros archivos de tópico (.HLP). Se usa el símbolo @ junto con el nombre del archivo, como anexo a la cadena de contexto (ambos deben tener formato oculto).

ComunicaciónCom\_dat@cap2.hlp

- 2) **Salto a ventanas secundarias.**- Se salta a un t3pico que se desplegar3 en una ventana secundaria. Se usa el signo > y el nombre de la ventana como anexo a la cadena de contexto (ambos deben tener formato oculto).

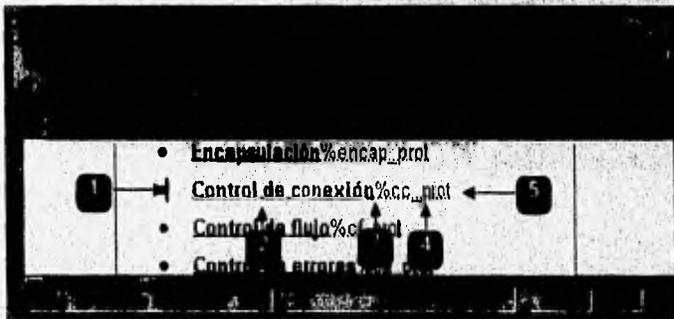
Comunicaci3nCom\_dat>com\_ventana

- 3) **Macro hotspots.**- Cuando el usuario hace "click" sobre un hotspot, la macrofunci3n es ejecutada. Se usa el signo ! inmediatamente despu3s del texto hotspot junto con la macrofunci3n a ejecutar con sus respectivos par3metros (ambos deben tener formato oculto).

Animaci3n!ExecProgram("anima1.exe",0)

### Agregar ventanas pop-up

Los pasos para crear una ventana pop-up se muestran en la figura 4-15



1. Colocar el punto de inserci3n y cambiar el formato de texto a subrayado.
2. Teclar la palabra o palabras pop-up.
3. Cambiar el formato de texto a oculto.
4. Teclar la cadena de contexto asociada con el t3pico que se va a desplegar.
5. Regresar al formato de texto normal.

Figura 4-15 Muestra como crear ventanas pop-up usando Word.

Tambi3n existen saltos especiales de ventanas pop-up y son los siguientes:

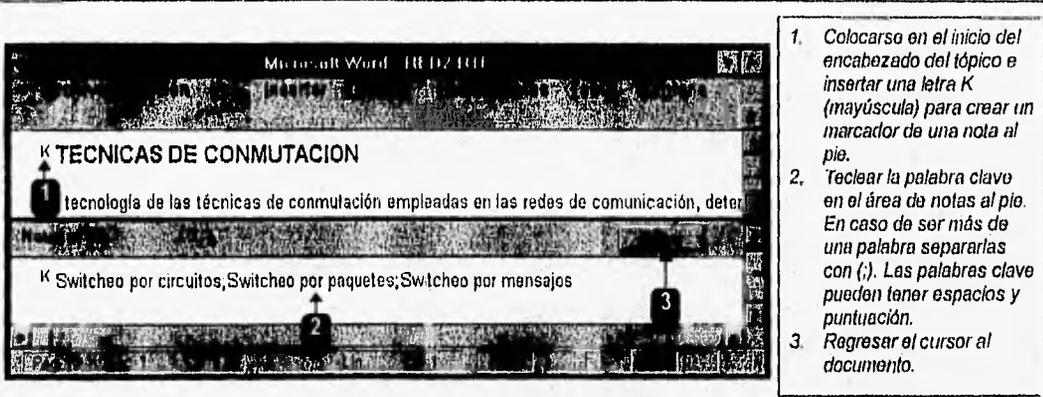
- 1) **Pop-up "Interarchivo".**- Se crean referencias a t3picos de informaci3n de otros archivos de t3pico (.HLP). Se usa el s3mbolo @ junto con el nombre del archivo y se anexa a la cadena de contexto (ambos deben tener formato oculto).

Comunicaci3nCom\_dat@cap2.hlp

- 2) **Macrohotspots.**- Para ejecutar una macrofunci3n, debe ser insertada junto con el signo ! inmediatamente despu3s del texto hotspot (ambos como texto oculto).

Animaci3n!ExecProgram("anima1.exe",0)

El proceso a seguir para agregar palabras clave se muestra en la figura 4-16.



1. Colocarse en el inicio del encabezado del tópico e insertar una letra K (mayúscula) para crear un marcador de una nota al pie.
2. Teclar la palabra clave en el área de notas al pie. En caso de ser más de una palabra separarlas con (;). Las palabras clave pueden tener espacios y puntuación.
3. Regresar el cursor al documento.

Figura 4-16 Muestra como crear palabras clave usando Word.

Una vez seleccionada una palabra clave en el cuadro de diálogos de búsqueda, se debe seleccionar un tópico específico con ayuda de su título.

**Agregar título al tópico**

El título del tópico es muy importante dado que identifica al tópico en tres instancias:

- 1) En la lista de tópicos encontrados en el cuadro de diálogo de búsqueda.
- 2) En la lista de historia.
- 3) En el menú de marcador de páginas.

La forma de insertar un título a un tópico se muestra en la figura 4-17.



1. Colocarse en el inicio del encabezado del tópico e insertar el signo \$ como un marcador de una nota al pie.
2. Teclar el título del tópico en el área de notas al pie, asegurandose que sea el mismo que se tecló en el encabezado del tópico.
3. Regresar el cursor al documento.

Figura 4-17 Muestra como agregar títulos al tópico usando Word.

Los títulos de los tópicos pueden ser de aproximadamente 128 caracteres.

**Agregar una secuencia de páginas**

El orden de los tópicos dentro de una secuencia de páginas es definida por el programador al seguir los siguientes pasos:

- ♦ Ordenar los tópicos y crear un plan de secuencia de páginas.

## CAPITULO 4: DISEÑO DE REDTUT

- ◆ Implementar las secuencias de páginas usando códigos de control.
- ◆ Si se desea agregar los botones anterior (<<) y siguiente (>>), se deben declarar en el archivo project (.HPJ) que se explicará posteriormente.

Las secuencias de páginas se pueden crear en nivel simple y multinivel.

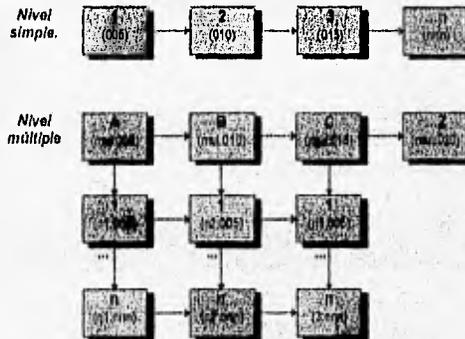


Figura 4-18 Niveles de secuencias de página.

En un nivel simple, la secuencia es de un tópico a otro únicamente. En cambio, en la secuencia multinivel, además de soportar la secuencia simple, cada tópico puede tener secuencias de páginas separadas (Fig. 4-18).

El proceso a seguir para crear una secuencia de páginas, se muestra en la figura 4-19:

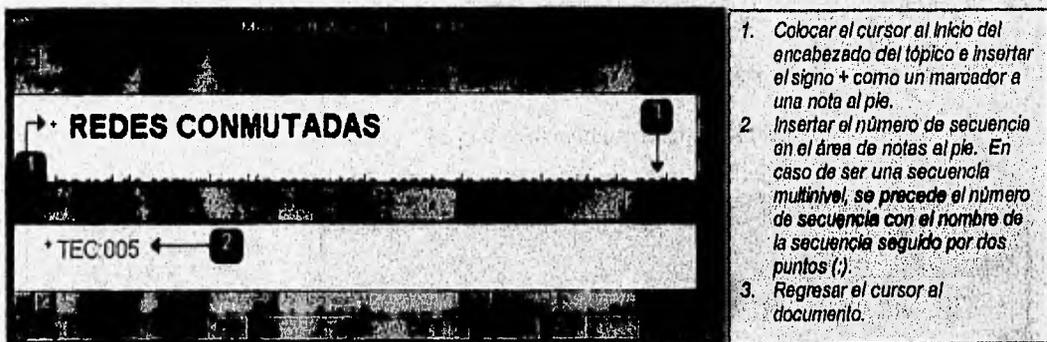


Figura 4-19 Muestra como crear secuencias de página usando Word.

Debido a que el compilador de herramientas de ayuda ordena las cadenas de secuencia de páginas en orden ascendente (de 0-9 o A-Z), se debe usar el mismo número de caracteres para cada número de secuencia. Por lo tanto, es recomendable poner números de secuencia con tres o más dígitos, según el número de tópicos en la secuencia. Además se recomienda, poner números con incrementos fijos (de 004, 005, etc.) con el fin de permitir insertar un tópico entre dos existentes; en un futuro. En caso de que no se quieran ordenar los tópicos numéricamente se propone lo siguiente:

- ◆ Si se desea una secuencia de tópicos ordenada alfabéticamente en lugar de números se insertan los títulos de los tópicos.
- ◆ Para ordenar, según aparezcan los tópicos en el archivo, sólo se pone el signo + sin número de secuencia.

### Agregar un indicador de etiquetas condicional

Con los indicadores de etiquetas condicionales, se puede controlar que tópicos están excluidos del sistema. Cada tópico puede tener uno o más indicadores de etiquetas. El implementar un indicador de etiquetas condicional se realiza en un proceso de dos pasos:

- 1) Insertar los indicadores de etiqueta dentro del archivo de tópicos.
- 2) Indicar al compilador de ayuda, en el archivo project (.HPJ), que tópicos se requieren incluir o excluir.

La forma de insertar un indicador de etiquetas se muestra en la fig. 4-20.



Figura 4-20 Agregando indicador de etiquetas usando Word.

### Agregar una referencia a una macro

Para crear referencias a las macros proporcionadas por las herramientas de ayuda se siguen los pasos mostrados en la figura 4-21. En el apéndice B se muestran todas las macros de ayuda proporcionadas por Windows.

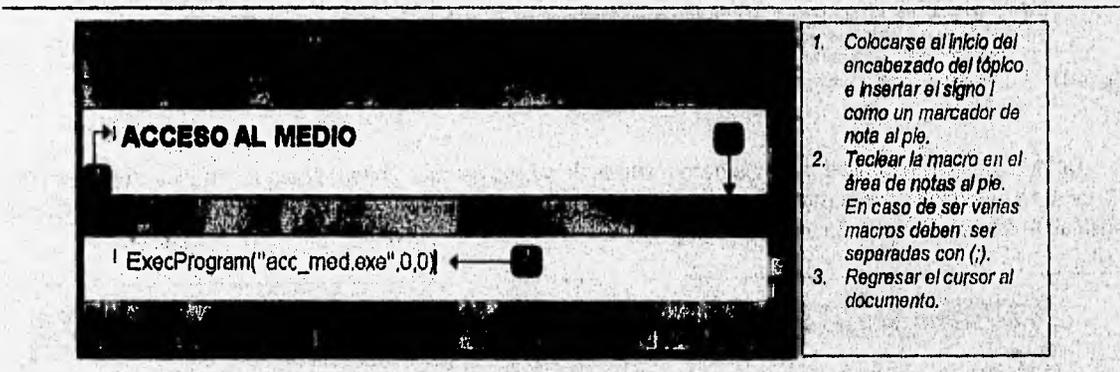


Figura 4-21 Muestra como agregar referencias a una macro usando Word.

### Agregar una región de no desplazamiento

Una región de no desplazamiento, es una región estática que permanece visible aún si se utilizan barras de desplazamiento en la ventana. Estas regiones son muy usadas para poner el título de los tópicos de información, gráficos y hotspots. Los pasos a seguir para insertar una región de no desplazamiento se muestran en la figura. 4-22.



1. Asegurarse que el texto o gráficos que se desea que aparezcan en el área de no desplazamiento, estén en el primer párrafo del tópico.
2. Seleccionar el párrafo.
3. Dar formato para prevenir un separador de páginas entre el párrafo seleccionado y el próximo.

Figura 4-22 Muestra como crear zonas de no desplazamiento usando Word.

### Agregando saltos en medio del tópico

Se les llama saltos en medio del tópico, a los saltos que se realizan dentro de un mismo tópico de información. Este tipo de saltos es muy usado cuando los tópicos son muy largos y se desean acceder varias partes del mismo, como es el caso de los glosarios. La forma de insertar un salto en medio del tópico se muestra en la figura 4-23.



1. Colocar el cursor en la posición dentro del tópico a donde se quiera saltar e insertar el signo # como nota al pie.
2. Teclar una cadena de contexto en el área de notas al pie.
3. Referenciar la cadena de contexto a un hotspot.

Fig. 4-23 Muestra como agregar saltos en medio del tópico.

Debido a que el compilador de las herramientas de ayuda es muy limitado, no soporta todo el formato de texto que proporciona Word para Windows. En la siguiente tabla (Tabla 4-2) se muestran algunas formas de sustituir el formato no soportado por el compilador.

Formato no soportado	Sugerencia
Comillas ( " ") y apóstrofes ( ' ' )	Usar signos de pulgadas " y '.
Guión largo (—)	Usar dos o más guiones cortos (---)
Texto en subíndice o superíndice	Crear un mapa de bits equivalente
Espacios de letras expandidos o condensados	Ninguna
Todo el texto en CAPS (mayúsculas)	Teclar el texto en mayúsculas con la tecla de cambio (shift) presionada.
Texto justificado	Ninguna
Tabulador decimal	Ninguna
Colorear o sombrear bordes	Usar mapas de bits
Bordes o sombras en tablas o celdas	Ninguna
Párrafo sombreado	Ninguna
Manejo de bullets	Usar mapas de bits

Tabla 4-2 Formato no soportado por el compilador y sugerencias al formato.

Además de conocer el formato no soportado por el compilador, es necesario conocer algunos tips utilizados en el tutorial, que facilitaron su compilación y su presentación. Esto se muestra en la tabla 4-3.

<i>Tips</i>
<ul style="list-style-type: none"> <li>◆ Crear alguna plantilla que permita crear los tópicos fácilmente.</li> <li>◆ Usar una fuente que generalmente esté contenida en la mayoría de las computadoras de los usuarios del tutorial (MS Serif, MS Sans Serif, Arial, etc.) .</li> <li>◆ Es posible agregarle color al texto, pero tratar de no ocasionar conflictos con los colores definidos en el panel de control.</li> <li>◆ Para cambiar o suprimir color a un hotspot, agregar al inicio de la cadena de contexto, los símbolos % y *, respectivamente (para el cambio de color el texto ya debe tener el color deseado).</li> <li>◆ Cambiar los marcadores de párrafo oculto a "no oculto" en Word, para no tener problemas con el compilador.</li> </ul>

Tabla 4-3 Tips para facilitar la compilación y presentación del sistema.

#### 4.2.2 Creación de Imágenes.

Anteriormente se ha mencionado que el formato de imágenes más adecuado para el tutorial, es el .BMP. Windows proporciona su propia aplicación para crear mapas de bits con formato BMP (Paint Brush), sin embargo, es muy rústico y de muy difícil manejo para crear imágenes que tengan cierto grado de complejidad (sombras, degradaciones de color, efectos, etc.). Es por esto que se necesita recurrir alguna aplicación que permita generar las imágenes con alta calidad y con la posibilidad de generarlas con formato BMP. Esta aplicación puede ser Corel Draw Ver. 3.0, que es la que se utilizará para tal fin (Fig. 4-24)



Figura 4-24 Corel Draw para Windows Ver. 3.0.

A pesar de que Corel Draw Ver. 3.0 cuenta con una opción para exportar sus archivos de imagen (.CDR) a un formato .BMP, ésta no es la mejor forma de darle dicho formato, dado que el número de colores soportado para el formato .BMP es reducido y esto ocasiona que las características de la imagen (diversidad

de colores, efectos, etc.) se pierdan. Por tal motivo, se hace uso de la captura de imágenes a través del portapapeles de Windows, con el fin de que la imagen no pierda sus características de presentación.

Con el uso del portapapeles, la imagen es copiada como mapa de bits (uno de los formatos estándares soportado por el portapapeles), dando la posibilidad de que la imagen sea transferida a cualquier aplicación de Windows.

Por lo tanto el proceso usado para crear las imágenes es el siguiente:

- 1) Crear la imagen en Corel Draw, haciendo uso de todas sus capacidades.
- 2) Una vez terminada la imagen, seleccionarla y copiarla al portapapeles de Windows.
- 3) Abrir Paint Brush y pegar el contenido del portapapeles.
- 4) Ajustar los atributos de la imagen (alto y ancho), de tal forma que el archivo contenga únicamente la imagen contenida en el portapapeles.
- 5) Retocar la imagen si es necesario.
- 6) Guardar la imagen con formato .BMP.

Siguiendo éste proceso, las imágenes conservan las mismas características con las que fueron creadas, lógicamente, con la calidad que un mapa de bits puede proporcionar.

### **Creación de hipergráficos**

Para la creación de hipergráficos se seleccionaron las imágenes (.BMP) del tutorial que requerían desplegar información adicional (texto u otra imagen), para asimilar mejor un concepto.

Los hipergráficos simples fueron creados al dar un formato especial a las referencias de gráficos insertadas en el texto (archivos .RTF). Estos hipergráficos pueden estar asociados a una ventana pop-up, a una ventana secundaria, a una macro-función o a un salto.

Los hipergráficos múltiples fueron creados con ayuda del editor "Hotspot", que permite la conversión de una imagen con formato .BMP a una con formato .SHG, que es la que almacena las ligas entre las "zonas calientes" y la información adicional que desplegará. Estos archivos son insertados como referencia en los archivos .RTF, de la misma forma que los archivos .BMP.

La ventaja de los hipergráficos múltiples es que una misma imagen puede estar asociada con diferentes tipos de ligas, ya sea a una ventana pop-up, a una ventana secundaria, a un macro-función o a un salto.

Para poder identificar, en REDTUT, los hipergráficos simples y los hipergráficos múltiples, se insertó en las imágenes un pequeño símbolo que mostrará la propiedad de la imagen (Fig. 4-25. En el caso de los hipergráficos simples una "manita" y en el caso de los hipergráficos múltiples "dos manitas superpuestas").

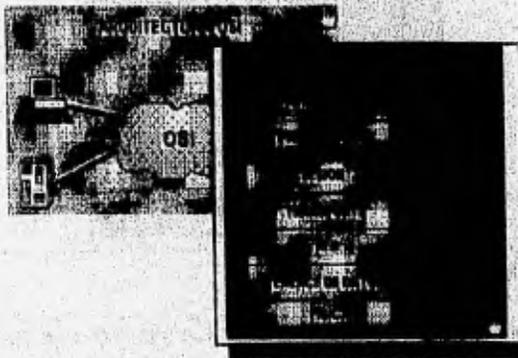


Figura 4.25 Ejemplo de hipergráficos simples y múltiples creados para REDTUT.

### Creación de imágenes e iconos

Las imágenes e iconos de REDTUT son los que nos permitirán llevar el control de la navegación por el tutorial. Por lo tanto deben tener las siguientes características:

- ◆ Representar gráficamente la acción que desempeñan.
- ◆ Tener colorido atractivo para que se localicen rápidamente.
- ◆ Ser de tamaño pequeño para que no compitan con otra información desplegada en pantalla.
- ◆ Ser uniformes para que puedan ser integradas en cualquier capítulo de REDTUT.

Las imágenes de REDTUT tienen como principal tarea identificar el capítulo seleccionado, por lo que están relacionadas con los títulos correspondientes a cada capítulo (Fig.4-26).



Figura 4-26 Imágenes que identifican los capítulos de REDTUT.

El tamaño de las imágenes de los capítulos son de 80 X 80 pixeles para desplegarse en la pantalla principal y de 35 X 35 pixeles en el caso de que se desplieguen dentro de los capítulos de REDTUT.

Debido a que los iconos deben presentar lo más cercanamente la acción que llevan implícita, se diseñaron como se muestra en la figura 4-27.



Figura. 4-27 Iconos que controlan la navegación en REDTUT.

En el caso específico de los iconos de navegación de REDTUT, el formato empleado es .BMP, debido a que se manejan como hipergráficos simples. Pero si su función es identificar la aplicación dentro del ambiente de Windows el formato debe ser .ICO, el cual es tratado como un recurso dentro de la programación en Windows.

El tamaño, tanto de los iconos de navegación como de los iconos que identificarán la aplicación, es de 35 X 35 pixeles.

### 4.2.3 Creación de animación.

Debido a que la animación que se integrará a REDTUT está basada en programación tradicional y corresponde a diversos procesos relacionados con el funcionamiento de redes de computadoras, pueden tratarse como programas independientes. Por lo tanto, un proceso individual será tratado como un programa ejecutable que posteriormente será integrado al tutorial con ayuda de macro-funciones.

Para que un proceso pueda ser comprendido en su totalidad es necesario mostrar, si las hay, las diferentes técnicas o modalidades empleadas para llegar a su finalidad. Es por esto que se toma la decisión de que un programa contenga todas las técnicas o modalidades relacionadas con un proceso, para que puedan ser comparadas.

Un ejemplo es la animación del proceso de "corrección de errores" en la comunicación de datos, la cual puede ser efectuada por diversas técnicas como son: "ARQ: alto y espera", "ARQ: continuo", "FEC: hacia adelante" y "FEC: convolucional". Debido a que la corrección de errores es solucionada con el envío y recepción de mensajes entre el emisor y el receptor, no se requiere de la animación de múltiples imágenes simultáneamente, más bien, del movimiento de sólo una imagen a la vez (la señal que viaja). Para estos fines la animación en tiempo real es más que suficiente. Asimismo, la duración del viaje de la señal que esta definida por la distancia que hay entre la imagen del transmisor y la imagen del receptor, puede ser solucionado con algoritmos basados en el reloj de la computadora. Por otro lado, para poder seleccionar y/o comparar entre una técnica y otra se requiere controlar la animación por procedimientos, que nos permiten interactuar con el programa al activar el botón que despliega los pasos realizados en las diferentes técnicas relacionadas con la "corrección de errores".

Una vez teniendo bien definidas las formas de control y las técnicas de animación específicas al proceso en estudio, se procede a la programación de eventos.

Primeramente se deben definir los elementos que integran el proceso de animación, en el caso específico del proceso de "corrección de errores" se definen los siguientes:

- ◆ Imágenes del emisor y receptor.
- ◆ Imágenes del buffer del emisor y receptor.
- ◆ Imagen de la señal en dirección del receptor.
- ◆ Imagen de la señal en dirección del emisor.
- ◆ Imagen de la señal afectada por el ruido.
- ◆ Imagen del contenido del buffer según el paso correspondiente a la técnica seleccionada.
- ◆ Imágenes que despliegan los títulos de las diferentes técnicas.
- ◆ Los botones que activan los pasos de las diferentes técnicas.
- ◆ Imágenes que indican que botón está activado o desactivado.
- ◆ Despliegue de texto que apoya las diferentes etapas del proceso.
- ◆ Imagen que indica el capítulo actual seleccionado.

La animación del proceso de corrección de errores puede dividirse en elementos estáticos y elementos dinámicos según el papel que desempeñen en la pantalla. Los elementos estáticos, son la mayor parte de los elementos anteriormente listados, a excepción de las imágenes de señal, el contenido de los buffers y el despliegue del texto, que varían según la técnica que los active. Por lo tanto, los elementos estáticos pueden ser desplegados desde la parte principal del programa y los elementos dinámicos pueden ser animados por medio de funciones de llamada controladas por el reloj de la computadora. El listado de los archivos fuente (.C), de definición (.DEF), de recursos (.RC), y proyecto (.PRJ) que componen el programa ejecutable (.EXE), se muestran en la figura 4-28.

```

/***** Archivo fuente (anm2.c) de la animación del proceso de "corrección de errores" *****/
#include<windows.h>
/** Constantes que identifican los botones de la pantalla principal ***/
#define IDB_P1 100
#define IDB_P2 101
#define IDB_P3 102
#define IDB_PA 103
#define IDB_PB 104
#define IDB_PC 105
#define IDB_P11 106
#define IDB_P12 107
#define IDB_PA1 108
#define IDB_PA2 109
#define IDB_CERRAR 200
/** Código virtual de teclas no definidas en windows.h ***/
#define VK_1 0x31
#define VK_2 0x32
#define VK_3 0x33
#define VK_A 0x41
#define VK_B 0x42
#define VK_C 0x43
#define VK_R 0x52
#define VK_W 0x57
#define VK_X 0x58
#define VK_Y 0x59
#define VK_Z 0x5A
/***** Declaración de variables globales *****/
char szProgName[]="Correrror", icon_anime[]="ANIMA";
static int ancho_pant,alto_pant;
HWND hPantalla;
HDC hPantIDC;
HBITMAP hBufy,hBit_prov,hOff,hOn;
IRGN hRgn_prev;

#pragma argsused
long FAR PASCAL WndProc(HWND,UINT,WPARAM,LPARAM);
WORD FAR PASCAL Anima_CE(HWND,WDRD,INT,DWORD);
/***** Inicio de la función principal *****/
int PASCAL WinMain(HINSTANCE hInst, HINSTANCE hPreInst, LPSTR lpszCmdLine, int nCmdShow)
{
    HWND hWnd;
    WNDCLASS wcApli;
    MSG mensaje;
    if (hPreInst){
        wcApli.lpszClassName = szProgName;
        wcApli.hInstance = hInst;
        wcApli.lpfnWndProc = WndProc;
        wcApli.hCursor = LoadCursor(NULL, IDC_ARROW);
        wcApli.hIcon = LoadIcon(hInst, icon_anime);
        wcApli.lpszMenuName = NULL;
        wcApli.hbrBackground = GetStockObject(BLACK_BRUSH);
        wcApli.style = CS_HREDRAW|CS_VREDRAW;
        wcApli.cbClsExtra = 0;
        wcApli.cbWndExtra = 0;
        if (!RegisterClass (&wcApli))
            return FALSE;
    }
    /** Determina las dimensiones de la pantalla ***/
    hPantalla=GetDesktopWindow();
    hPantIDC=GetDC(hPantalla);
    ancho_pant=GetDeviceCaps(hPantIDC,HORZRES);
    alto_pant=GetDeviceCaps(hPantIDC,VERTRES);
    ReleaseDC(hPantalla,hPantIDC);

    /** Crea la ventana principal ***/
    hWnd=CreateWindow(szProgName,"CORRECCION DE ERRORES (Animación)",WS_BORDER,0,0,ancho_pant,alto_pant,(HWND)NULL,
        (HMENU)NULL, (HANDLE)hInst, (LPSTR)NULL);
    ShowWindow(hWnd,nCmdShow);
    UpdateWindow(hWnd);
    while (GetMessage(&mensaje,NULL,NULL,NULL))
        TranslateMessage(&mensaje);
        DispatchMessage(&mensaje);
    return(mensaje.wParam);
}

/** Inicia función de procedimiento ***/
long FAR PASCAL WndProc(HWND hWnd,UINT msg,WPARAM wParam,LPARAM lParam)
{
    HWND hbP1,hbP2,hbP3,hbPa,hbPb,hbPc,hbP11,hbP12,hbPa1,hbPa2,hbCerrar;
    HDC hDC,hmetaMarco,hmetaBit,hmetaDC;
    FARPROC fpAnima_CE;
    static int Tempo;
    static HANDLE hMarco;
    PAINTSTRUCT ps;
    HRGN hmetaRgn;
    HBITMAP hld_Cap,hcanipu,h_arq1,h_arq2,h_atc1,h_atc2,hbm_Titulo,hbm_REOTUT;
    int x,y;
    switch (msg)
    {
        case WM_CREATE:
            /** Determina las dimensiones de la pantalla ***/
            hPantalla=GetDesktopWindow();
            hPantIDC=GetDC(hPantalla);

```

## CAPÍTULO 4: DISEÑO DE REDTUT

```

ancho_pant=GetDeviceCaps(hPantDC,HORZRES);
alto_pant=GetDeviceCaps(hPantDC,VERTRES);
ReleaseDC(hPantalla,hPantDC);
/** Creación de la caja tipográfica de la pantalla como metafile **/
hid_Cap=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"CAP1");
hbm_REDTUT=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"REDTUT");
hbm_Titulo=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"TITULO");
hmetaRgn=CreateRectRgn(0,50,ancho_pant,alto_pant-100);
hmetaMarco=CreateMetaFile(NULL);
if (hmetaMarco=NULL)
{
    hmetaBit=GetDC(hWnd);
    hmemDC=CreateCompatibleDC(hmetaBit);
    SelectObject(hmemDC,hid_Cap);
    BitBlt(hmetaMarco,(ancho_pant/2)-225,8,35,35,hmemDC,0,0,SRCCOPY);
    BitBlt(hmetaMarco,(ancho_pant/2)+191,8,35,35,hmemDC,0,0,SRCCOPY);
    SelectObject(hmemDC,hbm_Titulo);
    BitBlt(hmetaMarco,(ancho_pant/2)-181,2,362,46,hmemDC,0,0,SRCCOPY);
    SelectObject(hmemDC,hbm_REDTUT);
    SelectClipRgn(hmetaMarco,hmetaRgn);
    for (x=0,x<=ancho_pant;x+=100) {
        for (y=0,y<=alto_pant;y+=100)
            BitBlt(hmetaMarco,x,y,100,100,hmemDC,0,0,SRCCOPY);
    }
    SelectObject(hmetaMarco,CreateSolidBrush(RGB(255,255,255)));
    Rectangle(hmetaMarco,50,70,ancho_pant-50,alto_pant-120);
    hMarco=CloseMetaFile(hmetaMarco);
    DeleteDC(hmemDC);
    DeleteObject(hid_Cap);
    DeleteObject(hbm_Titulo);
    DeleteObject(hbm_REDTUT);
    DeleteObject(hmetaRgn);
    ReleaseDC(hWnd,hmetaBit);
}
/** Creación de botones **/
hbPa = CreateWindow ("BUTTON","&a",WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON,120,alto_pant-50,25,25,hWnd,IDB_PA,
    GetWindowWord(hWnd,GWW_HINSTANCE),NULL);
ShowWindow (hbPa,SW_SHOW);
hbPb = CreateWindow ("BUTTON","&b",WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON,150,alto_pant-50,25,25,hWnd,IDB_PB,
    GetWindowWord(hWnd,GWW_HINSTANCE),NULL);
ShowWindow (hbPb,SW_SHOW);
hbPc = CreateWindow ("BUTTON","&c",WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON,180,alto_pant-50,25,25,hWnd,IDB_PC,
    GetWindowWord(hWnd,GWW_HINSTANCE),NULL);
ShowWindow (hbPc,SW_SHOW);
hbP1 = CreateWindow ("BUTTON","&1",WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON,120,alto_pant-90,25,25,hWnd,IDB_P1,
    GetWindowWord(hWnd,GWW_HINSTANCE),NULL);
ShowWindow (hbP1,SW_SHOW);
hbP2 = CreateWindow ("BUTTON","&2",WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON,150,alto_pant-90,25,25,hWnd,IDB_P2,
    GetWindowWord(hWnd,GWW_HINSTANCE),NULL);
ShowWindow (hbP2,SW_SHOW);
hbP3 = CreateWindow ("BUTTON","&3",WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON,180,alto_pant-90,25,25,hWnd,IDB_P3,
    GetWindowWord(hWnd,GWW_HINSTANCE),NULL);
ShowWindow (hbP3,SW_SHOW);
hbP11 = CreateWindow ("BUTTON","&w",WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON,350,alto_pant-90,25,25,hWnd,IDB_P11,
    GetWindowWord(hWnd,GWW_HINSTANCE),NULL);
ShowWindow (hbP11,SW_SHOW);
hbP12 = CreateWindow ("BUTTON","&x",WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON,380,alto_pant-90,25,25,hWnd,IDB_P12,
    GetWindowWord(hWnd,GWW_HINSTANCE),NULL);
ShowWindow (hbP12,SW_SHOW);
hbPa1 = CreateWindow ("BUTTON","&y",WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON,350,alto_pant-50,25,25,hWnd,IDB_PA1,
    GetWindowWord(hWnd,GWW_HINSTANCE),NULL);
ShowWindow (hbPa1,SW_SHOW);
hbPa2 = CreateWindow ("BUTTON","&z",WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON,380,alto_pant-50,25,25,hWnd,IDB_PA2,
    GetWindowWord(hWnd,GWW_HINSTANCE),NULL);
ShowWindow (hbPa2,SW_SHOW);
hbCerrar = CreateWindow ("BUTTON","&Cerrar",WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON,ancho_pant-150,alto_pant-70,
    100,30,hWnd,IDB_CERRAR,GetWindowWord(hWnd,GWW_HINSTANCE),NULL);
ShowWindow (hbCerrar,SW_SHOW);
break;
case WM_COMMAND:
/** Crea la instancia para la función de animación **/
    fpAnima_CE=MakeProcInstanca((FARPROC)Anima_CE,GetWindowWord(hWnd,GWW_HINSTANCE));
/** Llamada a la función según el botón presionado con mouse **/
    switch(wParam)
    {
        case IDB_P1:
            if(!Tempo=SetTimer(hWnd,1,55,fpAnima_CE))
                MessageBox(hWnd,"No hay Tempo","Message",MB_OK);
            break;
        case IDB_P2:
            if(!Tempo=SetTimer(hWnd,2,55,fpAnima_CE))
                MessageBox(hWnd,"No hay Tempo","Message",MB_OK);
            break;
        case IDB_P3:
            if(!Tempo=SetTimer(hWnd,3,55,fpAnima_CE))
                MessageBox(hWnd,"No hay Tempo","Message",MB_OK);
            break;
        case IDB_PA:
            if(!Tempo=SetTimer(hWnd,4,55,fpAnima_CE))
                MessageBox(hWnd,"No hay Tempo","Message",MB_OK);
            break;
        case IDB_PB:
            if(!Tempo=SetTimer(hWnd,5,55,fpAnima_CE))
                MessageBox(hWnd,"No hay Tempo","Message",MB_OK);
    }
}

```

```

        break;
    case IDB_PC:
        if(!((Tempo=SetTimer(hWnd,6,55,fpAnima_CE)))
            MessageBox(hWnd,"No hay Temp","Message",MB_OK);
        break;
    case IDB_P11:
        if(!((Tempo=SetTimer(hWnd,7,55,fpAnima_CE)))
            MessageBox(hWnd,"No hay Temp","Message",MB_OK);
        break;
    case IDB_P12:
        if(!((Tempo=SetTimer(hWnd,8,55,fpAnima_CE)))
            MessageBox(hWnd,"No hay Temp","Message",MB_OK);
        break;
    case IDB_PA1:
        if(!((Tempo=SetTimer(hWnd,9,55,fpAnima_CE)))
            MessageBox(hWnd,"No hay Temp","Message",MB_OK);
        break;
    case IDB_PA2:
        if(!((Tempo=SetTimer(hWnd,10,55,fpAnima_CE)))
            MessageBox(hWnd,"No hay Temp","Message",MB_OK);
        break;
    case IDB_CERRAR:
        DestroyWindow(hWnd);
        break;
    }
    break;
case WM_KEYDOWN:
    /* Crea la instancia para la función de animación */
    fpAnima_CE=MakeProcInstance((FARPROC)Anima_CE,GetWindowWord(hWnd,GWW_HINSTANCE));
    /* Acción realizada por un botón al ser accionado por teclado */
    {
        switch(wpParam)
        {
            case VK_1:
                if(!((Tempo=SetTimer(hWnd,1,55,fpAnima_CE)))
                    MessageBox(hWnd,"No hay Temp","Message",MB_OK);
                break;
            case VK_2:
                if(!((Tempo=SetTimer(hWnd,2,55,fpAnima_CE)))
                    MessageBox(hWnd,"No hay Temp","Message",MB_OK);
                break;
            case VK_3:
                if(!((Tempo=SetTimer(hWnd,3,55,fpAnima_CE)))
                    MessageBox(hWnd,"No hay Temp","Message",MB_OK);
                break;
            /* VK_NUMPAD1, VK_NUMPAD2, VK_NUMPAD3 en caso del teclado numérico */
            case VK_NUMPAD1:
                if(!((Tempo=SetTimer(hWnd,1,65,fpAnima_CE)))
                    MessageBox(hWnd,"No hay Temp","Message",MB_OK);
                break;
            case VK_NUMPAD2:
                if(!((Tempo=SetTimer(hWnd,2,55,fpAnima_CE)))
                    MessageBox(hWnd,"No hay Temp","Message",MB_OK);
                break;
            case VK_NUMPAD3:
                if(!((Tempo=SetTimer(hWnd,3,55,fpAnima_CE)))
                    MessageBox(hWnd,"No hay Temp","Message",MB_OK);
                break;
            case VK_A:
                if(!((Tempo=SetTimer(hWnd,4,55,fpAnima_CE)))
                    MessageBox(hWnd,"No hay Temp","Message",MB_OK);
                break;
            case VK_B:
                if(!((Tempo=SetTimer(hWnd,5,55,fpAnima_CE)))
                    MessageBox(hWnd,"No hay Temp","Message",MB_OK);
                break;
            case VK_C:
                if(!((Tempo=SetTimer(hWnd,6,55,fpAnima_CE)))
                    MessageBox(hWnd,"No hay Temp","Message",MB_OK);
                break;
            case VK_W:
                if(!((Tempo=SetTimer(hWnd,7,55,fpAnima_CE)))
                    MessageBox(hWnd,"No hay Temp","Message",MB_OK);
                break;
            case VK_X:
                if(!((Tempo=SetTimer(hWnd,8,55,fpAnima_CE)))
                    MessageBox(hWnd,"No hay Temp","Message",MB_OK);
                break;
            case VK_Y:
                if(!((Tempo=SetTimer(hWnd,9,55,fpAnima_CE)))
                    MessageBox(hWnd,"No hay Temp","Message",MB_OK);
                break;
            case VK_Z:
                if(!((Tempo=SetTimer(hWnd,10,55,fpAnima_CE)))
                    MessageBox(hWnd,"No hay Temp","Message",MB_OK);
                break;
            case VK_R:
                DestroyWindow(hWnd);
                break;
        }
    }
}
break;
/* Despliegue de imágenes estáticas en la pantalla principal */
case WM_PAINT:

```

## CAPITULO 4. DISEÑO DE REDTUT

```

        HDC= BeginPaint(hWnd,&ps);
    /** Despliega el marco en la pantalla      ***
        PlayMetaFile(hDC,hMarco);
    /** Despliega triángonos da computadores y buffers      ****
        hmemDC=CroateCompatibleDC(hDC);
        hcompu=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"COMPU");
        h_arq1=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"ARQ1");
        h_arq2=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"ARQ2");
        h_atc1=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"ATC1");
        h_atc2=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"ATC2");
        hBufv=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUFV");
        hOff=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"OFF");
        hBit_prev=SelaciDbject(hmemDC,hcompu);
        BitBlt(hDC,60,(alto_pant/2)-60,125,108,hmemDC,0,0,SRCCOPY);
        BitBlt(hDC,0,ancho_pant-185,(alto_pant/2)-60,125,108,hmemDC,0,0,SRCCOPY);
        SelectObject(hmemDC,h_arq1);
        BitBlt(hDC,0,alto_pant-90,110,20,hmemDC,0,0,SRCCOPY);
        SelectObject(hmemDC,h_arq2);
        BitBlt(hDC,20,alto_pant-50,90,20,hmemDC,0,0,SRCCOPY);
        SelectObject(hmemDC,h_atc1);
        BitBlt(hDC,250,alto_pant-90,90,20,hmemDC,0,0,SRCCOPY);
        SelectObject(hmemDC,h_atc2);
        BitBlt(hDC,250,alto_pant-50,90,20,hmemDC,0,0,SRCCOPY);
        SelectObject(hmemDC,hOff);
        BitBlt(hDC,210,alto_pant-45,13,13,hmemDC,0,0,SRCCOPY);
        BitBlt(hDC,210,alto_pant-85,13,13,hmemDC,0,0,SRCCOPY);
        BitBlt(hDC,410,alto_pant-45,13,13,hmemDC,0,0,SRCCOPY);
        BitBlt(hDC,410,alto_pant-85,13,13,hmemDC,0,0,SRCCOPY);
        SelectObject(hmemDC,hBufv);
        BitBlt(hDC,ancho_pant-185,(alto_pant/2)+48,120,50,hmemDC,0,0,SRCCOPY);
        BitBlt(hDC,60,(alto_pant/2)+48,128,50,hmemDC,0,0,SRCCOPY);
        SelectObject(hmemDC,hBit_prev);
        DeleteDC(hmemDC);
        DeleteObject(hcompu);
        DeleteObject(h_arq1);
        DeleteObject(h_arq2);
        DeleteObject(h_atc1);
        DeleteObject(h_atc2);
        DeleteObject(hBufv);
        DeleteObject(hOff);
        DeleteObject(hOn);
        ReleaseDC(hWnd,hDC);
        EndPaint(hWnd,&ps);
        break;
    case WM_DESTROY:
        DeleteMetaFile(hMarco);
        PostQuitMessage(0);
        break;
    default:
        return(DefWindowProc(hWnd,msg,wParam,lParam));
}
return(0);
}

/****** Función de animación ******/
WORD FAR PASCAL Anima_CE(HWND hWnd,WORD wParam,int nDtiempo,DWORD wTiempo)
{
    HDC hdc,hmem,hMem;
    PAINTSTRUCT ps;
    hRgn_Anima,hRgn_Text;
    hAck,hNack,hNack2,hAck5,hDat1,hDat2,hDat3,hDat5,hDat6,
    hBuf1,hBuf2,hBuf3,hBuf5,hBuf6,hBuf7,hBuf8,hRuido,
    hBuf9;
    HPEN hPen;
    static int posx,posy,caso1=0,caso2=0,caso3=0,caso4=0,caso5=0,caso6=0,
    caso7=0,caso8=0,caso9=0,caso10=0;
    RECT rc;
    static char text1[]="LA ENTIDAD TRANSMISORA ENVIA UNA UNIDAD DE DATOS";
    text2[]="LA ENTIDAD RECEPTORA DETECTA QUE LA UNIDAD DE DATOS RECIBIDA ES ERRÓNEA, POR LO QUE TRANSMITE UN RECONOCIMIENTO NEGATIVO (NACK) INDICANDO QUE NO SE RECIBIO LA UNIDAD DE DATOS ENVIADA";
    text3[]="LA ENTIDAD TRANSMISORA AL RECIBIR EL RECONOCIMIENTO NEGATIVO (NACK) DEBE TRANSMITIR NUEVAMENTE LA UNIDAD DE DATOS ANTERIOR.";
    text4[]="LA ENTIDAD TRANSMISORA ENVIA VARIAS UNIDADES DE DATOS AL MISMO TIEMPO SIN NECESIDAD DE RECIBIR UN RECONOCIMIENTO (ACK) POR CADA UNIDAD DE DATOS.";
    text5[]="LA ENTIDAD RECEPTORA DETECTA QUE UNA UNIDAD DE DATOS RECIBIDA ES ERRÓNEA POR LO QUE ENVIA UN RECONOCIMIENTO NEGATIVO CON EL NUMERO CORRESPONDIENTE.";
    text6[]="LA ENTIDAD TRANSMISORA ENVIA UNIDADES DE DATOS, A PARTIR DE LA QUE TIENE EL NUMERO RECIBIDO EN EL RECONOCIMIENTO NEGATIVO.";
    text7[]="LA ENTIDAD TRANSMISORA ENVIA UNA SECUENCIA DE BITS, DONDE CADA BIT ES PRECEDIDO POR UN O MAS BITS REDUNDANTES (BR).";
    text8[]="LA ENTIDAD RECEPTORA RECIBE UNA SECUENCIA DE BITS ALGUNOS SON REDUNDANTES CON EL FIN DE NO ENVIAR ALGUN RECONOCIMIENTO O SOLICITAR RETRANSMISION EN CASO DE ERROR";
    text8a[]="A SE DETECTA QUE EN LOS PRIMEROS 3 BITS RECIBIDOS HAY UNA ALTERACION Y POR COMPARACION SE DETERMINA QUE EL VALOR ORIGINAL ES UN 0";
    text8b[]="B EN LOS SIGUIENTES BITS RECIBIDOS NO HAY ALTERACION POR LO QUE SE DETERMINA QUE EL VALOR ORIGINAL ES UN 1";
    text8c[]="C SE DETECTA UNA ALTERACION EN LOS SIGUIENTES BITS RECIBIDOS, SE REALIZA LA COMPARACION Y SE DETERMINA QUE EL VALOR ORIGINAL ES UN 1";
    text9[]="LA ENTIDAD TRANSMISORA ENVIA BLOQUES LIGADOS A UN GRUPO DE BITS REDUNDANTES (BR) BASADOS EN EL CONTENIDO DEL BLOQUE, LOS CUALES GARANTIZARAN QUE EL BLOQUE SEA RECIBIDO SIN ERROR";
    text10a[]="A LA ENTIDAD RECEPTORA RECIBE LOS BLOQUES CON SUS RESPECTIVOS GRUPOS DE BITS REDUNDANTES (BR);
    text10b[]="B CON AYUDA DE LOS 'BR' SE REALIZA UN PROCESAMIENTO PARA EXTRAER LOS BLOQUES ORIGINALES, AUN SI SE PRESENTAN ERRORES OGACIONADOS DURANTE LA TRANSMISION.";
    /** Determina las dimensiones de la pantalla      ***

```

```

hPantalla=GetDesktopWindow();
hPantDC=GetDC(hPantalla);
ancho_pant=GetDeviceCaps(hPantDC,HORZRES);
alto_pant=GetDeviceCaps(hPantDC,VERTRES);
ReleaseDC(hPantalla,hPantDC);
posy=(alto_pant/2);
/** Crea áreas de animación y Texto */
hDC=GetDC(hWnd);
hRgn_Anima=CreateRectRgn(185,81,ancho_pant-185,alto_pant-81);
hRgn_Text=CreateRectRgn(51,81,ancho_pant-51,(alto_pant/2)-61);
rc.left=61;rc.top=91;
rc.right=ancho_pant-61;
rc.bottom=(alto_pant/2)-61;
SetTextColor(hDC,RGB(0,0,128));
switch (nIDtempo)
{
/** Animación del paso 1 de "alto y espera" */
case 1:
if ((caso2==0)&(caso3==0)&(caso4==0)&(caso5==0)&(caso6==0)&(caso7==0)&
(caso8==0)&(caso9==0)&(caso10==0))
{
hMem=CreateCompatibleDC(hDC);
if (caso1==0)
{
DrawText(hDC,txt1,-1,&rc,DT_CENTER|DT_WORDBREAK);
hBuf6=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUF6");
hOn=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"ON");
hBit_prev=SelectObject(hMem,hBuf6);
BitBlt(hDC,60,(alto_pant/2)+48,120,50,hMem,0,0,SRCCOPY);
SelectObject(hMem,hOn);
BitBlt(hDC,210,alto_pant-85,13,13,hMem,0,0,SRCCOPY);
SelectObject(hMem,hBit_prev);
DeleteObject(hBuf6);
DeleteObject(hOn);
posx=86;
}
caso1++;
if (caso1==30)
{
hBuf7=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUF7");
hBit_prev=SelectObject(hMem,hBuf7);
BitBlt(hDC,60,(alto_pant/2)+48,120,50,hMem,0,0,SRCCOPY);
SelectObject(hMem,hBit_prev);
DeleteObject(hBuf7);
}
if ((posx<(ancho_pant/2)-50)&(caso1>=30))
{
hDat1=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"DAT1");
hBit_prev=SelectObject(hMem,hDat1);
hRgn_prev=SelectObject(hDC,hRgn_Anima);
BitBlt(hDC,posx,posy,100,30,hMem,0,0,SRCCOPY);
SelectObject(hMem,hBit_prev);
DeleteObject(hDat1);
posx+=2;
}
if ((posx>=(ancho_pant/2)-50)&(posx<ancho_pant-185))
{
hRuido=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"RUIDO");
hBit_prev=SelectObject(hMem,hRuido);
hRgn_prev=SelectObject(hDC,hRgn_Anima);
BitBlt(hDC,posx,posy-38,100,68,hMem,0,0,SRCCOPY);
SelectObject(hMem,hBit_prev);
DeleteObject(hRuido);
posx+=2;
}
DeleteDC(hMem);
if (posx>=ancho_pant-185)
{
ReleaseDC(hWnd,hDC);
hdc=GetDC(hWnd);
hmem=CreateCompatibleDC(hdc);
hBufv=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUFV");
hOff=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"OFF");
hBit_prev=SelectObject(hmem,hBufv);
BitBlt(hdc,60,(alto_pant/2)+48,120,50,hmem,0,0,SRCCOPY);
SelectObject(hmem,hOff);
BitBlt(hdc,210,alto_pant-85,13,13,hmem,0,0,SRCCOPY);
SelectObject(hmem,hBit_prev);
DeleteDC(hmem);
DeleteObject(hBufv);
DeleteObject(hOff);
hRgn_prev=SelectObject(hdc,hRgn_Text);
PaintRgn(hdc,hRgn_Text);
KillTimer(hWnd,nIDtempo);
ReleaseDC(hWnd,hdc);
caso1=0;
}
}
break;
/** Animación del paso 2 de "alto y espera" */
case 2:
if ((caso1==0)&(caso3==0)&(caso4==0)&(caso5==0)&(caso6==0)&(caso7==0)&
(caso8==0)&(caso9==0)&(caso10==0))

```

```

    hMem=CreateCompatibleDC(hdc);
    if (caso2==0)
    {
        DrawText(hdc,txt2,-1,&rc,DT_CENTER|DT_WORDBREAK);
        hBuf7=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUF7");
        hBuf1=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUF1E");
        hOn=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"ON");
        hBit_prev=SelectObject(hMem,hBuf7);
        BitBlt(hdc,60,(alto_pant/2)+40,120,50,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hBuf1);
        BitBlt(hdc,ancho_pant-185,(alto_pant/2)+48,120,50,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hOn);
        BitBlt(hdc,210,alto_pant-85,13,13,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hBit_prev);
        DeleteObject(hBuf7);
        DeleteObject(hBuf1);
        DeleteObject(hOn);
        posx=ancho_pant-185;
    }
    caso2++;
    if (posx>85)
    {
        hNack=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"NACK");
        hBit_prev=SelectObject(hMem,hNack);
        hRgn_prev=SelectObject(hdc,hRgn_Anima);
        BitBlt(hdc,posx,posy,100,30,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hBit_prev);
        DeleteObject(hNack);
        posx-=2;
    }
    DeleteDC(hMem);
    if (posx<=85)
    {
        ReleaseDC(hWnd,hdc);
        hdc=GetDC(hWnd);
        hmem=CreateCompatibleDC(hdc);
        hBufv=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUFV");
        hOff=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"OFF");
        hBit_prev=SelectObject(hmem,hBufv);
        BitBlt(hdc,60,(alto_pant/2)+48,120,50,hmem,0,0,SRCCOPY);
        BitBlt(hdc,ancho_pant-185,(alto_pant/2)+48,120,50,hmem,0,0,SRCCOPY);
        SelectObject(hmem,hOff);
        BitBlt(hdc,210,alto_pant-85,13,13,hmem,0,0,SRCCOPY);
        SelectObject(hmem,hBit_prev);
        DeleteDC(hmem);
        DeleteObject(hBufv);
        DeleteObject(hOff);
        hRgn_prev=SelectObject(hdc,hRgn_Text);
        PaintRgn(hdc,hRgn_Text);
        KillTimer(hWnd,nDtempo);
        ReleaseDC(hWnd,hdc);
        caso2=0;
    }
}
break;
/** Animación del paso 3 de "alto y espera"      **/
caso 3:
if ((caso1==0)&&(caso2==0)&&(caso4==0)&&(caso5==0)&&(caso6==0)&&(caso7==0)&&
(caso8==0)&&(caso9==0)&&(caso10==0))
{
    hMem=CreateCompatibleDC(hdc);
    if (caso3==0)
    {
        DrawText(hdc,txt3,-1,&rc,DT_CENTER|DT_WORDBREAK);
        hBuf7=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUF7");
        hOn=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"ON");
        hBit_prev=SelectObject(hMem,hBuf7);
        BitBlt(hdc,60,(alto_pant/2)+48,120,50,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hOn);
        BitBlt(hdc,210,alto_pant-85,13,13,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hBit_prev);
        DeleteObject(hBuf7);
        DeleteObject(hOn);
        posx=85;
    }
    caso3++;
    if (posx<ancho_pant-185)
    {
        hDat1=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"DAT1");
        hBit_prev=SelectObject(hMem,hDat1);
        hRgn_prev=SelectObject(hdc,hRgn_Anima);
        BitBlt(hdc,posx,posy,100,30,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hBit_prev);
        DeleteObject(hDat1);
        posx+=2;
    }
    DeleteDC(hMem);
    if (posx==ancho_pant-185)
    {
        ReleaseDC(hWnd,hdc);
        hdc=GetDC(hWnd);
        hmem=CreateCompatibleDC(hdc);
        hBufv=LoadBimap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUFV");

```

```

hOff=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"OFF");
hBit_prev=SelectObject(hMem,hBufv);
BitBlt(hdc,60,(alto_pant/2)+48,120,50,hmem,0,0,SRCCOPY);
SelectObject(hMem,hOff);
BitBlt(hdc,210,alto_pant-85,13,13,hmem,0,0,SRCCOPY);
SelectObject(hMem,hBit_prev);
DeleteDC(hMem);
DeleteObject(hBufv);
DeleteObject(hOff);
hRgn_prev=SelectObject(hdc,hRgn_Text);
PaintRgn(hdc,hRgn_Text);
KillTimer(hWnd,nIDtempo);
ReleaseDC(hWnd,hdc);
caso3=0;
}
}
break;
/** Animación del paso "A" de "ventanas deslizantes" */
case 4:
if ((caso1==0)&(caso2==0)&(caso3==0)&(caso5==0)&(caso6==0)&(caso7==0)&
(caso8==0)&(caso9==0)&(caso10==0))
{
hMem=CreateCompatibleDC(hdc);
if (caso4==0)
{
DrawText(hdc,txt4,-1,&rc,DT_CENTER|DT_WORDBREAK);
hBuf8=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUF8");
hOn=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"ON");
hBit_prev=SelectObject(hMem,hBuf8);
BitBlt(hdc,60,(alto_pant/2)+48,120,60,hMem,0,0,SRCCOPY);
SelectObject(hMem,hOn);
BitBlt(hdc,210,alto_pant-45,13,13,hMem,0,0,SRCCOPY);
SelectObject(hMem,hBit_prev);
DeleteObject(hBuf8);
DeleteObject(hOn);
posx=85;
}
caso4++;
if (caso4==30)
{
hBuf3=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUF3");
hBit_prev=SelectObject(hMem,hBuf3);
BitBlt(hdc,60,(alto_pant/2)+48,120,50,hMem,0,0,SRCCOPY);
SelectObject(hMem,hBit_prev);
DeleteObject(hBuf3);
}
if ((posx<(ancho_pant/2)-50)&(caso4>=30))
{
hDat5=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"DAT5");
hBit_prev=SelectObject(hMem,hDat5);
hRgn_prev=SelectObject(hdc,hRgn_Anima);
BitBlt(hdc,posx,posy,100,30,hMem,0,0,SRCCOPY);
SelectObject(hMem,hBit_prev);
DeleteObject(hDat5);
posx+=2;
}
if ((posx>=(ancho_pant/2)-50)&(posx<ancho_pant-185))
{
hRuido=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"RUIDO2");
hBit_prev=SelectObject(hMem,hRuido);
hRgn_prev=SelectObject(hdc,hRgn_Anima);
BitBlt(hdc,posx,posy-38,100,68,hMem,0,0,SRCCOPY);
SelectObject(hMem,hBit_prev);
DeleteObject(hRuido);
posx+=2;
}
DeleteDC(hMem);
if (posx>=ancho_pant-185)
{
ReleaseDC(hWnd,hdc);
hdc=GetDC(hWnd);
hMem=CreateCompatibleDC(hdc);
hBufv=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUFV");
hOff=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"OFF");
hBit_prev=SelectObject(hMem,hBufv);
BitBlt(hdc,60,(alto_pant/2)+48,120,50,hmem,0,0,SRCCOPY);
SelectObject(hMem,hOff);
BitBlt(hdc,210,alto_pant-45,13,13,hmem,0,0,SRCCOPY);
SelectObject(hMem,hBit_prev);
DeleteDC(hMem);
DeleteObject(hBufv);
DeleteObject(hOff);
hRgn_prev=SelectObject(hdc,hRgn_Text);
PaintRgn(hdc,hRgn_Text);
KillTimer(hWnd,nIDtempo);
ReleaseDC(hWnd,hdc);
caso4=0;
}
}
break;
/** Animación del paso "B" de "ventanas deslizantes" */
case 5:
if ((caso1==0)&(caso2==0)&(caso3==0)&(caso4==0)&(caso6==0)&(caso7==0)&
(caso8==0)&(caso9==0)&(caso10==0))

```

## CAPITULO 4: DISEÑO DE REDTUI

```

    hMem=CreateCompatibleDC(hDC);
    if (caso5==0)
    {
        DrawText(hDC,txt5,-1,&rc,DT_CENTER|DT_WORDBREAK);
        hBuf5=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUF5E");
        hBuf3=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUF3");
        hOn=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"ON");
        hBit_prev=SelectObject(hMem,hBuf3);
        BitBlt(hDC,60,(alto_pant/2)+48,120,50,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hBuf5);
        BitBlt(hDC,ancho_pant-185,(alto_pant/2)+48,120,50,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hOn);
        BitBlt(hDC,210,alto_pant-45,13,13,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hBit_prev);
        DeleteObject(hBuf3);
        DeleteObject(hBuf5);
        DeleteObject(hOn);
        posx=ancho_pant-185;
    }
    caso5++;
    if (posx>85)
    {
        hNack2=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"NACK2");
        hBit_prev=SelectObject(hMem,hNack2);
        hRgn_prev=SelectObject(hDC,hRgn_Anima);
        BitBlt(hDC,posx,100,30,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hBit_prev);
        DeleteObject(hNack2);
        posx-=2;
    }
    DeleteDC(hMem);
    if (posx<=85){
        ReleaseDC(hWnd,hDC);
        hdc=GetDC(hWnd);
        hmem=CreateCompatibleDC(hdc);
        hBufv=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUFV");
        hOf=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"OFF");
        hBit_prev=SelectObject(hmem,hBufv);
        BitBlt(hdc,ancho_pant-185,(alto_pant/2)+48,120,50,hmem,0,0,SRCCOPY);
        BitBlt(hdc,60,(alto_pant/2)+48,120,50,hmem,0,0,SRCCOPY);
        SelectObject(hmem,hOf);
        BitBlt(hdc,210,alto_pant-45,13,13,hmem,0,0,SRCCOPY);
        SelectObject(hmem,hBit_prev);
        DeleteDC(hmem);
        DeleteObject(hBufv);
        DeleteObject(hOf);
        hRgn_prev=SelectObject(hdc,hRgn_Text);
        PaintRgn(hdc,hRgn_Text);
        KillTimer(hWnd,niDiamo);
        ReleaseDC(hWnd,hdc);
        caso5=0;
    }
}
break;
/** Animación del paso "C" de "ventanas deslizantes" **/
case 8:
    if ((caso1==0)&&(caso2==0)&&(caso3==0)&&(caso4==0)&&(caso5==0)&&(caso7==0)&&
        (caso8==0)&&(caso9==0)&&(caso10==0))
    {
        hMem=CreateCompatibleDC(hDC);
        if (caso6==0)
        {
            DrawText(hDC,txt6,-1,&rc,DT_CENTER|DT_WORDBREAK);
            hBuf6=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUF6");
            hOn=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"ON");
            hBit_prev=SelectObject(hMem,hBuf6);
            BitBlt(hDC,80,(alto_pant/2)+48,120,50,hMem,0,0,SRCCOPY);
            SelectObject(hMem,hOn);
            BitBlt(hDC,210,alto_pant-45,13,13,hMem,0,0,SRCCOPY);
            SelectObject(hMem,hBit_prev);
            DeleteObject(hBuf6);
            DeleteObject(hOn);
            posx=85;
        }
        caso6++;
        if (caso6==30)
        {
            hBuf7=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUF_7");
            hBit_prev=SelectObject(hMem,hBuf7);
            BitBlt(hDC,60,(alto_pant/2)+48,120,50,hMem,0,0,SRCCOPY);
            SelectObject(hMem,hBit_prev);
            DeleteObject(hBuf7);
        }
        if ((posx<ancho_pant-185)&&(caso6==30))
        {
            hDat5=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"DAT5_2");
            hBit_prev=SelectObject(hMem,hDat5);
            hRgn_prev=SelectObject(hDC,hRgn_Anima);
            BitBlt(hDC,posx,100,30,hMem,0,0,SRCCOPY);
            SelectObject(hMem,hBit_prev);
            DeleteObject(hDat5);
            posx+=2;
        }
    }
}

```

```

DeleteDC(hMem);
if (posx >= ancho_pant-185)
{
    ReleaseDC(hWnd,hDC);
    hdc=GetDC(hWnd);
    hmem=CreateCompatibleDC(hdc);
    hBufv=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUFV");
    hOff=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"OFF");
    hBit_prev=SelectObject(hmem,hBufv);
    BitBlt(hdc,60,(alto_pant/2)+48,120,50,hmem,0,0,SRCCOPY);
    SelectObject(hmem,hOff);
    BitBlt(hdc,210,alto_pant-45,13,13,hmem,0,0,SRCCOPY);
    SelectObject(hmem,hBit_prev);
    DeleteDC(hmem);
    DeleteObject(hOff);
    DeleteObject(hBufv);
    hRgn_prev=SelectObject(hdc,hRgn_Text);
    PaintFgn(hdc,hRgn_Text);
    KillTimer(hWnd,niDlampo);
    ReleaseDC(hWnd,hdc);
    caso6=0;
}
}
break;
/** Animación del paso 1 de autocorrección convencional **/
caso 7:
if ((caso1==0)&(caso2==0)&(caso3==0)&(caso4==0)&(caso5==0)&(caso6==0)&
(caso8==0)&(caso9==0)&(caso10==0))
{
    hMem=CreateCompatibleDC(hdc);
    if (caso7==0)
    {
        DrawText(hdc,lexi7,-1,&rc_DT_CENTER|DT_WORDBREAK);
        hBufv=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUFBIN");
        hOn=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"ON");
        hBit_prev=SelectObject(hMem,hBufv);
        BitBlt(hdc,60,(alto_pant/2)+48,120,50,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hOn);
        BitBlt(hdc,410,alto_pant-85,13,13,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hBit_prev);
        DeleteObject(hBufv);
        DeleteObject(hOn);
        posx=85;
    }
    caso7++;
    if (caso7==30)
    {
        hBufv=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUFV");
        hBit_prev=SelectObject(hMem,hBufv);
        BitBlt(hdc,60,(alto_pant/2)+48,120,50,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hBit_prev);
        DeleteObject(hBufv);
    }
    if ((posx<(ancho_pant/2)-50)&(caso7>=30))
    {
        hDalb=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"DATABIN");
        hBit_prev=SelectObject(hMem,hDalb);
        hRgn_prev=SelectObject(hdc,hRgn_Anima);
        BitBlt(hdc,posx,posy,100,30,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hBit_prev);
        DeleteObject(hDalb);
        posx+=2;
    }
    if ((posx>=(ancho_pant/2)-50)&(posx<ancho_pant-185))
    {
        hRuIdo=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"RUIBIN");
        hBit_prev=SelectObject(hMem,hRuIdo);
        hRgn_prev=SelectObject(hdc,hRgn_Anima);
        BitBlt(hdc,posx,posy-38,100,68,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hBit_prev);
        DeleteObject(hRuIdo);
        posx+=2;
    }
}
DeleteDC(hMem);
if (posx >= ancho_pant-185)
{
    ReleaseDC(hWnd,hDC);
    hdc=GetDC(hWnd);
    hmem=CreateCompatibleDC(hdc);
    hBufv=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUFV");
    hOff=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"OFF");
    hBit_prev=SelectObject(hmem,hBufv);
    BitBlt(hdc,60,(alto_pant/2)+48,120,50,hmem,0,0,SRCCOPY);
    SelectObject(hmem,hOff);
    BitBlt(hdc,410,alto_pant-85,13,13,hmem,0,0,SRCCOPY);
    SelectObject(hmem,hBit_prev);
    DeleteDC(hmem);
    DeleteObject(hBufv);
    DeleteObject(hOff);
    hRgn_prev=SelectObject(hdc,hRgn_Text);
    PaintFgn(hdc,hRgn_Text);
    KillTimer(hWnd,niDlampo);
    ReleaseDC(hWnd,hdc);
    caso7=0;
}
}

```

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

## CAPITULO 4: DISEÑO DE REDTUT

```

    }
}
break;
/** Animación del paso 2 de autocorrección convolucional      **/
case 8:
    if ((caso1==0)&(caso2==0)&(caso3==0)&(caso4==0)&(caso5==0)&(caso6==0)&
        (caso7==0)&(caso8==0)&(caso10==0))
    {
        hMem=CreateCompatibleDC(hDC);
        Pen1=GetStockObject(NULL_PEN);
        if (caso8==0)
        {
            DrawText(hDC, text8, -1, &rc, DT_CENTER|DT_WORDBREAK);
            hBufb=LoadBitmap(GetWindowWord(hWnd, GWW_HINSTANCE), "BUFBI8");
            hOn=LoadBitmap(GetWindowWord(hWnd, GWW_HINSTANCE), "ON");
            hBit_prev=SelectObject(hMem, hBufb);
            BitBlt(hDC, ancho_pant-185, (alto_pant/2)+48, 120, 50, hMem, 0, 0, SRCCOPY);
            SelectObject(hMem, hOn);
            BitBlt(hDC, 410, alto_pant-85, 13, 13, hMem, 0, 0, SRCCOPY);
            SelectObject(hMem, hBit_prev);
            DeleteObject(hBufb);
            DeleteObject(hOn);
        }
        caso8++;
        if (caso8==200)
        {
            Penprev=SelectObject(hDC, Pen1);
            Rectangle(hDC, 51, 81, ancho_pant-51, (alto_pant/2)-61);
            DrawText(hDC, text8a, -1, &rc, DT_LEFT|DT_WORDBREAK);
        }
        if (caso8==380)
        {
            hBufb=LoadBitmap(GetWindowWord(hWnd, GWW_HINSTANCE), "BUFBI1");
            hBit_prev=SelectObject(hMem, hBufb);
            BitBlt(hDC, ancho_pant-185, (alto_pant/2)+48, 120, 50, hMem, 0, 0, SRCCOPY);
            SelectObject(hMem, hBit_prev);
            DeleteObject(hBufb);
        }
        if (caso8==400)
        {
            Penprev=SelectObject(hDC, Pen1);
            Rectangle(hDC, 51, 81, ancho_pant-51, (alto_pant/2)-61);
            DrawText(hDC, text8b, -1, &rc, DT_LEFT|DT_WORDBREAK);
        }
        if (caso8==580)
        {
            hBufb=LoadBitmap(GetWindowWord(hWnd, GWW_HINSTANCE), "BUFBI2");
            hBit_prev=SelectObject(hMem, hBufb);
            BitBlt(hDC, ancho_pant-185, (alto_pant/2)+48, 120, 50, hMem, 0, 0, SRCCOPY);
            SelectObject(hMem, hBit_prev);
            DeleteObject(hBufb);
        }
        if (caso8==600)
        {
            Penprev=SelectObject(hDC, Pen1);
            Rectangle(hDC, 51, 81, ancho_pant-51, (alto_pant/2)-61);
            DrawText(hDC, text8c, -1, &rc, DT_LEFT|DT_WORDBREAK);
        }
        if (caso8==780)
        {
            hBufb=LoadBitmap(GetWindowWord(hWnd, GWW_HINSTANCE), "BUFBI3");
            hBit_prev=SelectObject(hMem, hBufb);
            BitBlt(hDC, ancho_pant-185, (alto_pant/2)+48, 120, 50, hMem, 0, 0, SRCCOPY);
            SelectObject(hMem, hBit_prev);
            DeleteObject(hBufb);
        }
        if (caso8==800)
        {
            hBufb=LoadBitmap(GetWindowWord(hWnd, GWW_HINSTANCE), "BUFV");
            hOff=LoadBitmap(GetWindowWord(hWnd, GWW_HINSTANCE), "OFF");
            hBit_prev=SelectObject(hMem, hBufb);
            BitBlt(hDC, ancho_pant-185, (alto_pant/2)+48, 120, 50, hMem, 0, 0, SRCCOPY);
            SelectObject(hMem, hOff);
            BitBlt(hDC, 410, alto_pant-85, 13, 13, hMem, 0, 0, SRCCOPY);
            SelectObject(hMem, hBit_prev);
            DeleteDC(hMem);
            DeleteObject(hBufb);
            DeleteObject(hOff);
            SelectObject(hDC, hRgn_Text);
            PaintRgn(hDC, hRgn_Text);
            KillTimer(hWnd, niDtempo);
            caso8=0;
        }
        SelectObject(hDC, Penprev);
        DeleteObject(Pen1);
        DeleteDC(hMem);
    }
}
break;
/** Animación del paso 1 de autocorrección hacia adelante      **/
case 9:
    if ((caso1==0)&(caso2==0)&(caso3==0)&(caso4==0)&(caso5==0)&(caso6==0)&
        (caso7==0)&(caso8==0)&(caso10==0))
    {
        hMem=CreateCompatibleDC(hDC);
    }
}

```

```

if (caso9==0)
{
DrawText(hdc,txt9,-1,&rc,DT_CENTER|DT_WORDBREAK);
hBufb=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUFBL0");
hOn=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"ON");
hBit_prev=SelectObject(hMem,hBufb);
BitBlt(hdc,60,(alto_pant/2)+48,120,60,hMem,0,0,SRCCOPY);
SelectObject(hMem,hOn);
BitBlt(hdc,410,alto_pant-45,13,13,hMem,0,0,SRCCOPY);
SelectObject(hMem,hBit_prev);
DeleteObject(hBufb);
DeleteObject(hOn);
posx+=85;
}
caso9++;
if (caso9==30)
{
hBufv=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUFV");
hBit_prev=SelectObject(hMem,hBufv);
BitBlt(hdc,60,(alto_pant/2)+48,120,60,hMem,0,0,SRCCOPY);
SelectObject(hMem,hBit_prev);
DeleteObject(hBufv);
}
if ((posx<(ancho_pant/2)-50)&(caso9>=30))
{
hDatb=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"DATBL0");
hBit_prev=SelectObject(hMem,hDatb);
hRgn_prev=SelectObject(hdc,hRgn_Anima);
BitBlt(hdc,posx,posy,100,30,hMem,0,0,SRCCOPY);
SelectObject(hMem,hBit_prev);
DeleteObject(hDatb);
posx+=2;
}
if ((posx>=(ancho_pant/2)-50)&(posx<ancho_pant-185))
{
hRuldo=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"RULBL0");
hBit_prev=SelectObject(hMem,hRuldo);
hRgn_prev=SelectObject(hdc,hRgn_Anima);
BitBlt(hdc,posx,posy-38,100,88,hMem,0,0,SRCCOPY);
SelectObject(hMem,hBit_prev);
DeleteObject(hRuldo);
posx+=2;
}
DeleteDC(hMem);
if (posx==ancho_pant-185)
{
ReleaseDC(hWnd,hdc);
hdc=GetDC(hWnd);
hmem=CreateCompatibleDC(hdc);
hOff=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"OFF");
hBit_prev=SelectObject(hmem,hOff);
BitBlt(hdc,410,alto_pant-45,13,13,hmem,0,0,SRCCOPY);
SelectObject(hmem,hBit_prev);
DeleteDC(hmem);
DeleteObject(hOff);
hRgn_prev=SelectObject(hdc,hRgn_Text);
PaintRgn(hdc,hRgn_Text);
KillTimer(hWnd,nIDiempo);
ReleaseDC(hWnd,hdc);
caso9=0;
}
}
break;
/** Animación del paso 2 de autocorrección hacia adelante **/
caso 10:
if ((caso1==0)&(caso2==0)&(caso3==0)&(caso4==0)&(caso5==0)&(caso6==0)&
(caso7==0)&(caso8==0)&(caso9==0))
{
hMem=CreateCompatibleDC(hdc);
Pen1=GetStockObject(NULL_PEN);
if (caso10==0)
{
DrawText(hdc,tax10,-1,&rc,DT_LEFT|DT_WORDBREAK);
hBufb=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUFBR");
hOn=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"ON");
hBit_prev=SelectObject(hMem,hBufb);
BitBlt(hdc,ancho_pant-185,(alto_pant/2)+48,120,50,hMem,0,0,SRCCOPY);
SelectObject(hMem,hOn);
BitBlt(hdc,410,alto_pant-45,13,13,hMem,0,0,SRCCOPY);
SelectObject(hMem,hBit_prev);
DeleteObject(hBufb);
DeleteObject(hOn);
}
caso10++;
if (caso10==150)
{
Penprev=SelectObject(hdc,Pen1);
Rectangle(hdc,51,81,ancho_pant-51,(alto_pant/2)-61);
DrawText(hdc,tax10e,-1,&rc,DT_LEFT|DT_WORDBREAK);
}
if (caso10==350)
{
hBufb=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUFBR1");
}
}

```

## CAPITULO 4: DISEÑO DE REDTUT

```

        hBit_prev=SelectObject(hMem,hBufb);
        BitBlt(hDC,ancho_pant-185,(alto_pant/2)+48,120,50,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hBit_prev);
        DeleteObject(hBufb);
    }
    if (caso10==400)
    {
        hBufb=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUFBR2");
        hBit_prev=SelectObject(hMem,hBufb);
        BitBlt(hDC,ancho_pant-185,(alto_pant/2)+48,120,50,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hBit_prev);
        DeleteObject(hBufb);
    }
    if (caso10==450)
    {
        hBufb=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUFBL0");
        hBit_prev=SelectObject(hMem,hBufb);
        BitBlt(hDC,ancho_pant-185,(alto_pant/2)+48,120,50,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hBit_prev);
        DeleteObject(hBufb);
    }
    if (caso10==500)
    {
        hBufv=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"BUFV");
        hOf=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"OFF");
        hBit_prev=SelectObject(hMem,hBufv);
        BitBlt(hDC,ancho_pant-185,(alto_pant/2)+48,120,50,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hOf);
        BitBlt(hDC,410,alto_pant-45,13,13,hMem,0,0,SRCCOPY);
        SelectObject(hMem,hBit_prev);
        DeleteDC(hMem);
        DeleteObject(hBufv);
        DeleteObject(hOf);
        SelectObject(hDC,hRgn_Text);
        PaintRgn(hDC,hRgn_Text);
        KillTimer(hWnd,nIDtempo);
        caso10=0;
    }
    SelectObject(hDC,Penprev);
    DeleteObject(Pen1);
    DeleteDC(hMem);
}
break;
}
ReleaseDC(hWnd,hDC);
DeleteObject(hRgn_Anima);
DeleteObject(hRgn_Text);
}
/***** Archivo de definición (anm2.def) del proceso de animación de corrección de errores *****/
NAME Corrarior
DESCRIPTION 'Corrección de errores'
EXETYPE WINDOWS
CODE PRELOAD MOVABLE DISCARDABLE
DATA PRELOAD MOVEABLE MULTIPLE
HEAPSIZE 4096
STACKSIZE 9216
EXPORTS Anima_CE
/***** Archivo de recursos (anm2.rc) del proceso de animación de corrección de errores *****/
ANIMA ICON anima.ico
REDTUT BITMAP redtut2.bmp
TITULO BITMAP titulo.bmp
CAP1 BITMAP cap1.bmp
COMPU BITMAP comp.bmp
ACK BITMAP eck.bmp
NACK BITMAP nack.bmp
NACK2 BITMAP nack2.bmp
ACK5 BITMAP ack5.bmp
DAT1 BITMAP dato1.bmp
DAT2 BITMAP dato2.bmp
DAT3 BITMAP dato3.bmp
DAT5 BITMAP dato5.bmp
DAT5_2 BITMAP dato5_2.bmp
DATBIN BITMAP datobin.bmp
DATBLO BITMAP datoblo.bmp
RUIDO BITMAP ruido.bmp
RUIDO2 BITMAP ruido2.bmp
RUIBIN BITMAP ruibin.bmp
RUIBLO BITMAP ruiblo.bmp
BUFV BITMAP bufvec.bmp
BUF1 BITMAP buf1.bmp
BUF1E BITMAP buf1e.bmp
BUF3 BITMAP buf3.bmp
BUF5 BITMAP buf5.bmp
BUF5E BITMAP buf5e.bmp
BUF6 BITMAP buf6.bmp
BUF7 BITMAP buf7.bmp
BUF_7 BITMAP buf_7.bmp
BUF8 BITMAP buf8.bmp
BUFBIN BITMAP bufbn.bmp
BUFBINE BITMAP bufbine.bmp
BUFBIN1 BITMAP bufbine1.bmp
BUFBIN2 BITMAP bufbine2.bmp

```

```

BUFBLO BITMAP bufblo.bmp
BUFBR BITMAP bufbr.bmp
BUFBR1 BITMAP bufbr1.bmp
BUFBR2 BITMAP bufbr2.bmp
ARQ1 BITMAP arq1.bmp
ARQ2 BITMAP arq2.bmp
ATC1 BITMAP autoc1.bmp
ATC2 BITMAP autoc2.bmp
OFF BITMAP off.bmp
ON BITMAP on.bmp
/..... Archivo proyecto (anm2.pj) del proceso de animación de corrección de errores ...../
anm2.C
anm2.DEF
anm2.RC
    
```

Figura. 4-28 Archivos fuente (anm2.c), de definición (anm2.def), de recursos (anm2.rc) y proyecto (anm2.pj) de la animación del proceso de corrección de errores.

Los demás procesos son creados de forma similar, con la variante de que cada uno implementa la técnica o combinación de técnicas de animación que más le favorezcan.

### 4.3 Diseño de la navegación

Primeramente, debe entenderse como navegación, a la acción de viajar ordenadamente por los tópicos del tutorial con ayuda de los diferentes recursos de acceso a la información, proporcionados por las herramientas de ayuda, que gracias a su diversidad, proporcionan diferentes formas de navegación disponibles para el usuario.

#### 4.3.1 Navegación por temas

La navegación por temas debe ser diseñada para que el usuario pueda acceder los tópicos de información en un orden secuencial para obtener un aprendizaje gradual del contenido de REDTUT. Para éste propósito, es necesario definir cuales serán las propiedades de la navegación de los tópicos de información, los cuales se muestran en la tabla 4-4.

Tópicos de información	Propiedades de navegación
Tópico de contenido	<ul style="list-style-type: none"> <li>◆ No pueden regresar al tópico que les precede (si hay alguno).</li> <li>◆ Tiene la posibilidad de saltar a diferentes capítulos o subcapítulos.</li> <li>◆ Los tópicos de contenido de capítulos o subcapítulos serán accedados directamente por los tópicos de evaluación y tópicos de tema final.</li> </ul>
Tópico de introducción	<ul style="list-style-type: none"> <li>◆ Podrá regresar al tópico de contenido de capítulos.</li> <li>◆ El tópico siguiente será el tópico de contenido de subcapítulos.</li> </ul>
Tópico de tema	<ul style="list-style-type: none"> <li>◆ Pueden estar dentro de una secuencia de páginas.</li> <li>◆ Pueden estar aislado, con el fin de que sean desplegados por ventajas secundarias.</li> </ul>
Tópico de temas múltiples	<ul style="list-style-type: none"> <li>◆ Pueden estar dentro de una secuencia de páginas.</li> <li>◆ Pueden ser tópicos de tema final.</li> </ul>
Tópico de tema final	<ul style="list-style-type: none"> <li>◆ Es el último tópico de una secuencia de páginas.</li> <li>◆ Podrán regresar al tópico de contenido de subcapítulos.</li> <li>◆ Podrán acceder directamente los tópicos de evaluación.</li> <li>◆ Podrán regresar al tópico de contenido de capítulos.</li> </ul>
Tópico de evaluación	<ul style="list-style-type: none"> <li>◆ Debe permitir el acceso a tópicos de contenido de subcapítulos.</li> <li>◆ Debe permitir el acceso a tópicos de contenido de capítulos.</li> <li>◆ No deben ser accedados por índice, por búsqueda o cualquier tópico que no sea de tema final.</li> </ul>

Tabla 4-4 Propiedades de navegación de los tópicos de información del tutorial.

Una vez definidas las propiedades de navegación de los tópicos, es posible crear una navegación general que involucre todos los temas relacionados con el sistema tutorial.

Cabe señalar que dentro de la navegación fundamental del sistema se harán referencias a tópicos externos de otros capítulos, de otros archivos o programas de animación, vía hipertexto, con lo cual se obtiene una integración general del contenido del sistema. A pesar de que se tenga acceso a dichos tópicos externos del capítulo, la navegación del sistema no será interrumpida debido a que serán desplegados en ventanas secundarias o ventanas pop-up.

En la figura 4-29, se muestra la navegación de un subcapítulo del capítulo 1, la cual es semejante a los demás subcapítulos (Apéndice A).

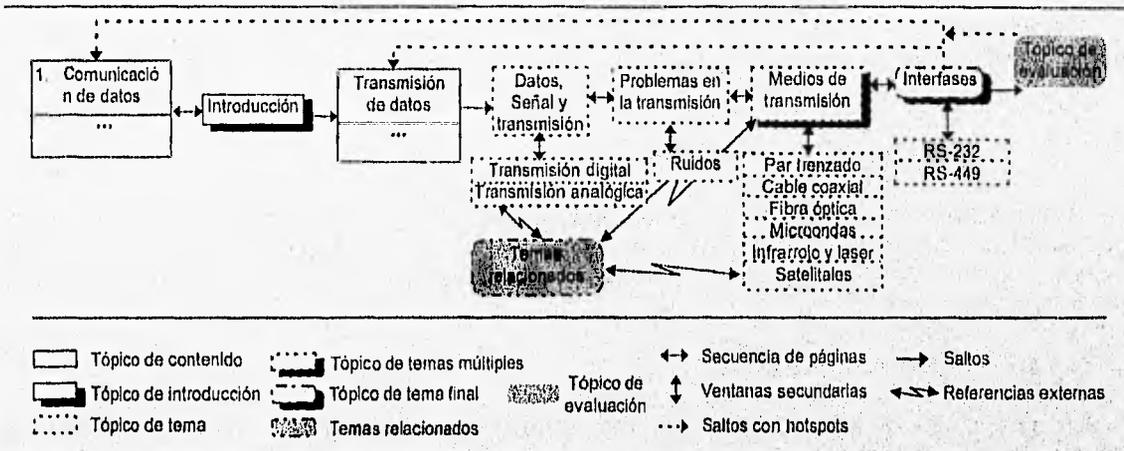


Figura 4-29 Navegación de un subcapítulo del tutorial de redes.

### 4.3.2 Navegación de la evaluación

Una de las partes fundamentales dentro de cualquier tutorial, es la evaluación del aprendizaje del usuario, por lo que se requiere que el diseño de la navegación de esta sección sea adecuada para que el usuario pueda rectificar y/o reafirmar su aprendizaje después de haber revisado el subcapítulo de interés.

Dado que el objetivo de esta sección de evaluación, es que el usuario pueda medir su aprendizaje, no se requiere una escala numérica que lo determine, más bien se debe dar a conocer que dada una respuesta errónea, todavía no se ha logrado la comprensión de un tema.

Como ya se ha mencionado, la forma más adecuada de evaluar al usuario de un tutorial es por medio de preguntas con opciones múltiples. Por lo tanto, nos serviremos de éste hecho, para diseñar la navegación de la parte de evaluación, así como de las siguientes consideraciones:

- ◆ El número de preguntas que contendrá el tópico de evaluación depende del subcapítulo que se va a evaluar (de tres en adelante).
- ◆ Se proporcionarán cinco opciones de respuesta, siendo sólo una la correcta.
- ◆ Las preguntas serán elaboradas de tal forma que involucren lo expuesto en los tópicos revisados y según lo que se considere necesario para conocer si un tema ha sido asimilado.
- ◆ Únicamente estarán visibles los números de preguntas, cuando se despliegue el tópico de evaluación.
- ◆ Las preguntas y las opciones de respuesta serán desplegadas en una ventana pop-up; permitiendo de esta manera cambiar de pregunta sin necesidad de contestar la seleccionada.
- ◆ Sólo existen dos estados posibles (correcto e incorrecto) al elegir una opción como respuesta. En la tabla 4-5 se muestra a donde más conduce cada estado.

Estado de elección	Consecuencias
Correcto	<ul style="list-style-type: none"> <li>◆ Nos mostrará en una ventana pop-up que la respuesta es acertada.</li> <li>◆ Nos permitirá continuar con el proceso de evaluación (con otra pregunta, si la hay).</li> </ul>
Incorrecto	<ul style="list-style-type: none"> <li>◆ El sistema saltará a una ventana secundaria con el tópico que muestra el tema no asimilado.</li> <li>◆ Una vez repasado el tema en cuestión, se permitirá regresar a la parte de evaluación.</li> </ul>

Tabla 4-5 Consecuencias de los posibles estados de elección.

Cabe señalar que no existe algún problema si se desea regresar a la sección de evaluación, cuando se ha elegido una respuesta errónea, dado que los temas de repaso son desplegados en ventanas secundarias que al ser cerradas nos regresan al tópico de evaluación.

Dentro del tópico de evaluación se proporcionan botones para saltar al tópico de contenido de subcapítulos o el de capítulos, los cuales están disponibles en cualquier momento durante la evaluación.

### 4.3.3 Navegación por índice

Para la navegación por índice se utiliza el tópico desplegado por el botón índice proporcionado por las herramientas de ayuda o por medio de la activación del botón "Índice" correspondiente al capítulo seleccionado en la pantalla principal de REDTUT. Este tópico contendrá un índice desglosado del contenido de un capítulo de REDTUT.

En la navegación por índice se da la posibilidad de consultar cualquier tema de interés, sin importar la lectura de tópicos en secuencia ordenada para un aprendizaje gradual, como en el caso de la navegación por temas.

Cuando se despliegue el tópico, sólo estarán visibles los subcapítulos y temas del contenido general.

Realmente la navegación por éste medio es muy sencilla, dado que sólo se desplegarán en ventanas secundarias los tópicos de interés seleccionados, dando la posibilidad de regresar al índice, una vez revisados los tópicos, con sólo activar el icono "cerrar" o por el cuadro de control proporcionado por Windows.

Además el tópico de índice podrá ser accesado por el botón "Índice" contenido en la barra de botones de cualquier capítulo de REDTUT en cualquier momento que se despliegue un tópico de información (Fig. 4-30).



Figura 4-30 La navegación por índice puede ser accesada por medio del botón "Índice" de la barra de botones.

**4.3.4 Navegación por búsqueda**

La navegación por búsqueda, se refiere a la forma proporcionada por el cuadro de búsqueda de las herramientas de ayuda para acceder la información.

Esta forma de acceder la información es muy útil cuando se desea acceder algún tema del cual no se tiene certeza si está contenido en el sistema, dado que en caso de que no estuviera contenido, se muestran otros temas relacionados que podrían ser útiles. El cuadro de búsqueda se despliega al activar el botón "buscar" (Fig. 4-31).

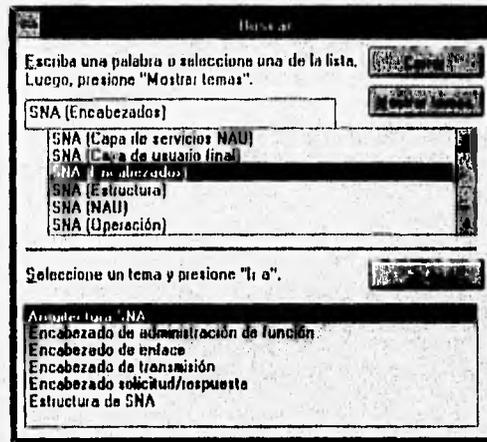


Figura 4-31 Cuadro de diálogo proporcionado por Windows, que permite la navegación por búsqueda.

**4.4 Creación de la ayuda en línea y glosario.**

**4.4.1 Ayuda en línea**

La ayuda en línea es una herramienta de gran utilidad en cualquier aplicación de Windows.

En el caso específico de REDTUT, la ayuda en línea es un recurso de utilidad, dado que existen diferentes formas de navegación y diferentes herramientas que nos permiten interactuar con el sistema tutorial (animación, edición, impresión, anotaciones, etc.).

El método utilizado para la creación de texto e imágenes de la ayuda en línea es el mismo que se describe para la generación de texto e imágenes de cualquiera de los capítulos de REDTUT. Esto es posible dado que la ayuda que es proporcionada, por el sistema de ayuda de Windows, al seleccionar el elemento de menú "Ayuda/Uso de la ayuda" de un archivo .HLP (cualquiera de los capítulos de REDTUT), está asociada también con otro archivo .HLP que en este caso específico contendrá la ayuda sobre el manejo de REDTUT.

Los elementos que debe proporcionar la ayuda en línea de REDTUT son:

- ◆ Descripción de los elementos de la pantalla principal.
- ◆ Ayuda para la navegación por temas.
- ◆ Ayuda para la navegación por índice.
- ◆ Ayuda para la navegación por búsqueda.

- ◆ Ayuda para la navegación de tópicos de evaluación.
- ◆ Descripción de los iconos de control de navegación.
- ◆ Ayuda sobre el manejo de los programas de animación.
- ◆ Ayuda para el uso de la barra de botones.
- ◆ Ayuda para el uso de la barra de menues.
- ◆ Ayuda para imprimir tópicos de información.
- ◆ Ayuda sobre el uso del glosario; etc.

La ayuda en línea tendrá que ser desplegada en una ventana secundaria al activar el elemento de menú "Ayuda/Uso de la ayuda" (de cualquier capítulo), para evitar confusiones dentro del ambiente de REDTUT, si se despliega el archivo de ayuda estandar (Fig.4-32). Asimismo, de poder ser accesada desde la pantalla principal de REDTUT.



Figura 4-32 Elemento de la barra de menú que desplegará la ayuda en línea de REDTUT.

#### 4.4.2 Glosario

El glosario debe desplegar conceptos breves relacionados con la información de los cuatro capítulos que forman el curso de redes de computadoras en REDTUT, con la finalidad de evitar referencias externas entre los diferentes capítulos, cuando sólo nos interesa conocer un concepto relacionado con el tema que se está estudiando.

Para que pueda ser rápido y fácil el acceso de los conceptos contenidos en el glosario, se proporciona un panel de botones nombrados con el alfabeto, que nos permite saltar a la lista de palabras que inician con la letra del botón activado (si las hay). En caso de que el glosario no contenga definición de palabras que inicien con la letra del botón activado, este nos saltará a listado de una letra alfabéticamente anterior a la seleccionada (Fig. 4-33).



Figura 4-33 Panel de botones del glosario de REDTUT.

Debido a que el glosario debe ser accesado por cualquiera de los cuatro capítulos o desde la pantalla principal, es necesario que se trate como un archivo independiente. En este caso también se manejará como un archivo con formato .HLP. que será desplegado en una ventana secundaria al ser activado el botón "Glosario" de la barra de botones de cada capítulo o de la pantalla principal.

# **CAPITULO 5**

## **INTEGRACION DE REDTUT**

5.1 Integración de texto, imágenes y animación.

5.2 Compilación de REDTUT.

## 5.1 INTEGRACION DE TEXTO, IMAGENES Y ANIMACION.

Debido a que el texto, las imágenes y la animación fueron creados en archivos independientes (.RTF, .BMP, .EXE; respectivamente), es necesario integrarlos en un solo archivo (.HLP) para la generación de REDTUT. Las herramientas de ayuda, con el uso de referencias externas a archivos y el uso de macros, nos permitirán realizar dicha integración, insertando las imágenes y la animación en el archivo de texto (.RTF).

### 5.1.1 Integración de imágenes al texto.

Las herramientas de ayuda permiten realizar la inserción de las imágenes al archivo de texto (.RTF) de dos formas distintas:

1. Inserción directa.
2. Inserción por referencia.

Si la inserción se hace de manera directa debe utilizarse la opción de copiar al portapapeles que proporciona Windows, para después pegar la imagen en la posición deseada en el archivo de texto (.RTF).

En el caso de que la inserción sea por referencia podemos usar 6 comandos que nos permiten insertar las imágenes alineadas con el texto en diferentes formas (Tabla 5-1).

Comando de referencia	Significado
{bmc nom_arch}	Alinea la imagen como un carácter de texto y están guardadas separadamente del texto.
{bml nom_arch}	Alinea la imagen a la izquierda de la pantalla y están guardadas separadamente del texto.
{bmr nom_arch}	Alinea la imagen a la derecha de la pantalla y están guardadas separadamente del texto.
{bmcwd nom_arch}	Alinea la imagen como un carácter de texto y la guarda junto con el texto.
{bmlwd nom_arch}	Alinea la imagen a la izquierda de la pantalla y la guarda junto con el texto.
{bmrwd nom_arch}	Alinea la imagen a la derecha de la pantalla y la guarda junto con el texto.

Tabla 5-1 Comandos de referencia para inserción de imágenes en el texto.

Como se muestra en la tabla V.1-1, las formas de alineación posibles son tres, pero con la variante de guardar la imagen dentro del archivo de texto (.RTF) o no. Obviamente si se elige guardar la imagen dentro del archivo de texto, tal archivo se hará más grande; pero, teniendo la ventaja de que sólo se tendrá un archivo con imágenes y no varios archivos separados.

*Imagen alineada a la izquierda*

La inserción de una imagen alineada a la izquierda, se muestra en la figura 5-1.

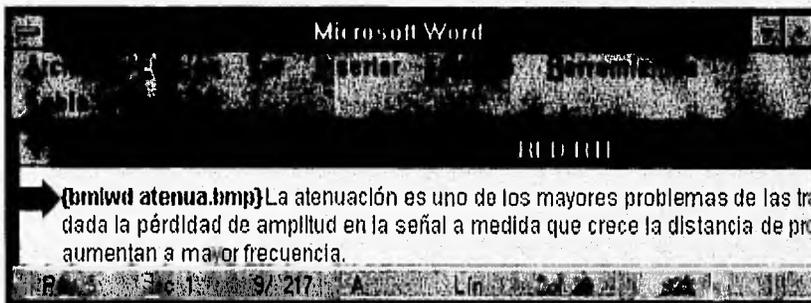


Figura 5-1 Inserción de una imagen alineada a la izquierda.

La forma como se desplegará al usuario se muestra en la figura 5-2.

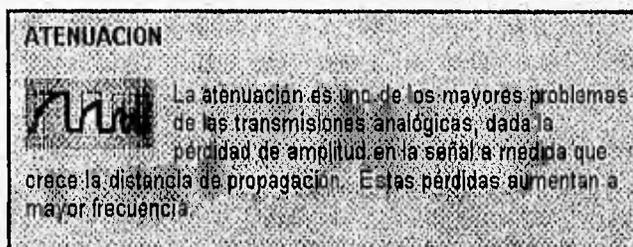


Figura 5-2 Imagen alineada a la izquierda.

*Imagen alineada a la derecha.*

La inserción de una imagen alineada a la derecha, se muestra en la figura 5-3.

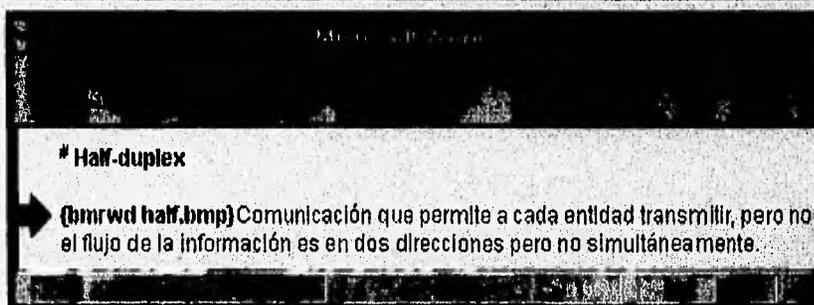


Figura 5-3 Inserción de una imagen a la derecha.

El uso de "retornos flexibles" (↵) es para tratar a las líneas de texto como un sólo párrafo. De esta manera es posible controlar la apariencia del texto y de la imagen. El efecto producido por la inserción de una imagen a la derecha y el uso de "retornos flexibles" se muestra en la figura 5-4.

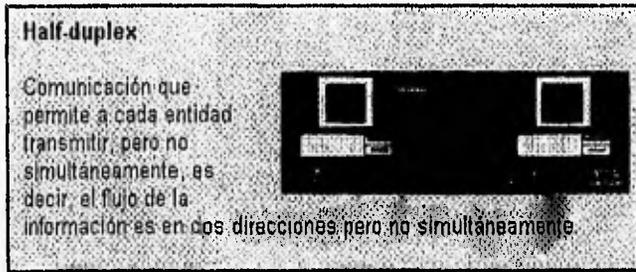


Figura 5-4 Imagen alineada a la derecha.

**Imagen alineada como un carácter de texto.**

La inserción de una imagen alineada como un carácter de texto se muestra en la figura 5-5.

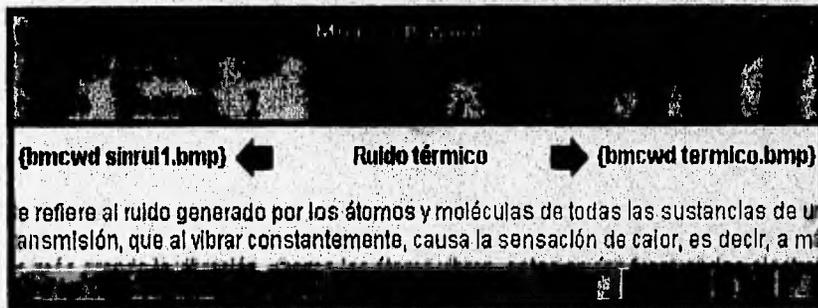


Figura 5-5 Inserción de una imagen alineada como carácter.

Debe considerarse que las imágenes, independientemente de la altura del texto en uso, quedarán alineadas con la línea base del texto. En caso de querer dar una apariencia distinta, se debe hacer uso de las propiedades para formatear párrafos. La forma como se desplegará al usuario se muestra en la figura 5-6.

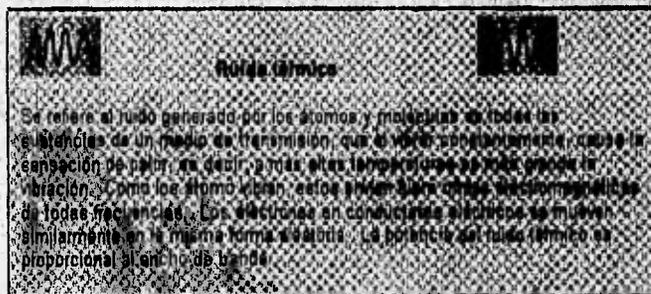


Figura 5-6 Imagen alineada como carácter.

El parámetro nom\_arch, en todos los casos, debe ser el nombre del archivo donde está guardada la imagen, con extensión. Si no se le antepone alguna ruta al archivo, se asume que está guardado en el mismo directorio donde está el archivo de texto (.RTF).

**5.1.2 Integración de animación al texto.**

Debido a que el diseño de la animación de REDTUT, está basado en técnicas de programación tradicional, donde el archivo resultante es ejecutable (.EXE); para la integración de la animación debe utilizarse de una macro que permita la ejecución de una aplicación de Windows, sin necesidad de salirse de la ventana principal proporcionada por las herramientas de ayuda. Nos estamos refiriendo a la macro: ExecProgram( ).

La macro ExecProgram, nos permite extender la potencia de los sistemas de ayuda al ejecutar cualquier aplicación de Windows.

La sintaxis de ExecProgram( ) se muestra en la tabla 5-2:

Sintaxis	Parámetros	
ExecProgram("línea_comandos",edo_despliegue )	línea_comandos	edo_despliegue
o EP("línea_comandos,edo_despliegue)	Línea de comandos de la aplicación que se quiere ejecutar. Puede contener parámetros separados por " / ".	Determina como aparecerá la aplicación al ser incluida: <u>Valor</u> <u>Acción</u> 0    despliegue normal 1    minimizada en un ícono 2    maximizada

Tabla 5-2 Sintaxis de la macro ExecProgram( ).

En el caso específico de REDTUT la macro ExecProgram se ejecutará cuando el usuario active un hotspot que tenga la macro asociada (botón con imagen de cámara de video), desplegando en una ventana secundaria el tópico de animación correspondiente al tema en cuestión (Fig.5-7).

Por lo tanto, la forma de insertar la animación al texto, depende de asociar un hotspot con una macro desde el editor de hotspots o al cambiar el formato de la referencia a una imagen, la cual posteriormente podrá ser insertada al texto como cualquier imagen.

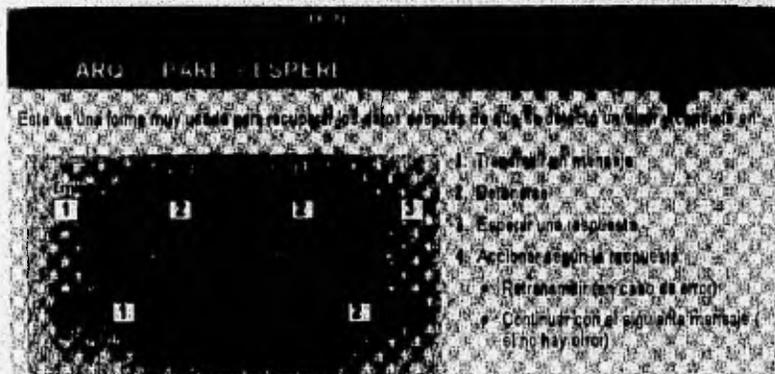


Figura 5-7 Animación a través de un hotspot.

## 5.2 COMPILACION DE REDTUT.

De acuerdo con el diseño del ambiente de REDTUT, el sistema está compuesto por dos partes fundamentales: la parte desarrollada en los archivos .HLP, cuyo contenido es texto, imágenes y navegación de REDTUT, y la parte que le da el ambiente de una aplicación en Windows (.EXE). Por tal motivo, la compilación de REDTUT se divide en dos etapas:

1. Generación de archivos .HLP.
2. Generación del archivo .EXE.

### 5.2.1 Generación de archivos .HLP

Los archivos .HLP son archivos binarios que pueden ser leídos por WINHELP.EXE y son construidos por el compilador de ayuda (HCP.EXE) con base en los archivos de tópicos (.RTF), de imágenes (.BMP) y del archivo proyecto (.HPJ).

El compilador de ayuda es proporcionado por varios productos de desarrollo de software como SDK para Windows, Microsoft C/C++, Microsoft Visual Basic, Borland Zortech, Borland C++, etc. Existen tres versiones disponibles que se muestran en la tabla 5-3.

Versión	Descripción
HC30.EXE	Compilador de ayuda para Windows 3.0. Los archivos generados son compatibles con Windows 3.0 y Windows 3.1.
HC31.EXE	Compilador de ayuda para Windows 3.1. Los archivos generados no son compatibles con Windows 3.0.
HCP.EXE	Compilador de ayuda para modo protegido de Windows 3.1. Hace mejor uso de la memoria. Los archivos generados no son compatibles con Windows 3.0.

Tabla 5-3 Versiones del compilador de ayuda en Windows.

Independientemente del compilador que se haya elegido, debe ser ejecutado desde el prompt del DOS; debido a que es una aplicación para MSDOS. En caso de querer ejecutar el compilador dentro de Windows, se requerirá de archivos de información de programa (PIF, *Program Information Files*) que lo permitan, como: HC30.PIF, HC31.PIF y/o HCP.PIF.

Varios elementos que componen la presentación de los archivos .HLP generados por el compilador, se definen en el archivo proyecto (.HPJ) que es un archivo de texto que controla como el compilador construirá los tópicos del archivo .HLP.

Cada sistema de ayuda debe tener su propio archivo proyecto, el cual puede ser creado en cualquier editor de texto y debe ser guardado con extensión .HPJ.

Los archivos proyecto se componen de nueve secciones que se muestran en la tabla 5-4.

Sección	Descripción
[ OPTIONS ]	Especifica varias opciones que controlan como el archivo de ayuda es construido. Debe ser la primera sección en el archivo. Es opcional.
[ FILES ]	Especifica los archivos tópicos (.RTF) incluidos en el archivo de ayuda.
[ BUILDTAGS ]	Especifica indicadores de etiquetas válidos para realizar una compilación condicional.
[ CONFIG ]	Especifica cualquier modificación de menús y botones en el archivo de ayuda y registra funciones DLL.
[ BITMAPS ]	Especifica los archivos de mapas de bits incluidos en el archivo de ayuda.
[ MAP ]	Asocia cadenas de contexto con números de contexto de una aplicación de Windows. En esta sección se pueden crear ligas entre la ayuda y las aplicaciones de Windows. Es opcional.
[ ALIAS ]	Asigna una o mas cadenas de contexto al mismo tópico. Es opcional.
[ WINDOWS ]	Controla la apariencia de la ventana principal de ayuda y cualquier ventana secundaria que se haya creado.
[ BAGGAGE ]	Lista archivos de multimedia que se queran insertar en los archivos de ayuda .HLP.

Tabla 5-4 Secciones componentes de un archivo proyecto.

En el caso específico de REDTUT se tuvieron que crear 4 archivos proyecto, uno por capítulo, con el fin de dar una apariencia distinta a cada capítulo.

En la figura 5-8 se muestra el archivo proyecto de un capítulo de REDTUT.

```

;DESARROLLO DEL CAPITULO 3: ARQUITECTURA DE REDES
[OPTIONS]
BMROOT=C:\USUARIO\AA\TESIS\CAP3
CONTENTS=INDICE
TITLE=CAPITULO 3: ARQUITECTURAS DE REDES
COMPRESS=HIGH
WARNING=3
REPORT=ON
ICON=CAP3.ICO
COPYRIGHT=©ALEJANDRO ALVARADO AZPEITIA (FI-UNAM)
OLDKEYPHRASE=ON

[FILES]
CAP3.RTF

;Creación de botones "anterior", "siguiente" y "glosario"
[CONFIG]
BrowseButtons()
CreateButton("btn_gls","&Glosario","JumpId('glosario.hlp>gls','GLOSARIO)")

; Mapeo de la cadena de contexto del tópico donde iniciará la navegación por temas
[MAP]
ARQUI_INT_2 30

;Asignación de las propiedades de las ventanas principal y secundarias del capítulo 3
[WINDOWS]
main=,1,(192,255,192),(0,128,0)
gls="GLOSARIO DE REDES DE COMPUTADORAS",(0,0,650,750),0,(255,255,255),(0,0,0),0
ind="CAPITULO 3:BUSQUEDA POR INDICE",(0,0,800,1000),1,(255,255,255),(0,128,0),0
evl="CAPITULO 3:EVALUACION [Repaso]",(0,0,800,1000),1,(255,255,255),(0,128,0),0
sub="CAPITULO 3:ARQUITECTURAS DE REDES [Subtemas]",(0,0,800,1000),1,(128,255,192),(0,128,0),0
    
```

Figura 5-8 Archivo proyecto del capítulo 3 de REDTUT.

En la sección [OPTIONS] existen varias opciones de control, pero las más importantes se muestran en la tabla 5-5.

Opción	Descripción	Sintaxis								
ROOT	Designa el directorio usado para almacenar tópicos y archivos gráficos listados en el archivo proyecto	ROOT = nom_ruta [, nom_ruta]								
BMROOT	Designa el directorio usado para almacenar mapas de bits localizados por referencia.(bmc, bml, etc.)	BMROOT = nom_ruta [,nom_ruta]								
COMPRESS	Especifica el tipo de compresión que se usará durante la compilación.	COMPRESS = nivel de compresión <table border="0"> <tr> <td>Nivel</td> <td></td> </tr> <tr> <td>ninguno</td> <td>0%</td> </tr> <tr> <td>medio</td> <td>40%</td> </tr> <tr> <td>alto</td> <td>50%</td> </tr> </table>	Nivel		ninguno	0%	medio	40%	alto	50%
Nivel										
ninguno	0%									
medio	40%									
alto	50%									
CONTENTS	Especifica la cadena de contexto del tópico de contenido.	CONTENTS = cadena de contexto								
TITLE	Determina el título del archivo de ayuda.	TITLE = nom_titulo								
ICON	Especifica el ícono usado cuando el archivo de ayuda es minimizado.	ICON = arch_icon.ico								
WARNING	Determina el nivel de mensajes de error que el compilador desplegará.	WARNING = nivel <table border="0"> <tr> <td>Nivel</td> <td></td> </tr> <tr> <td>1</td> <td>Solo errores más severos</td> </tr> <tr> <td>2</td> <td>No. Intermedio de errores.</td> </tr> <tr> <td>3</td> <td>Todos los errores y warnings.</td> </tr> </table>	Nivel		1	Solo errores más severos	2	No. Intermedio de errores.	3	Todos los errores y warnings.
Nivel										
1	Solo errores más severos									
2	No. Intermedio de errores.									
3	Todos los errores y warnings.									
ERRORLOG	Copia los mensajes de error a un archivo de texto.	ERRORLOG = arch_error								

Tabla 5-5 Algunas opciones que controlan la construcción del archivo de ayuda.

En la sección [FILES] son listados los archivos .RTF que van a ser compilados. Es posible el uso de nombres de rutas absolutas.

En la sección [BITMAPS] deben ir listados los archivos de mapas de bits, sin embargo, no es necesario usarla si en la sección [FILES] se utiliza la opción BMROOT.

En la sección [MAP] la sintaxis es la siguiente:

*cadena\_contexto = número\_contexto*

En la sección [ALIAS] la sintaxis es la siguiente:

*cadena\_contexto = alias*

El alias puede contener letras (A-Z), números (0-9), puntos y guiones inferiores ( \_ ).

En la sección [BUILDTAGS] se pueden listar hasta 30 indicadores de etiquetas.

En la sección [WINDOWS] la sintaxis es la siguiente:

Window\_name = "caption", (horiz\_pos, ver1\_pos, ancho, alto), estado, (RGB\_cliente), (no\_scroll(ng\_RGB), ontop\_state

La ventana principal se denomina con el *Window\_name = "main"*..

En la sección [CONFIG] se hace una lista de las macros que se van a utilizar en la ejecución de REDTUT.

En la sección [BAGGAGE] se hace una lista de mapas de bits y elementos de multimedia (como audio y animación).

Después de tener los archivos necesarios para la compilación en el mismo directorio, se tiene que teclear desde la línea de comandos de MSDOS el siguiente comando:

```
C:\>HCP archivo(hpj)
```

La extensión no es necesaria

Posteriormente se despliega una serie de mensajes que nos indican si la compilación fue exitosa o no. En caso de que no haya habido errores se genera un archivo con el mismo nombre del archivo proyecto, pero con extensión .HLP.

En la tabla 5-6 se muestran algunas consideraciones en el caso de que el archivo a compilar sea demasiado largo.

Consideraciones	
Memoria	Velocidad
<ul style="list-style-type: none"> <li>♦ Utilizar una máquina con 4 MB en RAM mínimo.</li> <li>♦ Utilizar la versión del compilador HCP.EXE, dado que hace mejor uso de memoria.</li> <li>♦ Cerrar todas las aplicaciones que no se estén usando antes de la compilación (en caso de estar en Windows).</li> <li>♦ Hacer las inserciones de mapas de bits por referencia.</li> <li>♦ Se debe reservar un espacio en disco duro, aproximado a cinco veces el tamaño del archivo de tópicos (.RTF).</li> </ul>	<ul style="list-style-type: none"> <li>♦ Utilizar la opción de compresión sólo cuando se quiera obtener la versión definitiva.</li> <li>♦ Utilizar una computadora rápida o el sistema operativo OS/2.</li> <li>♦ Dividir el contenido del sistema en archivos de tópicos (.RTF) separados.</li> </ul>

Tabla 5-6 Consideraciones para evitar problemas de memoria y velocidad en la compilación.

En el caso de que se presenten errores en el proceso de compilación, el compilador nos muestra mensajes de errores con números y categoría (Tabla 5-7).

Número del mensaje	Categoría de error
1000 - 1999	Preprocesamiento
2000 - 2999	Archivo proyecto
3000 - 3499	Indicadores de etiquetas
3500 - 3999	Macros
4000 - 4199	Cadenas de contexto
4200 - 4599	Códigos de control
4600 - 4999	Tópicos de archivo
5000 ó mayor	Misceláneos

Tabla 5-7 Categoría y número de los mensajes de error.

### 5.2.2 Generación del archivo ejecutable (.EXE)

El archivo ejecutable .EXE es el programa que nos va a permitir iniciar REDTUT, proporcionándonos la posibilidad de elegir cualquier capítulo (archivos .HLP) mediante la función WINHELP proporcionada por Windows (Apéndice C).

Este programa fue creado con un lenguaje de programación tradicional (lenguaje "C") y con el uso de funciones y librerías de Windows. El compilador empleado fue el Borland C++ para Windows, que al igual

que cualquier aplicación de Windows, cuenta con la ventajas y propiedades que proporciona el ambiente (Fig. 5-9)



Figura 5-9 Compilador Borland C++ para Windows ver. 3.1

Para el proceso de compilación se requiere de un archivo proyecto (.PRJ) que contenga las referencias al archivo fuente (.C), al archivo de definición (.DEF) y al archivo de recursos (.RC, si lo hay).

El archivo fuente debe contener el código del programa (Fig 5-10).

```
#include<windows.h>
/** Constantes que identifican los botones de la pantalla **/
#define IDB_C1 100
#define IDB_C2 101
#define IDB_C3 102
#define IDB_C4 103
#define IDB_C11 104
#define IDB_C21 105
#define IDB_C31 106
#define IDB_C41 107
#define IDB_C1B 108
#define IDB_C2B 109
#define IDB_C3B 110
#define IDB_C4B 111
#define IDB_AYUDA 112
#define IDB_GLOS 113
#define IDB_SALIR 200
/** Códigos virtuales de teclas no defintias en windows.h **/
#define VK_1 0x31
#define VK_2 0x32
#define VK_3 0x33
#define VK_4 0x34
#define VK_I 0x49
#define VK_N 0x4E
#define VK_D 0x44
#define VK_C 0x43
#define VK_B 0x42
#define VK_U 0x55
#define VK_A 0x41
#define VK_R 0x52
#define VK_G 0x47
#define VK_Y 0x59
#define VK_S 0x53
/***** Declaración de variables globales *****/
char szProgName[]="REDTUT", Icon_redtut[]="redtut";
static int ancho_pant, alto_pant;
HWND hWIND; hPantalla;
HDC hDC; hPantIDC;
HBITMAP hBil_prev;

#pragma argused
long FAR PASCAL WndProc(HWND,UINT,WPARAM,LPARAM);
/***** Inicio de la función principal *****/
```

## CAPITULO 5: INTEGRACION DE REDTUT

```

int PASCAL WinMain(HINSTANCE hInst, HINSTANCE hPrevInst, LPSTR lpszCmdLine, int nCmdShow)
{
    HWND hWnd;
    WNDCLASS wcApp;
    MSG mensaje;
    if (hPrevInst)
    {
        wcApp.lpszClassName = szProgName;
        wcApp.hInstance = hInst;
        wcApp.lpfnWndProc = WndProc;
        wcApp.hCursor = LoadCursor(NULL, IDC_ARROW);
        wcApp.hIcon = LoadIcon(hInst, icon_redtut);
        wcApp.lpszMenuName = NULL;
        wcApp.hbrBackground = GetStockObject(BLACK_BRUSH);
        wcApp.style = CS_HREDRAW|CS_VREDRAW;
        wcApp.cbClsExtra = 0;
        wcApp.cbWndExtra = 0;
        if (!RegisterClass(&wcApp))
            return FALSE;
    }
    /** Determina las dimensiones de la pantalla */
    hPantalla=GetDesktopWindow();
    hPantDC=GetDC(hPantalla);
    ancho_pant=GetDeviceCaps(hPantDC,HORZRES);
    alto_pant=GetDeviceCaps(hPantDC,VERTRES);
    ReleaseDC(hPantalla,hPantDC);

    /** Crea la ventana principal */

    hWnd=CreateWindow(szProgName,"REDTUT",WS_BORDER,0,0,ancho_pant,alto_pant,(HWND)NULL,(HMENU)NULL,(HANDLE)hInst,(LPSTR)NULL);
    ShowWindow(hWnd,nCmdShow);
    UpdateWindow(hWnd);

    while (GetMessage(&mensaje,NULL,NULL,NULL))
    {
        TranslateMessage(&mensaje);
        DispatchMessage(&mensaje);
    }
    return(mensaje.wParam);
}
long FAR PASCAL WndProc(HWND hWnd,UINT msg,WPARAM wParam,LPARAM lParam)
{
    HWND          hbCap1,hbCap2,hbCap3,hbCap4,hbCap11,hbCap21,hbCap31,hbCap41,hbCap1B,hbCap2B,hbCap3B,hbCap4B,hbAyuda,hbGlos,hbSalir;
    HDC           hDC,hmetaMarco,hmetaBil,hmemDC;
    static HANDLE hMarco;
    PAINTSTRUCT  ps;
    HMENU         hpan_Marco;
    HBITMAP       hbm_Fondo,hbm_REDUT,hbm_redtut,hbm_Cap1,hbm_Cap2,hbm_Cap3,hbm_Cap4,hbm_Titulo;
    HRGN          hmetaRgn;
    int           x=0,y=0;

    switch (msg)
    {
        case WM_CREATE:
            /** Determina las dimensiones de la pantalla */
            hPantalla=GetDesktopWindow();
            hPantDC=GetDC(hPantalla);
            ancho_pant=GetDeviceCaps(hPantDC,HORZRES);
            alto_pant=GetDeviceCaps(hPantDC,VERTRES);
            ReleaseDC(hPantalla,hPantDC);

            /** Creación de la caja tipográfica de la pantalla como metafile */
            hbm_REDUT=LoadBmpMap(GetWindowWord(hWnd,GWW_HINSTANCE),"REDTUT");
            hbm_redtut=LoadBmpMap(GetWindowWord(hWnd,GWW_HINSTANCE),"redtut2");
            hbm_Titulo=LoadBmpMap(GetWindowWord(hWnd,GWW_HINSTANCE),"TITULO");
            hmetaRgn=CreateRectRgn(0,60,ancho_pant,alto_pant-60);
            hpan_Marco=CreatePen(PS_SOLID,2,RGB(0,0,0));
            hmetaMarco=CreateMetaFile(NULL);
            if (hmetaMarco!=NULL)
            {
                hmetaBil=GetDC(hWnd);
                hmemDC=CreateCompatibleDC(hmetaBil);
                SelectObject(hmemDC,hbm_redtut);
                BitBlt(hmetaMarco,(ancho_pant/2)-230,6,35,35,
                    hmemDC,0,0,SRCCOPY);
                BitBlt(hmetaMarco,(ancho_pant/2)+185,6,35,35,
                    hmemDC,0,0,SRCCOPY);
                SelectObject(hmemDC,hbm_Titulo);
                BitBlt(hmetaMarco,(ancho_pant/2)-181,2,362,46,
                    hmemDC,0,0,SRCCOPY);
                SelectObject(hmemDC,hbm_REDUT);
                SelectClipRgn(hmetaMarco,hmetaRgn);
                for (x=0;x<=ancho_pant;x+=100)
                {
                    for (y=0,y<=alto_pant,y+=100)
                        BitBlt(hmetaMarco,x,y,100,100,hmemDC,0,0,SRCCOPY);
                }
                hMarco=CloseMetaFile(hmetaMarco);
                DeleteDC(hmemDC);
                DeleteObject(hbm_REDUT);
                DeleteObject(hbm_redtut);
                DeleteObject(hbm_Titulo);
                DeleteObject(hpan_Marco);
                ReleaseDC(hWnd,hmetaBil);
            }

            /** Creación de botones */

```

```

/* Botones para desplegar la navegaci3n por tomas */
hbCap1 = CreateWindow ("BUTTON", "&1. Comunicaci3n de datos", WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON, (ancho_pant5)-50, (alto_pant5)-30, 200, 25, hWnd, IDB_C1, GetWindowWord(hWnd, GWW_HINSTANCE), NULL);
ShowWindow (hbCap1, SW_SHOW);

hbCap2 = CreateWindow ("BUTTON", "&2. Redes de comunicaci3n", WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON, ancho_pant-2*(ancho_pant5), (alto_pant5)-30, 200, 25, hWnd, IDB_C2, GetWindowWord(hWnd, GWW_HINSTANCE), NULL);
ShowWindow (hbCap2, SW_SHOW);
hbCap3 = CreateWindow ("BUTTON", "&3. Arquitecturas de redes", WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON, (ancho_pant5)-50, alto_pant-(2*(alto_pant5)+30), 200, 25, hWnd, IDB_C3, GetWindowWord(hWnd, GWW_HINSTANCE), NULL);
ShowWindow (hbCap3, SW_SHOW);
hbCap4 = CreateWindow ("BUTTON", "&4. Dise1o de redes", WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON, ancho_pant-2*(ancho_pant5), alto_pant-(2*(alto_pant5)+30), 200, 25, hWnd, IDB_C4, GetWindowWord(hWnd, GWW_HINSTANCE), NULL);
ShowWindow (hbCap4, SW_SHOW);

/** Botones para desplegar navegaci3n por 3ndice */
hbCap11 = CreateWindow ("BUTTON", "&Indice", WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON, (ancho_pant5)+80, (alto_pant5)+10, 70, 25, hWnd, IDB_C11, GetWindowWord(hWnd, GWW_HINSTANCE), NULL);
ShowWindow (hbCap11, SW_SHOW);
hbCap21 = CreateWindow ("BUTTON", "&Indice", WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON, ancho_pant-(2*(ancho_pant5)-130), (alto_pant5)+10, 70, 25, hWnd, IDB_C21, GetWindowWord(hWnd, GWW_HINSTANCE), NULL);
ShowWindow (hbCap21, SW_SHOW);
hbCap31 = CreateWindow ("BUTTON", "&Indice", WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON, (ancho_pant5)+80, ancho_pant-(2*(alto_pant5)-10), 70, 25, hWnd, IDB_C31, GetWindowWord(hWnd, GWW_HINSTANCE), NULL);
ShowWindow (hbCap31, SW_SHOW);
hbCap41 = CreateWindow ("BUTTON", "&Indice", WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON, ancho_pant-(2*(ancho_pant5)-130), alto_pant-(2*(alto_pant5)-10), 70, 25, hWnd, IDB_C41, GetWindowWord(hWnd, GWW_HINSTANCE), NULL);
ShowWindow (hbCap41, SW_SHOW);

/** Botones para desplegar navegaci3n por b3squeda */
hbCap1B = CreateWindow ("BUTTON", "&Buscar", WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON, (ancho_pant5)+80, (alto_pant5)+45, 70, 25, hWnd, IDB_C1B, GetWindowWord(hWnd, GWW_HINSTANCE), NULL);
ShowWindow (hbCap1B, SW_SHOW);
hbCap2B = CreateWindow ("BUTTON", "&Buscar", WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON, ancho_pant-(2*(ancho_pant5)-130), (alto_pant5)+45, 70, 25, hWnd, IDB_C2B, GetWindowWord(hWnd, GWW_HINSTANCE), NULL);
ShowWindow (hbCap2B, SW_SHOW);
hbCap3B = CreateWindow ("BUTTON", "&Buscar", WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON, (ancho_pant5)+80, ancho_pant-(2*(alto_pant5)-45), 70, 25, hWnd, IDB_C3B, GetWindowWord(hWnd, GWW_HINSTANCE), NULL);
ShowWindow (hbCap3B, SW_SHOW);
hbCap4B = CreateWindow ("BUTTON", "&Buscar", WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON, ancho_pant-(2*(ancho_pant5)-130), alto_pant-(2*(alto_pant5)-45), 70, 25, hWnd, IDB_C4B, GetWindowWord(hWnd, GWW_HINSTANCE), NULL);
ShowWindow (hbCap4B, SW_SHOW);

/** Botones para desplegar ayuda y salir de REDTUT */
hbGlos = CreateWindow ("BUTTON", "&Glosario", WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON, ancho_pant-(2*(ancho_pant5)+120), alto_pant-60, 80, 25, hWnd, IDB_GLOS, GetWindowWord(hWnd, GWW_HINSTANCE), NULL);
ShowWindow (hbGlos, SW_SHOW);
hbAyuda = CreateWindow ("BUTTON", "&Ayuda", WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON, ancho_pant-(2*(ancho_pant5)), alto_pant-60, 80, 25, hWnd, IDB_AYUDA, GetWindowWord(hWnd, GWW_HINSTANCE), NULL);
ShowWindow (hbAyuda, SW_SHOW);
hbSalir = CreateWindow ("BUTTON", "&Salir", WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON, ancho_pant-(2*(ancho_pant5)-120), alto_pant-60, 80, 25, hWnd, IDB_SALIR, GetWindowWord(hWnd, GWW_HINSTANCE), NULL);
ShowWindow (hbSalir, SW_SHOW);
break;

case WM_COMMAND:
/** Ejecuta la funci3n WINHELP seg3n el bot3n presionado switch(wParam)
{
    case IDB_C1:
        WinHelp(hWnd, "cap1.hlp", HELP_CONTEXT, 10);
        break;
    case IDB_C2:
        WinHelp(hWnd, "cap2.hlp", HELP_CONTEXT, 20);
        break;
    case IDB_C3:
        WinHelp(hWnd, "cap3.hlp", HELP_CONTEXT, 30);
        break;
    case IDB_C4:
        WinHelp(hWnd, "cap4.hlp", HELP_CONTEXT, 40);
        break;

/** Navegaci3n por 3ndice
case IDB_C11:
    WinHelp(hWnd, "cap1.hlp", HELP_CONTENTS, NULL);
    break;
case IDB_C21:
    WinHelp(hWnd, "cap2.hlp", HELP_CONTENTS, NULL);
    break;
case IDB_C31:
    WinHelp(hWnd, "cap3.hlp", HELP_CONTENTS, NULL);
    break;
case IDB_C41:
    WinHelp(hWnd, "cap4.hlp", HELP_CONTENTS, NULL);
    break;

/** Navegaci3n por b3squeda
case IDB_C1B:
    WinHelp(hWnd, "cap1.hlp", HELP_PARTIALKEY, NULL);
    break;
case IDB_C2B:
    WinHelp(hWnd, "cap2.hlp", HELP_PARTIALKEY, NULL);
    break;
case IDB_C3B:
    WinHelp(hWnd, "cap3.hlp", HELP_PARTIALKEY, NULL);
    break;
case IDB_C4B:
    WinHelp(hWnd, "cap4.hlp", HELP_PARTIALKEY, NULL);
    break;
}

```

```

        break;
/** Ayuda en línea y salida de REDTUT ***
        case IDB_AYUDA:
            WinHelp(hWnd,"cap1.hlp",HELP_HELPONHELP,NULL);
            break;
        case IDB_GLOS:
            WinHelp(hWnd,"glosario.hlp",HELP_CONTENTS,NULL);
            break;
        case IDB_SALIR:
            DestroyWindow(hWnd);
            break;
    }
    break;
/** Activación de botones por teclado ***
    case WM_KEYDOWN:
        {
            switch(wParam)
            {
/** Navegación por temas ***
                case VK_1:
                    WinHelp(hWnd,"cap1.hlp",HELP_CONTEXT,10);
                    break;
                case VK_2:
                    WinHelp(hWnd,"cap2.hlp",HELP_CONTEXT,20);
                    break;
                case VK_3:
                    WinHelp(hWnd,"cap3.hlp",HELP_CONTEXT,30);
                    break;
                case VK_4:
                    WinHelp(hWnd,"cap4.hlp",HELP_CONTEXT,40);
                    break;
                /*VK_NUMPAD1,VK_NUMPAD2,VK_NUMPAD3,VK_NUMPAD4 en caso de teclado numérico*/
                case VK_NUMPAD1:
                    WinHelp(hWnd,"cap1.hlp",HELP_CONTEXT,10);
                    break;
                case VK_NUMPAD2:
                    WinHelp(hWnd,"cap2.hlp",HELP_CONTEXT,20);
                    break;
                case VK_NUMPAD3:
                    WinHelp(hWnd,"cap3.hlp",HELP_CONTEXT,30);
                    break;
                case VK_NUMPAD4:
                    WinHelp(hWnd,"cap4.hlp",HELP_CONTEXT,40);
                    break;
/** Navegación por Índice ***
                case VK_I:
                    WinHelp(hWnd,"cap1.hlp",HELP_CONTENTS,NULL);
                    break;
                case VK_N:
                    WinHelp(hWnd,"cap2.hlp",HELP_CONTENTS,NULL);
                    break;
                case VK_D:
                    WinHelp(hWnd,"cap3.hlp",HELP_CONTENTS,NULL);
                    break;
                case VK_C:
                    WinHelp(hWnd,"cap4.hlp",HELP_CONTENTS,NULL);
                    break;
/** Navegación por búsqueda ***
                case VK_B:
                    WinHelp(hWnd,"cap1.hlp",HELP_PARTIALKEY,NULL);
                    break;
                case VK_U:
                    WinHelp(hWnd,"cap2.hlp",HELP_PARTIALKEY,NULL);
                    break;
                case VK_A:
                    WinHelp(hWnd,"cap3.hlp",HELP_PARTIALKEY,NULL);
                    break;
                case VK_R:
                    WinHelp(hWnd,"cap4.hlp",HELP_PARTIALKEY,NULL);
                    break;
/** Ayuda en línea y salida de REDTUT ***
                case VK_Y:
                    WinHelp(hWnd,"cap1.hlp",HELP_HELPONHELP,NULL);
                    break;
                case VK_G:
                    WinHelp(hWnd,"glosario.hlp",HELP_CONTENTS,NULL);
                    break;
                case VK_S:
                    DestroyWindow(hWnd);
                    break;
            }
        }
    }
    break;
    case WM_PAINT:
        hdc= BeginPaint(hWnd,&ps);
/** Muestra el marco en la pantalla ***
        PlayMetaFile(hdc,hMarco);
/** Muestra íconos de REDTUT ***
        hmemDC=CreateCompatibleDC(hdc);
        hbm_Cap1=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"CAP1");
        hbm_Cap2=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"CAP2");
        hbm_Cap3=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"CAP3");
        hbm_Cap4=LoadBitmap(GetWindowWord(hWnd,GWW_HINSTANCE),"CAP4");
        hBR_prev=SelectObject(hmemDC,hbm_Cap1);

```

```

        BitBlt(hdc,(ancho_pant/5)-50,alto_pant/5,
        100,100,hmemDC,0,0,SRCCOPY);
        SelectObject(hmemDC,hbm_Cap2);
        BitBlt(hdc,ancho_pant-(2*(ancho_pant/5)),alto_pant/5,
        100,100,hmemDC,0,0,SRCCOPY);
        SelectObject(hmemDC,hbm_Cap3);
        BitBlt(hdc,(ancho_pant/5)-50,alto_pant-2*(alto_pant/5),
        100,100,hmemDC,0,0,SRCCOPY);
        SelectObject(hmemDC,hbm_Cap4);
        BitBlt(hdc,ancho_pant-(2*(ancho_pant/5)),
        alto_pant-2*(alto_pant/5),100,100,hmemDC,0,0,SRCCOPY);
        SelectObject(hmemDC,hBit_prev);
        DeleteDC(hmemDC);
        DeleteObject(hbm_Cap1);
        DeleteObject(hbm_Cap2);
        DeleteObject(hbm_Cap3);
        DeleteObject(hbm_Cap4);
        ReleaseDC(hwnd,hDC);
        EndPaint(hwnd,&ps);
        break;
    case WM_DESTROY:
        DeleteMetaFile(hMarco);
    /* Se cierran los capítulos de REDTUT */
        WinHelp(hwnd,"cap1.hlp",HELP_QUIT,NULL);
        WinHelp(hwnd,"cap2.hlp",HELP_QUIT,NULL);
        WinHelp(hwnd,"cap3.hlp",HELP_QUIT,NULL);
        WinHelp(hwnd,"cap4.hlp",HELP_QUIT,NULL);
        PostQuitMessage(0);
        break;
    default:
        return(DefWindowProc(hwnd,msg,wParam,lParam));
    }
    return(0L);
}

```

Figura 5-10 Archivo fuente (.C).

El archivo de definición de módulos (.DEF) contiene información necesaria para enlazar la aplicación (Fig. 5-11).

NAME	REDTUT
DESCRIPTION	'Tutorial de redes de computadoras'
EXETYPE	WINDOWS
CODE	PRELOAD MOVABLE DISCARDABLE
DATA	PRELOAD MOVEABLE MULTIPLE
HEAPSIZE	4096
STACKSIZE	9216

Figura 5-11 Archivo de definición de módulos (.DEF)

El archivo de recursos contiene un guión de todos los recursos de la aplicación como: archivos cabecera (.H), de iconos (.ICO), de cursores (.CUR), de matrices de puntos (.BMP), punteros (.PTR), etc. los cuales deberán incorporarse a la versión ejecutable (.EXE) del programa (Fig. 5-12).

redtut	ICON	redtut.ico
REDTUT	BITMAP	redtut2.bmp
redtut2	BITMAP	redtut.bmp
CAP1	BITMAP	cap1r.bmp
CAP2	BITMAP	cap2r.bmp
CAP3	BITMAP	cap3r.bmp
CAP4	BITMAP	cap4r.bmp
TITULO	BITMAP	titulo.bmp

Figura 5-12 Archivo de recursos (.RC).

Una vez creado el archivo proyecto (.PRJ), se puede realizar la compilación y enlazado de la nueva aplicación de Windows, la cual se logrará hasta que no existan errores en cualquiera de los archivos antes mencionados.

**P A R T E   I V**  
**R E S U L T A D O S**

## **CAPITULO 6**

# **EVALUACION DE FUNCIONALIDAD Y MECANICA DE ACTUALIZACION DE REDTUT**

**6.1 Evaluación de funcionalidad de REDTUT.**

**6.2 Mecánica de actualización de REDTUT.**

## 6.1 EVALUACION DE REDTUT.

En cualquier tipo de sistema, después de haber concluido las etapas de definición, análisis y desarrollo, es necesario presentar los resultados obtenidos y hacer una evaluación sobre ellos. La evaluación de REDTUT se hace mediante los resultados obtenidos en las diferentes formas de acceder la información a los 4 capítulos que contiene el sistema y sobre la función que desempeñan los elementos que integran su ambiente (barra de botones, barra de menús, animación, etc.).

### 6.1.1 Navegación por temas.

Accesar REDTUT por medio de la navegación por temas, es la forma propuesta para obtener un aprendizaje gradual sobre el curso de "Redes de computadoras". Para su evaluación es necesario presentar los elementos que caracterizan esta forma de navegación.

- ♦ *Pantalla principal* La pantalla principal ofrece la posibilidad de elegir el capítulo de interés (Fig. 6-1).
- ♦ *Icono "temas"* Permite elegir cualquier tema relacionado con el capítulo elegido previamente (Fig. 6-2).
- ♦ *Icono "salir"* Cierra el archivo que contiene el capítulo elegido (Fig. 6-2).
- ♦ *Icono "cerrar"* Cierra la ventana secundaria de subtemas para continuar con la navegación (Fig. 6-2).
- ♦ *Icono "evalúa"* Salta al tópico de evaluación correspondiente al tema elegido (Fig. 6-2).
- ♦ *Icono "animación"* Ejecuta un programa de animación relacionado con el tema actual (Fig. 6-2).
- ♦ *Botón "siguiente"* Permite el acceso al tema siguiente en el proceso de aprendizaje gradual (Fig. 6-2).
- ♦ *Botón "anterior"* Permite el acceso al tema anterior en el proceso de aprendizaje gradual (Fig. 6-2).

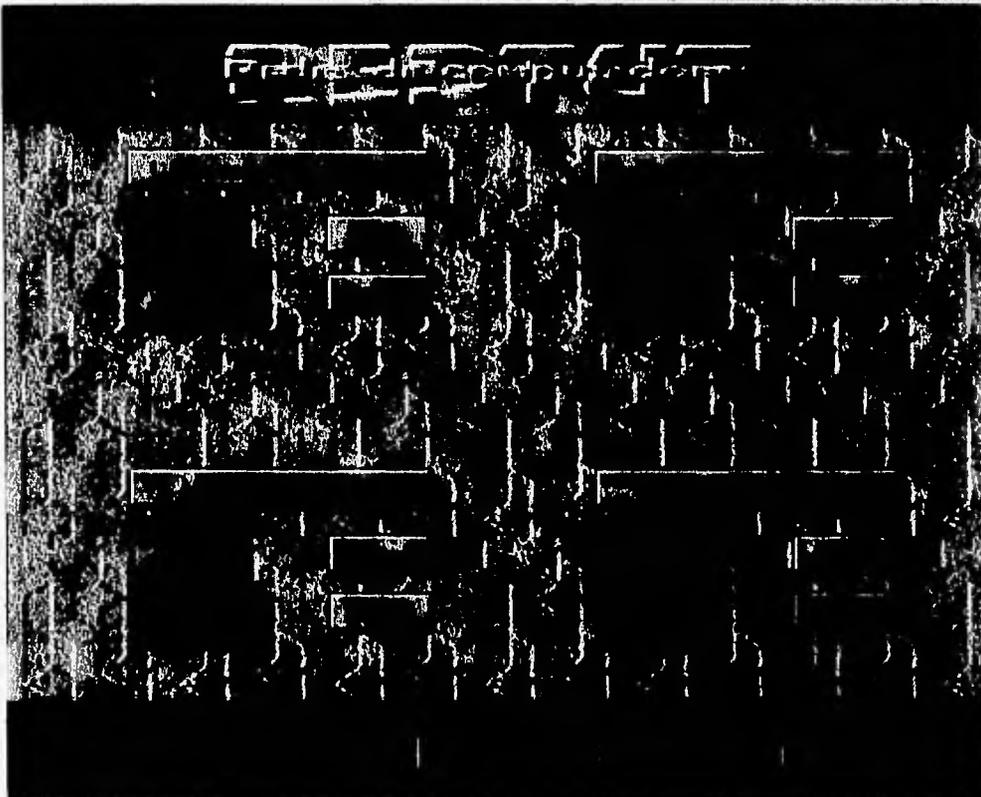


Fig. 6-1 Pantalla principal de REDTUT.

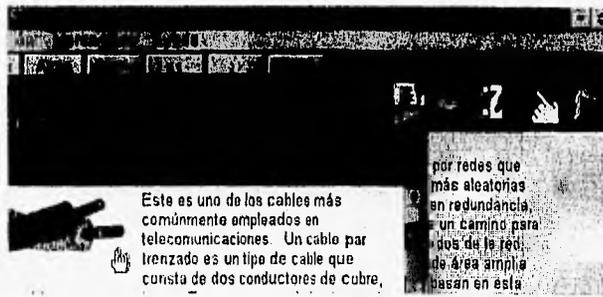


Fig. 6-2 Iconos que permiten la navegación por temas en REDTUT.

Las ventajas y desventajas de acceder la información a través de la navegación por tema se muestran en la tabla 6-1.

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>◆ Es posible acceder cualquier capítulo de interés sin necesidad de tener que empezar con el capítulo 1.</li> <li>◆ El tener el tópico de Introducción como el tópico de inicio, permite tomar la decisión de continuar con el capítulo seleccionado o elegir otro.</li> <li>◆ La secuencia que tienen los tópicos de temas, ayudan al aprendizaje gradual.</li> <li>◆ Permite el despliegue de subtemas en ventanas secundarias, sin perder la secuencia del aprendizaje.</li> <li>◆ Permite ejecutar los programas de animación dentro del tema que lo requiera.</li> <li>◆ Es la única forma posible acceder el tópico de evaluación que despliega en ventanas secundarias los temas que no han sido asimilados, sin perder la secuencia del aprendizaje.</li> <li>◆ La distribución de la pantalla está diseñada para no desplegar barras de desplazamiento verticales y/o horizontales.</li> <li>◆ Permite el despliegue del glosario en cualquier momento.</li> </ul>	<ul style="list-style-type: none"> <li>◆ No es posible imprimir las ventanas secundarias que despliegan los subtemas.</li> <li>◆ No es posible imprimir las ventanas instantáneas.</li> <li>◆ Existe la posibilidad de romper con la navegación por temas y cambiar a la navegación por búsqueda con el botón "buscar".</li> </ul>

Tabla 6-1 Ventajas y desventajas del acceso a REDTUT por medio de la navegación por temas.

### 6.1.2 Navegación por índice.

El acceso a cualquier tema por medio de la navegación por índice, es realizado al saltar al tópico de contenido del capítulo de interés. El tópico de contenido, tiene la característica de ser el primer tópico del archivo (Fig. 6-3), HLP que contiene el capítulo seleccionado. Por lo tanto, puede ser accedido de las siguientes formas:

1. Por medio del botón Índice del capítulo que esté seleccionado (Fig. 6-3).
2. Por medio de la pantalla principal de REDTUT, que proporciona el botón "Índice" que salta al tópico de contenido del capítulo seleccionado (Fig. 6-1).

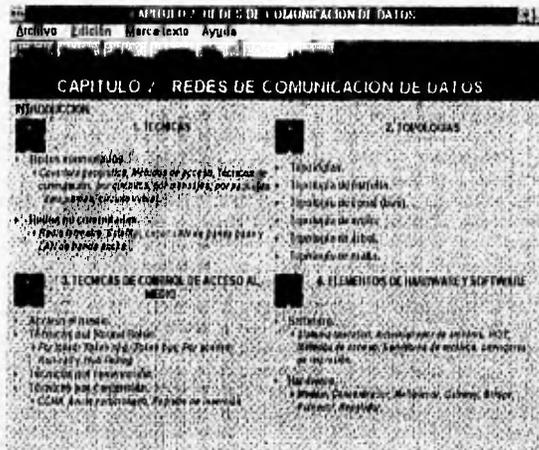


Figura 6-3 Topico de indice de REDTUT.

Las ventajas y desventajas de acceder información por medio de la navegación por Índice se muestran en la tabla 6-2.

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>◆ Permite tener una visión más amplia del contenido de un capítulo.</li> <li>◆ Seleccionar los temas que tienen relación con un mismo capítulo.</li> <li>◆ Es posible pasar de un tema a otro sin necesidad de llevar una secuencia gradual de aprendizaje (para usuarios que sólo buscan un tema de interés).</li> <li>◆ No existe la posibilidad de saltar a los tópicos diseñados para la navegación por temas (con iconos: "tema", "evalua", botones: "siguiente", "anterior", etc.), dado que todos los temas son desplegados en ventanas secundarias.</li> <li>◆ Es posible saltar de una ventana secundaria que contiene un tema a otra ventana secundaria con otro tema, sin moverse del tópico de contenido (el índice).</li> <li>◆ Permite el despliegue del glosario en cualquier momento.</li> <li>◆ La distribución de la pantalla es diseñada para no desplegar barras de desplazamiento verticales y/o horizontales.</li> </ul>	<ul style="list-style-type: none"> <li>◆ No se pueden imprimir los temas.</li> <li>◆ El índice sólo contiene los temas más relevantes de un capítulo.</li> <li>◆ Existen temas en ventanas secundarias que sólo pueden ser cerrados por medio del cuadro de control de la ventana.</li> </ul>

Tabla 6-2. Ventajas y desventajas del acceso a REDTUT por medio de la navegación por Índice.

### 6.1.3 Navegación por búsqueda

El acceso a REDTUT por medio de la navegación por búsqueda, tiene como fin principal:

1. Localizar información relacionada con una palabra o un tema especificado por el usuario.
2. Permitir la impresión de las ventanas instantáneas y secundarias que se relacionan con los de temas de un capítulo.

Estos fines se cumplen de las siguientes formas:

1. A través de la pantalla principal, que despliega el cuadro de diálogo de búsqueda correspondiente al capítulo seleccionado, al oprimir el botón "buscar" (Fig. 6-4).
2. Al oprimir el botón buscar de la barra de botones del capítulo seleccionado (Fig. 6-5).

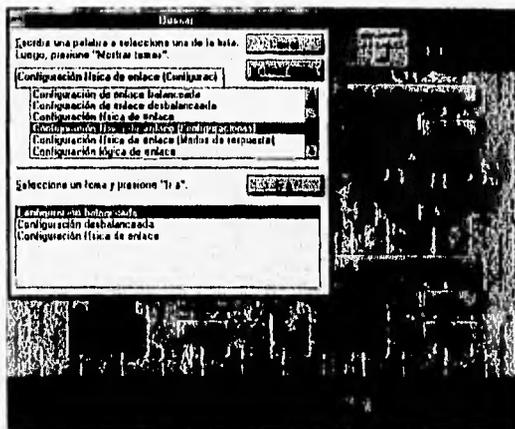


Figura 6-4 Navegación por búsqueda en REDTUT

Las ventajas y desventajas de la navegación por búsqueda se muestran en la tabla 6-3.

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>◆ Es posible localizar información relacionada con un tema de interés especificado por el usuario, con sólo teclear una palabra clave.</li> <li>◆ Muestra los temas que tienen relación con el tema especificado por el usuario.</li> <li>◆ Permite la impresión de ventanas instantáneas y secundarias relacionadas con los tópicos de temas.</li> <li>◆ Los temas de cada capítulo son listados en orden alfabético ascendente.</li> <li>◆ En caso de que la palabra clave especificada por el usuario no esté disponible en la lista de palabras claves, se muestra lo más cercano que se le parezca.</li> <li>◆ Permite el despliegue del glosario en cualquier momento.</li> <li>◆ Permite el acceso a bibliografía de apoyo correspondiente al capítulo seleccionado.</li> </ul>	<ul style="list-style-type: none"> <li>◆ Los tópicos diseñados para ventanas secundarias pueden presentar barras horizontales y/o verticales.</li> <li>◆ El icono "cerrar" diseñado para ventanas secundarias, no tiene ningún efecto al ser desplegado en la ventana principal.</li> <li>◆ Existe la posibilidad de acceder tópicos de tema final, los cuales pueden cambiar la navegación a la navegación por temas.</li> </ul>

Tabla 6-3. Ventajas y desventajas del acceso a REDTUT por medio de la navegación por búsqueda.

#### 6.1.4 Barra de menú en REDTUT.

La barra de menú que presenta un capítulo de REDTUT, es exactamente la misma que proporciona cualquier ayuda de alguna aplicación de Windows (Fig. 6-5). Sin embargo, los elementos de menú pueden realizar sus funciones adecuadamente adaptadas a los objetivos de REDTUT.



Fig. 6-5 Barra de menú y barra de botones de REDTUT.

Las ventajas y desventajas de la barra de menú en REDTUT se muestran en la tabla 6-4.

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>♦ En el menú "Archivo" es posible especificar la impresora, imprimir el tópico desplegado en pantalla y salir del capítulo seleccionado.</li> <li>♦ En el menú "Edición/Copiar..." es posible copiar al portapapeles de Windows el texto del tópico desplegado en pantalla, para que pueda ser editado en cualquier procesador de texto para Windows (Write, Word, etc.).</li> <li>♦ En el menú "Edición/Anotar..." es posible hacer anotaciones (textos de apoyo) que sirvan para mayor comprensión del tema en estudio.</li> <li>♦ En el menú "Marca-texto/Definir..." se puede hacer una lista de los temas más relevantes de un capítulo, con el fin de ser accedidos rápidamente.</li> <li>♦ En el menú "Ayuda" es posible acceder la ayuda en línea de REDTUT.</li> </ul>	<ul style="list-style-type: none"> <li>♦ El menú "Archivo/Abrir...", no tiene alguna aplicación dentro de REDTUT.</li> <li>♦ El menú "Archivo/Salir" está habilitado en cualquier tópico de tema seleccionado.</li> <li>♦ El menú "Ayuda/Acerca de..." no muestra el icono correspondiente a REDTUT.</li> </ul>

Tabla 6-4 Ventajas y desventajas de la barra de menú en REDTUT.

### 6.1.5 Barra de botones en REDTUT.

La barra de botones de REDTUT es la misma que presentan las ayudas en línea de las aplicaciones en Windows, con la diferencia de que se le agregó el botón para desplegar el glosario de REDTUT (Fig. 6-5). Los botones que no tienen una función específica en REDTUT ("Atrás" e "Historial"), pueden ser utilizados convenientemente cuando sean necesarios.

Las ventajas y desventajas de la barra de botones en REDTUT se muestran en la tabla 6-5.

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>♦ Los botones: "Índice", "Buscar", "Siguiente (&gt;)" y "Anterior (&lt;)" son adecuadamente adaptados en el diseño de REDTUT.</li> <li>♦ Se agrega el botón "Glosario" que está habilitado en cualquier tópico de temas.</li> <li>♦ El botón "Atrás" permite acceder el tópico anterior inmediato desplegado, aunque el tópico actualmente desplegado no forme parte del diseño de una secuencia de páginas ("&gt;","&lt;").</li> <li>♦ El botón "Atrás" no rompe con el acceso a REDTUT por la navegación por temas.</li> <li>♦ El botón "Historial" despliega una ventana secundaria que permite acceder no sólo el último tópico desplegado, sino hasta alguno de los 50 tópicos anteriormente desplegados.</li> </ul>	<ul style="list-style-type: none"> <li>♦ Los botones: "Índice", "Historial" y "Buscar", pueden romper con la navegación por temas de REDTUT.</li> </ul>

Tabla 6-5 Ventajas y desventajas de la barra de botones en REDTUT.

### 6.1.6 Ayuda en línea.

La ayuda en línea tiene gran importancia en cualquier aplicación que se desarrolle dentro del ambiente Windows, dado que cada aplicación utiliza sus propios elementos (menús, botones, íconos, etc.) que le permiten desempeñar adecuadamente sus funciones (Fig. 6-6).

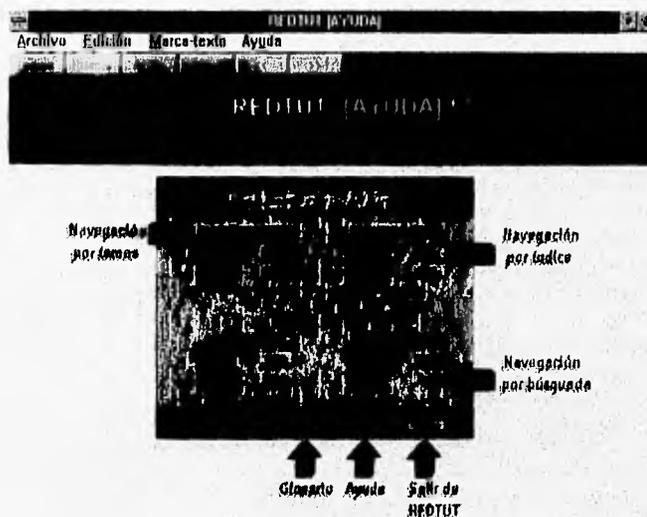


Fig. 6-6 Ayuda en línea de REDTUT desplegada al seleccionar el elemento de menú "Ayuda/Uso de la ayuda".

En el caso específico de REDTUT, la ayuda en línea es desplegada al seleccionar el elemento de menú "Ayuda/Uso de la ayuda", el cual ha sido adaptado para desplegar la ayuda correspondiente al uso de REDTUT, en vez de desplegar ayuda sobre el uso del sistema de ayuda en Windows (Fig.4-32). Además por medio del botón "Ayuda" en la pantalla principal de REDTUT (Fig. 6-1).

Las ventajas y desventajas de la ayuda en línea de REDTUT se muestran en la tabla 6-6.

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>◆ Proporciona la ayuda necesaria para navegar en REDTUT, de las tres formas posibles que se proponen.</li> <li>◆ Proporciona ayuda de como utilizar los elementos de menú o botones estándares del sistema de ayuda de Windows, adaptándolos al uso de REDTUT.</li> <li>◆ Es desplegado usando un elemento de menú estándar ("Ayuda/Uso de la ayuda"), evitando la creación de otro elemento de menú que podría ser inconveniente.</li> <li>◆ Es posible ser accesada desde la pantalla principal de REDTUT.</li> </ul>	<ul style="list-style-type: none"> <li>◆ No es posible cambiar el nombre del elemento de menú estándar utilizado para desplegar la ayuda de REDTUT.</li> </ul>

Tabla 6-6 Ventajas y desventajas de la ayuda en línea de REDTUT.

### 6.1.7 Glosario.

El glosario es una herramienta de gran utilidad en el proceso de aprendizaje. En REDTUT es proporcionado con la finalidad de acceder rápidamente el concepto de algún término relacionado con las redes de computadoras (Fig. 6.7). El glosario es un archivo .HLP independiente, por lo tanto puede ser desplegado en una ventana secundaria. El acceso al glosario puede ser:

- ◆ Al activar el botón "Glosario" de la barra de botones del capítulo seleccionado en cualquier momento de la navegación (Fig. 6-5).
- ◆ Al activar el botón glosario de la pantalla principal de REDTUT (Fig. 6-1)



Figura 6-7 Glosario de REDTUT.

Las ventajas y desventajas del uso del glosario dentro de REDTUT, se muestra en la tabla 6-7.

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>◆ Es posible acceder algún concepto relacionado con cualquiera de los capítulos de REDTUT.</li> <li>◆ El acceso del glosario es independiente al capítulo seleccionado.</li> <li>◆ El glosario es desplegado en una ventana secundaria, conservando de esta manera la secuencia de los tópicos de información de REDTUT.</li> <li>◆ Dado que el glosario es un archivo independiente, es posible ser modificado fácilmente sin tener que disponer de todos los recursos (imágenes, texto, etc.) que se requieren para compilar algún capítulo de REDTUT.</li> </ul>	<ul style="list-style-type: none"> <li>◆ Si el archivo que contiene el glosario (glosario.hlp), no está cargado en la misma ruta de acceso del capítulo seleccionado, no se puede desplegar.</li> </ul>

Tabla 6-7 Ventajas y desventajas del uso del glosario en REDTUT.

### 6.1.8 Animación.

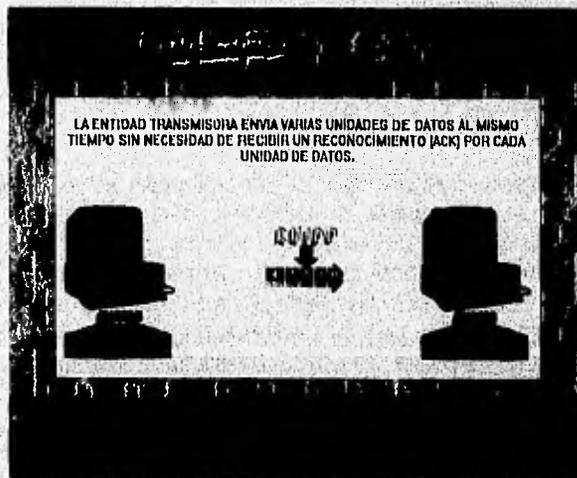


Figura 6-8 Pantalla de animación del proceso de corrección de errores en REDTUT.

La animación tiene la función de ejemplificar procesos relacionados con algún tema de redes de computadoras (Fig. 6-8 corrección de errores, control de flujo, etc). La animación se manejó como un programa independiente (.EXE), el cual es ejecutado dentro de un capítulo de REDTUT al activar el icono "animación" (Fig. 6-2).

Los programas de animación dentro de REDTUT tienen las ventajas y desventajas que se muestran en la tabla 6-8.

<b>Ventajas</b>	<b>Desventajas</b>
<ul style="list-style-type: none"> <li>◆ Ejemplifican de una forma más clara algún proceso relacionado con algún capítulo de REDTUT, que no es claramente entendido con sólo texto y/o imágenes.</li> <li>◆ Son programas independientes (archivos .EXE) que pueden ser ejecutados dentro de REDTUT con gran facilidad.</li> <li>◆ Al ser ejecutados no se pierde la continuidad de la navegación de REDTUT.</li> <li>◆ Un mismo programa de animación presenta las diferentes formas en que es posible realizar el proceso que se ejemplifica, con el fin de poder hacer comparaciones entre ellas y de evitar numerosos programas independientes que pueden generar más espacio en disco.</li> <li>◆ No requieren de espacio adicional en disco para ser ejecutados.</li> </ul>	<ul style="list-style-type: none"> <li>◆ Si el programa de animación (anm1.exe, anm2.exe, etc.), no está cargado en la misma ruta de acceso del capítulo seleccionado, no se puede desplegar.</li> <li>◆ Los programas de animación se hacen lentos si se ejecutan en drivers de discos flexibles (a, b, etc.).</li> </ul>

Tabla 6-8 Ventajas y desventajas de los programas de animación en REDTUT.

**6.1.9 Flexibilidad, transportabilidad y funcionalidad en REDTUT.**

REDTUT surge por la necesidad de proporcionar un apoyo adicional a los alumnos de la Facultad de Ingeniería que cursan la materia de "Redes de computadoras", por tal motivo es de gran importancia considerar estos tres aspectos que contribuyen a que cualquier aplicación cumpla con los fines para la que ha sido creada. REDTUT maneja estos aspectos convenientemente de la siguiente manera:

**Flexibilidad**

La flexibilidad que ofrece REDTUT puede ser tanto al usuario como al programador.

La flexibilidad que ofrece al usuario consiste en permitirle tomar parte de la edición del contenido de REDTUT de dos formas:

1. Usando el elemento de menú "Edición/Copiar", el cual copia el texto del tópico de información desplegado en pantalla al portapapeles de Windows, para después ser editado en cualquier procesador de texto (Write, Word, etc.) en ambiente Windows.
2. Usando el elemento de menú "Edición/Anotar...", el cual permite hacer anotaciones dentro de los mismos tópicos de información de REDTUT, las cuales se editan en una ventana secundaria que tiene la capacidad de guardar, copiar, pegar y eliminar la nota que se haya tecleado. En el momento que se guarda una nota hecha por el usuario se despliega la imagen de un "clip" en la esquina superior izquierda del área desplazable del tópico de información desplegado en la pantalla. Esta imagen tiene la finalidad de recordar al usuario que se ha hecho una anotación. Las notas que se guardan se graban dentro del mismo archivo de REDTUT y pueden ser accesadas mientras no se eliminen.

Por otro lado la flexibilidad que le ofrece al programador se debe a la facilidad con la que es posible hacer modificaciones a REDTUT, a pesar de la numerosa cantidad de archivos (.BMP, .SHG, .HPJ, etc.) que se requieren para poder compilar el archivo (.HLP) que se convertirá ejecutable para el sistema de ayuda en Windows.

### **Transportabilidad**

La transportabilidad que ofrece REDTUT es adquirida por los siguientes elementos:

- REDTUT no requiere de un equipo con características físicas muy sofisticadas, sólo requiere de un equipo que sea capaz de soportar Windows.
- REDTUT puede ser ejecutado en todas las versiones de Windows posteriores a la versión 3.0 (Windows 3.11, Windows 95, etc.), sin sufrir alteraciones que afecten el rendimiento del sistema.
- Solo requiere de aproximadamente 2.5 Mbytes libres en disco duro para ser cargado en su totalidad.
- En caso de que no se disponga de 2.5 Mbytes libres en disco duro, es posible ser ejecutado por módulos, es decir, capítulo por capítulo desde un drive externo (a, b, etc.).

### **Funcionalidad**

La funcionalidad que ofrece REDTUT es adoptada por los siguientes aspectos:

- Cumple con el objetivo de dar apoyo conceptual y visual con temas que se relacionan con las "Redes de computadoras".
- La facilidad de manejo que lleva implícita si el usuario es un usuario de Windows. En caso de que no lo sea, su manejo no tiene ninguna dificultad.
- La rapidez de acceso a la información proporcionada en cualquier forma de navegación, dada la característica de uso de hipertexto e hipergráficos.
- La posibilidad de obtener la información impresa de los tópicos que componen cualquier capítulo de REDTUT.
- La posibilidad de obtener sólo parte de la información contenida en REDTUT para ser procesada por el usuario desde algún procesador de texto en ambiente Windows.

## **6.2 ACTUALIZACION DE REDTUT.**

Todos los sistemas que proporcionan un servicio tienen que ser actualizados a las necesidades que vayan surgiendo con el tiempo, es por esto que es necesario conocer la forma de actualización del contenido de REDTUT.

### **6.2.1 Actualización de archivos .HLP**

Como ya es sabido, los archivos .HLP son parte fundamental de REDTUT, dado que son los que contienen el texto, las imágenes, el diseño de las tres diferentes formas de navegación y las funciones que permiten realizar tareas como impresión, edición, etc.

La actualización de .HLP depende de la necesidad de modificar el contenido de alguno de los capítulos que integran REDTUT; ya sea agregar información importante a un tema existente, agregar un tema nuevo, modificar la secuencia de navegación, etc. Independientemente de la necesidad que motive la actualización de REDTUT, ésta debe hacerse sobre los archivos que le dieron origen, que son:

- Archivos .RTF (cap1.rtf, cap2.rtf, cap3.rtf, cap4.rtf, glosario.rtf, etc.).

- ♦ Archivos .HPJ (cap1.hpj, cap2.hpj, cap3.hpj, cap4.hpj, glosario.hpj, etc.).
- ♦ Archivos .BMP (Imágenes de mapas de bits).
- ♦ Archivos .SHG (Imágenes con hotspots múltiples).

El método utilizado para la creación y/o modificación de cada uno de estos archivos se presentó en la parte de "Desarrollo" de los capítulos 4 y 5.

### **6.2.2 Actualización de archivos .EXE**

Los archivos ejecutables (.EXE) que forman parte de REDTUT son: la pantalla principal y los programas de animación. Debido a que estos archivos han sido generados con programación tradicional, su modificación debe hacerse sobre los siguientes archivos:

- ♦ Archivos fuente (redtut.c, anm1.c, anm2.c, etc.).
- ♦ Archivos de recursos (redtut.rc, anm1.rc, anm2.rc, etc.).
- ♦ Archivos de definición (redtut.def, anm1.def, anm2.def, etc.).
- ♦ Archivos project (redtut.prj, anm1.prj, anm2.prj, etc.).

El manejo adecuado de éstos archivos se describió en la parte de "Desarrollo" en el capítulo 5.

Asimismo, es posible crear otros programas de animación, o de otro tipo, que sirvan de apoyo para la comprensión de los temas tratados en el tutorial y posteriormente ser integrados de una manera sencilla al ambiente de REDTUT.

## CONCLUSIONES

## COMENTARIOS FINALES

Una de las tareas que caracterizan el desempeño de un ingeniero en computación es la capacidad de utilizar las herramientas que la tecnología proporciona, con el fin de satisfacer alguna necesidad práctica que surja dentro de cualquier ámbito.

El desarrollo del sistema tutorial de redes de computadoras (REDTUT), es el resultado obtenido en la tarea de satisfacer la necesidad práctica de contar con un apoyo adicional al curso de "Redes de computadoras" que se imparte en la Facultad de Ingeniería de la UNAM. Este sistema tutorial tiene la característica de estar desarrollado con herramientas soportadas por un ambiente operativo ya existente (Microsoft Windows).

El sistema tutorial de redes de computadoras (REDTUT), muestra las ventajas de introducir la computación en tareas relacionadas con el sector educativo, como son: el proceso enseñanza-aprendizaje y la consulta de información de un tema específico (Redes de computadoras); las cuales se ven beneficiadas con la implementación de hipertexto, hipergráficos, animación y el uso de un ambiente gráfico; elementos que proporcionan al usuario una interfase con:

- ◆ Presentación y acceso rápido de la información.
- ◆ Capacidad de ejemplificar procesos animados.
- ◆ Apoyo visual a través de imágenes y color.
- ◆ Facilidad de uso a través de botones, iconos, ventanas, etc.
- ◆ Capacidad de realizar tareas como: impresión y edición. etc.

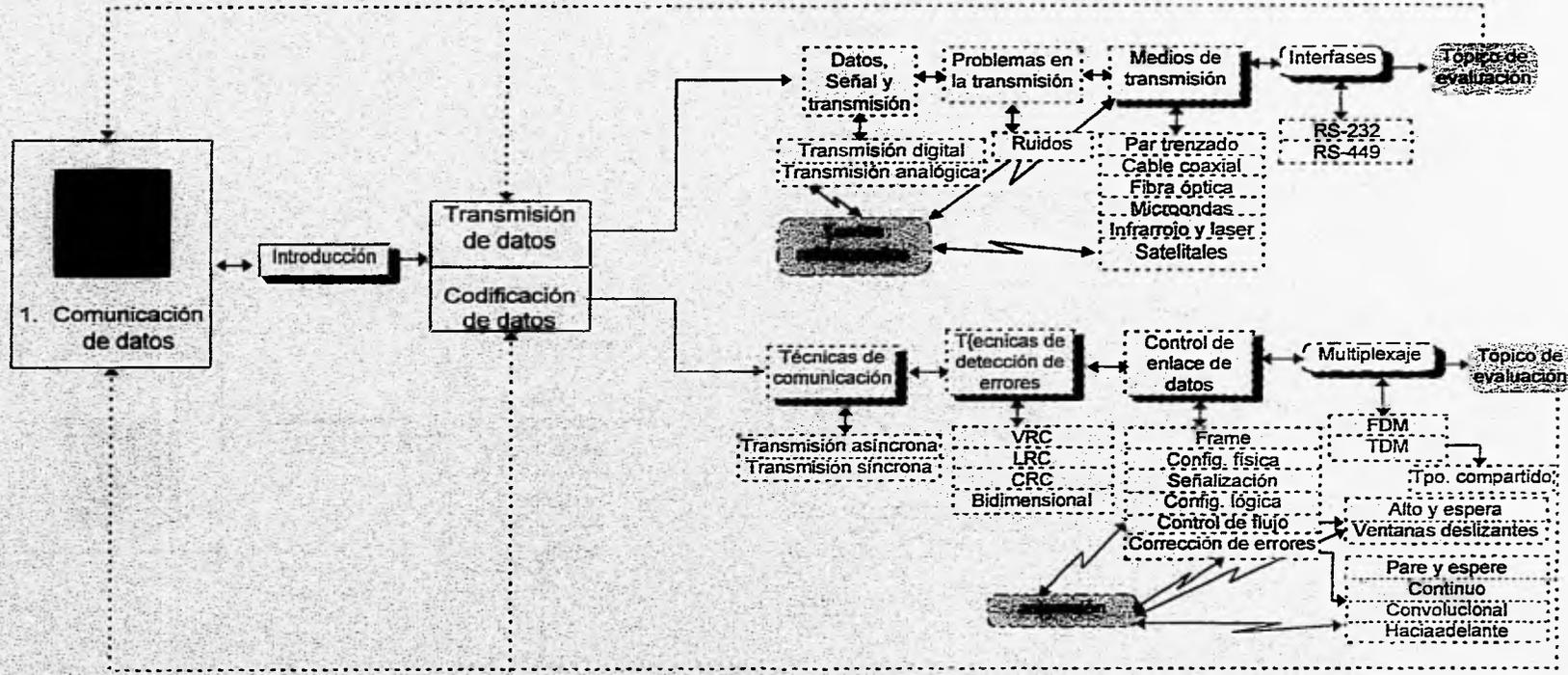
Sin embargo, a pesar de todas las ventajas proporcionadas en el sistema tutorial (REDTUT) con respecto a otros medios tradicionales empleados en la educación (libros, revistas, etc.), su función se concreta a ser únicamente un apoyo adicional, dado que no se puede prescindir del factor humano, que además de ser un elemento importante en la formación integral de un alumno, es determinante en su proceso de aprendizaje.

Por otro lado, cabe señalar que se carecen de medios específicos para difundir el tutorial en la comunidad estudiantil de la Facultad de Ingeniería, por lo cual dependerá en gran medida de los profesores que imparten la materia de redes de computadoras, así como de su disposición como material de apoyo en los espacios de cómputo y/o consulta con los que se cuentan actualmente en la Facultad de Ingeniería (laboratorios abiertos de computación, bibliotecas, etc).

**P A R T E      V**  
**A P E N D I C E S**

# APENDICE A

# NAVEGACION DEL CAPITULO 1 DE REDTUT: COMUNICACION DE DATOS.



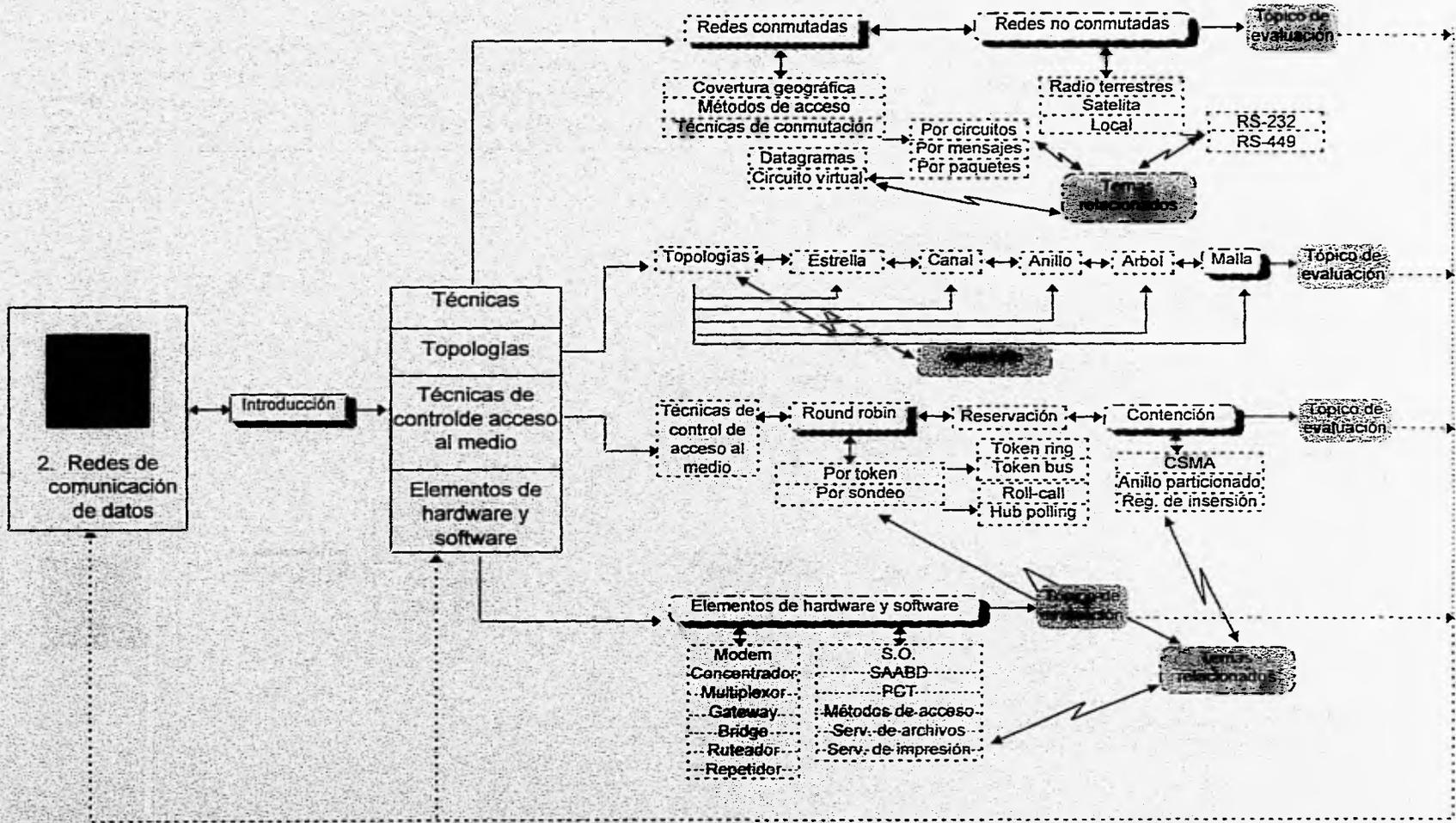
- Tópico de contenido
- Tópico de introducción
- Tópico de tema
- Tópico de temas múltiples

- Tópico de tema final
- Temas relacionados
- Tópico de evaluación

- ↔ Secuencia de páginas
- ↕ Ventanas secundarias
- Saltos con hotspots

- Saltos
- ↔ Referencias externas

## NAVEGACION DEL CAPITULO 2 DE REDTUT: REDES DE COMUNICACION DE DATOS.

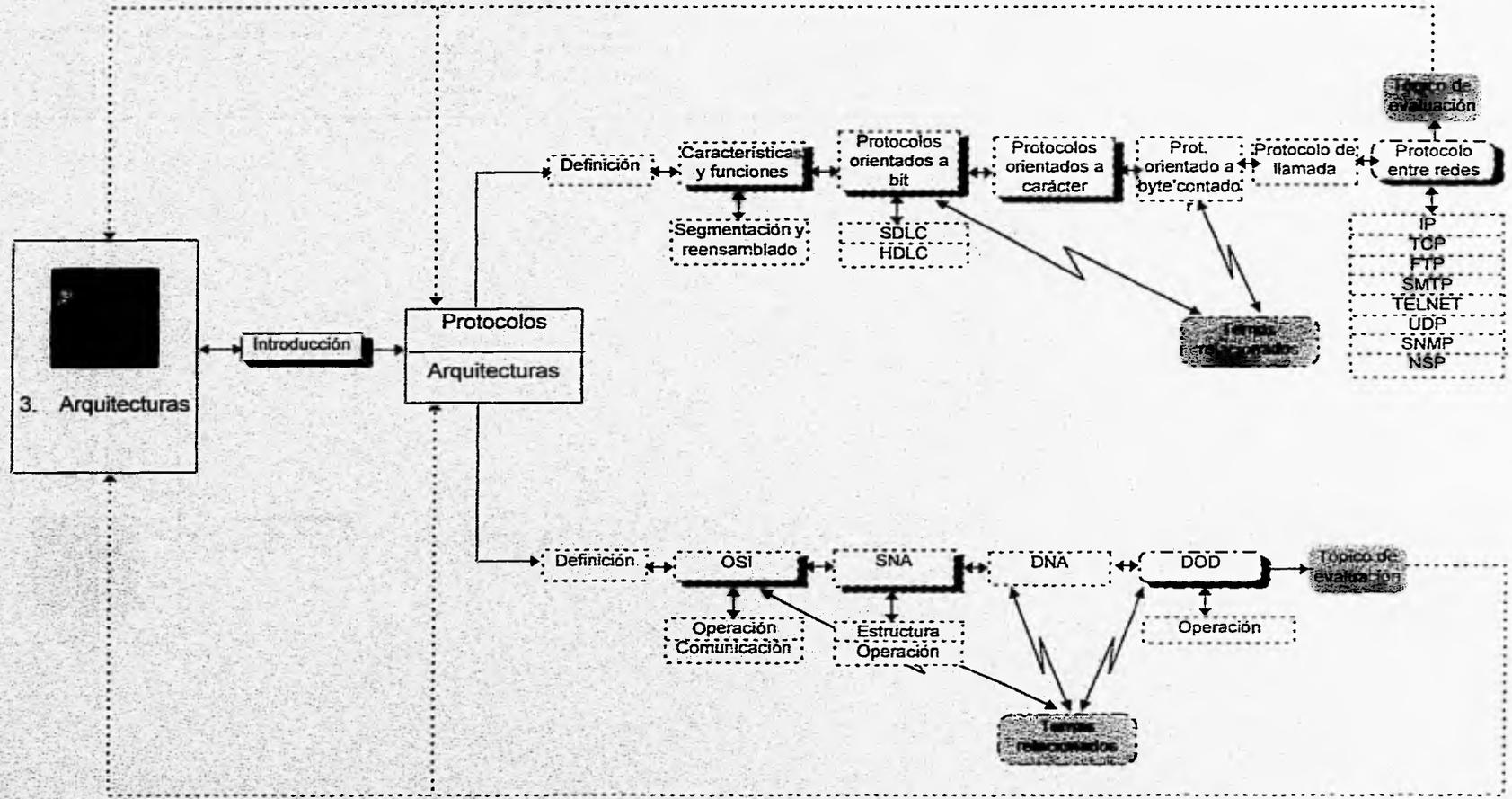


- Tópico de contenido
- Tópico de introducción
- Tópico de tema
- Tópico de temas múltiples

- Tópico de tema final
- Temas relacionados
- Tópico de evaluación

- ↔ Secuencia de páginas
- ↕ Ventanas secundarias
- Saltos
- ↔ Referencias externas
- Saltos con hotspots

# NAVEGACION DEL CAPITULO 3 DE REDTUT: ARQUITECTURAS.

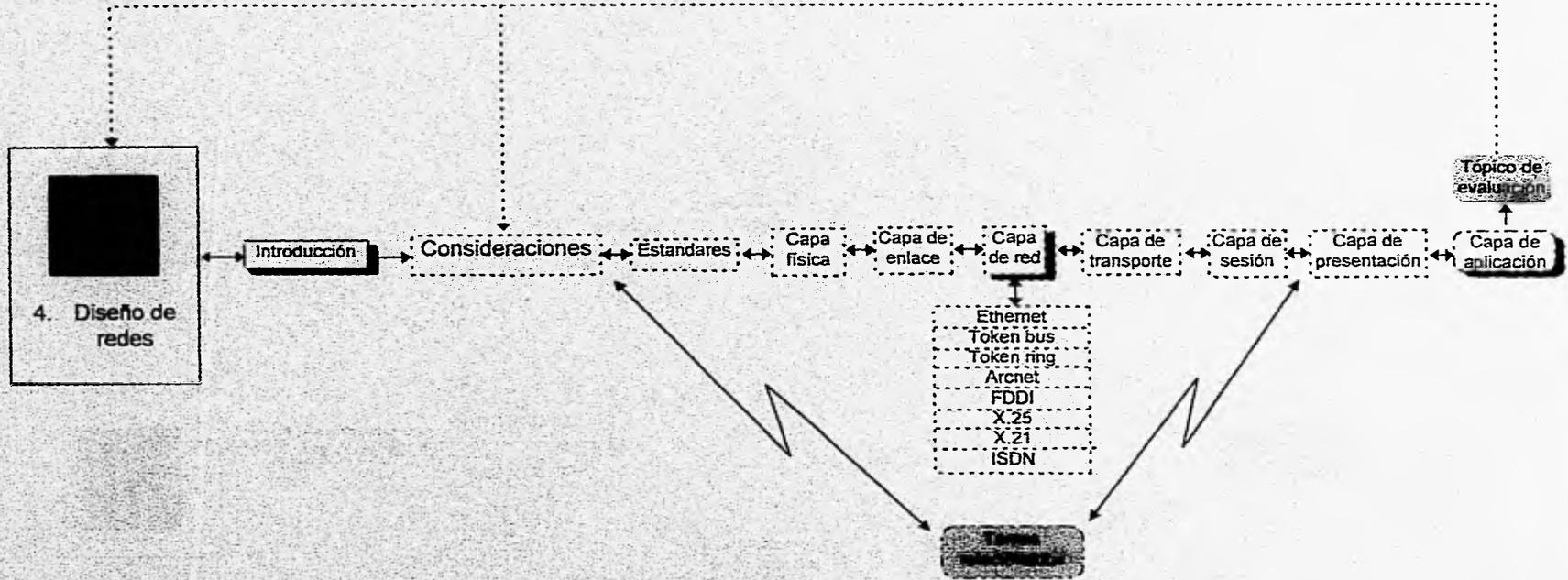


- Tópico de contenido
- Tópico de introducción
- Tópico de tema
- Tópico de temas múltiples

- Tópico de tema final
- Temas relacionados
- Tópico de evaluación

- Secuencia de páginas
- Ventanas secundarias
- Saltos con hotspots
- Saltos
- Referencias externas

## NAVEGACION DEL CAPITULO 4 DE REDTUT: DISEÑO DE REDES.



- |                           |                      |                      |                      |
|---------------------------|----------------------|----------------------|----------------------|
| Tópico de contenido       | Tópico de tema final | Secuencia de páginas | Saltos               |
| Tópico de introducción    | Temas relacionados   | Ventanas secundarias | Referencias externas |
| Tópico de tema            | Tópico de evaluación | Saltos con hotspots  |                      |
| Tópico de temas múltiples |                      |                      |                      |

## **APENDICE B**

## MACROFUNCIONES DE AYUDA EN WINDOWS.

### Macros de acceso a la barra de botones:

Back ( )	Despliega el t3pico previo en la lista de historia.
Contents ( )	Despliega el t3pico de contenido.
History ( )	Despliega la lista de historia.
Next ( )	Despliega el t3pico siguiente en la secuencia de p3ginas.
Prev ( )	Despliega el t3pico previo en una secuencia de p3ginas.
Search ( )	Despliega la caja de di3logo buscar.

### Macros que accesan los elementos de men3:

About ( )	Despliega la caja de di3logo acerca de...
Annotate ( )	Despliega la caja de di3logo de anotaci3n.
BookmarkDefine ( )	Despliega la caja de di3logo para definir marcadores de t3picos
BookmarkMore ( )	Despliega la caja de di3logo con mas marcadores de t3picos
CopyDialog ( )	Despliega la caja de di3logo copiar.
CopyTopic ( )	Copia el t3pico actual en el portapapeles.
Exit ( )	Sale del sistema de ayuda.
FileOpen ( )	Despliega la caja de di3logo abrir.
HelpOn ( )	Despliega ayuda sobre el uso del t3pico.
HelpOnTop ( )	Congela el estado de la ayuda a siempre visible.
Print ( )	Imprime el t3pico actual.
PrinterSetup ( )	Despliega la caja de di3logo para especificar impresoras.

### Macros para manejar barra de botones

BrowseButtons ( )	Agrega botones de secuencia.
ChangeButtonBinding ("id_bot3n","macro" )	Asigna una macro a un bot3n (pueden ser los est3ndar).
CreateButton ("id_bot3n","etiqueta","macro")	Agrega un nuevo bot3n en la barra de botones.
DestroyButton ("id_bot3n")	Remueve un bot3n de la barra de botones.
DisableButton ("id_bot3n")	Deshabilita un bot3n.
EnableButton ("id_bot3n")	Habilita un bot3n deshabilitado previamente.
SetContents ("nom_arch",num_contexto)	Despliega el t3pico de contenido.

**Macros para manejar la barra de menú**

AppendItem ( )	Agrega un elemento de menú.
ChangeItemBinding ("id_elemento","macro" )	Asigna una macro a un elemento de menú.
CheckItem ("id_elemento")	Coloca una marca de chequeo en un elemento de menú
DeleteItem ("id_elemento")	Remueve un elemento del menú.
DisableItem ("id_elemento")	Deshabilita un elemento de menú.
EnableItem ("id_elemento")	Habilita un elemento de menú.
InsertItem ("id_menu", "id_elemento", "nom_elemento", "macro", posición)	Inserta un elemento de menú.
InsertMenu ("id_menu", nom_menu, posición)	Inserta un nuevo menu
SetHelpOnFile (" nom_arch")	Designa un archivo de uso de ayuda específico.
UncheckItem ( "id_elemento")	Remueve la marca de chequeo de un elemento de menú.

**Macros de despliegan tópicos en ventanas pop-up.**

PopupContext ("nom_arch", num_contexto")	Despliega un tópico específico en una ventana instantánea.
PopupId ("nom_arch", "cadena_contexto")	Despliega un tópico específico en una ventana instantánea.

**Macros para extender la potencia del sistema de ayuda.**

ExecProgram ("línea de comando", "edo. de la pantalla)	Ejecuta un programa.
RegisterRoutine ("nom_DLL", "nom_función", "formato_parametros")	Registra una función DLL.

**Macros para saltar a tópicos de archivos.**

JumpContents ("nom_arch")	Salta al tópico de contenido del archivo dado.
JumpContext ("nom_arch", num_contexto)	Salta al número de contexto especificado en el archivo dado.
JumpHelpOn ( )	Salta al archivo del uso de la ayuda.
JumpId ("nom_arch", "cadena_contexto")	Salta al tópico especificado por la cadena de contexto.
JumpKeyword WinHelp macro	Salta al tópico que contiene la palabra clave

**Macros para manejar ventanas del sistema de ayuda.**

CloseWindow ("nom_ventana")	Cierra una ventana.
FocusWindow ("nom_ventana")	Cambia el foco de la ventana.
PositionWindow (horiz_pos, vert_pos, ancho, alto, edo_ventana, "nom_ventana")	Da tamaño y posición a la ventana especificada.

**Macros que trabajan con marcadores de texto.**

DeleteMark ("marcador")	Remueve un marcador de texto.
GoToMark ("marcador")	Salta a un marcador.
IfThen (IsMark("marcador"), "macro")	Ejecuta una macro si el marcador existe.
IfThenElse (IsMark("marcador"), "macro1", "macro2")	Ejecuta una de dos macros si el marcador existe.
IsMark ("marcador")	Prueba si existe o no el marcador dado.
Not(IsMark ("marcador"))	Invierte la condición de la macro IsMark.
SaveMark ("marcador" )	Guarda un marcador de texto.

**Macros que trabajan con aceleradores de teclado**

AddAccelerator (Tecla, edo.de cambio, "macro")	Asigna una macro a un acelerador de teclado.
RemoveAccelerator (tecla, edo_de cambio )	Remueve un acelerador de teclado.

# APENDICE C

**BOOL WinHelp(hwnd, lpszHelpFile, fuCommand, dwData)**

**HWND** *hwnd*; /\* encabezado de la ventana que solicita la ayuda \*/  
**LPCSTR** *lpszHelpFile*; /\* cadena que direcciona la ruta del directorio \*/  
**UINT** *fuCommand*; /\* tipo de ayuda \*/  
**DWORD** *dwData*; /\* dato adicional\*/

La función WinHelp inicia la ayuda en Windows (WINHELP.EXE) y pasa datos opcionales indicando la naturaleza de la ayuda solicitada por la aplicación. La aplicación especifica el nombre, y cuando es requerida, la ruta del archivo de ayuda que la aplicación desplegará.

Parámetro	Descripción
hwnd	Identifica la ventana que solicita la ayuda
lpszHelpFile	Puntero a una cadena que termina con NULL, si es necesario, y al nombre del archivo que desplegará la ayuda. El nombre del archivo debe ser seguido por el símbolo (>) y el nombre de una ventana secundaria si el tópico debe ser desplegado en una ventana secundaria en lugar de la ventana primaria. El nombre de la ventana secundaria debe estar definido en la sección [WINDOWS] del archivo proyecto (.HPJ).
fuCommand	Especifica el tipo de ayuda solicitado.
dwData	Especifica un dato adicional. Este valor depende del valor del parámetro fuCommand.

**Regresa**

Valor diferente de cero si la función fue exitosa. De otro modo, regresa el valor cero.

**Comentarios**

Antes de cerrar la ventana que solicita la ayuda, la aplicación debe llamar WinHelp con fuCommand en HELP\_QUIT.

**TIPOS DE AYUDA**

fuCommand	dwData	Acción
HELP_CONTEXT	Un entero largo sin signo contiene el número de contexto.	Despliega el tópico de ayuda identificado por el número de contexto definido en la sección [MAP] del archivo .HPJ.
HELP_CONTENTS	Ignorado; las aplicaciones deben ponerlo en 0L.	Despliega el tópico de contenido definido en la opción de contenido en la sección [OPTIONS] del archivo .HPJ.
HELP_SETCONTENTS	Un entero largo sin signo que contiene el número de contexto para el tópico de aplicación que se quiere designar como tópico de contenido.	Determina cual tópico de contenido de ayuda debe ser desplegado cuando el usuario presiona la tecla F1.
HELP_CONTEXTPOPUP	Un entero largo sin signo que contiene el número de contexto para un tópico.	Despliega en una ventana pop-up un tópico de ayuda particular identificado por un número de contexto que ha sido definido en la sección [MAP] de un archivo .HPJ.
HELP_KEY	Un puntero largo a una cadena que	Despliega el tópico encontrado en la lista

**FUNCION WINHELP**

	contiene una palabra clave para el t3pico deseado.	de palabras clave que coincide con la palabra clave pasada en el par3metro dwData, si es coincidente con s3lo uno. Si es coincidente con m3s de uno, despliega la caja de di3logo de b3squeda con los t3picos listados.
HELP_PARTIALKEY	Un puntero largo a una cadena que contiene una palabra clave para el t3pico deseado.	Despliega el t3pico encontrado en la lista de palabras clave que coincide con la palabra clave pasada en el par3metro dwData, si es coinciden con s3lo uno. Si es coincidente con m3s de uno, despliega la caja de ir a... con los t3picos listados. Si no coincide se despliega la caja de di3logo de b3squeda. Si se desea saltar a la caja de di3logo sin pasar una palabra clave (el tercer resultado), se debe usar un puntero largo a una cadena vac3a.
HELP_MULTIKEY	Un puntero largo a una estructura MULTIKEYHELP, definida en WINDOWS.H. Esta estructura especifica la tabla de caracteres de palabras clave de notas al pie.	Despliega el t3pico de ayuda definido por una palabra clave en una tabla de palabras claves alternativa.
HELP_COMMAND	Un puntero largo a una cadena que contiene una macro de ayuda a ser ejecutada.	Ejecuta una macro de ayuda.
HELP_SETWINPOS	Un puntero largo a una estructura HELPWININFO, la cual est3 definida en WINDOWS.H. Esta estructura especifica el tama3o y la posici3n de la ventana de ayuda principal o una ventana secundaria al ser desplegadas.	Despliega la ventana de ayuda si esta minimizada o en memoria, y la posiciona de acuerdo con el dato pasado.
HELP_FORCEFILE	Ignorado; las aplicaciones deben ponerlo en 0L.	Garantiza que WinHelp est3 desplegando el archivo de ayuda correcto. Si el archivo de ayuda desplegado es correcto, esta no act3a. Si el archivo de ayuda desplegado es incorrecto, WinHelp abre el archivo correcto.
HELP_HELPONHELP	Ignorado; las aplicaciones deben ponerlo en 0L.	Despliega el t3pico de contenido de el archivo de ayuda designado para el uso de ayuda.
HELP_QUIT	Ignorado; las aplicaciones deben ponerlo en 0L.	Informa a la aplicaci3n de ayuda que la ayuda ya no ser3 necesitada. Si no hay otras aplicaciones que lo requieran, Windows cierra la aplicaci3n de ayuda.

## APENDICE D

## CAPITULO 1: COMUNICACION DE DATOS

## INTRODUCCION

Al hablar de comunicación se asume que como mínimo deben estar involucradas un par de entidades. La transferencia de información ocurre a través de un canal o un medio. En cualquier momento una entidad puede ser:

- Un transmisor.- Es el dispositivo que envía la información.
- Un receptor.- Es el dispositivo que recibe la información.
- Un transceptor.- Es el dispositivo que envía y recibe información simultáneamente.

Existen 3 categorías que describen el tipo de comunicación entre entidades:

- Comunicación Simplex.- Comunicación en la cual una entidad es transmisora y la otra es receptora, es decir, el flujo de la información es en una sola dirección.
- Comunicación half-duplex.- Comunicación que permite a cada entidad transmitir, pero no simultáneamente, es decir, el flujo de la información es en dos direcciones pero no simultáneamente.
- Comunicación full-duplex.- Comunicación que permite a cada entidad transmitir y recibir simultáneamente, es decir, el flujo de la información es en dos direcciones simultáneamente.

Tradicionalmente se hace referencia a dos formas de conectar un sistema central con una o varias terminales: punto a punto y multipunto.

**PUNTO A PUNTO:** En este tipo de enlace, un enlace físico une sólo dos puntas de transmisión de datos, desde donde se puede enviar y/o recibir.

**MULTIPUNTO:** En este tipo de enlace se hace referencia a un sistema central que conecta varias terminales o sistemas secundarios.

**ENLACE PUNTO A PUNTO**

En un enlace punto a punto cada dispositivo tiene una conexión directa o línea dedicada con otro u otros dispositivos, es decir, se establece un único canal de comunicación entre dos sistemas de procesamiento de datos. Además cuenta con:

- Bajo costo.
- Permite la forma "conversacional" de comunicaciones.
- Apta para la transmisión de grandes volúmenes de datos.
- Permite fácil migración a otro medio físico de comunicación que se quiera utilizar.
- Admite la utilización de diferentes medios físicos en una red.
- Es de fácil implementación.

**ENLACE MULTIPUNTO**

En un enlace multipunto los dispositivos conectados comparten el mismo canal de comunicación, es por esto que una sola estación puede transmitir a la vez. El número de terminales permitido para compartir el medio, está en función de la capacidad del canal y del trabajo de carga en las terminales.

- Exige la utilización de una técnica de acceso al medio.
- Puede aumentar los tiempos de respuesta en comparación con el enlace punto a punto.
- Permite la conexión de más terminales por cada procesador central.
- El software y hardware es relativamente complejo.

**DATOS, SEÑALIZACION Y TRANSMISION****ANALOGICO Y DIGITAL**

Los términos analógico y digital corresponden aproximadamente, con continuo y discreto, respectivamente. Estos dos términos son frecuentemente usados en comunicación de datos en por lo menos tres contextos:

- Datos.- Son entidades que transmiten un significado.
- Señales.- Son los datos codificados eléctrica o magnéticamente. Pueden ser de dos tipos: analógicas y digitales.
- Transmisión.- Es la comunicación de datos por la propagación y procesamiento de señales.

En un sistema de comunicación se propagan datos desde un punto a otro por medio de señales eléctricas, independientemente el tipo de transmisión que se utilice, pudiendo ser:

Transmisión analógica

Transmisión digital

## TRANSMISION DIGITAL

Es la forma de transmisión de señales digitales. En la transmisión digital una cadena de pulsos on/off son enviados como datos viajando en circuitos electrónicos. Los pulsos son referidos como bits. Las señales digitales sólo pueden ser transmitidas a distancias limitadas, antes de que la atenuación ponga en peligro la integridad de los datos.

Características:

- Para lograr la transmisión digital a grandes distancias se utilizan repetidores que recuperan el patrón de 1's y 0's, retransmitiendo una nueva señal.
- Mantiene integridad en la información durante la transmisión.
- En el caso de que los datos sean analógicos, reciben el mismo tratamiento que los digitales.
- Generalmente es más barata que la transmisión analógica.

## TRANSMISION ANALOGICA

Es la forma de transmisión de señales analógicas. En una transmisión analógica la amplitud y la frecuencia de oscilación de la señal transmitida varía sobre un rango continuo. Las señales pueden representar datos analógicos (p.ej. voz) o datos digitales (los cuales antes deben pasar por un proceso de conversión).

Las señales analógicas sólo pueden ser transmitidas a distancias limitadas, antes de que la atenuación ponga en peligro la integridad de los datos.

Características:

- Para lograr grandes distancias de transmisión, los sistemas de transmisión analógica propagan la señal a través de amplificadores.
- En el caso de que los datos sean digitales, deben ser pasados por un modem que permita que reciban el mismo tratamiento que los analógicos.
- Los amplificadores usados para mantener la señal durante la transmisión, también inyectan componentes de ruido a la señal.

## PROBLEMAS EN LA TRANSMISION

Los medios físicos de transmisión no son perfectos debido a que algunos de ellos son muy afectados por los ruidos y distorsiones que se presentan en el medio ambiente.

### RUIDO

Conocer los diferentes tipos de ruidos es importante en la comunicación de datos, porque ellos son una de las principales causas de los errores en la transmisión. Existen varios tipos de ruido pero los que se presentan frecuentemente en la transmisión de datos son:

- Ruido térmico
- Ruido aleatorio
- Ruido de impulso
- Intermodulación
- Ruido de cruce

### RUIDO TERMICO

Se refiere al ruido generado por los átomos y moléculas de todas las sustancias de un medio de transmisión, que al vibrar constantemente generan calor, es decir, a más altas temperaturas es más grande la vibración. Como los átomos vibran, estos envían fuera ondas electromagnéticas de todas frecuencias. Los electrones en conductores eléctricos se mueven similarmente en la misma forma aleatoria. La potencia del ruido térmico es proporcional al ancho de banda.

### RUIDO ALEATORIO

Es el ruido que se extiende al azar por el espectro de frecuencias. Es causado por la actividad molecular del medio a través del cual el mensaje es transmitido. Se corrige aumentando en gran medida la relación Señal/Ruido.

### RUIDO DE IMPULSO

Se refiere a cualquier divergencia inesperada de la señal recibida, que va más allá de un cierto nivel de amplitud. El ruido de impulso puede tener picos de muy grande amplitud el cual sature el canal y deje afuera el dato.

Es causado por la actividad eléctrica, iluminación, etc. Esta es la causa principal de los errores de transmisión y se corrige con un buen aislamiento y el uso de equipo no defectuoso.

### INTERMODULACION

Se refiere a la combinación de 2 frecuencias  $F_1$  y  $F_2$  o sus armónicas  $nF_1$  y  $mF_2$ , que producen nuevas componentes de la señal de diferentes frecuencias. El ruido de intermodulación muchas veces es provocado, debido a que sobre un canal multiplexado muchas señales diferentes son amplificadas a la vez. Las señales desde dos canales independientes se intermodulan uno al otro para formar un producto, el cual cae dentro de una banda de frecuencias reservada para otra señal.

### RUIDO DE CRUCE

Este ruido se presenta cuando una señal capta alguna de las señales que está viajando sobre otro canal. Esto ocurre entre cables de par trenzado con señales separadas, ligas multiplexadas, ligas de microondas, etc.

Una de las más fuertes fuentes de ruido de cruce, es la inducción entre cables de circuitos separados. La posibilidad de que se haga presente depende del como se incremente la longitud del circuito, la proximidad entre los circuitos, la frecuencia de la señal y la amplitud de la señal.

### ATENUACION

La atenuación es uno de los mayores problemas en la transmisión de señales, tanto analógicas como digitales. La atenuación es la pérdida de amplitud en la señal a medida que crece la distancia de propagación. Estas pérdidas aumentan a mayor frecuencia.

### RETRASO

En la distorsión por retraso la fase de la señal no es transmitida linealmente. La señal es retardada más en algunas frecuencias que en otras. El retraso ocurre cuando una onda de banda angosta modulada en amplitud, se transmite a través de un circuito. Cuanto mayor sea la frecuencia de operación de un circuito mayor será el defasamiento.

### TRANSMISION vs. ERROR

La velocidad de transmisión afecta a la confiabilidad de la transmisión. A más alta velocidad de transmisión, hay mayor probabilidad de que los datos sean afectados, dado que si la señal se contamina con un ruido pequeño, se corre el riesgo de que este destruya una secuencia de bits.

## MEDIOS DE TRANSMISION

El medio de transmisión es la facilidad física usada para interconectar estaciones del usuario y dispositivos, creando una red de transporte de mensajes entre las mismas. La selección del medio físico depende de varios factores

Los medios físicos pueden clasificarse en dos formas:

#### MEDIOS GUIADOS

Par trenzado  
Cable coaxial  
Fibra óptica

#### MEDIOS NO GUIADOS

Microondas  
Infrarrojo y laser  
Satélites

### SELECCION DEL MEDIO DE TRANSMISION

- Tipo de ambiente donde se va a instalar.
- Tipo de equipo a usar.
- Tipo de aplicación y requerimientos.
- Capacidad económica (relación Costo/Beneficio esperada), etc.

### PAR TRENZADO

Este es uno de los cables más comúnmente empleados en telecomunicaciones. Un cable par trenzado es un tipo de cable que consta de dos conductores de cobre, cada uno cubierto por un material aislante y torcidos uno contra el otro. Es esta característica la que los distingue con el nombre de cables de par trenzado (Twisted Pair).

El par trenzado es susceptible a corrientes eléctricas no conductivas (como campos magnéticos), es por esto que se requiere de la envoltura del conductor de cobre.

El par trenzado a su vez se encuentra protegido por una cubierta aislante y protectora en la capa exterior denominada jacket (forro).

El par trenzado es capaz de soportar la transmisión de voz y de datos.

Existen diferentes tipos de par trenzado, sin embargo, todos tienen las mismas características eléctricas sus diferencias consisten en el diámetro del cable.

### DIAMETRO DEL CABLE PAR TRENZADO

El diámetro se mide en AWG (American Wire Gauge), los más comunes son de 22, 24 y 26 AWG. En este tipo de unidad de medición el cable de más alto número es el de menor diámetro.

- 22 AWG es usado para transmitir datos a alta velocidad.
- 24 AWG es usado por la mayoría de los teléfonos y aplicaciones de transmisión de datos.
- 26 AWG es usado para remiendo de equipo.

### CARACTERISTICAS DEL PAR TRENZADO

Los cables que son los conductores de cobre más delgados y menos protegidos por un jacket (forro) son denominados del tipo UTP (Unshielded Twisted Pair, cables de par trenzado sin blindar). Son sumamente baratos y flexibles dado que permiten manipular una señal a una distancia de 110 metros sin el uso de amplificadores.

Los cables de conductores más gruesos y muy bien cubiertos por un jacket (forro) son denominados del tipo STP (Shielded Twisted Pair, cables de par trenzado blindado). Estos últimos son más caros y menos flexibles que los UTP, pero permiten un rango de operación de hasta 500 metros, como es el caso de las instalaciones de tipo de Token Ring STP cuando se instalan redes con un número máximo de 15 unidades MAU (Múltiple Access Unit) con tarjetas de 4 Mbps.

Generalmente:

- Una red que utilice éste medio, puede tener conectados hasta 1000 dispositivos del usuario.
- Permite trabajar en half-duplex y en full-duplex

**VENTAJAS DEL PAR TRENZADO**

- Facilidad y rapidez de instalación
- Compatibilidad con Ethernet, Token Ring y Starlan.
- Ancho de banda de 10 Mbps.
- Distancias de hasta 110 metros con cables UTP y de hasta 500 metros con cables STP.
- Excelente relación de precio rendimiento.
- El precio promedio del cable UTP es de \$0.60 dólares por metro en tanto que el cable STP es de \$2.50 dólares por metro.
- Buena tolerancia a interferencias debidas a factores ambientales.
- Puede transportar de 12 a 24 canales de grado de voz
- Son válidos en cualquier topología: anillo, estrella, bus, árbol, etc.
- Transportan señales analógicas y digitales.
- Bajo costo.
- Soporta equipos del mercado

**DESVENTAJAS DEL PAR TRENZADO**

- Alta tasa de error a grandes velocidades.
- Baja inmunidad al ruido, interferencia, etc.
- Requiere de protección especial: blindaje, ductos, etc.
- limitaciones en distancia.
- La cantidad de información procesada en una unidad de tiempo es limitada

**CONDUCTOR DE COBRE (Par trenzado)**

Los conductores internos de cobre son usados para llevar la señal de comunicación. Generalmente tiene un diametro de 1mm. La forma trenzada de los conductores es usada para reducir interferencias eléctricas y ruido de cruce.

**MATERIAL AISLANTE (Par trenzado)**

Consiste de un material dieléctrico que no es capaz de conducir señales eléctricas. Este material puede ser PVC o algún otro plástico sintético.

**CABLE COAXIAL**

Está conformado por un alambre conductor básico cubierto por una placa o malla metálica que actúa como tierra. El alambre conductor y la tierra se encuentran separados por un aislante plástico, y finalmente, todo el conjunto está protegido por una cubierta exterior, también aislante, a la que por lo común se llama jacket (forro).

Los cables coaxiales pueden ser de varios tipos y anchos. Sin embargo, su principal característica es que pueden transportar una señal eléctrica a mayor distancia entre más grueso sea el conductor. El cable grueso suele ser más caro y menos flexible.

Es comunmente usado para conexiones de redes de datos y sistemas de seguridad. Existen diferentes tipos de cable coaxial pero generalmente se pueden dividir en:

Cable coaxial de banda angosta

Cable coaxial de banda ancha

**VENTAJAS DEL CABLE COAXIAL**

- Transmisión de voz, video y datos
- Fácil instalación.
- Compatibilidad con Ethernet y Arcnet
- Ancho de banda de 10 Mbps
- Distancias hasta 600 metros sin necesidad de repetidores
- Muy buena tolerancia a interferencias debidas a factores ambientales.
- Precio promedio \$1.00 dólar estadounidense por metro en el cable delgado y \$2.5 dólares por metro el cable grueso.
- Inmunidad a interferencias eléctricas
- Es la base de varias instalaciones.

**DESVENTAJAS DEL CABLE COAXIAL**

- El tamaño del cable.
- Costo de instalación.
- Opciones de configuración limitada.
- Dificultad de mantenimiento

**CABLE COAXIAL DE BANDA ANGOSTA**

- Transmite una señal digital simple, en half-duplex
- No hay modulación de frecuencia
- Se usan repetidores para alargar sus límites de distancia.
- Se usan para tipologías de bus, árbol y algunas veces anillo.
- Soportan de 200 a 1000 dispositivos.
- Alcance de 1 a 10 Km.
- Ancho de banda de 10 Mbps.
- Bajo costo en comparación con el de banda ancha.

**CABLE COAXIAL DE BANDA ANCHA**

- Es el mismo usado en redes de TV por cable.
- Se combina datos, voz y video simultáneamente.
- Se permite voz y video en tiempo real.
- Los datos deben ser modulados antes de la transmisión.
- Todas las señales se transmiten en half-duplex
- Se pueden conectar hasta 25 000 dispositivos con un alcance de 5 Km.
- Se usa en topologías de bus y árbol.
- Ancho de banda máximo: 400 Mhz y puede transmitir 100 % de su carga.
- Mejor inmunidad a los ruidos.
- No necesita conducto.
- Su costo es alto dado que se requieren de modems en cada estación del usuario.

**CONDUCTOR CENTRAL INTERNO (Cable coaxial)**

En algunos cables el conductor es de soldadura de cobre, es decir, es un cable de acero de alta resistencia recubierto de cobre.

**AISLANTE DIELECTRICO (Cable coaxial)**

Tiene la función de separar el conductor interno y conductor externo (la malla de cobre). El material puede ser polietileno. De este material depende la velocidad de propagación.

**MALLA DE COBRE (Cable coaxial)**

La función de la malla es impedir que el ruido eléctrico entre al cable. La malla también puede ser un tubo de aluminio sólido o una lámina de metal.

**CUBIERTA PLASTICA DE PROTECCION (Cable coaxial)**

Todo el conjunto está protegido por una cubierta exterior, también aislante, a la que por lo común se llama "jacket" (forro).

**FIBRA OPTICA**

El cable de fibra óptica se compone de una fibra muy delgada elaborada de dos tipos de vidrio o plástico con diferentes índices de refracción, uno para la parte interior y otro para la parte exterior evitando así la pérdida de la información. La fibra óptica, a su vez, se encuentra protegida por una placa aislante y protectora en la parte más exterior para darle mayor integridad estructural al cable. Es, sin embargo, extremadamente flexible ya que pueden realizar giros de hasta 360 grados sin problemas de afectación en el cable. Normalmente se utiliza por tres razones básicas:

- Para aquellos casos en donde las grandes distancias son un factor determinante para la implementación de una red local.
  - Cuando se requiere una alta capacidad de aplicaciones de comunicación.
  - Cuando el ruido o cualquier tipo de interferencia son factores a considerar.
- Para la transmisión de la información en redes locales via fibra óptica se utilizan una fibra como transmisor y otra como receptor. Es por esto que generalmente se producen en conjuntos de mínimo dos fibras por cable.

Existen numerosos tipos y variaciones de cable de fibra óptica, sin embargo, todos ellos contienen las mismas características de transmisión.

**CARACTERISTICAS DE LA FIBRA OPTICA**

- Cada fibra óptica provee un camino de transmisión único de extremo a extremo, unidireccional, es decir, half-duplex
- Los pulsos de luz se introducen en un extremo, usando un laser o LED.
- El tipo de enlace es punto a punto generalmente.
- El ancho de banda es mucho más alto que el de cualquier otro medio (aproximadamente 1Gbps a 20-30 Km., es decir, 60 veces más alto que el par trenzado, aproximadamente).
- Soporta transmisión de datos, voz y video.
- Debido a que durante la transmisión se presenta muy poca pérdida de señal se considera altamente confiable.

**VENTAJAS DE LA FIBRA OPTICA**

- Transmisión de voz, video y datos por el mismo canal.
- Aplicaciones de alta velocidad.
- No genera señales eléctricas o magnéticas.
- Inmune a interferencias y relámpagos.
- Puede propagar una señal sin necesidad de utilizar un amplificador a distancias de hasta 2000 metros
- Compatibilidad con Ethernet, Token Ring y FDDI (*Fiber Data Distributed Interface*; interfase de datos distribuidos por fibra)
- Excelente tolerancia a factores ambientales.
- Ofrece la mayor capacidad de adaptación a nuevas formas de rendimiento.

**DESVENTAJAS DE LA FIBRA OPTICA**

- Alto costo.
- No es tan fuerte y durable, en comparación con el par trenzado o el cable coaxial.
- Es difícil el modo de estandarización.

**FIBRA OPTICA**

Núcleo central, muy fino, de vidrio o plástico que tiene un alto índice de refracción.

Esta es la ruta por la cual una señal óptica es transmitida. Debe ser un material puro, dado que cualquier imperfección permitiría alguna distorsión de luz. Su diámetro puede estar dentro de un rango de 5 a 100 micras para la mayoría de fibras basadas en vidrio. Las fibras de plástico pueden estar entre 900 y 1000 micras.

#### **BUFFER INTERNO (Fibra óptica)**

Cubierta que rodea la fibra óptica, cuya propiedad de refracción es algo más baja que el núcleo central. Esta compuesto de vidrio o de plástico. La alta densidad del revestimiento sirve como un límite para las señales de luz que escapan del núcleo interno. Su diámetro está en el rango de 100 a 1000 micras.

#### **CUBIERTA EXTERIOR (Fibra óptica)**

Cubierta protectora que puede estar compuesta de muchos diferentes materiales que pueden ser: plástico, fibras de plástico trenzadas, blindaje de acero, acero trenzado y capa de teflón.

### **MICROONDAS**

Un sistema de microondas tiene dos partes básicas:

- **Unidad de procesamiento.-** Contiene un procesador de comunicaciones que es el responsable de generar e interpretar las señales de comunicación.
- **Unidad de transmisión.-** Es responsable de la transmisión y recepción de la señal.

Los sistemas de microondas usan implícitamente transmisión de radio-frecuencia (RF) entre dos estaciones, es decir, su transmisión es en forma muy semejante a cualquier otro medio de telecomunicaciones. En un sistema de microondas se usa el espacio aéreo como medio físico de transmisión. La información se transmite en forma digital a través de ondas de radio de muy corta longitud (centímetros). En estos sistemas es posible utilizar indistintamente enlace multipunto bien punto a punto. Cada sistema de microondas opera sobre un rango de frecuencias fijo que está entre 300 Mhz y 100 Ghz. Las distancias soportadas por un sistema de microondas varían ampliamente según la potencia de transmisión y la frecuencia de la señal.

Los puntos finales en un sistema de microondas deben tener línea de vista entre ellos para una comunicación apropiada. Las transmisiones de microondas más largas requieren del uso de estaciones de microondas intermedias comúnmente llamadas "repetidoras" (towers, torres).

### **INFRARROJO Y LASER**

La tecnología infrarrojo involucra la codificación de información en pulsos de luz infrarroja que es transmitida punto a punto entre dos entidades. Un sistema infrarrojo consiste de un transmisor de señales, un receptor de señales y una unidad de procesamiento. El uso de la luz infrarroja se puede considerar muy similar a la transmisión con microondas. En este caso el haz infrarrojo puede ser producido por un laser o un LED. Los dispositivos emisores y receptores deben estar ubicados "a la vista" uno del otro. La velocidad de transmisión llega a ser de hasta 100 kbps, a una distancia de hasta 16 Km. El tipo de enlace es punto a punto.

#### **VENTAJAS DEL MEDIO INFRARROJO-LASER**

- Inmunidad contra el ruido magnético, o sea, interferencia eléctrica.
- Arquitectura móvil.
- Diseño no terrestre.

#### **DESVENTAJAS DEL MEDIO INFRARROJO-LASER**

- El haz infrarrojo es afectado por el clima.
- Interferencia atmosférica y por obstáculos físicos.
- Distancia de transmisión limitada.
- Susceptible a interferencia de señales.
- Alto costo.
- Requerimiento de línea de vista.

### **SATELITES**

Actualmente es muy amplia la difusión del uso de satélites en redes de procesamiento de datos. El satélite de comunicaciones es un dispositivo que actúa principalmente como "reflector" de las emisiones terrenas. Los satélites reflejan un haz de microondas que transportan información codificada. Realmente la función de reflexión se compone de un receptor y un emisor que operan a diferentes frecuencias (el receptor a 6 Ghz y el emisor a 4Ghz, por ejemplo).

#### **Estaciones terrenas**

Anteriormente las estaciones usaban una antena del tipo "plato" de 10 m. de diámetro. Sin embargo, la reducción también llegaba a estos dispositivos y ahora miden 5 m. de diámetro y algunos llegan a medir 60 cm. de diámetro.

#### **SATELITE**

Físicamente los satélites giran alrededor de la tierra en forma sincrónica con ésta, a una altura de 35,680 km., en un arco directamente ubicado sobre el ecuador, para que el satélite pueda girar alrededor de la tierra en 24 hrs. Algo menos de la mitad del globo queda en el "cono de mira" de un satélite, con lo cual es obvia la importancia del alcance que tiene cada uno de estos dispositivos. el espaciamiento o separación entre dos satélites de comunicaciones es de 2880 km, que es equivalente a un ángulo de 4°, visto desde la tierra.

**CARACTERÍSTICAS DE ESTACIONES TERRENAS (Satélites)**

- Ubicables en oficinas o en domicilio particular.
- Eliminan las cargas de la conexión telefónica.
- Uso de microcomputadoras locales como inteligencia de control.
- Permiten el acceso "local" a archivos centralizados sin demoras producidas por compartir recursos.
- Una microestación se compone de:
  - Estación receptora.
  - Un segmento en el satélite.
  - Una estación emisora.

**INTERFASES**

Después de que el medio de transmisión ha sido seleccionado es necesario conectarlo al equipo de computadoras. Hay dos clases de equipos en comunicación de datos:

1. DCE (*Data Communications Equipment*, Equipo de comunicación de datos).
2. DTE (*Data terminal Equipment*, Equipo terminal de datos).

La interfase física es la forma con la cual se pueden conectar estas 2 clases de equipos.

La interfase entre DCE y DTE puede dividirse dentro de 4 aspectos:

- |  |  |
|--|--|
| <p><b>Mecánico.</b>- Tipo de conectores que se tienen que usar, número de pines de conexión, máxima longitud de cable permitida, etc</p> <p><b>Funcional.</b>- Se refiere a cuales señales son llevadas por cada pin ( temporizador, control, dato, etc)</p> | <p><b>Eléctrico.</b>- Línea de voltaje permitida, representación para varios niveles de voltaje, etc</p> <p><b>Procedural.</b>- Se define como las señales son intercambiadas y delimitadas al ambiente necesario para transmitir y recibir datos.</p> |
|--|--|

Interfases estandar:

RS-232-C

RS-449

**INTERFASE RS-232-C**

Esta es una de las interconexiones más difundida para enlazar equipos en transmisores de datos. Su nomenclatura internacional es CCITT V.24.

Consiste en la disposición de 25 circuitos de intercambio con una función en cada uno. Se implementa en un enchufe de 25 clavijas, de corte trapezoidal, para evitar un mal acoplamiento, que se asegura mediante dos tornillos, uno a cada lado.

Permite una velocidad máxima de 20 Kbps a una distancia máxima de 15 m.

No tiene prueba de mantenimiento

No especifica el tamaño o tipo de conector a usar en la interfase, siempre y cuando éste defina 25 señales, 3 de ellas sin asignar, 2 reservadas para prueba y las 20 restantes usadas para datos, control, temporización, etc.

**ANÁLISIS DE LAS PATILLAS DEL RS-232-C**

No. Patilla	EIA	CCITT	Descripción de la señal	Abreviatura común	De DCE	a DCE
1	AA	101	Tierra de protección (Chasis)	-	-	-
2	BA	103	Transmisión de datos	TD	-	X
3	BB	104	Recepción de datos	RD	X	-
4	CA	105	Petición de emisión	RTS	-	X
5	CB	108	Preparado para transmisión	CTS	X	-
6	CC	107	Equipo de datos preparado	DSR	X	-
7	AB	102	Tierra de señal o retorno común	SG	X	X
8	CF	109	Detector de señal de línea recibida	DGD	X	-
9	-	-	Voltaje (+)	-	-	-
10	-	-	Voltaje (-)	-	-	-
11	-	-	Sin asignar	-	-	-
12	SCF	122	Detector de señal de línea recibida secundaria	-	X	-
13	SCB	121	Preparado para transmisión secundaria	-	X	-
14	SBA	118	Transmisión de datos secundaria	-	X	-
15	DB	114	Sincronismo en transmisión por (DCE)	-	X	-
16	SBB	119	Recepción de datos secundaria	-	X	-
17	DD	-	Sincronización de recepción	-	X	-
18	-	115	Sin asignar	-	-	X
19	SCA	120	Petición de emisión secundaria	-	-	X
20	CD	108/2	Terminal de datos preparado	DTR	X	-
21	CG	110	Detector de calidad de la señal	SQ	X	-
22	GE	125	Indicador de señal de llamada	RI	-	X
23	CH-CI	111-112	Selector de velocidad binaria (DTE-DCE)	-	X	-
24	DA	113	Sincronismo de transmisión (por DTE)	-	-	X
25	-	-	Sin asignar	-	-	-

INTERFASE RS-449

Esta es la forma reemplazante de la RS-232-C, para redes analógicas con aplicaciones a largas distancias y altas velocidades. Se caracteriza por tener una función por circuito de intercambio, una velocidad máxima de 2 Mbps a una distancia de 1200 m.

No es una norma completa en si misma. Se complementa con los RS-442 / 423A.

La RS-422 especifica las características eléctricas para circuitos balanceados.

La RS-423-A especifica las características eléctricas para circuitos desbalanceados.

Dispone de un conector de 37 clavijas para dar cabida a más circuitos individuales que la RS-232-C, por lo tanto es más costosa y compleja.

ANALISIS DE LAS PATILLAS DEL RS-449

No. Patilla	Señal	Función	No. Patilla	Señal	Función
1	Pantalla	-	20	RC	Común de recepción
2	SI	Indicador de velocidad de señalización	21	Reservada	-
3	Reservada	-	22	SDb	Envío de datos (b)
4	SDa	Envío de datos (a)	23	STb	Temporización de datos (b)
5	STa	Temporización de envío (a)	24	RDb	Recepción de datos (b)
6	DRa	Recepción de datos (a)	25	RSb	Solicitud de envío (b)
7	RSa	Solicitud de envío (a)	26		Temporización de recepción (b)
8	RTa	Temporización de recepción	27	CSb	Listo para envío (b)
9	CSa	Listo para envío (a)	28	IS	Terminal en servicio
10	LL	Retorno local	29	DMb	Modo de datos (b)
11	DMa	Modo de datos (a)	30	TRb	Terminal listo (b)
12	TRa	Terminal listo (a)	31	RRB	Receptor listo (b)
13	RRa	Receptor listo (a)	32	SS	Selección de Standby
14	RL	Retorno remoto	33	SQ	Calidad de señal
15	IC	Entrada de llamada	34	NS	Nueva señal
16	SF/SR	Selección/secuencia/velocidad	35	TT	Temporizador de terminal
17	TTa	Temporizador de terminal	36	SB	Indicador de Standby
18	TM	Modo de prueba	37	SC	Común de envío
19	SG	Tierra de señal			

TECNICAS DE COMUNICACION

Para que dos dispositivos ligados por un medio de transmisión puedan intercambiar datos, se requiere de un alto grado de cooperación. Típicamente, los datos son transmitidos por bits sobre el medio, uno a la vez. La duración de esos bits debe ser la misma para transmitir que para recibir. A este proceso se le conoce con el nombre de sincronización. Las dos técnicas más comunes son:

- Transmisión asíncrona
- Transmisión síncrona

TRANSMISION ASINCRONA

A este tipo de transmisión también se le conoce con el nombre de *transmisión start-stop*. Esta técnica de transmisión es una solución a la tarea de sincronización y consiste en enviar muy pequeños bloques de bits y sincronizar el inicio de cada bloque. La transmisión asíncrona está orientada a carácter y los caracteres están formados de 5 a 8 bits.

Su funcionamiento es muy sencillo: Cuando todavía no se inicia la transmisión de un carácter, la línea entre el transmisor y el receptor se encuentra en un estado "de espera" (*idle*) que es el equivalente para el elemento de señalización con el "1" binario. El inicio del carácter es señalado por un bit de inicio con el valor binario "0". A esto le sigue el carácter que consta de 5 a 8 bits y en algunos casos, un bit de paridad que es usado por el receptor para el proceso de detección de error. Este último bit del carácter es seguido por un bit de "paro", el cual es un "1" binario. La longitud mínima para el bit de paro es especificado y puede ser 1, 1.5 ó 2 veces la duración de un bit ordinario. Se puede notar que el bit de "paro" es el mismo que el estado "idle", por lo tanto el transmisor seguirá enviando bits de paro hasta que se envíe el próximo carácter.

VENTAJAS DE LA TRANSMISION ASINCRONA

- Es simple y de bajo costo.
- No se requiere la misma sincronización de los relojes del receptor y del transmisor.

DESVENTAJAS DE LA TRANSMISION ASINCRONA

- Se requieren de 2 a 3 bits adicionales por carácter, por lo que se impide el uso completo de la capacidad de la línea para la transmisión de información usual.
- Entre más largo sea el bloque de bits existe mayor probabilidad de que se cause un error de recepción.

TRANSMISION SINCRONA

Esta es una de las técnicas más eficientes en comunicaciones. En este modo los bloques de caracteres o bits son transmitidos sin los bits de "inicio" y de "paro" y el tiempo de llegada o el de salida exacto de cada bit es predecible. Para coordinar el flujo entre el transmisor y el receptor, sus relojes deben ser sincronizados. Una posibilidad es el contar con un reloj separado en la línea entre el transmisor y el receptor. Por otro lado, la información del reloj debe estar contenida en la señal del dato.

Con la transmisión síncrona, se requiere de otro nivel de sincronización, que permita que el receptor determine el inicio y el fin del bloque de datos. Para anexar esto, cada bloque inicia con un patrón de bits (preámbulo) y termina con un patrón de bits (postámbulo). Estos patrones son información de control más que datos. Los datos anexados junto con la información de control reciben el nombre de "frame".

El formato exacto de un frame depende de la forma de transmisión:

- Si es orientada a bit
- Si es orientada a carácter

### TRANSMISION SINCRONA ORIENTADA A CARACTER

Con la transmisión orientada a carácter, el bloque de datos es tratado como una secuencia de caracteres (usualmente caracteres de 8 bits). Toda la información de control está en forma de carácter. El frame inicia con uno o más caracteres de sincronización. El carácter de sincronización, usualmente llamado SYN, es un patrón de bits único que señala al receptor que éste es el inicio del bloque. El postámbulo es el otro carácter único usado en algunas formas. El receptor es puesto en alerta cuando recibe los caracteres SYN y recibe datos hasta que el carácter postámbulo es detectado, entonces el receptor espera el próximo patrón de caracteres SYN. Alternativamente, se puede incluir dentro de la información de control de la longitud del frame y evitar el carácter postámbulo.

### TRANSMISION SINCRONA ORIENTADA A BIT

Con la transmisión orientada a bit, el bloque de datos es tratado como una secuencia de bits no se requiere de información de control para ser interpretados en unidades de caracteres de 8 bits. Se requiere de un patrón de bit especial que señale el inicio de un bloque. Su preámbulo tiene una longitud de 8 bits y es referido como una bandera. La misma bandera es usada como postámbulo. El receptor observa en la ocurrencia de la bandera patrón para señalar el inicio del frame. Esta es seguida por algún número de campo de control, luego un campo de datos de longitud variable, más campo de control y finalmente la bandera es repetida. La diferencia con la transmisión orientada a carácter consiste en el formato y la Interpretación de la información de control.

#### CARACTERISTICAS DE LA TRANSMISION SINCRONA

- Los datos se almacenan temporalmente en un buffer antes de su transmisión. Cuando todo el bloque está listo se intenta su envío.
- Los datos se transfieren en bloques y no en carácter.
- Existe un esquema definido y uniforme para la transmisión de los bits del mensaje.

#### VENTAJAS DE LA TRANSMISION SINCRONA

- No se usan bits de "inicio" y de "paro", por lo que el largo total es generalmente menor
- Usualmente permite mayores velocidades que la transmisión asíncrona
- Sólo se requiere de un 4% ó 5% del largo del bloque para los bits de encabezado

#### DESVENTAJAS DE LA TRANSMISION SINCRONA

- Se requiere de la sincronización de los relojes del transmisor y del receptor.

### TECNICAS DE DETECCION DE ERRORES

En una comunicación a través de un medio real existe una cierta probabilidad de que ocurran errores por diversas causas, normalmente debidas a dos tipos de fenómenos:

- Interferencias electromagnéticas que producen ruido en el medio físico.
- Incorrecto funcionamiento del modem.

Es por esto que sobre muchos sistemas de transmisión de datos la eliminación de errores es vitalmente importante. Hay dos caminos básicos para controlar los errores:

1. Detectar el error y retransmitir el mensaje erróneo.
2. Utilizar la forma de corrección de errores sin retransmisión.

Independientemente el tipo de control que se utilice, debe realizarse el proceso de detección de error. Dentro de las técnicas más usuales para la detección de errores se encuentran:

- Chequeo de paridad vertical (VRC)
- Chequeo bidimensional
- Chequeo de paridad longitudinal (LRC)
- Chequeo polinomial o de redundancia cíclica (CRC)

#### CHEQUEO DE PARIDAD VERTICAL (VRC)

Es un método simple aplicable a nivel de byte. Su uso está directamente relacionado con la codificación en ASCII. Por lo tanto si se define un carácter igual a un byte, podemos disponer de 7 bits para datos y un bit para control. Dicho bit de control representará la paridad del byte. Definiremos la paridad de carácter de dos formas:

- Paridad par.- Cuando el número total de bits que coinciden con "1" en el byte es par.
- Paridad impar.- Cuando el número total de bits que coinciden con "1" en el byte es impar.

En el momento de la transmisión, el extremo emisor calcula el bit de paridad de cada byte y lo anexa a los datos. El receptor recalcula la paridad del byte recibido y lo compara con el criterio utilizado. Es evidente que éste método no asegura que no hayan ocurrido errores. Basta que cambien su valor dos bits de datos simultáneamente para que la paridad sea correcta pero el dato no. Sin embargo, VRC disminuye la probabilidad de que el dato final sea erróneo.

**CODIFICACION EN ASCII**

El método de codificación de símbolos ASCII, utiliza los valores binarios que se pueden obtener con 7 bits para representar cualquier dato. Por lo tanto el máximo valor representable es 7F (hex)=127 (dec), es decir, se tienen 128 posibilidades distintas de representación ( de 0-127).

**VENTAJAS DE VRC**

- El cálculo de la paridad de los bits es muy sencillo.

**DESVENTAJAS DE VRC**

- Se ocupa demasiado espacio para el bit de paridad.

**CHEQUEO DE PARIDAD LONGITUDINAL (LRC)**

Si en lugar de considerar 7 bits como el dato a transmitir en el momento de calcular la paridad, consideramos un conjunto de caracteres (un bloque) con sus bits de VRC y sobre éstos calculamos la paridad, estaremos usando chequeo de paridad longitudinal (LRC).

Su funcionamiento es el siguiente:

- 1 Se toma el *i*-ésimo bit de cada byte ( $i=1..8$ ).
- 2 Se calcula el bit de paridad a partir de los bits tomados.
- 3 El bit resultante se coloca en el *i*-ésimo lugar ( $i=1..8$ ), de un carácter que se denomina BCC (*Block Check Character*).

Como se puede observar cuando se utiliza LRC, se agrega un carácter al final del mensaje que contiene todos los bits de paridad calculados.

**VENTAJAS DE LRC**

- Es más rápido debido a que se toman bloques en lugar de caracteres.

**DESVENTAJAS DE LRC**

- Se requiere de más espacio para el carácter BCC que contiene los cálculos de paridad.

**CHEQUEO BIDIMENSIONAL**

El chequeo bidimensional es el uso combinado de los métodos de chequeo vertical (VRC) y longitudinal (LRC). Con el vertical se obtiene la abscisa y con el longitudinal se obtiene la ordenada del punto "bit erróneo", con una gran probabilidad de acierto.

**VENTAJAS DE CHEQUEO BIDIMENSIONAL**

- Indica con exactitud que bit o bits son erróneos.

**DESVENTAJAS DE CHEQUEO BIDIMENSIONAL**

- Ocupa el doble de espacio de LRC para el cálculo de bits de paridad.

**CHEQUEO POLINOMIAL O REDUNDANCIA CICLICA (CRC)**

El método de redundancia cíclica CRC (*Cyclic Redundancy Check*) es otra técnica muy usada para detección de errores. Trabaja a nivel de mensajes agregando varios caracteres de control al final, siendo lo más común 2 ó 4 bytes de control.

El funcionamiento consiste en dividir la secuencia de bits a enviar por un número binario predeterminado. El resto de la división se adiciona al mensaje como secuencia de control.

Por una regla aritmética simple, si el divisor es un número de 16 bits, podemos tener la certeza que el residuo siempre podrá almacenarse en 2 bytes, en donde, agregando 2 caracteres a nuestro mensaje tendremos el método implementado. El extremo realiza el mismo cálculo que el emisor y compara el resultado obtenido con la secuencia de control recibida. Si no coinciden equivale a una indicación de error.

**VENTAJAS DE CRC**

- Ocupa menos espacio para detectar el error.

**DESVENTAJAS DE CRC**

- El cálculo que requiere es más complejo, comparado con VRC y LRC.

**CONTROL DE ENLACE DE DATOS**

El control de enlace de datos proporciona líneas de servicio que manejan el flujo de datos a través de una ruta de comunicaciones o enlace entre computadoras y terminales. Sus funciones son limitadas al enlace únicamente, es decir, el control de enlace es responsable sólo por el tráfico entre nodos adyacentes sobre la línea. La tarea del control de enlace inicia en el momento de que el dato es transmitido al nodo adyacente y termina cuando el transmisor recibe un reconocimiento de dicha transmisión. El control de enlace de datos proporciona las siguientes funciones:

1. Controlar la emisión y recepción de información en un enlace de comunicación.
2. Sincronizar la operación del transmisor con la operación del receptor.
3. Detectar y recuperar los errores entre dos puntos sobre el enlace.

Para llevar a cabo estas funciones, el control de enlace de datos establece los siguientes procedimientos:

- Un formato general de información
- Modos de señalización
- Control de flujo
- Configuraciones físicas de enlace
- Configuraciones lógicas de enlace
- Corrección de errores

## FORMATO GENERAL DE INFORMACION

El control de enlace de datos es transparente para un tipo especial de código, tal como el ASCII o el EBCDIC, para control y transferencia de mensajes de datos. Con propósitos de control se utiliza un grupo limitado de patrones de bits. La serie de bits que se transfieren entre estaciones de un enlace se ensamblan en grupos llamados "frames" (tramas). Cada frame se compone de cinco campos cuando sólo se está transfiriendo información de control de enlace. Se agrega un sexto campo cuando están transfiriendo datos. Los campos son los siguientes:

### SECUENCIA INDICADORA DE COMIENZO (Campo de frame)

Campo de 8 bits que identifica el comienzo del "frame". El patrón de bits indicador se reconoce en cualquier momento sin tener en cuenta la sincronización entre caracteres. La secuencia indicadora de comienzo tiene los siguientes usos:

- Se usa como referencia de la posición del campo de direccionamiento y el campo de control.
- Inicia la verificación de error de transmisión para el resto del frame.
- Se envía la secuencia indicadora en forma continua, para mantener el control del canal de comunicación cuando no se tienen frames disponibles para envío.

### DIRECCIONAMIENTO (Campo de frame)

Campo de 8 ó 16 bits. El contenido del campo de direccionamiento, ya sea ampliado o no, se usa para identificar:

- Una estación secundaria específica
- Un grupo de estaciones secundarias en el mismo enlace
- Todas las estaciones secundarias en el enlace

### DIRECCIONAMIENTO ESPECIFICO DE UNA ESTACION (Campo de direccionamiento)

Cada estación secundaria en un enlace tiene una dirección específica. Una estación secundaria puede recibir frames con el direccionamiento específico de alguna estación o con un direccionamiento común (en grupo o global), pero todos los frames enviados por una estación secundaria deben tener el direccionamiento específico de esa estación secundaria en el campo de direccionamiento.

### DIRECCIONAMIENTO EN GRUPO (Campo de direccionamiento)

Es un tipo de direccionamiento opcional usado por una estación primaria para enviar un frame a más de una estación secundaria en un enlace, pero no a todas. Puede ser útil cuando un grupo de estaciones secundarias necesitan realizar la misma función al mismo tiempo o necesitan recibir el mismo campo de información. Cuando una estación secundaria responde al frame que contiene un direccionamiento en grupo, el campo de direccionamiento de respuesta contiene el direccionamiento específico de la estación secundaria.

### DIRECCIONAMIENTO GLOBAL (Campo de Direccionamiento)

Todas las estaciones secundarias pueden reconocer el direccionamiento global. El direccionamiento global que está formado por 8 bits en "1", es usado por las estaciones primarias para enviar comandos o datos a todas las estaciones secundarias en el enlace, al mismo tiempo. Cuando una estación secundaria responde al frame que contiene un direccionamiento global, el campo de direccionamiento de respuesta contiene el direccionamiento específico de la estación secundaria.

### CONTROL (Campo de frame)

Campo de 8 ó 16 bits que contiene los comandos y respuestas requeridas para controlar la operación en un enlace de datos. Determina cómo se controla el proceso de comunicación. Este campo puede estar en uno o tres formatos:

- Información
- Supervisor
- Transferencia no numerada

### FORMATO DE INFORMACION (Campo de control)

Es usado para transmitir datos a un usuario final entre los dos dispositivos.

N(S) = Número de secuencia de envío  
N(R) = Número de secuencia de recibido

### FORMATO SUPERVISOR (Campo de control)

En éste caso se ejecutan funciones de control tales como el reconocimiento de frames, la respuesta de retransmisión de frames o suspensión temporal, etc.

N(R) = Número de secuencia de recibido  
SC = Código Supervisor  
P/F = Bit Poll/Final

**FORMATO NO NUMERADO (Campo de control)**

Es también usado con el propósito de control. Es usado para ejecutar inicialización de enlaces, desconexión de enlaces y otras funciones que controlan el enlace.

P/F = Bit Poil/Final

UN = Código no numerado

**INFORMACION (Campo de frame)**

Este campo contiene los datos que se están transfiriendo. Es de longitud variable dependiendo de las consideraciones del sistema tales como longitud normal de mensaje y velocidad de transferencia. Este campo no es usado en los frames en los que sólo contienen información de control de enlace.

**SECUENCIA DE VERIFICACION DE FRAME (Campo de frame)**

Se usa para asegurar que los datos enviados de una estación sean recibidos sin error por la estación receptora. Es un campo de 16 ó 32 bits que es el resultado de un proceso aritmético realizado sobre los campos de direccionamiento, de control y de Información (si lo hay).

**SECUENCIA INDICADORA DE FIN (Campo de frame)**

Campo de 8 bits que identifica el fin del frame. La confirmación de bits para la secuencia indicadora de comienzo y de fin es la misma. La secuencia indicadora de fin tiene los siguientes usos:

- Termina la verificación para los errores de transmisión.
- En algunos casos puede ser la indicadora de comienzo del frame siguiente.

**CONFIGURACION FISICA DE ENLACE**

Los enlaces físicos de datos se identifican por:

1. El tipo de circuito:
  - Half duplex
  - Full duplex
2. Los estados y modos de operación de la estación:
  - Modo de respuesta normal (NRM)
  - Modo de inicialización (IM)
  - Modo desconectado asincrónico (ADM)
  - Modo balanceado asincrónico (ABM)
  - Modo de desconectado normal (NDM)
  - Estado lógico desconectado (LDS)
3. La configuración del canal:
  - Configuración balanceada
  - Configuración desbalanceada

**MODO DE RESPUESTA NORMAL (NRM)**

El modo de respuesta normal (NRM, *Normal Response Mode*) es el modo de operación para estaciones primarias y secundarias que operan en el estado de transferencia de información (ITS, *Information Transfer State*). Permiten que las estaciones primarias controlen las transmisiones desde las estaciones secundarias, en enlaces multipunto. Una estación secundaria que opera en NRM no puede enviar frames a una estación primaria, hasta que la estación primaria se lo haya pedido.

**MODO BALANCEADO ASINCRONO (ABM)**

El modo balanceado asincrónico (ABM, *Asynchronous Balanced Mode*) es un modo de operación para estaciones combinadas que operan en el estado de transferencia de información (ITS, *Information Transfer State*).

El modo balanceado asincrónico se utiliza en enlaces punto a punto cuando las 2 estaciones en el enlace son estaciones combinadas. Cuando una estación combinada opera en ABM, es capaz de realizar funciones tales como las de las estaciones primarias y secundarias, es decir, de enviar comandos y recibir respuestas de la otra estación combinada.

**MODO DE INICIALIZACION (IM)**

Una estación secundaria es colocada en el modo de inicialización (IM, *Initialization Mode*) por una estación primaria, para cargar a la estación secundaria con software de operación. El modo de inicialización no se considera un modo de transferencia de información. Una estación que opera en IM se considera que está en el estado de inicialización (IS, *Initialization State*).

**MODO DESCONECTADO NORMAL (NDM)**

En el modo desconectado normal (NDM, *Normal Disconnected Mode*) una estación secundaria no puede enviar una respuesta hasta que reciba un pedido de transmisión de la estación primaria.

**MODO DESCONECTADO ASINCRONO (ADM)**

En el modo desconectado asincrónico (ADM, *Asynchronous Disconnected Mode*) una estación secundaria o combinada puede responder, sin recibir primero un pedido de transmisión de la estación primaria u otra estación combinada.

**ESTADO LOGICO DESCONECTADO (LDS)**

Cuando una estación secundaria o combinada se encuentra en estado lógico desconectado (LDS, *Logical Disconnected State*) la estación está desconectada lógicamente del enlace y no es capaz de transferir información del usuario. El estado LDS se presenta en los siguientes casos:

- Inicialmente la estación es encendida
- La energía de la estación ha sido temporalmente interrumpida.
- Los componentes físicos de la estación son reacondicionados manualmente.
- La estación cambia de una condición de espera a una condición de enlace de operación.
- La estación es colocada en estado desconectado por un comando de desconexión.
- La estación no es inicialmente capaz de entrar en estado de transferencia de información cuando sale de estado de inicialización.

### CONFIGURACION BALANCEADA

Un canal punto a punto balanceado puede tener sólo dos estaciones combinadas en el enlace donde cada estación combinada tiene igual responsabilidad para el control de enlace. Esto es, cualquiera de las dos estaciones puede realizar la función de una estación primaria, mientras la otra estación realiza las funciones de una secundaria. El enlace punto a punto balanceado puede tener circuitos "full-duplex" o "half-duplex".

### CONFIGURACION DESBALANCEADA

La configuración desbalanceada contiene por lo menos dos estaciones, siendo una de ellas responsable del control de enlace y transmisión de frames. Un enlace punto a punto desbalanceado tiene sólo dos estaciones: una estación primaria y una secundaria. Las estaciones pueden estar conectadas ya sea por circuitos "full-duplex" o "half-duplex". Un circuito punto a punto desbalanceado conecta una estación primaria con dos o más secundarias. Las estaciones secundarias están conectadas en forma paralela. Las estaciones pueden estar conectadas tanto por un circuito "full-duplex" como por un "half-duplex".

### MODOS DE SEÑALIZACION

Algunos de los métodos más usados de codificación de datos son los siguientes:

- **No retorno a cero (NRZ)** Usado cuando se emplean modems síncronos, o con enlaces "sin modems" si se dispone de una fuente externa de sincronización.
- **No retorno a cero Invertido (NRZI)** Usado cuando se emplean modems asíncronos, o con enlaces "sin modems" si no se dispone de fuente de sincronización externa.
- **Manchester** Usado para enlace dentro del dominio.

En cada caso, la técnica de codificación determina el formato, los datos que aparecen en la salida del adaptador-emisor y la entrada del adaptador-receptor.

#### SEÑALIZACION NRZ (NO RETORNO A CERO)

Con este método de codificación, la señal de datos se fija en un caso para representar un bit "0", y en otro caso para representar un bit "1". La señal de datos no cambia si se debe representar una sucesión de bits similares.

La codificación NRZ funciona bien con modems síncronos dado que los modems desarrollarán y sincronizarán las señales de sincronización de recepción. En caso de que el adaptador receptor tenga que suministrar la sincronización surge un problema. El adaptador debe ver cambios periódicos en la señal de datos para sincronizar su pulso con el comienzo de un tiempo de bit. Si el campo de información del frame contuviera una larga sucesión de bits "0", la señal de datos no cambiaría estados y se podría perder la sincronización del pulso. Si el campo de información contiene una sucesión de bits "1", la técnica de inserción de ceros proporciona los cambios periódicos que serían necesarios.

#### SEÑALIZACION NRZI (NO RETORNO A CERO INVERTIDO)

El objetivo de esta técnica de codificación es eliminar el período sin transición extendido, si el campo de información contiene sucesivos bits "0".

El estado "alto" o "bajo" de la señal de datos no significa nada en la codificación NRZI. La señal siempre cambiará estados para representar un bit "0" y no cambiará para representar un bit "1". Este método mantendrá la sincronización del pulso del adaptador-receptor una vez que se haya establecido. El cambio en el estado de la señal que se produce por cada bit "0" proporciona un medio para ajustar automáticamente la sincronización de datos de la estación receptora durante la transmisión del frame.

#### SEÑALIZACION MANCHESTER

La codificación Manchester, que se usa para enlaces en el domicilio, elimina por completo los períodos sin transición.

Este método de codificación proporciona un cambio por cada bit de información. La polaridad o dirección del cambio identificará el bit como "1" o "0". Una transición de dirección positiva representará un bit "0" y una transición de dirección negativa representará un bit "1". Si se deben transmitir dos bits "similares" en sucesión tendrá lugar una transición extra en el medio del tiempo de un bit. Esta transición de fase extra no será reconocida como datos por el receptor.

### CONFIGURACIONES LOGICAS DE ENLACE

Las configuraciones lógicas más utilizadas son:

#### PUNTO A PUNTO                      MULTIPUNTO

- Con estaciones secundarias individuales
- Con estaciones secundarias agrupadas y continuidad entre canales externos e internos
- Con estaciones secundarias agrupadas sin continuidad entre canales externos e internos

## CONTROL DE FLUJO

El control de flujo es una técnica para asegurar que una entidad transmisora no sature a una entidad receptora con datos. Típicamente la unidad receptora contiene un buffer de datos con suficiente capacidad. Cuando los datos son recibidos esta debe hacer un pequeño procesamiento antes de pasar el dato a la capa superior del usuario. En ausencia de control de flujo, el buffer del receptor puede saturarse y desbordarse mientras se está procesando un dato anterior. Dentro de las técnicas más utilizadas para el control de flujo se encuentran:

1. "Alto y espera"

2. "Ventanas deslizantes"

### CONTROL DE FLUJO: ALTO Y ESPERA.

La forma más simple de control de flujo se conoce como "alto y espera".

Su funcionamiento es el siguiente:

1. Una entidad fuente transmite una entidad de dato.
2. Después de la recepción, la entidad de destino indica su disponibilidad para aceptar otra unidad de dato por medio del envío del reconocimiento de la unidad de dato recibida.
3. La entidad fuente debe esperar hasta recibir el reconocimiento antes de enviar la próxima unidad de datos.

Una de las tareas del transmisor es decidir cuando un mensaje debe ser enviado como un bloque continuo o como unidad de datos (según el protocolo empleado). Sin embargo en este caso la entidad transmisora divide un bloque largo de datos en bloques más pequeños y los envía uno a la vez. Esto es aceptado por las siguientes razones:

- En una transmisión muy larga es muy común la ocurrencia de algún error y en caso de que se diera se requiere de la transmisión del bloque completo.
- En bloques más pequeños es menos común que la ocurrencia de algún error y en caso de que se diera se retransmitiría el dato necesario.

### CARACTERÍSTICAS DE ALTO Y ESPERA

- En este caso la entidad de destino puede parar el flujo de datos por simplemente detener el reconocimiento.
- En enlaces multipuntos, es usual que no se permita a una estación ocupar la línea por períodos muy largos, dado que esto ocasiona grandes retardos para las demás estaciones.
- El tamaño del buffer del receptor puede ser limitado.
- Con el uso de múltiples unidades de datos para un mensaje simple, la técnica "alto y espera" puede ser inadecuada, dado que sólo una unidad de datos a la vez puede estar en tránsito.

### CONTROL DE FLUJO: VENTANAS DESLIZANTES.

En esta técnica si es posible que múltiples unidades de datos se encuentren en tránsito a la vez. Su funcionamiento es el siguiente:

Se supone que se tienen 2 estaciones "A" y "B" conectadas por medio del enlace full-duplex. La estación "B" contiene 7 buffers para recepción de unidades de datos y "A" está capacitada para enviar 7 unidades de datos sin necesidad de esperar algún reconocimiento. Para saber cuales unidades de datos han sido reconocidos, a cada uno se le etiqueta con una secuencia de números del rango del 0-7 (se le denomina del módulo 8). "B" indica que reconoce una unidad de dato al enviar un reconocimiento que incluye el número de la secuencia de la próxima unidad de datos especificada. Es decir, si "B" regresa el número de secuencia 5, éste indica que ha recibido la unidad de datos con número 4 y declara que "B" ahora está esperando la unidad de datos con número 5. Esta forma puede ser usada para reconocer múltiples unidades de datos.

Por ejemplo: "B" puede recibir unidades de datos con los números 2, 3 y 4, y detener el reconocimiento hasta que llegue la unidad de datos 4, para luego enviar el número de secuencia 5, o sea, "B" reconoce las unidades de datos 2, 3 y 4 a la vez. "A" mantiene una lista de números de secuencia que éste tiene permitido enviar y "B" mantiene una lista de números de secuencia que esta preparada a recibir. A cada una de las listas se les puede asemejar como ventanas de unidades de datos.

## TECNICAS DE CORRECCION DE ERRORES

Existen diversos métodos para la corrección de errores como:

### Métodos de corrección de repetición automática (ARQ, Automatic Request for Repeat)

- ARQ: "pare y espere"
- ARQ: "continuo"

### Métodos de autocorrección (FEC, Forward Error Correction)

- Convolutional
- Corrección hacia adelante por bloques

### MÉTODOS DE CORRECCION DE REPETICION AUTOMATICA (ARQ)

Todas las formas de corrección ARQ usan el fenómeno de reenvío del mensaje para intentar subsanar el problema. Dentro de los métodos de corrección de repetición automática se encuentran:

#### Desventajas:

- Pérdida de tiempo
- Sobrecarga de las líneas
- Determinación del criterio de selección del número de retransmisiones

**ARQ: " PARE Y ESPERE "**

Esta es una forma muy usada para recuperar los datos después de que se detectó un error y consiste en:

1. Transmitir un mensaje
2. Detenerse
3. Esperar una respuesta
4. Accionar según la respuesta:
  - Retransmitir (en caso de error)
  - Continuar con el siguiente mensaje ( si no hay error)

Existen caracteres de control en el código ASCII destinados para tales efectos.

El número de retransmisiones normalmente es un parámetro programable en los adaptadores de comunicaciones o en el software central.

**CARACTERES DE CONTROL EN EL CODIGO ASCII**

CARACTER	NOMBRE	PROPOSITO
ACK	Reconocimiento positivo	1) Paridad correcta 2) Listo para recibir mensajes
NAK	Reconocimiento negativo	1) Paridad incorrecta 2) No listo para recibir
EOT	Fin de transmisión	1) Fin de la secuencia de transmisión ;desconexión 2) No hay mensaje listo para envío
ENQ	Preguntar	1) Usado luego de la expiración de un periodo de tiempo (tiempo cumplido), para efectos de requerir la respuesta adecuada

**ARQ: " CONTINUO "**

Se utiliza con modalidad de transmisión Full-duplex. En este caso la unidad transmisora no necesita esperar un reconocimiento para después enviar el bloque, sino que inmediatamente envía el siguiente bloque. Mientras los bloques son inicialmente transmitidos, la cadena de reconocimientos es examinada por la unidad de transmisión. Cuando un reconocimiento negativo es recibido la unidad determinará cual bloque fue incorrecto. Los bloques deben ser anteriormente numerados. El reconocimiento negativo continuará con el número del bloque al que está referido para que la unidad transmisora pueda identificarlo. Existen diversas formas de implementar el método ARQ continuo entre ellas están:

- Retroceda 2
- Retroceda n
- Repetición selectiva

**RETROCEDA 2 (ARQ: Continuo)**

Se envía una respuesta de reconocimiento por cada 2 mensajes transmitidos. Es decir, mientras se está enviando una, se está reconociendo por la otra vía el anterior, con el consiguiente ahorro del tiempo.

**RETROCEDA N (ARQ: Continuo)**

Es el método utilizado en los protocolos orientados a bit. Se establece a priori un módulo que indica cada cuantos mensajes transmitidos se va a enviar una respuesta de reconocimiento positivo. En caso de un error en los datos recibidos, se pide que se retransmita la secuencia a partir del mensaje  $x$  retrocediendo  $n = m - x$ , con  $m = \text{módulo}$

**REPETICION SELECTIVA (ARQ: Continuo)**

En esta forma de corrección en lugar de solicitar la repetición parcial o total de una secuencia de mensajes, se pide la retransmisión de una en particular "seleccionándola" por un número correlativo, dentro de la secuencia recibida.

**METODOS DE AUTOCORRECCION (FEC)**

En general hace referencia a un método de codificación de datos, para efectos de asegurar una mayor confiabilidad a los mensajes transmitidos.

El método de autocorrección a diferencia del de repetición, no requiere reconocimientos ni retransmisión de mensajes.

Con el aumento del uso de canales de alta velocidad, el efecto negativo debido al uso de un gran número de bits redundantes, se ve disminuido, al punto de que su "mala influencia" en los tiempos de respuesta, es un precio muy bajo comparado al beneficio que representa. La sobrecarga provocada oscila entre 7% y 50%.

**FEC: CONVOLUCIONAL**

Mediante la codificación convolucional (o corrección de errores hacia adelante por repliegue), cada bit de una secuencia del usuario es comparada con uno o más bits enviados inmediatamente antes. El valor de cada bit es ligado con el valor de otros bits. Además un bit redundante es agregado en cada grupo de bits comparado de ésta manera.

## REDTUT: CAPITULO 1

Cuando un bit es comparado solamente con el bit que precede, el número de bits redundantes requerido para asegurar la decodificación en el receptor es muy alto, aunque la complejidad del procesamiento es minimizada.

Inversamente cuando el bit es comparado con un gran número de bits previamente transmitido, el número de bits redundantes es minimizado pero la complejidad de procesamiento en ambos extremos es más alta.

### FEC: HACIA ADELANTE POR BLOQUES

A diferencia del esquema convolucional, bloques enteros de datos se cargan en registros, donde se procesan como un todo.

Bits redundantes son agregados basándose en el contenido del bloque completo. Según la codificación de bloque parece agregar menos sobrecarga de bits redundantes, a ser enviados como datos del usuario.

Debido a la forma de transmitir en bloques, los bloques del usuario resultan más largos que con esquemas convolucionales. La "duración de un bit" es la forma estándar de medir esta demora, y es transparente de la velocidad de frecuencia empleada.

### MULTIPLEXAJE

La tecnología del multiplexaje permite la transmisión de múltiples señales sobre un mismo enlace. En un sistema de multiplexaje dos o más señales son combinadas para que puedan ser transmitidas juntas sobre un medio físico guiado o no guiado. Los dos tipos de multiplexaje principales son:

- Multiplexaje por división de frecuencia (FDM)
- Multiplexaje por división de tiempo (TDM)

Existen dos razones principales para usar multiplexaje:

- 1) El multiplexaje puede ser usado para garantizar que el medio de transmisión se está operando en su tasa de información más económica. La mayoría de los medios de transmisión, junto con sus equipos terminales asociados, requerirán de un mínimo costo en tasa de transmisión
- 2) En sistemas donde los usuarios reciben todas las señales por multiplexaje, se le permite al usuario seleccionar la señal requerida.

### MULTIPLEXAJE POR DIVISION DE FRECUENCIA (FDM)

El tipo de multiplexaje por división de frecuencia (FDM), consiste en dividir el canal que se va a multiplexar en un número único de frecuencias y cada frecuencia es asignada a un par de entidades de comunicación. FDM puede ser usado sólo si el ancho de banda disponible sobre el canal multiplexado excede al ancho de banda requerido por todas las entidades de comunicación.

Un número de señales puede ser transmitida simultáneamente si cada señal es modulada sobre una frecuencia diferente de transmisión y estas frecuencias de transmisión están suficientemente separadas para que los anchos de bandas de las señales no se traslapen. Es decir, que cada señal requiere de un cierto ancho de banda centrado alrededor de su frecuencia de transmisión para prevenir interferencias.

Un multiplexor recibe datos para transmisión, el dato es transmitido por el multiplexor sobre la frecuencia permitida para la entidad de transmisión. El multiplexor que recibe envía la información recibida sobre una frecuencia específica para el destino asociado con dicha frecuencia.

#### CARACTERISTICAS DE FDM

- Se divide el ancho de banda disponible en canales paralelos.
- El ancho de banda de cada subcanal es directamente proporcional a la velocidad.
- Para evitar interferencias entre subcanales se utilizan "bandas de guarda" ( anchos de banda sin uso) de protección.
- La capacidad del canal está limitado por el ancho de banda.

#### VENTAJAS DE FDM

- A cada equipo terminal de datos (DTE) se le asigna una única frecuencia la cual puede ser tratada como un canal no compartido.
- Bajo costo.
- Posibilidad de conectarse en cascada.
- Poder realizar un enlace full-duplex sobre dos hilos.
- Lleva el modem incorporado.

#### DESVENTAJAS DE FDM

- No es ampliamente usado en comunicación de datos por el costo que representa tener hardware que permita transmitir y recibir señales sobre una variedad de frecuencias.
- Baja eficacia.
- Presenta un número limitado de canales.
- Velocidad limitada por el canal.
- Altamente inflexibles.
- Tiene controles muy limitados.

### MULTIPLEXAJE POR DIVISION DE TIEMPO (TDM)

Para el multiplexaje por división de tiempo (TDM), se requiere de multiplexores de tiempo compartido entre el canal y todos los equipos terminales de datos (DTE's) involucrados en la comunicación.

El multiplexaje TDM es usado cuando la tasa de bits del medio excede la tasa de datos requeridos para una señal digital. Múltiples señales digitales pueden ser llevadas por una ruta simple de transmisión en varias porciones de cada señal a la vez. Estas porciones pueden ser a nivel de bits, bytes, bloques o cantidades más largas.

En un momento específico, el multiplexor remoto enviará un byte dado por el DTE 1; en la próxima instancia, un byte del DTE 2 será enviado y así hasta que todos los DTE's hayan sido sondeados; el ciclo es luego repetido. Para evitar la situación en la cual la información llegue a un DTE equivocado, ambos multiplexores deben ser sincronizados. Un camino para sincronizar los multiplexores consiste en usar un patrón de bits especial para indicar el inicio de un nuevo ciclo.

### MULTIPLEXOR DE TIEMPO COMPARTIDO

Un multiplexor de tiempo compartido debe sondear los DTE's para determinar si hay información para enviar. Sin embargo, surge un problema si uno de los DTE's no tiene información para enviar. En éste caso se propone lo siguiente:

- Se utilice un patrón de bits reservado (carácter NUL) que se envíe en caso de que un DTE no tenga información para transmitir.
- Transmitir nueve bits entre los multiplexores, 8 bits para datos y el noveno para indicar si el byte contiene datos o no.
- Hacer uso del multiplexaje estadístico

### MULTIPLEXAJE ESTADÍSTICO (TDM)

Es una variación del multiplexaje TDM que propone una solución al problema de los DTE's que no tienen información a enviar, de tal manera que sólo envía información desde un DTE cuando la información está disponible. Los multiplexores deben ser capaces de determinar el destino de cada byte que es recibido, para esto se usa un identificador único que identifica el origen o el destino del byte y debe ser incluido con cada byte. Esto implica que:

- El ancho de banda es reducido porque el identificador es enviado con cada byte.
- El multiplexor debe conocer el destino del byte.

En adición a estas razones, el multiplexaje estadístico difiere del TDM, en que si ninguno de los DTE's está activado, el canal multiplexado estará ocioso. Por otro lado, si todos los dispositivos conectados al multiplexor transmiten simultáneamente por un período prolongado, puede suceder que el ancho de banda no sea suficiente para soportar el tráfico.

### CARACTERÍSTICAS DE TDM

- División del tiempo en intervalos.
- Muestreo de las líneas.
- "Tiempos de guarda" (tiempos en los que no se envía nada) para evitar interferencias.
- Necesidad del empleo de modems.

### BIBLIOGRAFIA: COMUNICACION DE DATOS

- Comunicaciones y redes de procesamiento de datos.  
*Nestor Gonzalez.*
- Communications wiring and interconnection.  
*Fred J. Mc Climans.*
- Data and computer communications.  
*William Stallings, 1988. (2ª edición)*
- Data communications.  
*Larry Hughes, 1992.*
- Data communications & teleprocessing system.  
*Trevar Housley, 1987.*
- Sistemas de comunicaciones.  
*Jose Manuel Huldobro, 1993.*
- Telecommunications and computer.  
*James Martin, 1976. (2ª edición).*

## CAPITULO 2: REDES DE COMUNICACION DE DATOS

### INTRODUCCION

La mayoría de los servicios de comunicación, transfieren datos a través de un sólo enlace de comunicación. Esto limita la posibilidad de que los usuarios se comuniquen con otros a través de canales múltiples. Una red de comunicación de datos consta de 2 o más nodos que están interconectados a través de enlaces de transmisión basados sobre una o varias de las muchas tecnologías y topologías de red disponibles. Las redes pueden ser caracterizadas por una jerarquía de nodos y/o enlaces.

Existen dos tipos básicos de redes de comunicación de datos:

- Conmutadas (switched)
- No conmutadas (non switched)

#### NODO

Es un punto de unión de enlaces. Desde el punto de vista del flujo de los datos: es un punto de conmutación de la ruta que siguen los mensajes.

#### ENLACE

Es un circuito físico entre los puntos, o bien, un circuito lógico o conceptual entre dos usuarios de una red conmutada u otro tipo de red de comunicaciones que les permite comunicarse entre sí.

### REDES CONMUTADAS

Una red de comunicación conmutada, consta de una colección de nodos interconectados, en los cuales los datos son transmitidos desde un origen a un destino al ser dirigidos a través de los nodos de la red. Los nodos son conectados por rutas de transmisión.

Las redes conmutadas ("switched") de comunicación pueden clasificarse por:

- Cobertura geográfica
- Métodos de acceso
- Técnicas de conmutación empleadas en la red

#### CARACTERISTICAS DE REDES CONMUTADAS

- Algunos nodos conectan sólo otros nodos, su tarea sólo consiste en conmutar datos.
- Los enlaces nodo-estación son generalmente enlaces punto a punto dedicados.
- La topología de la red de nodos puede ser completamente o parcialmente conectada.

#### COBERTURA GEOGRAFICA

Hay tres tipos de redes caracterizadas por su cobertura geográfica:

1. Redes de área local (LAN, Local Area Network).- Cerca del 60% de todas las redes están dentro de la categoría de las LAN
2. Redes de área metropolitana (MAN, Metropolitan Area Network).- Cerca del 22% de todas las redes cubren distancias de alrededor de 60 Km.
3. Redes de área ancha (WAN, Wide Area Network).- Cerca del 18% de todas las redes caen dentro de la categoría de WAN's, cubren distancias de alrededor de 500 Km.

#### MÉTODOS DE ACCESO

Cada nodo suscrito debe ser diseñado para acceder a la red conmutada para servicio. Básicamente hay tres métodos de acceso distinto:

- Redes de acceso por demanda
- Redes de acceso por sondeo ( Polled )
- Redes de multiacceso

#### REDES DE ACCESO POR DEMANDA

Este es generalmente empleado en la mayoría de las redes de comunicación de voz y de datos. De acuerdo con esta técnica, basada en interrupciones, el suscriptor cambia al estado de "descolgado" para demandar servicio. El nodo de servicio proporciona el "tono para marcar" inmediatamente después de descubrir la condición de "descolgado". El "tono para marcar" es una invitación para un diálogo futuro que en última instancia resulta en una ruta conmutada entre el suscriptor llamado y al que llamó.

#### REDES DE ACCESO POR SONDEO

Este es otro método popular para permitir a la "computadora controlada" por el nodo conmutador, interrogar a cada suscriptor sobre la necesidad de intercambiar datos con otro suscriptor.

#### REDES DE MULTIACCESO

En este caso un nodo origen inicia la transmisión sin tener este un mensaje o un paquete en su memoria. Si una colisión ocurre entre dos mensajes transmitidos por dos orígenes, cada nodo retransmite después de esperar un período aleatorio. En algunos sistemas el origen retransmite si éste no recibe un reconocimiento positivo desde el nodo destino dentro de un período de tiempo.

## TECNICAS DE CONMUTACION

Las técnicas de conmutación empleadas en las redes de comunicación, determinan la manera en la cual los recursos de la red son compartidos con los usuarios. Los datos que entran a la red desde una estación son dirigidos a su destino por medio de conmutación de un nodo a otro. Para el desempeño de ésta tarea se proponen tres tipos de técnicas de conmutación:

1. Conmutación por circuito
2. Conmutación por mensajes
3. Conmutación por paquetes

El elegir una u otra técnica de conmutación depende además de:

- El número de estaciones.
- El número de nodos.
- Carga total sobre el sistema.
- La distancia típica entre dos estaciones.

## CONMUTACION POR CIRCUITO

La técnica de conmutación por circuito (CS, Circuit Switching), implica que existe una ruta de comunicación dedicada entre dos estaciones, es decir, la ruta a seguir es una secuencia de enlaces conectados por los nodos. Sobre cada enlace físico, un canal es dedicado a la conexión. La comunicación vía conmutación por circuito involucra tres fases:

- 1) Establecimiento del circuito
- 2) Transferencia de datos
- 3) Desconexión del circuito.

Dado que la ruta de conexión es establecida antes de que la transmisión inicie, entonces el canal debe estar disponible y reservado entre cada par de nodos en la ruta y cada nodo debe tener capacidad de conmutación interna para realizar la conexión. Los conmutadores deben tener la inteligencia para realizar esas localizaciones y de generar una ruta a través de la red.

### ESTABLECIMIENTO DEL CIRCUITO

Antes de que cualquier dato sea transmitido, un circuito estación-a-estación debe ser establecido.

### TRANSFERENCIA DE DATOS

La señalización y transmisión puede ser digital o analógica. Generalmente la conexión es full-duplex y el dato puede ser transmitido en ambas direcciones.

### DESCONEXION POR CIRCUITO

Después de un período de transferencia, la conexión es finalizada, usualmente por la acción de una de las dos estaciones.

## DESVENTAJAS DE LA CONMUTACION POR CIRCUITO

- La capacidad del canal es ocupada por todo el tiempo que dura la conexión, aún si los datos no han sido iniciados para ser transferidos.
- Para una conexión de voz, la utilización del canal puede ser algo alta, pero aún así no se aprovecha al 100%.
- Para una conexión terminal-a-computadora, la capacidad del canal puede estar ociosa durante la mayor parte del tiempo de la conexión.
- Existe un retardo antes de la transferencia de datos por el establecimiento de llamada.
- Ambas estaciones deben estar disponibles al mismo tiempo para el intercambio de datos.
- Los recursos deben estar disponibles y dedicados a través de la red entre las dos estaciones.

## VENTAJAS DE LA CONMUTACION POR CIRCUITOS

- La red es efectivamente transparente a los usuarios desde el momento que se establece el circuito.
- Los datos son transmitidos en una tasa de datos fija sin otro retardo que el retardo de propagación a través de los enlaces de transmisión.
- El retardo en cada nodo es insignificante.
- Es una técnica apropiada en el caso de intercambio de datos que involucran un flujo continuo, tal como voz, algunas formas de sensores, antradas de telemetría, etc.
- Para cargas muy pesadas entre dos estaciones, es la técnica de menor costo.

## CONMUTACION POR MENSAJES

La técnica de conmutación por mensajes (MS, Message Switching) surge como una alternativa a la técnica de conmutación por circuito. Generalmente es apropiado para intercambio de datos digitales. En este caso a las unidades lógicas de datos se les denomina "mensajes". En la conmutación por mensajes no es necesario establecer una ruta dedicada entre dos estaciones. En este caso si una estación desea enviar algún mensaje se realiza lo siguiente:

- 1) La estación agrega una dirección de destino al mensaje.
- 2) El mensaje es pasado a través de la red desde un nodo a otro.
- 3) En cada nodo el mensaje es recibido, almacenado brevemente y luego transmitido al próximo nodo hasta llegar a su destino.

En esta técnica de conmutación cada nodo debe ser típicamente una microcomputadora de propósito general, con un buffer suficiente para almacenar los mensajes que llegan. El almacenamiento es basado en disco si el tiempo de recuperación debe ser corto; o es basado en cintas magnéticas si el tiempo de recuperación no debe ser tan corto. Un mensaje es retardado en cada nodo el tiempo requerido para recibir todos los bits del mensaje, más el retardo en la cola de espera para tener oportunidad de retransmitir al próximo nodo. Este sistema también es conocido como un sistema de mensajes de "almacena y adelanta"

#### **DESVENTAJAS DE LA CONMUTACION POR MENSAJES**

- La principal desventaja es que no es apta para tiempo real o tráfico interactivo.
- El retardo a través de la red es relativamente largo y tiene alta varianza.
- No puede ser usado para conexiones de voz.
- No es apta para conexiones interactivas terminal-estación.
- Sobrecarga de bits en cada mensaje.
- Los nodos deben ser centros de conmutación de mensajes con muchas facilidades.
- La sobrecarga incrementa el retardo del mensaje.
- No hay señal de reconocimiento de mensaje.

#### **VENTAJAS DE LA CONMUTACION POR MENSAJES**

- La eficiencia de la línea es mayor, debido a que un simple canal nodo-a-nodo puede ser compartido por varios mensajes a la vez.
- La habilidad simultánea de enviar y recibir no es requerida. La red puede almacenar el mensaje dejando pendiente la recepción del mismo.
- Cuando el tráfico es pesado los mensajes son aún aceptados, pero se incrementa el retardo.
- Se puede enviar un mensaje a muchos destinos.
- Se puede establecer prioridad en los mensajes.
- El procedimiento de control y recuperación de errores sobre un mensaje, puede ser construido dentro de la red.
- Puede realizarse fácilmente conversión de formatos de los datos (ASCII, EBCDIC, etc.)
- Los mensajes enviados a terminales que no están operando, pueden ser interceptados y almacenados o redirigidos a otras terminales.

#### **CONMUTACION POR PAQUETES**

La técnica de comunicación por paquetes (PS, Packet Switching) es una combinación de las ventajas de la conmutación por circuito y la conmutación por mensajes. La principal diferencia externa, es que la longitud de la unidad de datos es limitada. Es decir, si algún mensaje es demasiado largo, éste se divide en unidades más pequeñas que reciben el nombre de "paquetes". El paquete contiene datos más una dirección de destino.

A diferencia de la conmutación por circuitos, se emplea la técnica de "almacena y adelanta" para conmutar cada paquete con el propósito de dirigirlos hacia su destino sobre el circuito establecido antes de iniciar la transmisión.

La diferencia clave con respecto a la conmutación por mensajes, consiste en la división de la unidad lógica de datos en tamaños más pequeños los cuales tienen un importante efecto en el buen desempeño de la técnica. Los dos procedimientos más comunes para realizar la entrada de los mensajes sobre una red conmutada por paquetes son:

1. Por datagramas
2. Por circuito virtual

#### **DATAGRAMAS**

En el procedimiento por datagramas, cada paquete es tratado independientemente, es decir, dado que cada paquete tiene la dirección de destino no importa cual sea la ruta que siga. Además, es posible que la estación destino reciba los paquetes en una secuencia diferente a la que fueron enviados. El ordenarlos correctamente es una tarea de la estación destino. En esta técnica cada paquete, tratado independientemente, es referido como un "datagrama".

#### **VENTAJAS DEL USO DE DATAGRAMAS**

- Los nodos pueden ser conmutadores pequeños.
- Es muy flexible.
- El datagrama es inherentemente más confiable.
- Si un nodo falla se busca otra ruta.
- Es eficiente para mensajes cortos.
- Los datagramas pueden ser almacenados hasta que sea posible su entrega.

#### **DESVENTAJAS DEL USO DE DATAGRAMAS**

- No son apropiados para tráfico interactivo.
- La sobrecarga incrementa el retardo del paquete.

#### **CIRCUITO VIRTUAL**

En el procedimiento de circuitos virtuales, una conexión lógica es establecida antes de que cualquier paquete sea enviado. Cada paquete contiene un identificador de circuito virtual más dato. Cada nodo sobre la ruta preestablecida sabe a donde dirigir cada paquete, no se requiere de decisiones de ruteo. Eventualmente las estaciones de destino terminan la conexión con un paquete de "respuesta limpia". En cualquier momento, cada estación puede tener más de un circuito virtual para cualquier otra estación.

La principal característica de la técnica circuito virtual es que el encaminamiento entre estaciones es dado antes de la transferencia de datos. Esto significa que se requiere de una ruta dedicada. La diferencia con el procedimiento de datagramas es que los nodos no requieren tomar decisiones de encaminamiento para cada paquete.

Un circuito virtual fácilmente puede proporcionar un número de servicios diferentes, como el de secuenciación, control de errores, control de flujo, etc.

#### VENTAJAS DEL USO DE CIRCUITOS VIRTUALES

- Es eficiente para intercambios largos.
- Los nodos pueden ser conmutadores pequeños.
- Uso dinámico del ancho de banda.

#### DESVENTAJAS DEL USO DE CIRCUITOS VIRTUALES

- Si un nodo falla todos los circuitos virtuales que pasan sobre él se pierden.
- La sobrecarga de canal puede bloquear la llamada para inicio de la transmisión; entonces se incrementa el retardo del paquete.
- Sobrecarga de bits en cada paquete.

### REDES NO CONMUTADAS (BROADCAST)

Las redes de comunicación no conmutadas no tienen nodos de conmutación intermedios. Cada estación es conectada a un transceptor que se comunica sobre un medio compartido por otras estaciones. De esta manera una transmisión desde cualquier otra estación es emitida y recibida por todas las demás estaciones.

Las redes no conmutadas más comunes son:

- Red de radio terrestre
- Red satelital
- Red local

La diferencia principal entre éstos tres tipos de redes no conmutadas es la tasa de datos a transmitir y la distancia entre las estaciones.

#### CARACTERISTICAS DE LAS REDES NO CONMUTADAS

- Por arquitectura básica, no tiene dispositivos de conmutación.
- Los datos transmitidos por una estación son recibidos por las demás estaciones conectadas a la red.
- Las estaciones comparten un medio de transmisión común, por lo tanto se requiere de técnicas de control de acceso.
- Sólo un puerto de Entrada/Salida es requerido en cada estación.
- Dado que la transmisión facilitada es compartida, sólo una estación puede transmitir a la vez.

### REDES DE RADIO TERRESTRES

En las redes de radio terrestres las estaciones transmiten y reciben vía antena y todas ellas comparten el mismo canal de radiofrecuencia. Las estaciones están dentro del rango de transmisión de cada otra y transmiten directamente.

El tipo de transmisión en las redes de radio terrestres tiende a ser omnidireccional (para todas direcciones); por lo tanto, la transmisión desde un nodo puede ser recibida por todos los demás nodos en la red. El tiempo de propagación en éste tipo de redes pocas veces exceden el rango de milisegundos.

La arquitectura de redes de radio terrestres puede ser clasificada de las dos formas siguientes:

1. Centralizada.
2. Distribuida

#### REDES DE RADIO TERRESTRES CENTRALIZADAS

En una red centralizada, hay un transceptor central, conectado a un recurso central. Todos los demás nodos se comunican sólo con el nodo central. La comunicación nodo-a-nodo es indirecta, mediado por el nodo central. En un sistema centralizado dos canales de radio son requeridos. Los nodos individuales envían paquetes al nodo central sobre un canal y el nodo central transmite los paquetes sobre el otro canal. Dado que la transmisión de radio es omnidireccional, los paquetes transmitidos por el nodo central, son captados por todos los nodos.

#### REDES DE RADIO TERRESTRE DISTRIBUIDAS

En las redes distribuidas no existe la estación central y las terminales se comunican entre sí, emitiendo sus mensajes al medio. Los principales aspectos a tener en cuenta en este tipo de redes son:

- Las frecuencias de transmisión y recepción.
- Proveer mecanismos para evitar colisiones.
- Utilización de estaciones repetidoras con funciones de encaminamiento para extender la cobertura de la red.

Las redes distribuidas toman completa ventaja de la propiedad omnidireccional de la transmisión de radio. Un canal es usado por todas las transmisiones y cada transmisión es captada por todos los demás nodos.

**COLISION**

Se le llama colisión al problema que se presenta cuando dos o más estaciones terminales, conectadas a un mismo medio físico, en una misma red, desean transmitir simultáneamente.

**REDES SATELITALES**

En las redes satelitales todos los nodos conectados, están habilitados para captar cada transmisión desde el satélite. Esto es debido a que todos los nodos comparten el mismo canal de frecuencia. Cada estación transmite al satélite, el satélite repite la transmisión la cual es recibida por múltiples estaciones. La transmisión y la recepción se realiza vía antena.

Los enlaces satelitales son usados en una amplia variedad de configuraciones, las cuales son determinadas, primeramente, por la naturaleza de las estaciones terrestres y la interconexión proporcionada por el enlace del satélite.

Los diferentes tipos de estaciones terrestres son:

- Estaciones multiplexadas
- Estaciones individuales
- Estaciones transportadoras de asignación fija. Los tipos de enlaces satelitales más comunes son:
  - Punto a punto
  - Punto a muchos
  - Muchos a muchos
  - Muchos a punto

**PROPIEDADES DE LAS REDES SATELITALES**

- Un satélite en órbita geosíncrona es visible alrededor de la cuarta parte de la superficie de la tierra.
- El costo de la transmisión es independiente de la distancia, dentro del área de cobertura del satélite.
- Ambas aplicaciones: punto a punto y radiodifusión son posibles
- Permite el uso de muy altos anchos de banda.
- Tienen un retardo de propagación tierra-satélite-tierra, aproximado a la cuarta parte de un segundo.
- Una estación en transmisión puede en muchos casos recibir su propia transmisión.

**REDES DE AREA LOCAL (LAN)**

Una red de área local (LAN) es una red de comunicaciones que permite la interconexión, en un medio común, de una variedad de dispositivos de procesamiento de información dentro de un área relativamente pequeña. Las redes de área local se originaron como un medio para compartir dispositivos periféricos en una organización.

Las LAN, normalmente son semejantes a las redes de conmutación de paquetes, donde el enfoque de almacenamiento y reenvío generalmente no se usa, por lo que no hay nodos de conmutación, sino que el computador se conecta directamente a la red por medio de un nodo que realiza las funciones necesarias para que el computador reciba y transmita los paquetes.

Los cuatro componentes principales de una LAN son:

1. El canal
2. La interfase física
3. El protocolo
4. La estación de usuario

Las dos técnicas de transmisión de redes locales más usadas son:

1. De banda base (Baseband)
2. De banda ancha (Broadband)

**CANAL**

Consiste de un cable coaxial de TV, coaxial bandabase, par trenzado o fibra óptica.

**INTERFASE FISICA**

La interfase entre la ruta y la estación de usuario puede tomar varias formas. Estas pueden ser un simple cable de TV, diodos infrarrojos, antenas de microondas o LED's para fibras ópticas.

**PROTOCOLO**

El protocolo controla lógicamente los componentes de control de la LAN y proporciona a los usuarios acceso a la red.

**ESTACION DE USUARIO**

Esta puede ser cualquier cosa desde un procesador de palabras hasta una computadora mainframe.

**BANDA BASE (BASEBAND)**

Esta técnica usa todo el ancho de banda disponible para formar un canal de datos. Las señales de datos son transmitidas directamente sobre el enlace sin ser moduladas sobre cualquier frecuencia portadora. Los niveles específicos de voltaje son usualmente escogidos para representar los bits "1" y "0", y la señalización puede ser considerada como una secuencia de pulsos de voltaje sobre la línea.

Debido a que no se requiere modulación de portadora, se pueden usar líneas muy simples sobre enlaces punto a punto. Esto define los límites de distancia que pueden ser cubiertos sin usar repetidores para mantener el nivel de la señal reconocible sobre el ruido.

Las transmisiones banda base pueden también ser usadas por redes multipunto. De ser así, los conductores de línea deben ser reemplazadas por transceptores más sofisticados para garantizar niveles de señales que se pueden mantener a través de los múltiples dispositivos compartidos. Las limitaciones de distancia fueron impuestas para mantener los niveles de las señales y tiempos de propagación dentro de límites aceptables.

#### VENTAJAS DE LA TRANSMISION BANDA BASE

- Los conductores y transceptores son dispositivos relativamente simples.
- Cualquier topología puede ser soportada.
- La elección del medio incluye: par trenzado de cobre, cable coaxial y fibra óptica.
- Pueden soportar tasas de datos altas (de 1-10 Mbps).

#### DESVENTAJAS DE LA TRANSMISION BANDA BASE

- Limitaciones de distancia.
- Susceptibilidad al ruido de cruce y otros ruidos típicos en los cables.
- Dificultad de hacer múltiples derivaciones con fibras ópticas.

#### BANDA ANCHA (BROADBAND)

En esta técnica el ancho de banda es dividido en frecuencias dentro de canales múltiples, las cuales pueden ser usadas independientemente. El cable coaxial es especialmente aceptable para transmisión de banda ancha. Los modems de radiofrecuencia son requeridos para modular transmisiones sobre frecuencias portadoras y de esta manera garantizar que no sucedan interferencias con cualquier otra transmisión.

Las fibras ópticas y microondas de radio tienen anchos de banda más anchos que el cable coaxial; pero cada uno requiere diferentes tipos de transceptores.

La transmisión es unidireccional sobre el canal. La tasa de transmisión máxima de un canal es proporcional a su ancho de banda.

#### VENTAJAS DE LA TRANSMISION DE BANDA ANCHA

- Capacidad de transmisión más grande que la usada bajo tecnologías actuales.
- Habilidad para distribuir canales del mismo medio para todas las clases diferentes de tráfico simultaneo, como datos, TV, y voz.

#### DESVENTAJAS DE LA TRANSMISION DE BANDA ANCHA

- Requiere de un diseño más experto para garantizar que la red cumple con los requerimientos de varios tipos de componentes de transmisión.
- Necesita más experiencia técnica para entender y proporcionar apoyo de mantenimiento para los componentes de radiofrecuencia.
- Las correcciones son más costosas para una red de banda ancha.

#### CARACTERISTICAS DE LAS LAN

- **Tamaño.**- El tamaño de una LAN es normalmente determinado por la organización que la usa. La aplicación más pequeña puede ser el caso donde se utilice una LAN para compartir los dispositivos periféricos (no más de 50 m de largo). La aplicación más grande podría ser el de una LAN situada en una Universidad o en una planta industrial, en donde varios edificios son conectados (de 10 a 20 Km de longitud).
- **Costo.**- El costo de conexión a una LAN depende del tipo de computador que se conecta.
- **Velocidad.**- La capacidad de transmisión es usualmente más grande que el de las WAN's. Típicamente el rango de tasa de transmisión de bits es de 1 Mbps hasta 100 Mbps. El retardo de una LAN normalmente es pequeño, entre 10 y 100 microsegundos, es menor que el de una WAN.
- **Tasa de errores.**- La tasa de errores en una LAN es baja debido a que las distancias cubiertas por una LAN son pequeñas y se pueden usar cables de razonable calidad sin incrementar el costo.
- **Es privada.**- Atiende las necesidades de un sólo grupo de usuarios y es controlada por dichos usuarios.
- **Tiene capacidad de transmitir datos, voz y video entre las estaciones de usuarios y computadoras.**
- **Transmite paquete de datos.**
- **Es conectable, es decir, puede conectar cada dispositivo de la red a cualquier otro dispositivo de la red.**
- **Disponibilidad comercial.**

#### VENTAJAS DE LAS LAN

- El 90% de requerimientos de procesamiento de cualquier usuario común se resuelven con el uso de una LAN.
- Tiene capacidad de compartir recursos.
- Compatibilidad de equipos
- Procesamiento distribuido (no depende de un elemento central)
- Alta velocidad, tasa de error baja, distancias de alcance bastante aceptables.
- Capacidad de transmisión simultanea de información de distinta naturaleza.
- Distribución adecuada del hardware.
- Simplicidad y flexibilidad de modificaciones de la configuración de la red.
- Alta rentabilidad e integridad.
- Son expandibles.
- Son fáciles de acceder.
- Adaptables a diferentes aplicaciones.

**DESVENTAJAS DE LAS LAN**

- El no poder abarcar una cobertura geográfica más amplia.
- Existencia de varios estándares.
- La interoperabilidad no es garantizada: software, datos, etc.
- En bases de datos distribuidas surgen problemas de integridad, seguridad y privacidad
- Escalación progresiva: más equipo de lo que se necesita actualmente será requerido.
- Pérdida de control: Cada vez más difícil para manejar y forzar estándares.

**TOPOLOGIAS**

La topología de una red normalmente es una descripción del cableado que conecta los nodos de la red. Sin embargo, no siempre describe el camino que toman los paquetes cuando viajan a través de la red. Dado que el tamaño de las LAN es pequeño en comparación con las WAN, no se puede usar la estructura de una red conmutada, sino se han desarrollado algunas topologías en las que no se requiera, o se requiera sólo un nodo de conmutación. Las topologías más usadas son:

- Topología estrella.
- Topología de canal (Bus).
- Topología de anillo (Ring).
- Topología de árbol (jerárquica).
- Topología de malla. (usada principalmente en WAN's).

Estos tipos de topologías pueden ser usados para construir bloques de redes con topologías más complejas.

**TOPOLOGIA ESTRELLA**

Una red en estrella emplea un nodo central de conmutación al cual se conectan todos los nodos de la red por medio de enlaces bidireccionales punto a punto. La comunicación entre cualquiera dos estaciones es vía conmutación por circuito generalmente. Para que una estación transmita un dato, éste debe primero enviar una solicitud al nodo de conmutación central y preguntar por una conexión a alguna estación destino. Desde que el circuito es realizado, el dato puede ser intercambiado entre las dos estaciones como si estuvieran conectadas por un enlace punto a punto dedicado.

En la topología estrella todas las comunicaciones son controladas por el nodo de conmutación central el cual debe iniciar y mantener un número de rutas de datos concurrentes. Consecuentemente el nodo central de conmutación es más complejo.

**VENTAJAS DE LA TOPOLOGIA ESTRELLA**

- Su estructura es relativamente simple.
- Pueden soportar gran capacidad de procesamiento.
- Pueden representar una importante topología para comunicaciones vía satélite.
- Para anexas otro nodo-estación a la red sólo se requiere la solicitud y aceptación de un enlace punto a punto.

**DESVENTAJAS DE LA TOPOLOGIA ESTRELLA**

- Si el nodo de conmutación central se detiene, la red deja de funcionar.
- Limitaciones en cuanto a rendimiento y confiabilidad.
- La red puede crecer sólo hasta alcanzar la capacidad del nodo de conmutación central.

**TOPOLOGIA DE CANAL (BUS)**

Esta es la más sencilla de las topologías dado que se usa un medio de comunicación común al cual se conectan todos los nodos de la red en enlace multipunto. La conexión al nivel físico es tan simple que sólo se debe conectar el dispositivo al medio (sin uso de repetidores o conmutadores). Debido a que todos los nodos comparten el mismo canal, sólo una estación puede transmitir a la vez. Para esto se requiere de técnicas de control de acceso al medio. Normalmente el canal se encuentra en estado "pasivo" (no contiene cableado activo para amplificar señales). Esto significa que los canales son inherentemente fiables, pero con longitud limitada, ya que los transmisores deben ser capaces de enviar la señal a lo largo de todo el canal.

Cuando una estación transmite un paquete todos los dispositivos conectados a la red lo pueden recibir. Cuando un nodo reconoce que un mensaje va dirigido a él, lo saca del canal. Por lo tanto, el canal requiere que cada nodo tenga la capacidad de transmitir, recibir y resolver problemas.

**VENTAJAS DE LA TOPOLOGIA DE CANAL**

- El control es distribuido.
- Alta confiabilidad.
- La interconexión de la red e instalación es muy sencilla.

**DESVENTAJAS DE LA TOPOLOGIA DE CANAL**

- Generalmente debe haber una distancia mínima entre estaciones para evitar interferencias en las señales.
- El uso de la topología se limita a las LAN.
- Requiere de una técnica de control de acceso al medio.
- A distancias grandes se requiere del uso de repetidores.

## TOPOLOGIA DE ANILLO (RING)

En la topología de anillo, la red local consiste de un conjunto de repetidores unidos por enlace punto a punto en un lazo cerrado. Los datos fluyen en una sola dirección alrededor del anillo, y los dispositivos conectados al anillo pueden recibir los datos de él. Para transmitir es necesario que el dispositivo interrumpa los datos del anillo para poder transmitir los suyos. Cada repetidor participa en dos enlaces. El repetidor es un dispositivo comparativamente simple, capaz de recibir datos de un enlace y transmitirlos bit por bit sobre el otro enlace tan rápido como estos son recibidos. No se requiere de un buffer en el repetidor.

Cada estación se conecta a la red a través de un repetidor. Los datos son transmitidos en paquetes. Cuando un paquete se transmite, este circulará indefinidamente si no se quita del medio. Cualquier paquete que se transmita puede ser visto por todos los nodos de la red.

Una característica importante es el tener el control distribuido, dado que cada elemento de la red, generalmente, tiene la misma jerarquía que los demás en lo que respecta a sus facultades de comunicaciones. En algunas implementaciones se usa un segundo canal que permite que el flujo de datos sea en ambas direcciones.

### VENTAJAS DE LA TOPOLOGIA DE ANILLO

- Se requiere de mínima inteligencia para su funcionamiento.
- Es de costo aceptable
- El tipo de control es distribuido.
- Es flexible y confiable
- Existen diversas formas para el control de la transmisión de paquetes

### DESVENTAJAS DE LA TOPOLOGIA DE ANILLO

- Si falla un elemento del anillo, toda la red deja de funcionar.
- En la medida en que se pasan los mensajes, se puede disminuir notablemente la velocidad de la red.
- El mensaje que entra en la red debe tener un grupo de bits adicionales para la dirección de su destino.

## TOPOLOGIA DE ARBOL (JERARQUICA)

Esta topología es una derivación de la topología de canal. En esta topología se cuenta con un nodo de partida al cual se le denomina como "nodo raíz". Directamente conectados a éste nodo puede haber otros en un segundo nivel y a su vez cada uno de éstos puede tener varios nodos conectados en cascada. Nuevamente, una transmisión desde cualquier estación se propaga sobre el medio y puede ser recibida por las demás estaciones.

El enlace que se utiliza es multipunto. Existen dos clases principales de la topología de árbol:

1. Forma jerárquica
2. Forma canal / árbol

### FORMA JERARQUICA (Topología)

En esta forma los nodos son conectados en forma jerárquica. Esta variante de la topología de árbol es usada en ciertos tipos de redes de área ancha, por su simplicidad en el control de encaminamiento. Dado que sólo existe una ruta entre cada dos nodos, entonces toda la información de encaminamiento requerida, puede estar contenida dentro de las direcciones de encaminamiento.

### FORMA DE CANAL / ARBOL (Topología)

La topología de canal / árbol encuentra su principal uso en las LAN. Se puede considerar que es una extensión de la topología de canal.

En esta forma de la topología, para simplificar la interconexión entre ramas, no más de dos nodos son permitidos entre los repetidores que enlazan la rama de la red.

### VENTAJAS DE LA TOPOLOGIA DE ARBOL

- El nodo central es menos costoso que uno de la topología de estrella.
- El tiempo de estancia de una transferencia de datos es menor que el número de mensajes que permanecen dentro de la rama.

### DESVENTAJAS DE LA TOPOLOGIA DE ARBOL

- La transmisión se propaga por todas las ramas, haciéndola más lenta
- Sólo una estación puede transmitir a la vez
- Se requieren de técnicas de control de acceso al medio.
- La expansión y reconfiguración de la red ocasiona grandes problemas
- Se provoca congestión en la raíz y en los nodos de más alto nivel

## TOPOLOGIA DE MALLA

Esta topología es usada por redes que permiten interconexiones más aleatorias. Las redes de malla permiten redundancia, ya que puede haber más de un camino para los paquetes entre dos nodos de la red. Por esta razón las redes de área amplia (WAN) normalmente se basan en esta topología. Para poder aplicar la topología de malla a redes de área local (LAN), los nodos de conmutación no deben operar según el principio de almacenamiento y reenvío, ya que aumentarían el retardo de la red.

También una red con esta topología puede ser conectada completamente, pero tiene la desventaja de requerir para  $n$  nodos,  $n(n-1)/2$  enlaces.

**VENTAJAS DE LA TOPOLOGIA DE MALLA**

- Si es una red completamente conectada no se requieren de nodos intermedios.
- Mas de una estación puede comunicarse a la red simultáneamente.
- Se pueden seguir varias rutas a un mismo destino.

**DESVENTAJAS DE LA TOPOLOGIA DE MALLA**

- Se requiere de gran número de enlaces, en caso de ser una red conectada completamente, por lo que se ocasionan problemas de tiempo.
- La expansión de la red es costosa.
- Económicamente es aceptable solo para un número de nodos muy pequeño.

**TECNICAS DE CONTROL DE ACCESO AL MEDIO**

Todas las redes locales constan de una colección de dispositivos que deben compartir las capacidades de transmisión de la red. Para realizar esta tarea se requiere del uso de alguna técnica que controle el acceso al medio de transmisión, sólo de ésta manera dos dispositivos podrán intercambiar datos.

El determinar cual técnica se debe utilizar depende de donde y como se va a emplear. En el caso de que se disponga de una LAN centralizada se diseña un controlador, el cual autoriza o no el acceso a la red. En cambio, en una LAN distribuída las estaciones desempeñan la función de controlar el acceso al medio, determinando el orden en el que las estaciones deben transmitir. Además se deben considerar una serie de factores como: la topología de la red, el costo, el desempeño y complejidad, etc. Generalmente las técnicas de control de acceso al medio se pueden categorizar como:

- Técnicas síncronas
- Técnicas asíncronas

Con las técnicas síncronas una capacidad específica está dedicada a una conexión. Tales técnicas no son óptimas en LAN's debido a que se requieren estaciones que son inestables generalmente. Las técnicas asíncronas se dividen en 3 categorías:

1. Round robin
2. Reservación
3. Contención

**CONTROL DE ACCESO AL MEDIO POR ROUND ROBIN**

Las técnicas round robin son sencillas, dado que estan basadas en la filosofía de "dar a todos un turno". A cada estación en turno se le da una oportunidad para transmitir. Durante esta oportunidad, la estación puede realizar una transmisión o puede transmitir sujeta, ya sea a una máxima cantidad de datos o un período de tiempo definido para esta oportunidad. De cualquier modo cuando la estación termina debe renunciar a su turno, y el derecho para transmitir pasa a la próxima estación en la secuencia lógica. El control de turnos puede ser centralizado o distribuído.

Cuando muchas estaciones tienen datos a transmitir sobre un periodo largo de tiempo, las técnicas round robin pueden ser muy eficientes. Las técnicas principales bajo la filosofía de round robin son:

- Control de acceso por token
- Control de acceso por sondeo

**CONTROL DE ACCESO POR TOKEN**

Esta es una de las formas para controlar el acceso a un medio de transmisión compartido. Un token es pasado de una estación a otra de acuerdo con un conjunto de reglas definidas, entendidas y aceptadas por todas las estaciones conectadas al medio. Una estación sólo puede transmitir un frame cuando éste posee un token y, después de la transmisión del frame, ésta libera el token para permitir que otra estación accese el medio transmisión. La secuencia de operación es la siguiente:

1. Primeramente se establece un anillo lógico, el cual enlaza todas las estaciones conectadas al medio físico y un simple control de token es creado.
2. Un token es pasado desde una estación a otra alrededor del anillo lógico hasta que éste es recibido por una estación que desea enviar algun frame.
3. La estación en espera envía los frames esperados usando el medio físico; después pasa el token de control a la próxima estación en el anillo lógico.

Las funciones de monitoreo dentro de las estaciones activas conectadas al medio físico, proporcionan una base para inicialización y recuperación de la conexión del anillo lógico y del token. A pesar que las funciones de monitoreo están normalmente repartidas a lo largo de todas las estaciones sobre el medio, sólo una estación, a la vez, carga con la responsabilidad de recuperación y reinicialización. Las topologías que se requieren para el uso de esta técnica son la de anillo y la de canal y se les conoce como:

- Token ring
- Token bus

**TOKEN RING**

La técnica de token ring es una variante de token bus, donde se hace coincidir el anillo lógico con el anillo físico, evitando de esta manera los complejos procedimientos de inicialización y mantenimiento del token, utilizados en la técnica token bus.

El funcionamiento de token ring es el siguiente:

- 1) Una estación debe esperar el token para poder enviar un frame. Cuando se recibe el token se inicializa la transmisión del frame, el cual incluye la dirección de destino en su encabezado.

- 2) El frame es repetido (cada bit es recibido y luego retransmitido) por todas las estaciones en el anillo.
- 3) Cuando una estación descubre que es el destino del frame, realiza una copia del frame e indica que éste ha sido recibido al insertar bits de respuesta al final del frame.
- 4) Luego el frame sigue circulando de regreso a la estación de inicialización, donde es eliminado. La estación libera el token de dos formas posibles, dependiendo de la velocidad de transmisión usada en el anillo:
  - a) En anillos lentos (4 Mbps aprox.), el token se libera sólo después de que los bits de respuesta han sido recibidos.
  - b) En anillos de alta velocidad (16 Mbps aprox.), el token se libera después de la transmisión del último bit de un frame.

#### VENTAJAS DEL ACCESO AL MEDIO POR TOKEN RING

- Su principal ventaja es que el tráfico puede ser regulado
- Permite a las estaciones transmitir diferentes cantidades de datos cuando estas reciben el token
- Asignación de prioridades a las estaciones, para que a las de más alta prioridad aseguren la posesión del token.

#### DESVENTAJAS DEL ACCESO AL MEDIO POR TOKEN RING

- Su principal desventaja es el requerimiento de mantenimiento del token.
- Requerimiento de técnicas de manejo de fallas.

#### TOKEN BUS

En la operación de token bus, los procedimientos usados para la administración del anillo lógico son diferentes a los usados para administración del anillo físico. Las redes que utilizan token bus normalmente usan cable coaxial como medio de transmisión.

En token bus, sólo la estación que posee el token puede transmitir algún frame. Todas las estaciones que pueden iniciar la transmisión de un frame están enlazadas en la forma de un anillo lógico.

- 1) El token es pasado, usando el canal físicamente, alrededor del anillo lógico.
- 2) Al recibir el token desde su predecesor en el anillo, una estación puede transmitir frames en espera hasta un máximo definido.
- 3) Después pasa el token a su sucesor conocido, sobre el anillo lógico.

#### CONSIDERACIONES EN TOKEN BUS

- Bajo una operación normal el token es pasado desde una estación, sobre el anillo lógico, a otra estación usando un frame de control corto como token.
- Cada estación requiere conocer la dirección de la estación anterior y siguiente en el anillo lógico.
- Si una estación falla al recibir el token, la estación que lo envía usa una serie de procedimientos de recuperación, para encontrar un nuevo sucesor.
- Existen otros procedimientos dedicados a dar de alta o baja alguna estación para mantener una operación correcta, para inicialización del anillo, etc.
- Las estaciones que no usan el token están permitidas sobre el canal, pero sólo pueden responder a reconocimientos y sondeos.
- Las estaciones conectadas al anillo lógico deben realizar las siguientes funciones:
  - Inicialización del anillo.
  - Adición al anillo.
  - Borrado desde el anillo.
  - Administración de fallas.

#### VENTAJAS DEL ACCESO POR TOKEN BUS

- Aún con tráfico intenso todas las estaciones pueden enviar y recibir mensajes, esperando a lo sumo M-1 "glros" (para M estaciones).
- Es uno de los más eficientes, especialmente con alto tráfico.
- No requiere detección de colisiones.
- El uso de canales punto a punto, lo sitúa en posición de privilegio para la implementación o la migración al uso de fibras ópticas.
- El anillo lógico no está necesariamente relacionado con el anillo físico.
- Su topología lógica proporciona gran flexibilidad.

#### DESVENTAJAS DEL ACCESO POR TOKEN BUS

- Se requiere de procedimientos de técnicas de contención para inicialización, para establecer el anillo lógico, para mantenimiento del token, etc.
- El tiempo invertido en inicialización y mantenimiento es muy alto.

#### CONTROL DE ACCESO POR SONDEO (POLLING)

La técnica de sondeo se utiliza en redes centralizadas. Su función es evitar que las estaciones intenten enviar información al mismo tiempo. Esta técnica puede ser empleada en redes con enlace punto a punto o multipunto. En caso de redes con enlaces punto a punto, los mensajes generados por sondeo no necesitan tener dirección de ubicación, puesto que la estación sólo recibe los sondeos que le están dirigidos. En este caso el sondeo no se requiere para evitar el caos en las líneas. En caso de que las redes tengan enlace multipunto se definen dos métodos de sondeo:

1. Sondeo por llamada a lista (roll-call polling)
2. Sondeo por secuencia (Hub polling)

Independientemente de cuál de las dos técnicas anteriores se utilice, el control de acceso por sondeo se puede esquematizar como sigue:

1. Se define una tabla (I) con las direcciones de las estaciones que van a intervenir.
2. Se define el orden de intervención de las mismas, pudiendo establecer prioridades (Tabla II).
3. Se definen los elementos de control y almacenamiento que se usaran para manejar el tráfico.

4. El elemento de control invita, de una a la vez, a las estaciones a transmitir, según el orden de prioridades.
5. Estas responden negativamente a la invitación, o hacen uso de ella enviando los datos.
6. Se reitera el procedimiento mientras existan entradas en la tabla (II).

### SONDEO POR LLAMADA A LISTA (ROLL-CALL POLLING)

Consiste de un controlador central encargado de enviar, simplemente, un mensaje por turno a cada una de las estaciones, preguntando si la estación tiene datos a transmitir. Los mensajes de sondeo tienen la dirección de la estación a la que van dirigidos. Cada estación conoce su dirección y sólo responde cuando la dirección del mensaje coincide con la suya. Si la estación que se interroga tiene alguna información para enviar, la envía; pero si no, devuelve un mensaje especial que indica el "rechazo del sondeo". Ante la negativa, el controlador central envía un mensaje de sondeo a la siguiente estación.

### SONDEO POR SECUENCIA (HUB POLLING)

En el sondeo por secuencia, el sistema central parte un mensaje de sondeo destinado a la primera estación del enlace y ésta, después de reconocerlo, si no tiene datos para transmitir, le cambiará la dirección de destino y lo enviará a la segunda estación. La secuencia se repite hasta agotar las estaciones en la lista o hasta que una contesta. Con esta técnica de sondeo se obtiene un mejor rendimiento del sistema central, pero se requiere de más inteligencia en la interconexión de cada estación, lo que aumenta los costos.

### CONTROL DE ACCESO AL MEDIO POR RESERVACION

Para el flujo de tráfico, las técnicas de reservación son aceptables. En general, para estas técnicas, el tiempo sobre el medio de comunicación es dividido en intervalos como en el caso de la técnica de multiplexaje por división de tiempo. Para que una estación pueda transmitir reserva intervalos de tiempo posteriores por un período extendido o indefinido. Estas reservaciones pueden ser realizadas tanto en redes centralizadas como en redes distribuidas.

### ACCESO AL MEDIO POR CONTENCION

Para explotar el tráfico, las técnicas de contención son usualmente apropiadas. Con estas técnicas, no se ejecuta algún control para determinar a quien le corresponde el turno; sino todas las estaciones compiten por tiempo de acceso al medio. Estas técnicas son de naturaleza distribuidas necesariamente. Su principal ventaja es que son simples de implantar y eficientes para moderar carga. Dentro de las técnicas que se rigen bajo esta filosofía se tienen:

- CSMA
- Anillo particionado
- Registro de inserción

### CONTROL DE ACCESO AL MEDIO POR CSMA

El método de acceso CSMA (Acceso múltiple con censado de portadora: Carrier Sense Multiple Acces) es usado sólo por redes con topología de canal. El medio común usado por las redes con topología de canal está diseñado para operar en un modo de acceso múltiple. Existen diferentes variantes con respecto a esta técnica de control de acceso al medio y son:

1. CSMA 1-persistente
2. CSMA no persistente
3. CSMA p-persistente
4. CSMA/CD

En general, los protocolos CSMA aseguran que ninguna estación comience a transmitir información cuando el canal está ocupado.

#### CSMA 1-PERSISTENTE

Esta técnica se le denomina "1-persistente", dado que la estación transmite con una probabilidad de 1 cuando encuentra el canal desocupado.

Para evitar el tiempo ocioso en el canal, el CSMA 1-persistente puede ser usado. Si una estación quiere transmitir sensa el medio y sigue las siguientes reglas:

- 1) Si el medio está desocupado transmite
- 2) Si el medio está ocupado, continúa sensando hasta que el canal es detectado desocupado; luego inmediatamente transmite
- 3) Si hay una colisión, espera un período de tiempo y repite el paso 1.

En CSMA 1-persistente, si dos o más estaciones esperan para transmitir, seguramente ocurrirá una colisión.

#### CSMA NO PERSISTENTE

Una estación requiere, para poder transmitir, sensar el canal y seguir las siguientes reglas:

- 1) Si el medio está desocupado, transmite.
  - 2) Si el medio está ocupado, espera un período de tiempo corrido (determinado con probabilidad de distribución) y repite el paso 1.
- El uso de tiempos de retransmisión aleatorios reduce la probabilidad de colisiones.

#### CSMA P-PERSISTENTE

Sigue el siguiente procedimiento:

- 1) Si el medio está desocupado, transmite con probabilidad "p", y retarda una unidad de tiempo con probabilidad "1-p". La unidad de tiempo es típicamente igual al retardo de propagación máximo.
- 2) Si la transmisión es retardada una unidad de tiempo, se repite el paso 1.
- 3) El principal problema en esta técnica es la inestabilidad bajo cargas pesadas de transmisión.

**CSMA/CD**

El método de acceso CSMA/CD (Carrier Sense Múltiple Access with Collision Detection) resulta apropiado para una topología de canal. Con este estilo de operación es posible que dos estaciones intenten transmitir un frame sobre el canal al mismo tiempo, causando que el dato de ambas estaciones sea corrompido. Para evitar este problema, antes de transmitirse un frame, la estación origen debe primero verificar si un frame está en transmisión. Si existe algún frame en transmisión, la estación retarda su transmisión hasta que el frame actual ha sido transmitido y sólo después de esto intenta enviar su frame. Sin embargo si dos estaciones intentan transmitir al detectar simultáneamente que el medio está desocupado ocurre una colisión. En caso de que esto se presente se realiza lo siguiente:

1. Si una colisión es detectada durante la transmisión, inmediatamente cesa la transmisión del frame, y se transmite una señal de aviso (breve interferencia) para garantizar que todas las estaciones conocen que ha ocurrido una colisión
2. Después de transmitir la señal de aviso, espera un período de tiempo aleatorio y luego se intenta transmitir nuevamente usando CSMA.

La dirección de la estación origen es incluida como parte del encabezado del frame para que la estación receptora pueda directamente responder a la estación origen.

**CARACTERISTICAS DE CSMA/CD**

- El buen funcionamiento de la técnica CSMA/CD depende de la carga del canal.
- El acceso por CSMA/CD es probabilístico.
- Si la tasa de transmisión es muy alta (más de 10Mbps) la sobrecarga de la red tiende a ser baja.
- La transmisión de un frame sólo se intenta si el canal está inactivo.
- La probabilidad de que se presente una colisión es baja.

**ANILLO PARTICIONADO**

Esta técnica es sólo utilizada para controlar el acceso al medio de redes con topología de anillo. Primero se define un número fijo de dígitos binarios para que circulen continuamente alrededor del anillo desde una estación a otra. El anillo es adaptado para contener un número fijo de particiones, cada una formada por un conjunto de bits capaces de llenar un frame de tamaño fijo. En esta técnica cada partición contiene un bit para designar que la partición está llena o vacía. Todas las particiones son inicialmente marcadas como vacías. El funcionamiento es el siguiente:

- 1) Una estación con datos para transmitir debe dividirlos en frames con longitud mayor a la fija.
- 2) La estación debe esperar la llegada de una partición vacía. Al llegar lo marca como partición llena e inserta el frame en la partición completando las direcciones de origen y destino.
- 3) La partición que contiene el frame circula alrededor del anillo físico de una estación a otra.
- 4) Si una estación detecta que el frame le corresponde hace una copia del frame, modifica los bits de respuesta y repite el frame sin modificaciones para hacerlo llegar a la estación origen que marcará la partición como vacía.
- 5) Desde que la partición es declarada nuevamente como vacía, la estación está libre para volver a transmitir.

Cada estación conoce el número de particiones sobre el anillo y puede cambiar la propiedad del "bit lleno/vacío" conforme va pasando.

**VENTAJAS DE ACCESO AL MEDIO POR ANILLO PARTICIONADO**

- La principal ventaja es su simplicidad.
- La interacción con el anillo en cada nodo es minimizada.
- Mejora la rentabilidad del sistema.

**DESVENTAJAS DE ACCESO AL MEDIO POR ANILLO PARTICIONADO**

- Su principal desventaja es que es un despilfarrador de ancho de banda.
- Cada frame contiene más de 16 bits para el encabezado y esto ocasiona sobrecarga.
- Una estación puede enviar sólo un frame por cada recorrido.
- Si sólo una estación tiene frames para transmitir, muchas de las particiones circulan vacías.
- Se requiere de un nodo especial (monitor) para mantener la estructura de anillo básica.
- La transmisión de cada frame completo en el nivel de enlace, normalmente requiere de múltiples particiones, dado que cada partición puede llevar 16 bits de información usual.

**REGISTRO DE INSERION**

Es una técnica particularmente utilizable en LAN's con topología de anillo. En esta técnica, cuando un dispositivo conectado a una estación tiene información a transmitir, la carga dentro de un registro deslizante. Este registro debe ser del tamaño del frame de mayor longitud y es usado para controlar temporalmente los frames que pasan circuiando por el nodo. En adición, el nodo tiene un buffer para almacenar localmente los frames producidos.

En la técnica de registro de inserción se cuenta con un registro de cambio y un buffer en un nodo. El funcionamiento es diferente dependiendo si se tienen o no datos a transmitir:

- Si la estación tiene datos a transmitir
1. El frame a ser transmitido es localizado en el buffer de salida.
  2. Si la línea está desocupada y el registro de cambio está vacío, el frame puede ser transferido inmediatamente al registro de cambio.

3. Si el frame tiene una longitud de n bits, menor que el tamaño del frame más largo, y si menos de n bits están vacíos en el registro de cambio, los n bits son transferidos paralelamente a la parte vacía, inmediatamente adyacente a la parte llena, en el registro de cambio.
4. El puntero de entrada es ajustado de acuerdo con la operación.
  - Si la estación no tiene datos a transmitir
1. Cuando el anillo está desocupado, la parte más a la derecha del registro de cambio, desde la posición del puntero de entrada, indica que está desocupado
2. Cuando el dato llega desde el anillo, es insertado bit por bit en el registro de cambio y al mismo tiempo se va desplazando el puntero de entrada hacia la izquierda por cada bit.
3. El frame inicia con un campo de dirección. Tan rápido como la toda la dirección está en el registro, la estación puede determinar si ésta es la dirección destino.
4. Si no es la dirección el frame es sacado con sólo desplazar hacia afuera un bit sobre la derecha por cada nuevo bit que llegue por la izquierda, dejando el puntero de entrada estático.
5. Después de que ha llegado el último bit del frame, la estación continúa sacando bits fuera hacia la derecha hasta que el frame es completado.
6. Si durante este tiempo, no llegan frames adicionales, el puntero de entrada regresará a su posición inicial.
7. Por otro lado, un segundo frame iniciará a acumularse en el registro y es sacado como el primero.

**VENTAJAS DEL ACCESO AL MEDIO POR REGISTRO DE INSERION**

- La principal ventaja es que se logra la máxima utilización del anillo (comparado con otros métodos).
- La técnica de registro de inserción representa una forma eficiente de equidad.

**DESVENTAJAS DEL ACCESO AL MEDIO POR REGISTRO DE INSERION**

- El requerimiento de un mecanismo de purga. Esto se refiere que al permitir múltiples frames sobre el anillo se requiere del reconocimiento de una dirección anterior para el traslado del frame
- Si el campo de dirección de un frame está dañado, éste puede circular indefinidamente.
- Se presenta un retardo en cada estación.

**ELEMENTOS DE HARDWARE Y SOFTWARE**

Toda implementación de una red de comunicación de datos involucra una filosofía de diseño y un conjunto de componentes físicos que materializan dicha filosofía.

El software y hardware para la comunicación de datos, tienen diferentes formas y niveles de sofisticación. Sin embargo en los niveles básicos de transmisión, existen conceptos y dispositivos que son estándares y abarcan todas las plataformas de comunicación de datos en redes de computadoras.

HARDWARE	SOFTWARE
Modem	Sistema operativo
Concentrador	Sistema administrador de archivos y de base de datos
Multiplexor	Procesos de control de transacción
Gateway (Pasarela)	Métodos de acceso
Bridge (Puente)	Servidor de archivos
Routeador	Servidor de impresión
Repetidor	

**MODEM**

El nombre de éstos dispositivos se deriva de la función que realizan (MODulador/DEModulador). Los modems son dispositivos digitales destinados principalmente a la conversión de señales digitales a analógicas y viceversa. Pueden ser externos, independientes o residir dentro del gabinete del procesador central. Se distinguen por sincrónicos o asincrónicos dependiendo del tipo de mensaje a transmitir. Pueden tener diagnósticos residentes y disponer de mecanismos de detección y corrección de errores. La rapidez de reacción de los circuitos del modem, es una variable importante para los tiempos de respuesta de las terminales remotas. Pueden proveer la sincronización de la señal. A los modems que transmiten por dos líneas se les denomina bicanalizador. Cuando se combinan los modems con los multiplexores se les denomina modem multicanalizador. Cuando un modem tiene múltiples conexiones se le denomina modem maestro.

**TIPOS DE MODEMS**

- Modem óptico
- Modem de corto alcance
- Modem acústico
- Modem inteligente
- Modem digital
- Modem V.32

**MODEM SINCRONO**

Usan relojes sincronizados para alcanzar tasas de transmisión bastante altas sin afectar el uso de la línea. Sin embargo, la velocidad es limitada principalmente por la precisión y velocidad de los relojes. Una desventaja es que se requiere anexar información extra a los paquetes y esto ocasiona una cantidad mayor de datos de control que la requerida por los modems asíncronos.

**MODEM ASINCRONO**

Los modems asíncronos son relativamente baratos. Los relojes son usados para determinar donde un bit termina y donde comienza otro. La tolerancia en sus variaciones de ritmo de transmisión no es controlada, lo cual es aceptable debido a que cada 9 ó 10 bits transmitidos se ocupan para reorganizar el transmisor y el receptor. Esto genera un 10% de datos de control en la sobrecarga. Este tipo de modems no hacen mención a cualquier forma de detección o corrección de errores.

**MODEM OPTICO**

Un modem óptico transmite datos sobre líneas de fibra óptica. Este tipo de modem convierte señales eléctricas en pulsos de luz, en el fin transmisor, para ser transmitidos sobre líneas de fibra óptica. En el fin receptor, el modem recibe los pulsos de luz y los vuelve a convertir a señales eléctricas que la computadora pueda entender. Este modem opera usando modos de transmisión síncrona o asíncrona.

**MODEM DE CORTO ALCANCE**

Este tipo de modem utiliza cable par trenzado para transmitir señales eléctricas cuando las distancias abarcan aproximadamente 20 Km o menos. Este modem transmite a velocidades de 9,600 bps hasta los 5 Km, 4,800 bps hasta los 10 Km y finalmente 2,400 bps en distancias de 10 a 20 Km. Normalmente este tipo de modem es usado para conectar computadoras entre diferentes oficinas en un mismo edificio.

**MODEM ACUSTICO**

Este es el tipo de modem más viejo y es también llamado acoplador acústico. Este modem es una interfase con cualquier teléfono para dialogar con otra computadora.

**MODEM INTELIGENTE**

Un modem inteligente puede ejecutar funciones por el uso de un lenguaje de comandos. El lenguaje puede ser accesado a través de un programa de comunicación y poner en funcionalidad al modem. Con este tipo de modems se puede contestar o llamar a otros modems, conmutar parámetros de comunicación, proporcionar volumen a los comunicadores de los modems y muchas otras funciones bajo software de control.

**MODEM DIGITAL**

Si en vez del uso de conversión analógica, los circuitos de comunicación usan transmisión digital, por lo tanto un modem digital es usado. Este tipo de modem modifica los bits digitales que requiera. Esta función es convertir señales digitales EIA-232 dentro de señales más convenientes para la transmisión.

**MODEM V.32**

Este modem trabaja en full duplex a una velocidad de 9,600 bps sobre líneas telefónicas normales. Este modem es típicamente usado para respaldar líneas telefónicas dedicadas sobre redes. Esto significa que si la transmisión de datos es interrumpida, un modem V.32 y una línea de teléfono normal, pueden ser usada como un remplazo temporal para la línea dedicada.

**CAPACIDADES DE LOS MODEMS**

- Velocidad.- Todos los modems están diseñados para operar en una velocidad específica a un rango discreto de velocidades. Puede ser tipo switch o bajo un programa de control. Las velocidades de los modems bajo el control de una microcomputadora son típicamente de 300, 1200, 2400, 4800 y 9600 bps.
- Opciones de teléfono.- La mayoría de los modems nuevos pueden reaccionar al anillo indicador sobre la línea y automáticamente responder una llamada. Algunos modems pueden marcar números telefónicos bajo un programa de control. Si el modem encuentra una línea ocupada, este inmediatamente remarca el número hasta que la conexión es hecha.
- Autoprueba.- El modo autoprueba incluye una prueba de ciclo de regreso, en la cual la señal de salida del modem es circulada de regreso al mismo lugar, chequea el diagnóstico de memoria y prueba las transmisiones modem-a-modem.
- Datos sobre voz.- Permiten transmisión simultánea de voz y datos. Esto permite tomar lugar una conversación mientras un dato es inicialmente transmitido sobre la misma línea telefónica.
- Compatibilidad.- Existe una gran variedad de modems estándar y el adherirse a un estándar aceptado ampliamente ayuda a establecer compatibilidad en los modems.
- Costo.- El precio de los modems es mucho menor que el medio de transmisión.

**CONCENTRADOR**

Es un dispositivo inteligente de línea compartida basado en un microprocesador, cuyo objetivo es concentrar líneas de comunicación para economizar: modems, líneas, adaptadores y puertos de conexión central. El concentrador permite a múltiples dispositivos compartir circuitos de comunicación. Debido a que los concentradores disponen de un microprocesador, pueden ser usados uno a la vez, a diferencia de los multiplexores que se usan por pares. Su uso puede ser local o remoto. El uso de concentradores reduce el trabajo de sondeo al procesador central, dado que en lugar de invitar a transmitir a "n" estaciones, sólo tiene que invitar a un concentrador, es decir, "n-1" secuencias de sondeo son evitadas.

Los concentradores pueden ser programados para transmitir datos sólo desde terminales que están activas. Gracias al uso de memoria en los concentradores se pueden modificar la velocidad de las terminales, el formato del dato, los procedimientos de comunicaciones y el número de terminales. También puede ser programado para hacer más eficiente la utilización de enlaces de datos de alta velocidad a través de la compresión de datos.

#### **FUNCIONES PRINCIPALES DE LOS CONCENTRADORES**

- Sondeo de estaciones
- Conversión de protocolos
- Conversión de códigos
- Elaboración de formatos de mensajes
- Recolección local de los datos
- Conversión de velocidades
- Compactación de datos
- Control de errores
- Reingreso automático de los datos capturados.
- Diagnósticos

#### **MULTIPLEXOR**

Un multiplexor es un dispositivo que permite la transmisión de múltiples señales sobre un solo medio de comunicación, es decir, permiten el remplazo de múltiples líneas de transmisión de baja velocidad, con una sola línea de transmisión de alta velocidad.

La principal finalidad de un multiplexor es proporcionar al usuario reducción en sus costos de comunicación. El uso de multiplexores debe ser considerado cuando un gran número de terminales de datos están comunicadas dentro de una área geográfica similar o cuando varias líneas corren en paralelo en cualquier distancia. Varias líneas de comunicación entran al multiplexor desde una estación. El multiplexor combina el dato desde todas las líneas que llegan y lo transmite por una línea a otro multiplexor en el fin receptor. El multiplexor recibe los datos separados y los distribuye a lo largo de las líneas terminales de salida. El número de líneas que entran al multiplexor desde una estación es el mismo número de líneas que salen hacia las terminales de la parte remota.

En general, cualquier dispositivo que transmite o recibe una cadena de datos en forma serial, puede ser considerado un candidato para realizar la función de multiplexor.

#### **TIPOS DE MULTIPLEXORES**

- Multiplexor inverso
- Multiplexor T-1
- Multiplexor multipuertos
- Multiplexor de fibra óptica

#### **MULTIPLEXOR INVERSO**

Proporciona una ruta de datos de alta velocidad entre computadoras. Este toma una línea de alta velocidad de una computadora emisora y la separa en múltiples líneas de baja velocidad. Estas son recombinadas por otros multiplexores inversos antes de la conexión con la computadora receptora.

#### **MULTIPLEXOR T-1**

Es un tipo especial de multiplexor combinado con una unidad de servicios de alta capacidad, que maneja los fines de un enlace T-1. Un enlace T-1 es un enlace de comunicación que transmite a 1544 Mbps. Además los circuitos T-1 pueden cargar 24 canales de 64 kbps.

#### **MULTIPLEXOR MULTIPUERTOS**

Combina un módem y un equipo de multiplexaje por división de tiempo en un solo dispositivo. Toda la línea del módem puede ser de varias velocidades de transmisión.

#### **MULTIPLEXOR DE FIBRA OPTICA**

Toma múltiples canales de datos, con una velocidad de 64 Kbps por canal y multiplexa estos canales en una línea de fibra óptica de 14 Mbps.

#### **GATEWAY (PASARELA)**

Un gateway es un dispositivo cuyo servicio es interconectar sistemas diferentes(LAN-WAN). Su propósito no es realizar, a través de los sistemas, intercambio de mensajes personales o conferencias; mas bien, es hacer conexiones de tiempo real entre un usuario y una de las varias estaciones remotas.

Los gateway proporcionan servicios de login remoto a estaciones distantes desde un mismo punto, permitiendo a los usuarios establecer contacto con varias estaciones, sin tener que rendirles cuentas. Con un Gateway es posible conectar una red Novell basada en PC's con una red SNA o una red Ethernet y hacer la compartición de recursos transparente al usuario.

Los componentes funcionales de un Gateway son las Interfases de redes y los procesos de translación. Los gateway no encaminan paquetes de datos dentro de la red.

#### **FUNCIONES DE UN GATEWAY**

- Conversión de protocolos.
- Capacidad de operación desde la capa de transporte hasta la capa de aplicación.
- Conversión de código de carácter.

- Conversión de atributos de pantallas.
- Mapeo de teclas para generar diferentes códigos de teclas cuando otras teclas son presionadas.
- Mostrar transparencia al usuario de las conexiones que realiza.

### BRIDGE (PUENTE)

Un bridge es un dispositivo usado para conectar dos redes locales dentro de una proximidad cerrada una de otra, usando el mismo protocolo y el mismo medio de transmisión, generalmente. Un bridge debe ser configurado para conocer o tener la habilidad de aprender cuales direcciones estan sobre cada red para saber cuales frames pasar. En otras palabras, un bridge regula el tráfico desde una red a otra basado en las direcciones de destino de cada frame. Los dos tipos de bridges más usados son:

- Bridge transparente
- Bridge fuente de rutas

En general la función de un bridge se concreta en lo siguiente:

1. Leer todos los frames transmitidos sobre una red "A" y aceptar esas direcciones para una red "B".
2. Usando el protocolo de control de acceso al medio de la red "B", retransmitir los frames hacia "B".
3. Hacer lo mismo para el tráfico de la red "B" a la red "A".

### BRIDGE TRANSPARENTE

Un bridge transparente monitorea el flujo de los datos y automáticamente construye un directorio de direcciones. Debido a la velocidad del flujo de los datos en una LAN, los bridges transparentes pueden rápidamente determinar el origen de las direcciones de las estaciones de trabajo sobre la LAN. Cuando se encuentra un paquete que tiene una dirección de destino que no está incluida sobre la LAN, el bridge sacará el paquete. Si una red tiene más de un bridge transparente, otros bridges pueden recibir copias múltiples del paquete, luego cada bridge que recibió una copia del paquete examinará la dirección destino y si no está incluida dentro de su subred sacará el paquete. Como un paquete fluye a través de una red a su destino, cada bridge graba la dirección de la localización de cada estación permitiendo al proceso ser automatizado desde que el primer paquete es entregado a su dirección.

### BRIDGE FUENTE DE RUTAS

Los bridges que funcionan bajo esta filosofía requieren de una estación de trabajo para iniciar la acción. Una estación de trabajo inicia el proceso de descubrimiento de una ruta cuando éste intenta iniciar una sesión con otra estación. En el proceso de descubrimiento de la ruta, el paquete de la estación origen, es transmitido por cada bridge, lo cual permite al paquete atravesar todas las rutas posibles entre dos estaciones. Como los paquetes son retransmitidos por cada bridge, la información de encaminamiento es agregada al campo de información de encaminamiento en cada paquete. Cuando los paquetes llegan a su destino el proceso es invertido y la respuesta es retransmitida de regreso a la estación origen. Posteriormente el software residente en la estación origen, determina la ruta óptima.

### CONSIDERACIONES PARA LOS BRIDGES

- Un bridge no modifica el contenido o el formato del frame que recibe, sino lo encapsula con un encabezado original.
- Puede ser internamente concebido por el uso de otro protocolo.
- Debe contener suficiente espacio en el buffer para satisfacer las demandas. Debido a que por un período de tiempo la llegada de los frames puede ser más rápida que su transmisión.
- Debe poseer inteligencia de direccionamiento y de encaminamiento. Por lo menos debe conocer cuales son las direcciones de cada red para saber cuales frames pasar.
- Puede conectar más de dos redes
- Los bridges también pueden ser usados para interconectar redes con diferentes protocolos de comunicación de bajo nivel.
- Proporcionan transparencia en protocolos de alto nivel

### VENTAJAS DEL USO DE BRIDGES

- Rentabilidad.- Por el uso de los bridges, la red puede ser particionada dentro de unidades autosuficientes.
- Seguridad.- Es proporcionado por el aislamiento de porciones de la red.
- Conveniencia.- Puede realizar enlaces más sencillos que con la utilización de medios físicos.
- Cobertura geográfica.- Pueden enlazar redes ampliamente separadas.
- Algunos mantienen el estado de información de otros bridges, permitiéndoles desempeñar una función de encaminamiento dinámico.
- Algunos cuentan con un mecanismo de control que maneja el buffer de paquetes en cada bridge para solventar la congestión.

### RUTEADOR

Un ruteador es un dispositivo que puede ser usado para interconectar redes que usan el mismo protocolo, sobre el mismo o diferente medio físico de comunicación. Algunos ruteadores son capaces de soportar dos o más protocolos y son llamados ruteadores multi-protocolos. Los ruteadores pueden ser usados para ligar geográficamente redes dispersas al mismo tiempo.

Los ruteadores examinan los frames que fluyen sobre una red y si están destinados a otra red, consulta su tabla de ruteo. La tabla contiene una lista de varios nodos residentes sobre redes diferentes, las rutas entre nodos y el número de nodos intermedios requeridos para que un frame llegue a su destino sobre otra red. El ruteador normalmente seleccionará la ruta con el menor número de "saltos".

Los ruteadores son similares a los bridges en el sentido de que ellos pueden operar en un modo local o modo remoto o soportar ambos modos.

### RUTEADOR EN MODO LOCAL

Un ruteador de modo local funciona como bridge inteligente y tiene la habilidad para convertir paquetes de frame desde un tipo de red al formato del frame soportado por una red de área local diferente.

**RUTEADOR EN MODO REMOTO**

Un ruteador en modo remoto permite a las LAN ser interconectadas en una vía común, facilitando la comunicación y habilitando a las estaciones de trabajo sobre redes diferentes para comunicarse con cualquier otra vía o una red WAN.

**VENTAJAS DE UN RUTEADOR**

- Los ruteadores determinan las rutas de datos más eficientes entre dos redes.
- Los ruteadores ignoran las topologías y los niveles de acceso usados por la red.
- Los ruteadores no están restringidos por el medio de comunicación o por el protocolo de comunicación.

**REPETIDOR**

Los repetidores son dispositivos que consisten de amplificadores y otros equipos con el propósito de reestablecer la señal básica atenuada. Los repetidores están espaciados a distancias iguales a lo largo de la línea, del sistema de radio u otro medio de transmisión.

Un repetidor no reconoce mensajes, sólo señales. Este regenera fielmente y retransmite cualquier señal que recibe. El objetivo de los repetidores, en ésta instancia, es permitir una longitud de cable más larga. Los repetidores se colocan a una distancia determinada por la configuración utilizada. Estas distancias pueden ser hasta de 500 m.

**SISTEMA OPERATIVO**

El sistema operativo de red, es el software necesario para integrar los diversos componentes de una red en un sistema, al cual pueda tener acceso un usuario final. Las funciones principales de un sistema operativo son:

- Funciones de interfase
- Funciones de proceso
- Funciones de administración
- Funciones de administración de archivos

El sistema operativo ejecuta todas estas funciones de una forma ampliamente transparente al programa de aplicación y al programador. El sistema operativo ejecuta varias funciones al ejecutar un programa, pero éstas funciones son llamadas sin que el programador las mande llamar explícitamente. El sistema operativo es el administrador de todos los sistemas de computación. Este es cargado cuando el sistema es iniciado y porciones de éste permanecen residentes en memoria con el propósito de que siempre sea capaz de proporcionar sus funciones de administración e interfaces.

**FUNCIONES DE INTERFASE DEL SISTEMA OPERATIVO**

- Para usuarios
- Para sistemas de Entrada/Salida
- Para sistemas de archivo

**FUNCIONES DE PROCESO DEL SISTEMA OPERATIVO**

- Procesos de ejecución
- Procesos de inicio/paro
- Impone prioridades de proceso
- Previene procesos desde la interferencia con cualquier otro.
- Permite multiprocesamiento
- Permite multitarea

**FUNCIONES DE ADMINISTRACION DEL SISTEMA OPERATIVO**

- Administra memoria
- Administra el sistema de entrada/salida y dispositivos
- Administra el acceso al CPU
- Administra el acceso de usuario a través de provisiones de seguridad.

**FUNCIONES DE ADMINISTRACION DE ARCHIVOS DEL SISTEMA OPERATIVO**

- Localiza espacio en disco
- Mantiene los directorios en el disco
- Maneja atributos de archivo
- Provee archivo de seguridad

**CRITERIOS DE SELECCION DE UN SISTEMA OPERATIVO DE RED**

- Compatibilidad con las aplicaciones.- Las aplicaciones deben poder ejecutarse en un entorno de red.
- Manejabilidad.- Deben poderse crear nuevos usuarios, crear directorios, agregar y quitar aplicaciones, actualizar y reconfigurar el sistema de manera sencilla, etc.
- Facilidad de uso.- Debe ser de fácil manejo para el usuario.
- Fiabilidad.- Que de preferencia disponga de un sistema tolerante a fallos, con el cual se permita seguir trabajando en caso de que el sistema principal deje de funcionar.
- Seguridad.- Debe proporcionar niveles de seguridad para cualquier situación en formas de derechos de acceso (lectura, escritura, etc) o passwords.

- **Funcionamiento.**- Es necesario que la velocidad del sistema operativo de red esté en consonancia con los recursos de hardware disponibles.
- **Ampliación.**- El sistema operativo debe proporcionar la facilidad de ampliar la red sin tener que recurrir a medios costosos, tanto técnicos como en equipos.

## SISTEMA ADMINISTRADOR DE ARCHIVOS Y DE BASE DE DATOS

Un sistema de administración de base de datos (DBMS, DataBase Management System) organiza los datos dentro de registros, organiza los registros dentro de archivos y proporciona acceso a la base de datos sobre uno o más claves de acceso. Un DBMS también proporciona un mecanismo para relacionar un archivo con otro.

Un sistema de administración de archivos (FMS, File Management System) proporciona de un subconjunto de las capacidades encontradas en un DBMS. Un FMS está básicamente orientado al manejo de archivos. Los sistemas de administración de bases de datos y archivos proporcionan servicios de datos para procesos de aplicación.

## PROCESOS DE CONTROL DE TRANSACCION

Mientras que los DBMS proveen a una aplicación con acceso a datos, el acceso de aplicación a terminales u otros nodos es proporcionado por un proceso de control de transacción (TCP, transaction Control Process), también referido como un "sistema de control de mensajes" (MCS, Message Control System). Sin un TCP, un método de acceso de comunicación de datos puede parcialmente cumplir esta función. El TCP habilita diferentes terminales para servir como interfase de múltiples aplicaciones, aislandolas de las diferencias físicas entre terminales y entre nodos de redes. El TCP usa diferentes métodos de acceso, permitiendo de esta manera el acceso a diferentes tipos de terminales.

### FUNCIONES DE UN TCP

- Mantenimiento del contexto.
- Administración de memoria.
- Transacción de rutas.
- Seguridad y estadísticas.
- Prioridad de mensajes.
- Interfase de operaciones entre aplicaciones y terminales.

### METODOS DE ACCESO

Generalmente los métodos de acceso separan la aplicación de las características físicas del dato o dispositivo que está accediendo.

Los métodos de acceso de comunicación de datos dan a los sistemas de usuario fácil acceso a los dispositivos terminales. Los métodos de acceso liberan a los usuarios de los atributos específicos de los dispositivos de las terminales y proporcionan conexión, desconexión y servicios de transferencia de datos a las aplicaciones. Existen dos formas principales de acceso:

- Acceso por conexión aplicación-terminal
- Acceso por terminal

### ACCESO POR CONEXION APLICACION-TERMINAL

Una de las funciones de un método de acceso es proporcionar conexiones aplicación-terminal. Esta es realizada al tener una combinación de aplicaciones y una combinación de terminales disponibles. El método de acceso sirve como un conmutador para conectar solicitudes de terminales con la propia aplicación.

### ACCESO POR TERMINAL

Debido a que un método de acceso separa el programa de aplicación del acceso lógico de la terminal, pueden ser usados con o sin TCP, dependiendo del ambiente. El método de acceso desempeña pocas funciones cuando usa un TCP, dado que algunas funciones, tales como el encaminamiento de mensajes y edición de datos, son desempeñados por el TCP.

El primer requerimiento de acceso a una terminal desde un programa es el conectarlos. El método de acceso sirve como un intermediario en este caso. Desde que una conexión ha sido realizada, existe una ruta de comunicación, por lo tanto la terminal y la aplicación pueden intercambiar datos.

Sin un TCP el método de acceso realiza la conexión entre un programa de aplicación y una terminal. La conexión puede ser estática; es decir, la aplicación y la terminal son conectadas a cada otro y la terminal puede correr sólo las transacciones proporcionadas por la aplicación particular. Para que el usuario de la terminal pueda acceder otro proceso de aplicación, la terminal debe primero ser desconectada desde su acceso de aplicación actual y luego reconectada al nuevo.

## SERVIDOR DE ARCHIVOS

La función primaria de un servidor es proporcionar un servicio a los usuarios de la red. Los dos tipos más comunes son los servidores de archivos y los servidores de impresión. Los servidores de archivos permiten a los usuarios compartir uno o más drives de discos. Para proporcionar efectivamente compartición de disco, el servidor de archivo debe desempeñar las siguientes funciones:

- Controlar el acceso actual
- Convenio de licencia
- Optimización de acceso
- Rentabilidad
- Acceso transparente
- Interfases
- Transferencia de archivos
- Comunicación nodo-a-nodo
- Administración de red
- Seguridad
- Interfases para programas de aplicación (API)

**SERVIDOR DE IMPRESION**

Este servidor es requerido debido a que, generalmente, sólo un trabajo puede ser impreso a la vez y debido a que muchos usuarios pueden simultáneamente requerir el uso de la misma impresora. Un servidor de impresora es un sistema de software llamado "spooler". Un spooler es una interfase entre usuarios que requieren el servicio de impresión e impresoras. El spooler maneja la impresión de trabajos por la posición que tengan en la cola de impresión, establece prioridades de impresión de trabajos, imprime trabajos desde la cola sobre la impresora disponible y proporciona un administrador de impresión de trabajos.

Cualquier recurso compartido requiere administración y control, y la administración de impresión es una de las más difíciles de manejar debido a que la relativamente baja velocidad de las impresoras, ocasionan alta demanda de su uso.

**BIBLIOGRAFIA: REDES DE COMUNICACION DE DATOS**

- Business data communications.  
David A. Stamper, 1991. (3ª edición).
- Computer communications.  
K.G. Beauchamp, 1990. (2ª edición).
- Computer mediated communications.  
Matthew Rapaport, 1991.
- Contemporary data communications.  
Emilio Ramos, AL Schroeder, 1994.
- Data communication, computer networks and open systems.  
Fred Halsall, 1992.
- Data communication.  
Lynn A. De Noia, 1987.
- Data communications networking devices.  
Gilbert Held, 1992. (3ª edición).
- Diseño de redes locales.  
Hopper/Temple/Willanson, 1989.
- Local and metropolitan area networks.  
William Stallings, 1993. (4ª edición).
- Network topology optimization.  
Roshan L. Sharma, 1990.
- Telecommunications protocols and design.  
John D. Spraging, 1991.

## CAPITULO 3: ARQUITECTURAS DE REDES

## INTRODUCCION

La mayoría de las redes se organizan en una serie de capas o niveles, con el objeto de reducir la complejidad de su diseño. El número de capas, el nombre, contenido y función de cada una varían de una red a otra. Sin embargo, en cualquier red, el propósito de cada capa es ofrecer servicios a las capas superiores. Para poder realizar la comunicación entre capas, se requiere del uso de reglas y convenciones que se les denomina protocolos. Normalmente cada capa pasa la información de datos y control, a la capa inmediatamente inferior hasta llegar a la capa localizada en la parte más baja de la estructura o viceversa. De bajo de la primera capa está el medio físico, a través de la cual se realiza la comunicación real.

Entre cada par de capas adyacentes hay una interfase, la cual define los servicios y operaciones que la capa inferior ofrece a la capa superior.

Al conjunto de capas y protocolos se le denomina arquitectura de la red, las especificaciones de ésta deberán contener la información suficiente que le permita al diseñador escribir un programa o construir el hardware correspondiente a cada capa.

## PROTOSCOLOS

Un protocolo es un conjunto de reglas que gobierna el intercambio de datos entre dos entidades, independientemente si el formato empleado para los datos son paquetes o mensajes. Un protocolo formaliza la interacción entre capas de redes mediante la estandarización del uso del canal de comunicación. Los protocolos tienen que estar relacionados con los métodos usados para:

- Inicialización y terminación de intercambio de datos.
- Sincronización de transmisores y receptores
- Detección y corrección de errores transmitidos
- Formateo y codificación de datos.

La definición de un protocolo puede ser comparada con la definición de un lenguaje, dado que contiene:

- Sintaxis.- Se refiere a la definición de un vocabulario, que consiste en:
  - Dar un formato a los datos
  - Codificación de datos
  - Asignación de niveles a la señal
- Gramática.- Se refiere a la colección de reglas de procedimientos.
- Semántica.- Se refiere a la información de control para coordinación y manejo de errores.

## PROTOSCOLOS (CARACTERISTICAS Y FUNCIONES)

Las funciones y características principales de un protocolo son las siguientes:

CARACTERISTICAS	FUNCIONES
<ul style="list-style-type: none"> <li>• Directo/Indirecto</li> <li>• Monolítico/Estructurado</li> <li>• Simétrico/Asimétrico</li> <li>• Estandar/No estandar</li> <li>• Transparencia</li> <li>• Independencia de código</li> <li>• Múltiples configuraciones</li> <li>• Especificaciones</li> </ul>	<ul style="list-style-type: none"> <li>• Segmentación y reensamblado</li> <li>• Encapsulación</li> <li>• Control de conexión</li> <li>• Control de flujo</li> <li>• Control de errores</li> <li>• Sincronización</li> <li>• Secuenciación</li> <li>• Direccionamiento</li> <li>• Multiplexaje</li> <li>• Servicios de transmisión</li> </ul>

## PROTOSCOLO DIRECTO / INDIRECTO

Si dos sistemas comparten un enlace punto a punto o multipunto, las entidades de esos sistemas pueden comunicarse en forma directa, al pasar la información de control y los datos directamente entre entidades sin la intervención de un agente activo. En cambio, en el caso en que dos entidades no compartan la misma red conmutada, pero estén indirectamente conectadas a través de dos o más redes, la comunicación es indirecta.

## PROTOSCOLO MONOLITICO / ESTRUCTURADO

Un protocolo es monolítico cuando incluye en el formato de datos la lógica del enlace para coordinar la comunicación con otra red.

Un protocolo es estructurado cuando exhibe una jerarquía o una estructura de capas o niveles, donde las funciones más primitivas son implementadas en las entidades de niveles más bajos para proporcionar servicios a las entidades de los niveles más altos.

## PROTOSCOLO SIMETRICO / ASIMETRICO

Un protocolo es simétrico si permite la comunicación entre entidades iguales.

Un protocolo es asimétrico cuando puede ser ordenado por la lógica de un intercambio o por el deseo de mantener una de las entidades o sistemas tan simple como sea posible.

**PROTOCOLO ESTANDAR / NO ESTANDAR**

Un protocolo estandar es compartido por varios sistemas.

Un protocolo no estandar es construido para una situación de comunicación específica o para un modelo de computadoras particular.

**TRANSPARENCIA (Protocolos)**

Un protocolo debe tener la habilidad de transmitir cualquier patron de bits como dato proporcionando la seguridad de que haya sido aceptado correctamente.

**INDEPENDENCIA DE CODIGO (Protocolos)**

Un protocolo debe tener la habilidad suficiente para transmitir datos con cualquier forma de codificación, como ASCII, EBCDIC, etc.

**MULTIPLES CONFIGURACIONES (Protocolos)**

Un protocolo debe tener la capacidad de múltiples configuraciones, para permitir al diseñador del sistema planear una topología de red que vaya de acuerdo con las aplicaciones.

**ESPECIFICACIONES (Protocolos)**

Existen cinco elementos principales para la especificación de un protocolo:

- 1) El servicio que debe ser proporcionado por el protocolo.
- 2) La noción acerca del ambiente en el cual el protocolo es ejecutado.
- 3) El vocabulario de mensajes usado para implementar el protocolo.
- 4) La codificación (formato) de cada mensaje en el vocabulario.
- 5) Las reglas de los procedimientos para mantener la consistencia del mensaje intercambiado.

**SEGMENTACION Y REENSAMBLADO (Protocolos)**

Segmentación es la acción de fragmentar los datos en unidades lógicas de tamaño limitado (bloques, mensajes, paquetes), de acuerdo con el nivel en que trabaje el protocolo, para intercambio de datos entre dos entidades.

El reensamblado es la contraparte de la fragmentación y consiste en unir nuevamente los datos fragmentados dentro de mensajes apropiados al nivel de aplicación.

**VENTAJAS DE LA SEGMENTACION Y REENSAMBLADO (Protocolos)**

- La red de comunicación sólo acepta bloques de datos de un máximo tamaño limitado.
- El control de errores es más eficiente con unidades lógicas de datos más pequeñas.
- Hace más equitativo el acceso al medio en sistemas de transmisión compartida.
- Una unidad lógica de datos pequeña permite a las entidades contener buffers pequeños.

**DESVENTAJAS DE LA SEGMENTACION Y REENSAMBLADO (Protocolos)**

- Cada unidad lógica de datos contiene una cantidad fija de información de control.
- Al llegar una unidad lógica de datos puede generar una interrupción que debe ser atendida.
- Más tiempo es invertido para procesamientos más pequeños.
- Más numerosas unidades lógicas de datos.

**ENCAPSULACION (Protocolos)**

Es la adición de información de control al dato a transmitir. Cada unidad lógica de datos, no solo contiene datos, sino también información de control. Existen algunas unidades lógicas de datos que sólo contienen información de control y no de datos. La información de control se usa en tres categorías generales:

- a) Dirección.- Indica la dirección del transmisor y/o receptor.
- b) Código de detección de error.- Una pequeña parte del frame chequea la secuencia del frame, si es incluida la detección de error.
- c) Control de protocolo.- Se incluye información adicional para implementar funciones propias del protocolo.

**CONTROL DE CONEXION (Protocolos)**

Se refiere a la función de establecer una asociación lógica o conexión entre entidades. Para esto se consideran tres fases:

- 1) Establecimiento de conexión
- 2) Transferencia de datos
- 3) Terminación de la conexión

**CONTROL DE FLUJO (Protocolos)**

El control de flujo es una función desempeñada por una entidad receptora para limitar la cantidad o rango de datos que le son enviados por una entidad transmisora. El control de flujo es también requerido sobre protocolos orientados a aplicación.

**CONTROL DE ERROR (Protocolos)**

Esta función, al igual que el control de flujo, debe ser desempeñada por protocolos de diferentes niveles, para garantizar que el dato sea intercambiado con éxito entre una capa y otra.

**SINCRONIZACION (Protocolos)**

Para realizar la sincronización los protocolos de las entidades deben conocer el estado de la entidad con la que se desea entablar comunicación, con el fin de realizar funciones como inicialización, transferencia de datos y terminación.

La dificultad de la sincronización radica en el hecho de que una entidad conoce el estado de la otra sólo en virtud de las unidades lógicas de datos recibidas, las cuales no llegan instantáneamente, sino que se toman algún tiempo para viajar del transmisor al receptor, corriendo el riesgo de perderse.

**SECUENCIACION (Protocolos)**

La función de secuenciación de un protocolo consiste en identificar el orden en el cual las unidades lógicas de datos, que contienen datos, fueron enviadas para numerarlas. Esta función sólo tiene sentido en un contexto de transferencia de datos orientada a conexión. La secuenciación sirve a tres propósitos principales:

- 1) Entrega ordenada
- 2) Control de flujo
- 3) Control de errores.

**DIRECCIONAMIENTO (Protocolos)**

Para comunicar dos entidades sobre un enlace que no sea punto-a-punto, ambos deben ser capaces de identificarse. Una distinción es generalmente hecha por medio de nombres, direcciones y rutas. El nombre especifica qué es el objeto, una dirección específica donde está el objeto y una ruta específica como llegar al objeto. Para éste propósito se usan:

- Nombres locales.- Nombre por el cual una entidad es identificada dentro de su propio sistema.
- Nombres globales.- Nombre por el cual una entidad es conocida fuera de su sistema.

Para asignar un nombre único a una entidad se debe considerar:

- La estructura del nombre.- Puede ser jerárquica (red.sistema.entidad) o bien plana (cada entidad tiene un nombre global único).
- Conocimiento del nombre.- Una entidad sólo puede solicitar o enviar datos a una estación cuyo nombre conozca.
- Nombres de conexión.- Se usa en transmisión orientada a conexión en la fase de transferencia de datos.
- Nombres de puertos.- Se refiere al nombre de una entidad global
- Nombres de grupos.- Nombre que se refiere a más de una entidad o puertos.

**MULTIPLEXAJE (Protocolos)**

Se refiere al multiplexaje de transferencia de datos dentro de una entidad. Esto se puede realizar por el uso de nombres de conexión, lo cual permite el uso de múltiples conexiones simultáneas. También se usa para el mapeo de conexiones desde un nivel a otro.

**SERVICIOS DE TRANSMISION (Protocolos)**

Un protocolo proporciona servicios a las entidades que lo usan como:

- Prioridad.- Puede ser asignada sobre mensajes base o sobre una conexión base.
- Grado de servicio.- Dado que ciertas clases de datos pueden requerir una mínima cantidad de información procesada, en una unidad de tiempo, o un tiempo máximo de retardo.
- Seguridad.- los mecanismos de seguridad y los accesos restringidos pueden ser invocados.

**PROTOCOLOS ORIENTADOS A BIT**

Un protocolo orientado a bit transmite datos como una cadena de bits. Para permitir al receptor reconocer donde un mensaje inicia y termina en la cadena de bits es usado un pequeño patrón de bits único ( se le llama bandera). Por supuesto que ese patrón de bits, puede ser parte de un dato de usuario, pero algo tiene que hacerse para garantizar que el patrón siempre sea interpretado apropiadamente. Si una bandera de formateo es definida como una serie de 6 bits "1" encerrados en bits "0" (0111110), las series de 6 unos adyacentes en los datos de usuario deben ser interceptados. Esto puede ser aceptado por la inserción de un cero extra después de cada serie de 5 bits "1's" en el dato de usuario.

El receptor ahora puede correctamente detectar la estructura de las banderas en la cadena de bits por la inspección del primer bit después de cada serie de cinco "1's": si es cero éste debe ser borrado, de lo contrario el patrón es parte del delimitador del frame.

Dentro de los protocolos orientados a bit más usados se tienen:

- Protocolo SDLC ( Synchronous Data Link Control )
- Protocolo HDLC ( High-level Data Link Control )

**PROTOCOLO SDLC (Synchronous Data Link Control)**

El protocolo SDLC opera en modo full-duplex sobre líneas no conmutadas en configuración punto a punto o multipunto normalmente. En este protocolo los datos son transmitidos en una dirección alrededor de la configuración de lazo cerrado. En todas las configuraciones, incluyendo punto a punto, una estación es designada como la estación primaria y las demás como estaciones secundarias. La estación primaria controla el enlace y determina a cual estación se le permite transmitir.

La unidad básica de transmisión en SDLC es el frame. SDLC es un protocolo posicional. Esto significa que cada campo, excepto el campo de datos, tienen una longitud específica y una posición relativa a los campos adyacentes, es decir, no se usan caracteres especiales de control para delimitar el dato o los encabezados del mensaje.

**FRAME DEL PROTOCOLO SDLC / HDLC**

**CAMPO DE BANDERA**

Es usado para indicar el inicio y el fin del frame. El patron de bits usado es " 01111110 ", el cual es reservado.

**CAMPO DE DIRECCION**

Campo de 8 bits, es decir, un máximo de 256 direcciones únicas son posibles.

**CAMPO DE CONTROL**

Campo de 8 ó 16 bits que contiene los comandos y respuestas requeridas para controlar la operación en un enlace de datos. Determina como se controla el proceso de comunicación. Este campo puede estar en uno o tres formatos:

Formato de Información	Formato supervisor	Formato no numerado
Es usado para transmitir datos a un usuario final entre los dos dispositivos. Se usan los campos N(S) y N(R) para numerar los mensajes: N(S) = Número de secuencia de envío de 3 bits N(R) = Número de secuencia de recibido de 3 bits	Tiene 2 bits disponibles. En éste caso se ejecutan funciones de control tales como el reconocimiento de frames, la respuesta de retransmisión de frames o suspensión temporal, etc. N(R) = Número de secuencia de recibido SC = Código Supervisor P/F = Bit Poll/Final	Tiene 5 bits disponibles para identificar la función de control. Es usado para ejecutar inicialización de enlaces, desconexión de enlaces y otras funciones que controlan el enlace. P/F = Bit Poll/Final UN = Código no numerado

**CAMPO DE DATOS**

Este campo es opcional. Siempre es omitido por los frames supervisor, es opcional para los frames no numerados y está usualmente presente en los frames de Información. El número de bits debe ser múltiplo de 8.

**CAMPO VERIFICADOR DE SECUENCIA DEL FRAME**

Campo de 16 bits que verifica la secuencia del frame

**CARACTERISTICAS DEL PROTOCOLO SDLC**

- Detección de error: SDLC usa CRC-16 para detectar errores.
- Transparencia: Dado que el patron usado para las banderas debe evitarse en el campo de datos se inserta un bit "0" después de encontrar 5 bits "1's" consecutivos en el dato.
- Direccionamiento: SDLC es limitado en capacidades de direccionamiento dado que sólo puede usar 8 bits para direccionar.
- Independencia de código: Todos los protocolos orientados a bit proporcionan independencia de código. La única restricción es que se debe manejar el campo de datos por octetos.
- Configuraciones: SDLC soporta casi todas las combinaciones del half-duplex, full-duplex, punto a punto, multipunto, líneas dedicadas no conmutadas y todo tipo de configuraciones cerradas.
- Eficiencia: SDLC maneja encabezados fijos. Cada frame de Información tiene 48 bits usados por banderas, control, direccionamiento y detección de error.
- Crecimiento: SDLC tiene la capacidad de ser flexible para satisfacer los nuevos desarrollos.

**PROTOCOLO HDLC (High-level Data Link Control)**

El protocolo HDLC está orientado a bit. HDLC soporta modos de transmisión half-duplex y full-duplex; configuraciones punto a punto y multipunto y líneas conmutadas y no conmutadas. HDLC define tres tipos de estaciones, tres configuraciones de enlace y tres modos de transferencia de datos.

Estaciones	Configuraciones	Modos de transferencia
1. Estación primaria	1. Configuración desbalanceada	1. Modo de respuesta normal
2. Estación secundaria	2. Configuración balanceada	2. Modo de respuesta asíncrono
3. Estación combinada	3. Configuración simétrica	3. Modo balanceado asíncrono

El formato del frame usado por HDLC es el mismo que el usado por SDLC.

La operación de HDLC consiste en el intercambio de frames de información, frames supervisores y frames no numerados entre una estación primaria y una secundaria o entre dos estaciones primarias.

**ESTACION PRIMARIA**

Tiene la responsabilidad de controlar la operación del enlace. Los frames emitidos por la estación primaria son llamados "comandos", los cuales son enviados a las estaciones secundarias sobre el canal. La estación primaria mantiene un enlace lógico separado con cada estación secundaria sobre la línea.

**ESTACION SECUNDARIA**

La estación secundaria opera bajo el control de la estación primaria. Los frames emitidos por la estación secundaria son llamados "respuestas". Estas mantienen sólo una sesión con la estación primaria y no tienen responsabilidad de control sobre el enlace. Una estación secundaria no se puede comunicar directamente con las demás, lo tienen que hacer a través de la estación primaria.

**ESTACION COMBINADA**

La estación combinada combina las operaciones de la estación primaria y la estación secundaria. Una estación combinada puede emitir "comandos" y "respuestas". Estas mantienen una sesión con otra estación combinada.

**CONFIGURACION BALANCEADA**

Es usada sólo en operación punto a punto. Esta configuración consta de dos estaciones combinadas y soporta transmisión half-duplex y full-duplex sobre líneas conmutadas o no conmutadas. Cada estación tienen la responsabilidad del control de enlace.

**CONFIGURACION DESBALANCEADA**

Usada en operaciones punto a punto o multipunto. Esta configuración consiste de una estación primaria y una o más secundarias y soporta modos de transmisión full-duplex y half-duplex en líneas conmutadas o no conmutadas. La configuración es llamada desbalanceada debido a que la estación primaria es responsable del control de cada estación secundaria y de establecer y mantener el enlace.

**CONFIGURACION SIMETRICA**

Es muy poco usada. La configuración proporciona dos configuraciones de estaciones desbalanceadas con operación punto a punto independientes entre sí. Cada estación contiene un estado primario y uno secundario, por lo tanto cada estación es considerada lógicamente dos estaciones; una primaria y una secundaria. La estación primaria transmite comandos a la estación secundaria en el otro fin del canal y viceversa.

**MODO DE RESPUESTA NORMAL (NRM)**

Esta es una configuración desbalanceada. La estación primaria puede iniciar la transferencia de los datos a la secundaria, pero la secundaria sólo puede transmitir datos en respuesta a un sondeo de la primaria.

**MODO DE RESPUESTA ASINCRONO (ARM)**

Esta es una configuración desbalanceada. En este modo, la estación secundaria puede iniciar la transmisión sin permiso explícito de la estación primaria. La estación primaria aún retiene la responsabilidad de la línea, incluyendo inicialización, recuperación de error y desconexión lógica.

**MODO BALANCEADO ASINCRONO**

Esta es una configuración balanceada. Cada estación combinada puede iniciar la transmisión sin recibir permiso de otra estación combinada.

**PROTOCOLOS ORIENTADOS A CARACTER**

En un protocolo orientado a carácter el uso de una estructura mínima es inevitable sobre la cadena de bits. Si el número de bits por carácter está fijo a  $n$  bits (7 ó 8 típicamente), toda la comunicación toma lugar en múltiplos de  $n$  bits. Esas unidades de datos luego son usadas para codificar los datos de usuario y códigos de control (STX, inicio de texto y ETX fin de texto) que pueden servir como delimitadores y pueden ser usados para encerrar el dato de usuario.

Si una secuencia de datos es transmitida, se debe poner atención para que los delimitadores no ocurran accidentalmente en los datos de usuario. Esto puede solucionarse si todos los caracteres de control son precedidos por un carácter de escape de enlace de datos (DLE). El código DLE es interpretado por el receptor como un código de control que cambia a "off" cualquier significado especial del primer carácter que lo siga. El receptor borra el primer código DLE que ve en la cadena de caracteres y pasa el siguiente carácter no interpretado. Sólo si el significado especial de un STX o ETX no es suprimido por un carácter predecesor DLE, éste se interpreta como un delimitador.

Un ejemplo de un protocolo orientado a carácter es:

- Protocolo BSC ( Binary Synchronous Communications )

**PROTOCOLO BSC (Binary Synchronous Communications)**

El protocolo BSC o BISYNC usa un modelo de protocolo síncrono orientado a carácter. El protocolo BSC transmite datos en un bloque a la vez. Para esto se requiere que el transmisor y el receptor estén en sincronía entre sí. Para mantener la sincronización, se usan modems síncronos los cuales contienen sus relojes sincronizados durante la transmisión. Sólo tres códigos de datos son soportados por BSC: Transcode de 6-bits, ASCII y EBCDIC.

Para sincronizar los modems del receptor y del transmisor se transmiten uno o más caracteres de sincronización (SYN) en el inicio de cada bloque de transmisión. Para mantener la sincronización en transmisión de bloques muy largos se insertan caracteres SYN, en intervalos regulares. Las configuraciones posibles en BSC son:

- BSC en modo punto a punto
- BSC en modo multipunto

Una limitación de BSC es que es esencialmente un protocolo half-duplex, por lo que cada mensaje transmitido debe ser reconocido por el receptor antes de que el próximo mensaje sea enviado. Además esta característica hace al BSC ineficiente en aplicaciones conversacionales.

**BSC EN MODO PUNTO A PUNTO**

Se usa la contención para determinar cual estación tiene garantizado el derecho de transmitir. Para transmitir un mensaje una estación debe ganar el control del enlace. Las conexiones punto a punto pueden ser establecidas ya sea por líneas conmutadas o líneas dedicadas.

**BSC EN MODO MULTIPUNTO**

En este caso una estación es designada como supervisor y las otras como estaciones secundarias. El supervisor está al tanto de todas las estaciones sobre la línea; las estaciones secundarias ven un enlace punto a punto con el supervisor. El supervisor mantiene absoluto control sobre el enlace; las estaciones secundarias permanecen pasivas y monitoreando el enlace. Las estaciones secundarias sólo pueden transmitir cuando se lo permite el supervisor. Es decir, todas las transferencias son iniciadas por el supervisor.

**CARACTERISTICAS DEL PROTOCOLO BSC**

- Control del mensaje.- Cada bloque transmitido puede tener un campo de encabezado opcional para control de mensajes como: información de ruta, prioridad y tipo de mensaje. El inicio y el fin del mensaje son identificados por la inserción de los caracteres de control.
- Control de error.- El control de error es ya sea CRC o LRC con VRC. CRC es usado con EBCDIC, con SBT y con ASCII, LRC y VRC son usados con ASCII no transparente.
- Transparencia.- La transmisión transparente con BSC es algo engorrosa, incluyendo la inserción de caracteres extras en el mensaje, dado que incrementan la sobrecarga del protocolo.
- Direccionamiento.- Es posible direccionar con BSC un rango ancho, desde 1 a 7 caracteres en longitud.
- Independencia de código.- No existe independencia de código debido a que sólo maneja SBT, ASCII y EBCDIC.
- Configuraciones.- Son limitadas a los modos multipunto y punto a punto sobre líneas conmutadas o líneas dedicadas.
- Eficiencia.- BSC es muy eficiente con respecto al número de caracteres de control requeridos para transmitir el mensaje.
- Crecimiento.- Es limitado a configuraciones punto a punto y multipunto.

**PROTOCOLOS ORIENTADOS A BYTE-CONTADOR**

Las banderas de un protocolo orientado a bit y los caracteres de control de un protocolo orientado a carácter son usados para estructurar una secuencia de cadenas de datos dentro de fragmentos muy largos. Esta estructuración se realiza para indicar al receptor donde una cadena de datos inicia y termina. En un protocolo orientado a byte-contador un método con poca diferencia es usado. En un lugar conocido, después del carácter de control STX, el transmisor incluye el número preciso de bytes que el mensaje contiene. Un carácter ETX es ahora superfluo y las técnicas orientadas a bit y a carácter ya no son requeridas. La mayoría de los protocolos de uso actual son de este tipo.

Un protocolo orientado a byte-contador es orientado a carácter. Este protocolo recibe este nombre debido a que el número de caracteres en el mensaje es dado en el encabezado del mensaje requerido.

**VENTAJAS**

- Es transparente
- Es obvio donde inicia y termina el mensaje
- Cualquier patrón de bits puede ser representado dentro de la cadena de datos.
- Tiene un buen desempeño.

**PROTOCOLOS DE LLAMADA (X.25)**

El protocolo X.25 es usado para proporcionar una interfase entre los equipos de usuario (DTE's) y una red de comunicaciones (DCE) conmutada por paquetes. El protocolo X.25 está orientado a conexión, es decir, se debe establecer un circuito virtual entre los DTE's, antes de la transmisión de cualquier dato. El protocolo X.25 encuentra su principal aplicación en redes de área ancha (WAN's), aunque también puede ser usada por algunas LAN's via un gateway. El protocolo X.25 es un conjunto de 3 protocolos:

- Protocolo en el nivel físico
- Protocolo en el nivel de control de enlace
- Protocolo en el nivel de red

**CIRCUITO VIRTUAL EN X.25**

Un circuito virtual (CV) es un circuito transparente bidireccional con rutas de flujo controlado entre un par de puertos físicos o lógicos. Sus características principales son:

- Establecimiento y terminación de un circuito virtual
- Facilidad de registro
- Transferencia de datos en forma transparente
- Recuperación de errores
- Alineación con los servicios del modelo OSI.

**PROTOCOLO EN EL NIVEL FISICO (X.25)**

Un protocolo entre las entidades del nivel físico en el DTE y el DCE. Su función es proporcionar transmisión bit serial full-duplex, rutas de transmisión síncronas, etc.

**PROTOCOLO EN EL NIVEL DE CONTROL DE ENLACE (X.25)**

Un protocolo entre las entidades del nivel de control de enlace en el DTE y el nodo de la red.

La capa de enlace de datos garantiza transferencia de información rentable. La estructura del frame, los procedimientos de control y el control de flujo están basados sobre el protocolo HDLC. Para proporcionar una transmisión de información secuencial libre de errores entre

un DTE y un DCE, se usa una versión de las técnicas ARQ para corrección de errores. Se emplea el modo de operación balanceado asíncrono. Ambos fines del enlace de comunicación pueden operar sincrónicamente, por lo que ambos pueden iniciar la transmisión de comandos y de respuestas en cualquier momento.

### PROCOLO EN EL NIVEL DE RED (X.25)

Un protocolo entre las entidades del nivel de red conmutada por paquetes en el DTE y el nodo de la red.

En la capa de red dos formas de conexión son proporcionadas: las llamadas virtuales y las llamadas virtuales permanentes. Las llamadas virtuales son semejantes a las llamadas telefónicas ordinarias. Una llamada virtual permanente es parecida a una línea dedicada con conexión permanentemente establecida, entonces cada entidad de comunicación puede enviar la información cuando sea requerida.

### SERVICIOS DE X.25

El protocolo X.25 provee de acceso a los siguientes servicios de redes, los cuales pueden ser proporcionados por una red privada o pública de datos:

- 1) Circuitos virtuales conmutados (SVC's, Switched Virtual Circuits).- También se les conoce como llamadas virtuales. Un SVC es una asociación temporal entre dos DTE's, la cual es iniciada por la señalización de un DTE de una solicitud de llamada a la red.
- 2) Circuitos virtuales permanentes (PVC, Permanent Virtual Circuits).- Un PVC es una asociación permanente existente entre dos DTE's, los cuales no requieren iniciar una llamada o terminarla.

### FACILIDADES DE X.25

El protocolo X.25 es diseñado para proporcionar facilidad de servicios como:

- Selección de comunicación rápida
- Numeración de secuencia extendida
- Retransmisión de paquetes al DTE
- Grupo de usuarios cerrado
- Reserva de cargas de llamadas
- Redirección de la llamada
- Modificación de la dirección de la línea de llamada

### PROTOCOLOS ENTRE REDES (TCP/IP)

Muchos sistemas de computadoras conectados a LAN's o WAN's usan el protocolo entre-redes (IP, Internet Protocol) y el protocolo de control de transmisión (TCP, Transmission Control Protocol) en los niveles más altos. Esos protocolos son colectivamente conocidos como TCP/IP. El conjunto de protocolos TCP/IP fué desarrollado al principio de los 70's por el Depto. de Defensa Norteamericano para conformar la base de un sistema de redes interconectadas de diversos tipos (Internet).

TCP/IP incluye protocolos orientados a red y protocolos que soportan aplicaciones. TCP proporciona los servicios a nivel de transporte y IP a nivel de red. El protocolo TCP/IP define básicamente la forma de interconectar subredes y encaminar el tráfico entre ellas. Los protocolos que conforman la jerarquía de TCP/IP son:

- |                  |                    |                 |                  |
|------------------|--------------------|-----------------|------------------|
| • Protocolo IP   | • Protocolo TCP    | • Protocolo FTP | • Protocolo SNMP |
| • Protocolo SMTP | • Protocolo TELNET | • Protocolo UDP | • Protocolo NSP  |

### PROCOLO ENTRE-RED (IP)

El servicio de entre-red se refiere a un conjunto de redes interconectadas. Cada red componente soporta comunicación limitada por el número de dispositivos conectados. Las redes son conectadas por gateways. El protocolo IP no está orientado a conexión. Es por esto, que proporciona lo que se conoce como un servicio de "sin conexión poco fiable", es decir, puede ser que una unidad de datos nunca sea entregada a su destino o, sea entregada fuera de la secuencia en la que fue enviada.

### DIRECCIONAMIENTO EN IP

En el campo de direccionamiento existen 2 campos de 32 bits cada uno, que proporcionan las direcciones de las terminales origen y destino. Estas direcciones identifican a una subred y a una terminal conectada a ella en forma universal, es decir, son únicas dentro del entorno Internet para una cierta terminal.

Las clases de dirección IP para una determinada estación, dependerá de la capacidad de crecimiento que pueda desarrollar la subred a la cual se conecta. Cuando una estación cambia de subred, también debe cambiar su dirección IP. Por otro lado si una estación está conectada a más de una subred, tendrá una dirección IP por cada conexión.

Debido a que las direcciones a las cuales se podrán comunicar las terminales son direcciones físicas, se hace necesario implantar un mecanismo para trasladar la dirección IP de una estación a su dirección física de conexión. Este mecanismo se denomina resolución de dirección.

### OPERACION EN IP

Considerando dos estaciones A y B sobre redes diferentes en la entre-red.

- 1) La estación A envía datos a la estación B. Se inicia el proceso en la estación A.
- 2) El módulo IP en la estación A construye un datagrama que consta de datos desde el protocolo TCP más información de control usada por IP.
- 3) El datagrama es enviado a través de la red de A al gateway apropiado en forma transparente.
- 4) Cuando el gateway recibe el datagrama éste debe hacer una decisión de ruteo.
- 5) Existen 2 posibilidades;

- Si la estación destino B está conectada directamente a una de las redes, a la cual el gateway está conectado, el módulo IP en el gateway envía el datagrama a través de la red a B.
- Si al llegar a la estación uno o más gateways adicionales deben ser recorridos, el datagrama se envía a través de la red al próximo gateway sobre la ruta apropiada.

Es decir, el módulo IP en una estación o gateway debe tener información que le permita tomar decisiones de ruteo. Además debe estar habilitado para emplear el protocolo de acceso a la red apropiado para poder enviar el dato a través de las redes a las cuales está conectado.

### PROTOCOLO DE CONTROL DE TRANSMISION

El protocolo TCP provee un mecanismo rentable para el intercambio de datos entre procesos en diferentes computadoras. El protocolo garantiza que el dato sea entregado libre de error, en secuencia, sin pérdidas y sin duplicación. El servicio básico proporcionado por TCP es la transferencia de datos entre usuarios por medio de protocolos de transferencia de archivos (FTP) o protocolos de transferencia de correo simple (SMTP).

Las unidades de datos del TCP tienen un formato de 20 bytes para todos los intercambios.

### FUNCIONES DE TCP

- Fragmentación de mensajes
- Retransmisión de segmentos
- Reordenamiento y establecimiento de prioridades
- Definición del formato de los datos
- Establecimiento y terminación de conexiones.
- TCP tiene la finalidad de lograr un servicio orientado a conexión y proporcionar fiabilidad de transferencia de datos.

### SERVICIOS DE TCP

- Calidad de servicio.- TCP permite al usuario de transporte especificar la calidad de servicio de transmisión proporcionada. TCP intentará optimizar el uso del IP y los recursos de red ya sea manejando prioridades o manejando retardos.
- Entrega urgente.- TCP intentará hacer la transmisión facilitando la transferencia de datos tan rápido como sea posible en caso de ser urgente. En el receptor TCP notificará al usuario del recibo del dato urgente.
- Seguridad.- Una clasificación o rango de seguridad puede ser usado para etiquetar los datos proporcionados por TCP. Esto puede influenciar la ruta tomada para el dato y el como sea encapsulado.

### OPERACION DE TCP

- Los datos son pasados desde un usuario de transporte al TCP.
- TCP encapsula esos datos dentro de un segmento, que contiene datos de usuario más información de control.
- Para cumplir con una transferencia de datos rentable, los segmentos que salen son numerados secuencialmente y posteriormente reconocidos por número por el módulo TCP del destino.
- Si los segmentos llegan fuera de orden, pueden ser reordenados basándose en el número de secuencia. Si un segmento se pierde, no será reconocido, y el módulo transmisor del TCP lo retransmitirá.

### PROTOCOLO DE TRANSFERENCIA DE ARCHIVOS(FTP)

El propósito de FTP es transferir un archivo o parte de un archivo, desde un sistema a otro, bajo el comando de un usuario de FTP. Típicamente FTP soporta 3 estructuras de archivos y 4 tipos de datos. La comunicación de los usuarios con FTP es mediada por el sistema operativo, el cual contiene manejadores de Entrada/Salida.

El protocolo FTP debe interactuar con 3 entidades:

- Debe ser interfase de usuario para aceptar solicitudes desde un usuario interactivo o posiblemente un programa.
- Debe estar disponible para comunicarse con otro FTP para cumplir con la transferencia de archivos (esto es implementado por el uso de servicios de TCP).
- Para transferir un archivo, FTP debe estar habilitado para conseguir el archivo. Para esto se requiere de una interfase en el sistema de administración de archivos local.

### ESTRUCTURAS DE ARCHIVOS SOPORTADAS POR FTP

FTP permite al usuario especificar la estructura del archivo a emplear. Existen 3 estructuras posibles:

- Archivo no estructurado.-Puede contener cualquier tipo de datos (binarios o texto) y es transferido por entre las dos entidades de protocolos FTP como una cadena de bits transparente.
- Archivo estructurado.-Consiste de una secuencia de registros de tamaño fijo de un tipo definido de datos. Dado el contenido de estos archivos, son transferidos normalmente como una cadena de bloques de tamaño fijo. También pueden ser transferidos en forma compresada.
- Archivos de acceso aleatorio.-Son registros comprimidos de tamaño variable. Normalmente a los registros se les llama páginas y al archivo, archivo paginado. Cada registro/página tiene un encabezado asociado el cual incluye longitud, el tipo de campo e información posicional que indica la posición del registro/página relativo al contenido del archivo total.

### TIPOS DE DATOS SOPORTADOS POR FTP

FTP permite al usuario especificar el tipo de datos a emplear. Los tipos de datos son:

- Binario de 8 bits
- Texto ASCII
- Texto EBCDIC

- Binario de longitud variable

#### SERVICIOS DE FTP

- Compresión de datos para reducir costos de comunicación
- Provee de mecanismos para controlar el acceso de los usuarios a los archivos sobre el sistema por medio de identificación de password.
- Permite el uso de una cadena transparente de bits para permitir el envío de cualquier orden de datos o archivos de texto.

#### OPERACION DE FTP

Si un usuario sobre un sistema A desea acceder un archivo de un sistema B, entonces se deben comunicar el FTP de A y el FTP de B de la siguiente manera:

- 1) El usuario se conecta al FTP local para transferir todo o parte de un archivo.
- 2) Hay 3 posibilidades:
  - a) El usuario de A desea que un archivo de B sea transferido al sistema A. En este caso se usará el acceso local del usuario para guardar el contenido del archivo.
  - b) El usuario de A tiene un archivo local preparado para enviarlo al sistema B
  - c) El usuario de A puede solicitar que un archivo sea intercambiado entre un sistema B y un sistema C.

#### PROTOCOLO DE ADMINISTRACION DE RED SIMPLE (SNMP)

El protocolo SNMP (Simple Network Management Protocol) no está relacionado con los servicios de usuario, sino con la administración de todos los protocolos de comunicación dentro de cada estación y los diferentes elementos de interconexión de redes que proporcionan dichos servicios. Es decir, administra todo el ambiente de interconexión de redes.

El protocolo SNMP ha sido definido para ayudar al manejador de la red a realizar las funciones de administración de desempeño y fallas.

SNMP es un protocolo de aplicación, por lo que una plataforma de comunicación estándar debe ser usada para permitir que los mensajes asociados sean transferidos concurrentemente con el mensaje relacionado con el servicio de usuario.

El papel de SNMP es proporcionar un camino para el administrador de procesos (en el administrador de estación), para intercambiar mensajes relacionados con la administración, ejecutando procesos de administración en los diferentes elementos que se utilizan: gateways, estaciones, etc.

#### PROTOCOLO DE TRANSFERENCIA DE CORREO SIMPLE (SMTP)

El protocolo SMTP (Simple Mail Transfer Protocol) proporciona las bases para facilitar un correo electrónico por red. El correo electrónico es probablemente el servicio más ampliamente usado asociado con redes de computadoras. Típicamente un correo electrónico se ejecuta sobre un sólo sistema, donde cada usuario tiene un buzón.

Cuando un usuario se manifiesta sobre un sistema, puede enviar correo con sólo colocar un mensaje en el buzón de otro usuario y recibir correo al leer los mensajes de su propio buzón. Generalmente los buzones son mantenidos por el sistema administrador de archivos. Cada buzón es un directorio que puede contener archivos de texto, que son mensajes. Los usuarios pueden preparar mensajes con ayuda de un editor o un procesador de palabras.

El protocolo SMTP proporciona un mecanismo para transferir mensajes entre sistemas separados, pero no es responsable de aceptar correo de usuarios locales o de distribuir el correo recibido a los usuarios locales. Con SMTP un usuario puede enviar correo no sólo a otro usuario sobre el mismo sistema, sino a cualquier usuario en la entre-red. El estándar SMTP no especifica la interfase de usuario.

#### PROTOCOLO TELNET

TELNET es un protocolo usado para ligar terminales a aplicaciones. TELNET actualmente es implementado en dos módulos:

- 1) TELNET usuario.- Interactúa con el módulo E/S de la terminal sobre el sistema. Este módulo convierte las características reales de las terminales al estándar de la red y viceversa.
- 2) TELNET servidor.- Interactúa con los procesos y aplicaciones. Este módulo actúa como un manejador de terminal suplente, dado que las terminales remotas aparecen como locales para el proceso o aplicación.

TELNET es accesado por medio de un sistema operativo local, ya sea por un proceso de aplicación o por un usuario en alguna terminal. TELNET presta servicios para iniciar el acceso a un usuario con el sistema operativo de una máquina remota, con el fin de iniciar la ejecución de un programa o proceso sobre la máquina para luego interactuar con éste como si el usuario terminal/proceso estuviera conectado/ejecutando sobre la misma máquina.

#### CARACTERISTICAS DE TELNET

TELNET puede ser caracterizado como sigue:

- Especifica una terminal estándar en la red, es decir, las características de las terminales específicas son mapeadas dentro del estándar. Esto permite que las terminales de diferentes vendedores se puedan conectar a una variedad de estaciones.
- Especifica el protocolo entre terminales y estaciones. Esto permite que ciertas características de las terminales sean negociadas (ancho de línea, tamaño de página, tipo de enlace, etc.)
- Proporciona intercambio de datos fiablemente, por medio de TCP
- Permite al usuario de una terminal controlar una aplicación en una estación remota como si el usuario fuera un usuario local de la estación.

#### OPERACION DE TELNET

- 1) Todos los comandos y datos que entran en la terminal del usuario, son pasados por el sistema operativo local al módulo TELNET usuario.

- 2) El módulo TELNET usuario pasa los comandos y datos al módulo TELNET servidor, usando un servicio prestado por TCP.
- 3) El módulo TELNET servidor entrega los comandos y datos dados por el usuario, a través del sistema operativo local, al proceso interactivo.
- 4) Cualquier dato salido del proceso interactivo es regresado por el mismo camino para ser desplegado sobre alguna terminal del usuario o para ser interpretados por los procesos de aplicación del usuario.

### PROTOCOLO PARA USUARIOS DE DATAGRAMA (UDP)

TCP/IP proporciona un protocolo de transporte sin conexión conocido como UDP. Dado la forma que usa UDP para transmitir la parte de dato de usuario, de su respectiva unidad de datos, al protocolo de la aplicación apropiada, es necesario incorporar un campo de dirección adicional en el encabezado de cada unidad de datos.

Debido a que UDP es un protocolo sin conexión, sólo una unidad de datos asociada con el protocolo es transmitida en el campo de datos de usuario de un datagrama IP:

- El puerto origen.- Es la dirección origen del puerto del protocolo de aplicación (16 bits).
- El puerto destino.- Es la dirección destino del puerto del protocolo de aplicación.
- El campo de longitud.- Es una cuenta del número total de octetos en el datagrama completo (UDP), incluyendo el encabezado.
- El campo "checksum" en el encabezado UDP relaciona el datagrama UDP completo.

UDP es usado ya sea cuando no se requiere corrección de error o para un simple intercambio de un mensaje solicitud/respuesta corto entre dos protocolos de aplicación.

### PROTOCOLO SERVIDOR DE NOMBRE (NSP)

El protocolo NSP garantiza que los nombres de los procesos de aplicación sean únicos dentro de un ambiente particular, al proporcionar el manejo de localización de nombres de los procesos de aplicación.

## ARQUITECTURAS DE REDES

Dado que las redes de computadoras actuales han incrementado su capacidad y complejidad, las funciones relacionadas con la transmisión de datos son desempeñadas por hardware complejo, software y firmware operando en los diferentes dispositivos que componen la red. Para realizar esta tarea con mayor facilidad se usa una arquitectura de red.

Una arquitectura de red es un plan comprensible que gobierna el diseño de los elementos de software, hardware y firmware que componen una red de computadoras. La meta de una arquitectura es ocultar la complejidad de la comunicación, a los usuarios de la red, es decir, hacer una red transparente. El acceso transparente a los servicios que proporcione la red, es cumplido por dividir sus tareas dentro de una serie de niveles o capas. Cada capa cumple un conjunto de tareas específicas, las cuales son luego ofrecidas como servicio a los usuarios de la capa.

### CONSIDERACIONES DE LAS ARQUITECTURAS DE REDES

- Definir los protocolos de las capas, las interfaces concretas y las interfaces abstractas entre las capas funcionales.
- Proporcionar al usuario variedad de opciones de configuración de redes
- Permitir a los usuarios cambiar una configuración con relativa facilidad.
- Proporcionar reglas para el desarrollo de nuevos productos. Las reglas pueden cambiar.
- Definir una base que permita a la arquitectura evolucionar y cambiar, para soportar nuevas tecnologías.

## ARQUITECTURA OSI

La arquitectura proporcionada por OSI (Open Systems Interconnection) es organizada por capas. En cada capa de una arquitectura OSI se define un servicio y una o más especificaciones de protocolos. El modelo OSI divide una red dentro de 7 capas de funciones y capacidades específicas. Cada capa desempeña un conjunto de funciones diferentes y son independientes entre sí. Las capas de la arquitectura OSI son:

- Capa de aplicación
- Capa de presentación
- Capa de sesión
- Capa de transporte
- Capa de red
- Capa de enlace de datos
- Capa física

### SERVICIO

Un servicio consiste de la definición de los servicios específicos de una capa proporcionados a las capas que lo rodean, pero no dice cómo esos servicios deben ser proporcionados.

### ESPECIFICACION DE UN PROTOCOLO

Una especificación de protocolo describe el formato de la unidad de datos intercambiada y especifica los procedimientos que debe desempeñar una capa al intercambiar esas unidades de datos en proporción de los servicios de aquella capa.

## CAPA FISICA DE OSI

Es la capa más baja del modelo OSI y sus funciones características son:

- Intercambiar entre máquinas de comunicación señales eléctricas, señales ópticas o señales de radio.
- Direccional cables, conectores, modems y otros dispositivos usados para conectar máquinas.
- A la capa física no le interesa cuantos bits componen una unidad de datos o que significado tiene el dato transmitido.
- En esta capa el transmisor, simplemente transmite una señal y el receptor la detecta.

## CAPA DE ENLACE DE DATOS DE OSI

Esta capa se encarga de la transmisión de unidades de datos sobre un circuito físico. Las funciones que operan en la capa de enlace de datos permiten al dato ser transmitido, en una manera relativamente libre de error, algunas veces sobre circuitos físicos que tienden a transmitir erróneamente. Otras características son:

- A esta capa le interesa conocer como los bits son agrupados dentro de una colección.
- Desempeña funciones de sincronización con respecto a las falias que ocurren en la capa física.
- Implementa mecanismos de detección de error.
- Puede desempeñar procedimientos de control de flujo

## CAPA DE RED DE OSI

La capa de red suple un servicio que capas más altas emplean para mover bits desde un nodo final (nodo origen o destino de un dato) a otro, donde los bits pueden fluir a través de cualquier nodo intermedio (nodo intermediario que dirige los datos). Sus características principales son:

- El servicio básico de la capa de red es proporcionar transferencia transparente de datos entre entidades de transporte.
- La capa de red proporciona tareas más complejas de transmisión de datos entre dos nodos de la red cualquiera.
- El servicio de la capa de red es responsable de establecer, mantener y terminar conexiones.

## CAPA DE TRANSPORTE DE OSI

La capa de transporte se construye sobre los servicios de la capa de red y las capas que le anteceden para formar la capa más alta de un servicio rentable de transporte de datos fin-a-fin. Sus características principales son:

- El propósito principal es proporcionar un mecanismo rentable para el intercambio de datos entre procesos en sistemas diferentes.
- La capa de transporte garantiza que las unidades de datos sean liberadas de error, en secuencia, sin pérdidas o duplicaciones.
- Las funciones de la capa de transporte encabezan el direccionamiento de los procesos.
- Puede controlar la tasa en la cual los mensajes fluyen a través de la red para controlar y prevenir una congestión.
- La capa de red proporciona un servicio fin-a-fin que los programas puedan usar para mover datos hacia atrás y hacia adelante entre ellos.

## CAPA DE SESION DE OSI

La capa de sesión proporciona el mecanismo para controlar el diálogo entre aplicaciones. Otras características son:

- Es la capa más baja de las asociadas con la aplicación de programas.
- Proporciona el medio para que dos procesos de aplicaciones establezcan y usen una conexión llamada "sesión".
- La capa de sesión define 3 tipos de diálogo: Interacción en dos caminos simultáneos, Interacción en dos caminos alternados e interacción en 1 camino.
- Establece puntos de sincronización dentro de un diálogo, permitiendo a un diálogo ser interrumpido y resumido desde un punto de sincronización.
- Proporciona el mecanismo de recuperación.

## CAPA DE PRESENTACION DE OSI

La capa de presentación tiene que ver con la sintaxis de los datos intercambiados entre entidades de aplicación. Su propósito es resolver diferencias en formatos y representación de datos.

La capa de presentación define la sintaxis usada entre entidades de aplicación y proporciona procedimientos para la selección y modificación de la representación a usar.

Los tipos de sintaxis pueden ser:

- 1) Sintaxis abstracta.- Es una definición formal del contenido de la información de los dos programas de datos en intercambio.
- 2) Sintaxis concreta local.- Define como el contenido de la información de los datos es actualmente representado en un sistema de computación.
- 3) Sintaxis de transferencia.- Define como el contenido de la información de los datos es codificado para transmisión sobre la red.

## CAPA DE APLICACION DE OSI

La capa de aplicación proporciona un medio para que los procesos de aplicaciones accedan el ambiente OSI. Esta capa contiene funciones de administración y generalmente hace uso de mecanismos para soportar aplicaciones distribuidas.

Esta capa proporciona todas las funciones relacionadas con la comunicación de sistemas no proporcionadas por las capas inferiores como: administración de trabajo, transferencia de archivos, correo electrónico e intercambio de datos particulares de la aplicación.

## OPERACION DE LA ARQUITECTURA DE OSI

El flujo de datos en una arquitectura basada en el modelo de OSI es

- 1) El dato de usuario es presentado, por una aplicación de usuario, a la capa de aplicación
- 2) La capa de aplicación agrega información de control del protocolo que usa (encabezado), al dato de usuario y luego pasa su encabezado y el dato de usuario a la siguiente capa inferior.
- 3) Cada capa debe encapsular el dato.
- 4) El dato encapsulado se transporta a través del circuito de comunicaciones a la estación receptora.
- 5) Posteriormente el dato viaja de las capas inferiores a las capas superiores y el encabezado creado por la capa de transmisión equivalente es usado por la capa de recepción equivalente para invocar una función de servicio.
- 6) Conforme un dato pasa hacia las capas superiores se le van quitando los encabezados después de que han sido usados. A este proceso se le llama desencapsular.

En cualquier capa de la estructura de OSI, dos formas básicas de operación son posibles:

- 1) Operaciones orientadas a conexión
- 2) Operaciones de modo sin conexión

### ENCAPSULACION EN OSI

El proceso de "encapsulación" en OSI, consiste en agregar encabezados al dato de usuario. En ocasiones éste término es usado inadecuadamente, dado que el dato desde las capas superiores sólo tienen encabezados agregados en un lado extremo del frame. La única capa que encapsula el dato completamente es la capa de enlace de datos, la cual agrega información en el encabezado y en la parte final.

### OPERACION EN OSI ORIENTADA A CONEXION

Se establece una conexión lógica entre entidades equivalentes antes del intercambio de datos. Sus características principales son:

- Tiempo de vida claramente distinguible
- Realiza la comunicación en tres pasos: establecimiento, transferencia de datos y terminación.
- Identificadores de conexión.
- Navegación de unidades de datos
- Mapeo de conexión a través de la red
- Direcccionamiento abreviado.
- Encaminamiento entre redes usualmente fijo
- Proporciona responsabilidad.

### OPERACION DE MODO SIN CONEXION EN OSI

Cada unidad de dato transmitida es independiente de unidades de datos previas o siguientes y no se establece una conexión lógica. Sus características principales son:

- Servicio de acceso simple
- Independencia de unidades de datos
- Unidades de datos autosuficientes
- Mapeo delimitado fin-a-fin
- Direcccionamiento completo con unidad de datos.
- Puede usar ruteo alternado
- Responsabilidad limitada

### COMUNICACION ENTRE CAPAS EN OSI

Los usuarios se comunican con el proveedor de servicios a través de una dirección o de un identificador comúnmente conocido como "sitio de acceso de servicio" (SAP, Service Access Point). A través del uso de 4 tipos de transacciones, llamadas primitivas, el proveedor de servicios coordina y maneja el proceso de comunicación entre los usuarios. Las primitivas son:

- 1) Solicitud.- Se utiliza para invocar la información.
- 2) Indicación.- Se utiliza para invocar una función o para indicar a una función que ha sido invocada en un sitio de acceso de servicio.
- 3) Respuesta.- Sirve para completar una función previamente invocada por una indicación en el SAP.
- 4) Confirma.- Se utiliza para completar una función previamente invocada por una solicitud en SAP.

Una primitiva es codificada en software con un formato muy específico. Una primitiva es usada por una capa para invocar servicio a las entidades y crear cualquier encabezado que será usado por una capa equivalente en una estación remota. Es decir, las primitivas son usadas por capas adyacentes en un sitio local para crear encabezados usados por las capas equivalentes en sitios remotos.

## ARQUITECTURA SNA

SNA (Systems Network Architecture) es una especificación de IBM que describe la arquitectura para una red distribuida de procesamiento de datos. SNA define las reglas y protocolos para la interacción de los componentes en la red. La arquitectura SNA fue desarrollada para satisfacer las necesidades de flexibilidad en la comunicación para el ambiente de procesamiento de datos. SNA es una arquitectura jerárquica que se compone por 7 capas:

- Capa de usuario final

- Capa de servicios NAU
- Capa de control de flujo de datos
- Capa de control de transmisión
- Capa de control de ruta
- Capa de control de enlace de datos
- Capa de control físico

### CAPA DE USUARIO FINAL EN SNA

Esta capa representa al usuario de la red SNA, sobre una estación final, alguna aplicación u otros sistemas.

### CAPA DE SERVICIOS NAU EN SNA

Las funciones definidas en la capa de servicios de unidades direccionables de red (NAU, Network Addressable Unit) se incluyen servicios de presentación de sesión, los cuales cumplen con consideraciones de formateo de datos y las interfaces de programación entre el usuario final y la capa de servicios.

También presta servicios de transacción los cuales definen el esqueleto para transacciones distribuidas y servicios especiales como distribución de documentación. También incluye un manejador de servicios para manejar servicios proporcionados por la capa.

### SERVICIOS DE PRESENTACION EN SNA

La capa de servicios de funciones de administración de datos FDM (Funcion Management Data), cumple un conjunto de funciones y servicios proporcionados al usuario final. Los servicios de presentación incluyen los siguiente:

- Traslación de formato.- Este servicio permite tener en cada final, una diferente visión del intercambio de datos.
- Compresión y compactación.- El dato puede ser comprimido en el nivel de bit o byte usando procedimientos específicos para reducir el volumen de transmisión.
- Soporte de programa de transacción.- Este servicio controla la comunicación del nivel de conversación entre programas de transacción por:
  - a) Programas de transacción para invocar y cargar.
  - b) Mantener los protocolos en modo envía/recibe.
  - c) Exigir el uso correcto de parámetros y restricciones de secuenciación.

### SERVICIOS DE TRANSACCION EN SNA

Este servicio sólo se aplica en comunicación de programa-a-programa. Incluye lo siguiente:

- Servicios de configuración.- Permiten al operador iniciar o reconfigurar una red por la activación y desactivación de enlaces.
- Servicios de operador de red.- Incluye funciones como colección y despliegue de estadísticas de la red y la comunicación de datos desde usuarios y procesos al operador de la red.
- Servicios de sesión.- Soportan la activación de una sesión de parte de los usuarios finales y aplicaciones.
- Mantenimiento y administración de servicios.- Proporciona para probar la red, facilidades y asistencia en fallas de comunicación e identificación.

### CAPA DE CONTROL DE FLUJO DE DATOS EN SNA

Esta capa proporciona protocolos para controlar el flujo de datos dentro de una sesión. Los protocolos definidos en ésta capa tienen significado sólo para una sesión específica y no tiene efecto sobre otras sesiones. Sus funciones principales están en las siguientes categorías.

- Modo envía/recibe.- Se refiere al tipo de enlace usado, que es principalmente half-duplex y full-duplex.
- Encadenación.- Es un mecanismo para trazar secuencias de transmisión con el propósito de recuperación. Este protocolo permite patrones de sesión, para limitar el tamaño del mensaje más largo que pueda ser intercambiado en la sesión.
- Agrupación.- Esta tarea se realiza de acuerdo con una secuencia de intercambios. Este concepto es usado para definir y controlar secuencias de transacción.

Opciones de respuesta.- Se pueden especificar tres modos:

- 1) No envía una respuesta.
- 2) Envía sólo una respuesta, en caso de una excepción.
- 3) Siempre envía una respuesta.

Inactivar/cerrar.- Un paro permanente o temporal para el flujo de datos puede ser solicitado.

### CAPA DE CONTROL DE TRANSMISION EN SNA

La capa de control de transmisión es responsable del establecimiento, mantenimiento y terminación de las sesiones SNA. Una sesión es una relación lógica entre puntos finales. Esta capa puede establecer una sesión, en respuesta a una solicitud desde la capa superior, desde un proceso de aplicación o para propósitos de su propio control. Esta capa está compuesta por dos módulos: el manejador de conexión (CPMGR) el cual maneja la transferencia de datos individualmente y el control de sesión el cual maneja asuntos del nivel de sesión. Las funciones principales desempeñadas en ésta capa son:

- Encaminamiento.- Esta es esencialmente una función de demultiplexión. Las unidades de datos que entran son ruteadas a la entidad apropiada, la cual puede estar en la misma capa o en alguna superior.
- Encapsulación.- Cada mensaje que sale es encapsulado en unidades de datos. El encabezado que se agrega al dato, contiene información de control para expedir, reparar, rutear, encriptar y controlar otras funciones.

- Paseo.- Este es el mecanismo usado en rutas virtuales. Sólo usado por puntos finales para controlar el flujo de las unidades de datos. Proporciona números de secuencia en cada sesión para el chequeo del mensaje y procedimientos para controlar un máximo número de mensajes relacionados a una sesión en cualquier momento.

### **CAPA DE CONTROL DE RUTA EN SNA**

La capa de control de ruta tiene dos tareas principales: el encaminamiento de datos y el control de flujo. El manejo de rutas a través de los nodos SNA es cumplido por la capa de control de ruta al examinar los nombres de la red en el mensaje y determinar el enlace apropiado para que el sistema sea llevado a su destino.

La capa de control de ruta crea canales lógicos entre puntos finales, referidos como unidades direccionables de red (NAU, Network Adressable Unit). Una NAU es una entidad de nivel de aplicación, capaz de ser direccionada y de intercambiar datos con otras entidades.

El control de ruta está basado sobre los conceptos de:

- Grupo de transmisión.- Un grupo de transmisión es un conjunto de uno o más enlaces físicos entre nodos adyacentes en la red.
- Ruta explícita.- Una ruta explícita es una ruta entre dos puntos finales definido como una secuencia ordenada de grupos de transmisión.
- Ruta virtual.- Una ruta virtual es una conexión lógica entre dos puntos finales que es dinámicamente asignada a una ruta explícita.

Esta capa provee de protocolos para rutear los datos a través de la red y manejar y controlar las rutas y congestión de la red. Esta capa también proporciona un protocolo llamado segmentación, el cual permite el envío de largos mensajes como segmentos múltiples que posteriormente son reensamblados por el elemento de control de ruta en el nodo receptor.

### **CAPA DE CONTROL DE ENLACE DE DATOS EN SNA**

Esta capa proporciona procedimientos para manejar un enlace y transferir cadenas de bits libres de error a través del enlace. Los procedimientos incluyen administración de transmisión half-duplex y full-duplex; recuperación del error desde el enlace donde ocurrió el error; control de líneas multipunto y control de flujo a nivel de enlace. El protocolo SNA para el control de enlace es SDLC.

### **CAPA DE CONTROL FISICO EN SNA**

Representa la interfase física con el medio de transmisión. SNA no es arquitecto de algún nuevo estándar en esta capa. En la capa de control físico se usan los estándares EIA RS232, V.24, X.21, etc. Se especifican dos tipos de interfase:

- Para enlace de comunicación serial
- Para enlaces paralelos.

Las interfases para enlaces de comunicación serial son usadas por la mayoría de las conexiones nodo-a-nodo. Las redes SNA también pueden incluir enlaces paralelos de alta velocidad entre un mainframe y un procesador de comunicación front-end.

### **TAREAS Y ESPECIFICACIONES DE SNA**

- Facilidad para compartir recursos
- Facilitar dispositivos para conmutar entre programas de aplicación.
- Facilitar el tipo de mezcla terminal/dispositivos para coexistir sobre el mismo enlace de comunicación.

Especificaciones de SNA:

- Las funciones deben ser proporcionadas para soportar comunicación de datos entre las diferentes entidades que son servidas.
- El protocolo es seguido cuando dos entidades desean comunicarse entre en una forma libre de error y más tarde desean terminar su comunicación.
- Los comandos deben estar asociados con los procedimientos de conexión.

### **ESTRUCTURA DE SNA**

En una red SNA toda la comunicación entre usuarios finales es vía un puerto dentro de la red llamado "unidad lógica" (LU, Logical Unit). Las LU manejan todas las funciones relacionadas con la comunicación entre usuarios finales. Las LU son interconectadas a través de una red de control de ruta, que desempeña todas las funciones de red no especificadas por la sesión, pero orientadas a la transportación de mensajes desde una unidad lógica a otra. Las otras unidades direccionables de red (NAU) son:

- Unidades físicas (UF)
- Sitios de control de los servicios del sistema (SSCP)

La comunicación entre NAU's toma lugar a través de mensajes que en SNA son llamados unidades de solicitud y las respuestas a éstas son llamadas unidades de respuestas. Los encabezados que contienen información relevante para una capa en particular en SNA son agregados a esas unidades de solicitud/respuesta. Hay 4 tipos de encabezados:

- Encabezado de administración de función (AF).
- Encabezado solicitud/respuesta (SR).
- Encabezado de transmisión (T).
- Encabezado de enlace (E).

### **UNIDADES FISICAS (SNA)**

Son interfases que controlan las funciones de una parte física particular del equipo hacia la red.

### **SSCP (SNA)**

Los sitios de control de los servicios del sistema (SSCP, Systems Services Control Points), son usados para administración y control de los recursos de la red

**ENCABEZADO SOLICITUD/RESPUESTA (SNA)**

Contiene información de control referente al estado de la conversación entre dos entidades de comunicación.

**ENCABEZADO DE TRANSMISION (SNA)**

Contiene información de control relacionada con el transporte de los mensajes a través de la ruta de control de la red.

**ENCABEZADO DE ENLACE (SNA)**

Contiene información de control referente al movimiento de los mensajes en un enlace particular (normalmente SDLC).

**ENCABEZADO DE ADMINISTRACION DE FUNCION (SNA)**

Contiene información de control que especifica acciones que tienen que ser extraídas del contenido del dato en la unidad de solicitud o de respuesta.

**OPERACION EN SNA**

Una regla fundamental en SNA es que no puede haber comunicación si no hay alguna sesión entre entidades de comunicación (NAU).

En la red SNA, la arquitectura consiste de una "subárea" y nodos "periféricos". Los nodos de subárea son interconectados para formar la base de SNA, mientras que los nodos periféricos se conectan a una subárea de la red. Cada NAU tiene una dirección de red que consta de dos partes: una subárea y elementos de dirección. El elemento de dirección debe ser único para cada subárea y cuando hay comunicación entre subáreas, la dirección de la subárea es usada para el encaminamiento.

La estructura de los datos en SNA se forma al agregar un encabezado al dato de usuario, al pasar de capa en capa, para cumplir el proceso de comunicación.

**SESION (SNA)**

Una sesión en SNA es el inicio y el diálogo subsecuente de una comunicación entre dos NAU's enlazados previamente. La sesión debe iniciar con una entidad que desea comunicarse, usando una sesión con los puntos de control de los servicios del sistema (SSCP) para solicitar permiso a otras sesiones. Una sesión se compone de dos medias sesiones. Una media sesión soporta el conjunto de funciones usado por el NAU para proporcionar los protocolos requeridos para soportar comunicación dentro de una sesión particular. Dos medias sesiones son interconectadas por el control de ruta de la red. Un NAU debe ser capaz de soportar múltiples medias sesiones concurrentes con otros NAU's.

**ARQUITECTURA DNA**

DNA (Digital Network Architecture) es una arquitectura de red proporcionada por DEC (Digital Equipment Corporation) y desarrollada como una red distribuida, que soporta un amplio rango de aplicaciones y programas. DNA no debe confundirse con DECNET, dado que éste se refiere a un conjunto específico de productos software y hardware que implementan la arquitectura DNA.

DNA fue diseñada para cumplir un conjunto de metas, en base al concepto de arquitecturas por capas jerárquicas. DNA consta de 8 capas:

- Capa de usuario
- Capa de administración de red
- Capa de aplicación de red
- Capa de control de sesión
- Capa de comunicaciones finales
- Capa de rutas
- Capa de enlace de datos
- Capa física

**METAS DE LA ARQUITECTURA DNA**

- Crear una interfase de usuario común que contemple implementaciones y aplicaciones variadas.
- Proporcionar capacidad de recurso-compartido a los datos, computadoras y dispositivos periféricos.
- Soportar computación distribuida, permitiendo ejecutar programas de cooperación en diferentes computadoras en una red.
- Soportar un amplio rango de facilidades de comunicación como Ethernet y X.25.
- Mantener un alto nivel de disponibilidad, en cada evento de nodo o enlace fallido.

**CAPA DE USUARIO EN DNA**

Contiene funciones definidas por el usuario como programas de aplicación. En esta capa se proporcionan interfaces a las tres capas inferiores inmediatas a ésta. Los programas de aplicación pueden directamente acceder la capa de control de sesión para una comunicación programa-a-programa. Contiene algunos sistemas DECNET, como programas de control de red. Tiene acceso directo a las capas de administración de red, aplicación de red y control de sesión.

**CAPA DE ADMINISTRACION DE RED EN DNA**

Esta capa proporciona administración descentralizada para una red DNA y es usada principalmente para el control y monitoreo de las operaciones de la red. Los módulos dentro de un nodo son responsables de dos funciones:

- Coordinar la administración del nodo
- Comunicarse con módulos iguales en nodos remotos para cumplir la administración descentralizada. Esta capa recibe servicio de tres capas más bajas:
- La capa de aplicación de red proporciona acceso a archivos remotos.
- La capa de control de sesión comunica con entidades iguales para el manejo de funciones descentralizadas.
- La capa de enlace de datos encabeza funciones de administración simple entre nodos adyacentes.

### CAPA DE APLICACION DE RED EN DNA

La capa de aplicación de red proporciona servicios genéricos a la capa de usuario y administración de red. Los servicios incluyen acceso y transferencia de archivos remotos, acceso a terminales interactivas remotas, acceso a gateways o sistemas que no sean DNA. Los módulos en ésta capa operan independientemente y asincrónicamente.

### CAPA DE CONTROL DE SESION EN DNA

La capa de control de sesión proporciona los siguientes servicios:

- Es responsable de la comunicación lógica entre usuarios.
- Proporciona el mapeo de los nombres de los nodos, para especificar la dirección de los nodos.
- Proporciona funciones de comunicación proceso-a-proceso, para procesos residentes en las capas de usuario, administración de red y aplicación de red.
- Sus servicios están orientados a conexión.
- Valida sesiones entre usuarios
- Activa y desactiva sesiones

### CAPA DE COMUNICACIONES FINALES EN DNA

La capa de comunicaciones finales es un servicio de comunicación fin-a-fin de capas superiores orientado a conexión. Sus funciones principales son:

- Aísla las capas superiores de los errores transitorios.
- Es responsable del control de error, segmentación, reensamblado y control de flujo de una comunicación fin-a-fin.
- Multiplexa múltiples conexiones llamados enlaces lógicos, entre pares de nodos o entre un nodo o nodos múltiples.

### CAPA DE RUTAS EN DNA

La capa de ruteo proporciona un servicio de entrega de mensajes en redes anchas. Esta capa implementa un servicio de datagramas que reparte paquetes sobre la base de "mejor-esfuerzo" no garantizado, dado que el paquete puede perderse o ser entregado fuera de orden.

- La capa de ruteo proporciona funciones de ruteo entre nodos.
- Es responsable de control de congestión de la red.
- Determina el tiempo ocurrido de un paquete dentro de la red.

### CAPA DE ENLACE DE DATOS EN DNA

La capa de enlace de datos garantiza que una transmisión libre de error pase al otro lado por un enlace con muchos errores. Además controla la ruta de comunicaciones entre nodos adyacentes. Esta capa incluye especificaciones de arquitectura para cinco formas de enlaces de datos:

- 1) HDLC
- 2) LAPB (Link Access Procedures-Balanced)
- 3) CSMA/CD LAN
- 4) LAN FDDI
- 5) DDCMP (Digital Data Communication Message Protocol)

### CAPA FISICA EN DNA

En este caso cualquiera de la gran variedad de las capas físicas estándar pueden ser usadas. DNA incluye especificaciones de arquitectura que definen tres formas mejores de enlaces físicos:

- 1) Conexión de modem
- 2) LAN CSM/CD
- 3) LAN FDDI

### ESTRUCTURA DE LA ARQUITECTURA DNA

En la arquitectura DNA se distinguen dos tipos de nodos:

- 1) Nodos finales.- Tienen una conexión a la red pero no calculan rutas o conmutan paquetes a otros nodos.
- 2) Nodos para rutas.- Soportan múltiples enlaces y tráfico de datos; por lo tanto calculan rutas

Las interfaces entre capas son más flexibles que en el modelo OSI, la capa de usuario tiene acceso directo a las capas de administración de la red, aplicación de la red y control de sesión; la capa de administración de red, tiene acceso directo a las capas de aplicación de red, control de sesión y enlace de datos. DNA está disponible para usar 3 diferentes tecnologías de subredes:

- 1) DDCMP (Digital Data Communications Message Protocol): es usado para formar enlaces punto-a-punto entre dos computadoras. El medio físico usado por DDCMP puede ser un simple cable RS-232-C o puede consistir de un enlace de comunicación de satélite, microondas o de una WAN.

- 2) Ethernet: Consiste de protocolos de enlace físico y datos. Una de sus principales características es que muchos usuarios pueden coexistir sobre el medio.
- 3) X.25: Es una arquitectura de comunicaciones de datos conmutada por paquetes.

## ARQUITECTURA DOD

La arquitectura de comunicación DOD (Department Of Defense) está basada sobre un punto de vista de comunicación de datos que involucra tres agentes:

- Procesos.- Son las entidades fundamentales para la comunicación
- Computadoras.- Son capaces de ofrecer soporte múltiple a procesos simultáneos
- Redes.- Es el medio de conectar las computadoras y el medio por el cual el dato es transmitido desde una computadora a otra.

La arquitectura DOD organiza sus tareas de comunicación en cuatro capas relativamente independientes:

- Capa de acceso a la red
- Capa entre-redes (Internet)
- Capa estación-a-estación
- Capa de proceso.

### CAPA DE ACCESO A LA RED EN DOD

Tiene que ver con el intercambio de datos entre una estación a otra y la red a la cual está conectada. La estación transmisora debe proveer a la red con la dirección de la estación destino, para que la red pueda dirigir el dato apropiadamente.

El protocolo específico usado en ésta capa dependerá del tipo de red que sea usado. Diferentes protocolos se han desarrollado para redes conmutadas por circuitos (protocolo X.21), redes conmutadas por paquetes (protocolo X.25), redes locales (protocolos IEEE 802) y otros.

### CAPA ENTRE-REDES (INTERNET) EN DOD

Esta capa maneja los procedimientos que se requieren para el encaminamiento de datos entre dos estaciones conectadas en distintas redes. En esta capa se usa un protocolo entre redes (IP, Internet Protocol) para proporcionar la función de encaminamiento a través de redes múltiples. Este protocolo no sólo es implementado en una estación sino también en gateways.

### CAPA ESTACION-A-ESTACION EN DOD

La función de esta capa es garantizar que el intercambio de datos sea rentable, es decir, debe asegurar que todos los datos transmitidos lleguen a su destino y en el mismo orden en el cual fueron enviados. El protocolo usado es TCP.

### CAPA DE PROCESO EN DOD

Esta capa contiene los protocolos requeridos para soportar varias aplicaciones diferentes.

### OPERACION DE LA ARQUITECTURA DOD

Para lograr la comunicación entre estaciones, todas las entidades que están conectadas deben tener una dirección única. Para esto se requieren dos niveles de direccionamiento o puertos:

- 1) Dirección global
- 2) Dirección local

La operación en DOD es muy simple:

Suponiendo que un proceso asociado con el puerto 1 de una estación A, desea enviar un mensaje a otro proceso asociado con un puerto 2 en una estación B:

- 1) El proceso en A maneja el mensaje sobre la capa estación-a-estación con instrucciones para mandarlo a la estación B, puerto 2.
- 2) La capa estación-a-estación maneja el mensaje sobre la capa entre redes con instrucciones para enviarlo al host B.
- 3) La capa entre-redes maneja el mensaje sobre la capa de acceso a la red con instrucciones para enviarlo al gateway X.

### DIRECCION GLOBAL EN DOD

Cada estación sobre la red debe tener una dirección global de red única. Esto permite al dato ser entregado a la propia estación

### DIRECCION LOCAL EN DOD

Cada proceso dentro de una estación debe tener una dirección que sea única dentro de la estación, esto permite al protocolo estación-a-estación entregar los datos a los propios procesos.

### CONTROL DE OPERACION DE DOD

Para controlar la operación de DOD, la información de control al igual que los datos de usuario deben ser transmitidos.

- La información de control es generada como un bloque de datos por los procesos de transmisión
- El protocolo de la capa estación-a-estación (TCP) puede dividir el bloque dentro de dos piezas más pequeñas para hacerlo más manejable. Para cada pieza TCP agrega un encabezado.
- El protocolo TCP debe manejar cada segmento sobre la capa entre-red con ayuda de los gateways intermedios

- El protocolo IP agrega un encabezado a cada segmento recibido desde el protocolo TCP. El resultado se llama datagrama IP.
- Finalmente cada datagrama IP es presentado a la capa de acceso de red para su transmisión.
- La capa de acceso de red agrega su propio encabezado creando un "paquete".
- Cuando el dato es recibido en la otra estación, ocurren los procesos invertidos.
- En cada capa el encabezado correspondiente es eliminado y el resto es pasado a la próxima capa superior hasta que los datos del proceso original son entregados al proceso destino.

**BIBLIOGRAFIA: ARQUITECTURAS DE REDES**

- Computer communications.  
K.G. Beauchamp, 1990. (2ª edición).
- Decnet phase V an OSI implementation.  
James Martin, 1992.
- Design and validation of computer protocols.  
Gerard J. Holzmann, 1991.
- Internetworking.
- SNA architecture, protocols and implementation.  
Atul Kapoor, 1992.
- Telecommunications protocols and design.  
John D. Spraying, 1991.

## CAPITULO 4: DISEÑO DE REDES

## INTRODUCCION

La planeación y el diseño son aspectos importantes en el desarrollo de todos los proyectos, incluyendo los sistemas de computadoras con una red de comunicación de datos.

Los sistemas actuales de comunicación basados en computadoras, presentan frecuentemente fallas en su sistema. Generalmente las fallas ocurren por:

- 1) El hardware y el software involucrados no son realmente diseñados para sistemas basados en comunicación.
- 2) Tienen poca o ninguna experiencia para resolver problemas.

Un sistema útil no sólo debe satisfacer los requerimientos funcionales definidos en las especificaciones de aplicación, sino que debe considerar otros factores como: tiempos de respuesta, interfaces eficientes con el usuario, costo, expandibilidad, etc.

Generalmente las redes de comunicación están compuestas de:

**ARQUITECTURA (Elemento de red)**

Una arquitectura se une junto con todos los componentes, dentro de un conjunto. La arquitectura define sus funciones y distribución. Al igual que los protocolos debe observar el modo en el cual debe operar al comunicarse con otra.

**SOFTWARE (Elemento de red)**

Se refiere al software que se ejecuta sobre los nodos que incluyen desde manejadores hasta los programas necesarios para implementar un estado de funcionalidad.

**NODOS (Elemento de red)**

Los nodos sean inteligentes o no, actúan como conmutadores. Los nodos son puntos claves en el proceso de diseño. Los nodos deben soportar algunas características como: "almacenar y adelantar", control de error, control de transmisión retrasada, radiodifusión, etc.

**MEDIOS DE TRANSMISION (Elemento de red)**

Los medios de transmisión son enlaces sobre los cuales la información es transmitida. Pueden ser ondas de radio terrestre, satélites, fibra óptica, cable coaxial, par trenzado u otro medio. Pueden ser de banda ancha o de banda base, trabajar en modo digital o analógico, etc.

**CONSIDERACIONES EN EL DISEÑO DE REDES**

Para realizar un buen diseño de una red de computadoras se debe considerar los siguientes parámetros:

- Requerimientos del usuario
- Expandibilidad, escalabilidad y evaluación
- Determinación del tráfico
- Aspectos de servicio
- Elección de protocolos
- Alcance de acción
- Consideraciones de tiempo y retard
- Determinismo
- Conectividad
- Costo
- Disponibilidad, rentabilidad y mantenimiento

**REQUERIMIENTOS DE USUARIO (Diseño de redes)**

El análisis de requerimientos es el primero y el más importante paso en el diseño de redes. No hay algún otro aspecto del diseño más importante que el de entender completamente las necesidades del usuario, dado que de éstas se determina la tecnología, protocolos, hardware, software y recursos que se deben emplear.

**DETERMINACION DEL TRAFICO (Diseño de redes)**

El diseñador debe identificar las características adecuadas del tráfico según cada aplicación planeada para la red. El tráfico puede ser de diversas formas y tamaños, conforme a muchos protocolos y formatos, con varios tipos de patrones para traslado, y requiere métodos especiales de procesamiento y encabezados.

**ELECCION DE PROTOCOLOS (Diseño de redes)**

La elección de protocolos debe ser definida para aplicaciones ya existentes y aplicaciones planeadas. Esto incluye todos los protocolos para cada arquitectura soportada.

**CONSIDERACIONES DE TIEMPO Y RETARDO (Diseño de redes)**

El tiempo es uno de los más importantes aspectos a considerar en el diseño de redes de comunicación. Decrementar el tiempo de transmisión y el retardo de la red, es una responsabilidad del diseñador. Los principales parámetros a considerar son:

- Tiempo de acceso
- Tiempo de respuesta
- Retardo vs. cantidad de información procesada
- Bloqueo vs. almacenamiento.
- Interrupciones y reinicios.

**CONECTIVIDAD (Diseño de redes)**

Dado que existen diferentes tipos de características de tráfico, el usuario requiere para transportar diferentes tipos de conectividad de red y usuario. Para lo cual debe considerar:

- Conectividad usuario-red y red-red
- Requerimientos geográficos
- Topología
- Infraestructura

**DISPONIBILIDAD, RENTABILIDAD Y MANTENIMIENTO (Diseño de redes)**

Se refiere a la capacidad de la red de ser suficiente para transmitir mensajes en un tiempo considerable y el de mantener el estado de operación de la red eficientemente.

**EXPANDIBILIDAD, ESCALABILIDAD Y EVALUACION (Diseño de redes)**

La red debe estar disponible para reaccionar al cambio. También puede ser diseñada con la capacidad de expansión, y debe tener la opción de ser escalada hacia arriba o hacia abajo. El plan de la red debe definir un ciclo evolutivo.

**ASPECTOS DE SERVICIOS (Diseño de redes)**

Deben considerarse tópicos importantes como:

- Administración de red
- Publicidad
- Seguridad
- Apoyo al usuario
- Recuperación de desastres, etc.

**ALCANCE DE ACCION (Diseño de redes)**

Una arquitectura de red debe resolver cualquier problema general que sea posible. Es decir, debe ser diseñado para soportar un rango amplio de aplicaciones y un amplio rango de tecnologías inferiores.

**DETERMINISMO (Diseño de redes)**

De acuerdo con la propiedad de determinismo, condiciones idénticas producirán resultados idénticos. En el diseño de una red determinística las rutas deben ser idénticas a la topología física usada.

**COSTO (Diseño de redes)**

El costo es usualmente el factor primordial en el diseño del sistema. Para poder estimar el costo del sistema se debe considerar algunas variables importantes:

- Tamaño y velocidad del CPU
- Tamaño y velocidad del disco
- Tipo de líneas de comunicación (punto-a-punto, multipunto, full-duplex, etc.)
- Técnicas de control de error
- Tipo de sondeo
- Tipo de conmutación
- Alcance de la red
- Tipo de medio de comunicación, etc.

**ESTANDARES PARA REDES**

Actualmente es aceptado en la industria de la comunicación que se requieren de estándares que gobiernen las características físicas, eléctricas y procedurales de los equipos de comunicación. Un "estandar" se define como la interfase o el conjunto de interfaces que se usa entre varios equipos. Un estandar describe las funciones de cada uno de los equipos requeridos para su desempeño y garantiza que las señales que viajan entre los equipos sean ajustadas para su propósito. La aplicación o potencial de cualquier estandar particular, es el factor determinante de como muchas partes requieren ser involucradas en su desarrollo y documentación.

Existen diversos grupos que han desarrollado diferentes conjuntos de estándares. Los principales son:

**CCITT**  
**ANSI**

**ECMA**  
**IEEE**

**ISO**  
**EIA**

**VENTAJAS DE UN ESTANDAR DE RED**

- Un estandar asegura que tendrá gran mercado para una pieza particular de hardware o software.
- Permiten productos de comunicación de diferentes vendedores, dando al comprador más flexibilidad en la selección y uso de equipo.

### DESVENTAJAS DE UN ESTANDAR DE RED

- Un estandar tiende a "congelar" la tecnología. Una vez que un estandar es desarrollado, está sujeto a revisar y comprometerse a usar las técnicas más eficientes que sea posible.
- Hay múltiples estándares para un mismo elemento de red.

### ISO

La Organización Internacional de Normalización (ISO, International Standards Organization) produce un gran número de estándares actualmente. Los miembros de ISO son organizaciones estándares individuales a nivel nacional. El comité técnico de ISO, que no sólo crea estándares sino también determina que estándares producir, está compuesto de miles de voluntarios desde fabricantes de computadoras, proveedores de productos de comunicación, usuarios de computadoras, gobiernos y organizaciones de consultoría.

El objetivo de ISO, al desarrollar su modelo de referencia (OSI), es simplemente definir un conjunto de mecanismos que hagan posible la interconexión de sistemas Informáticos heterogéneos, utilizando los medios públicos de transmisión de datos. En la elaboración del modelo de referencia (OSI), ISO ha tomado en cuenta la posibilidad de que su arquitectura permitiera la fácil utilización de las diferentes normas emitidas por otros organismos internacionales, en especial el CCITT.

### CCITT

El Comité Consultivo Internacional Telegráfico y Telefónico (CCITT, Consultative Committee for International Telephone and Telegraph). CCITT forma parte de la Unión Internacional de Telecomunicaciones (UIT, Union International Telecommunication). CCITT se relaciona con los estándares de interconexión de redes telefónicas y con los sistemas de señalización usados por modems para la transmisión de datos de computadoras sobre líneas telefónicas.

CCITT llama a los estándares que produce recomendaciones (recomendación X.25, recomendación X.21, recomendación X.400, etc.). Los contribuidores principales del CCITT, son las organizaciones de telecomunicación públicas y privadas. CCITT está compuesta por 18 grupos (SG's) con la responsabilidad de cubrir un área particular, ya sean voz, datos o servicios integrados.

### GRUPOS QUE INTEGRAN CCITT

- 1.- SGI      influyen estándares relacionados con servicios telegráficos y telemáticos.
- 2.- SGII     dirigen operaciones de sistemas telefónicos.
- 3.- SGIII    dirigen principios de tarifa.
- 4.- SGIV     dirigen mantenimiento de transmisión de circuitos internacionales.
- 5.- SGV      dirigen protección acerca de peligros y disturbios en la transmisión.
- 6.- SGVI     dirigen la protección y especificación de cubiertas de cables y su alcance.
- 7.- SGVII    dirigen redes públicas de datos.
- 8.- SGVIII   dirigen telegrafía y DTE's relacionados.
- 9.- SGIX     dirigen la calidad de transmisión telegráfica
- 10.- SGX     dirigen nodos telegráficos.
- 11.- SGXI    dirigen señalización y conmutación telegráfica.
- 12.- SGXII   se relacionan con la eficiencia de transmisión telefónica.
- 13.- SGXIII   se relacionan con redes telefónicas públicas automáticas y semiautomáticas.
- 14.- SGXIV   se relacionan con equipo y transmisión telegráfica.
- 15.- SGXV    se relaciona con sistemas de transmisión.
- 16.- SGXVI   se relaciona con redes telefónicas.
- 17.- SGXVII   se relaciona con la transmisión sobre redes telefónicas y telex.
- 18.- SGXVIII se relaciona con redes digitales, como es ISDN.

### ECMA

La Asociación Europea de Manufacturas de Computadoras (ECMA) fué originalmente formada por un grupo de compañías europeas, pero ha ido creciendo hasta llegar a ser internacional e incluir miembros representantes de organizaciones como: IBM, Digital, AT&T, British Telecom y Toshiba. ECMA es considerada una organización estandar regional, que desarrolla información tecnológica estandar para la región europea específicamente. Los estándares de ECMA son frecuentemente expedidos a ISO para desarrollar estándares a nivel internacional.

### ANSI

ANSI es la organización nacional de normalización de los Estados Unidos. ANSI asigna comités para elaborar estándares en áreas relacionadas con información tecnológica. Los principales comités (ASC's) son los siguientes:

- JTC1 TAG.      Este grupo proporciona un lugar a los Estados Unidos en los estandars JTC1 de ISO.
- ASC X3.        Este comité produce aproximadamente el 90 % de los estándares en información tecnológica.
- ASC T1.        Este grupo integra el cuerpo del mercado de estándares para la industria de telecomunicaciones.
- ASC X12.       Este grupo es responsable de los estandares relacionados con el intercambio de datos electrónico.

ANSI no es una organización gubernamental; está es fundada por sus miembros y por la demanda de estándares.

### IEEE

El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) es una organización de normalización y desarrollo en el área de ingeniería, y especialmente en redes locales. IEEE es una sociedad profesional donde sus miembros son ingenieros individuales. La mayoría de sus actividades sólo se relacionan con información tecnológica. IEEE son de las organizaciones autorizadas que operan sobre la línea de ANSI cuando desarrollan estándares.

IEEE es la organización central para el desarrollo de estándares de redes de área local (LAN's) bajo su proyecto 802. El trabajo del comité 802 es actualmente dividido en subcomités que se relacionan con un estándar de LAN específico:

- 1.- IEEE 802.1 se relaciona con el estándar de interfaces para las capas más altas (HLL).
- 2.- IEEE 802.2 se relaciona con el estándar de control de enlace lógico (LLC).
- 3.- IEEE 802.3 se relaciona con el estándar CSMA/CD.
- 4.- IEEE 802.4 se relaciona con el estándar de token bus.
- 5.- IEEE 802.5 se relaciona con el estándar de token ring.
- 6.- IEEE 802.6 se relaciona con el estándar de redes MAN's.

El proyecto IEEE 802 es compatible con las capas más bajas del modelo OSI.

## EIA

La Asociación de Industrias de Electrónica (EIA), es una asociación de compañías que se responsabilizan del desarrollo y publicación de estándares para la comunicación de terminales. EIA emprende algunos proyectos de estandarización y los opera como una organización acreditada bajo las normas de ANSI. Los estándares desarrollados por EIA están relacionados primordialmente con las interfaces físicas de comunicación y señalización eléctrica. Un estándar muy conocido de EIA es: EIA-232-D, el cual documenta el camino en el cual una terminal o computadora es conectada a un modem.

## ESTANDARES PARA LA CAPA FISICA

La capa física se relaciona con los siguientes aspectos:

- Eléctrico (Tasas de datos, niveles de voltajes, etc.).
- Funcional/procedural (Definición de datos, temporización, señales de control, etc.).
- Mecánico (Tipo de conector, número de patillas, etc.).

Por lo tanto, la conexión física para el medio de transmisión es definida por la capa física. La interfase estándar más usada en la capa física es RS-232-C definida por EIA (Electronic Industries Association) para ANSI (American National Standards Institute). El mismo estándar es adoptado por CCITT como V.24 y V.28, y es adoptado por ISO como 2110.

RS-232-C es remplazado en algunas ocasiones por RS-449, que es adoptado por CCITT con el nombre V.35.

## ESTANDARES PARA LA CAPA DE ENLACE DE DATOS

La capa de enlace de datos desempeña las siguientes funciones:

- Mantener la conexión de enlace de datos.
- Procesos de activación y desactivación de enlaces.
- Mapear unidades de datos con unidades de enlace de datos.
- Multiplexar unidades de datos a conexiones físicas múltiples.
- Delimitar las unidades de enlaces de datos.
- Detección y recuperación de errores y notificación a usuarios finales.
- Crear identificadores de red (ID) e Intercambiar parámetros con partes iguales (capa de enlace) entre nodos de red compatibles.

La capa de enlace de datos se divide en dos subcapas. La más baja de las dos, controla el acceso al medio físico y puede ser diferente en cada tipo de red. La parte más alta de la capa de enlace de datos desempeña la parte del trabajo que es común a todos los tipos de red. La mayoría de las opciones en ésta subcapa están basadas sobre un mismo estándar llamado HDLC (High-level Data Link Control).

La definición principal del servicio de la capa de enlace de datos está en ISO 8808, la cual es equivalente a CCITT X.212. HDLC. Asimismo es definida en una serie de documentos incluyendo: ISO 3309, ISO 4335, ISO 7776 y ISO 7809.

## ESTANDARES PARA LA CAPA DE RED

La capa de red proporciona las siguientes capacidades:

- Proporciona direcciones de red e identificadores de red (ID) en puntos finales.
- Multiplexaje de conexiones de red a la capa de enlace de datos.
- Segmentación/encapsulación de cadenas de datos.
- Selección de servicios apropiados.
- Selección de la calidad de servicio deseado.
- Detección y recuperación de errores y notificación a usuarios finales.
- Ejecución de transferencia de datos.
- Reinicio de conexión con la pérdida de datos y notificación a usuarios finales.

La capa de red depende del sistema de red que sea usado. Los estándares de OSI cubren los dos tipos de red principales LAN's y WAN's. El servicio de red básico es definido en ISO 8348. La estructura interna es dada en ISO 8648.

IEEE proporciona los estándares relacionados con redes de área local (LAN's) y el CCITT proporciona los estándares relacionados con las redes de área ancha (WAN's).

Los estándares más importantes relacionados con LAN's son:

- Ethernet
- Token bus
- Token ring
- ARCnet
- FDDI

Los estándares más importantes relacionados con WAN's son:

- X.25
- X.21
- ISDN

## ETHERNET

Ethernet es un estándar diseñado por Xerox, Intel y DEC, cuyas especificaciones están diseñadas para satisfacer los requerimientos principales de una LAN. Ethernet es la base del estándar IEEE 802.3 y forma parte del estándar ISO 8802/3. El uso de éste estándar ha sido ampliamente adoptado y adaptado para sistemas de oficinas y fábricas. Ethernet está disponible en redes 10BASE2, 10BASE5, las cuales usan cable coaxial, y la más actual 10BASET, que usa cable par trenzado.

La conexión de Ethernet al cable coaxial se denomina transceptor; tanto 10BASE2 como 10BASE5 se refieren a él como MAU (Medium Attachment Unit), unidad de acceso al medio.

Ethernet 10BASE5 llama al cable entre el transceptor y la tarjeta adaptadora de red, cable AUI (Attachment Unit Interfaz) o también cable "drop". Ethernet 10BASE2, al conectarse directamente a la tarjeta no requiere un cable AUI.

### REQUERIMIENTOS PRINCIPALES DE UNA LAN

- Tasa de transmisión de datos alta.
- Distancias considerables.
- Habilidad para soportar varios cientos de dispositivos.
- Simplicidad.
- Tasa de error baja y detección de errores eficiente.
- Uso eficiente de recursos compartidos.
- Estabilidad bajo cargas altas.
- Fácil acceso al medio.
- Fácil de instalación.
- Fácil de reconfigurar.
- Bajo costo.

### ETHERNET 802.3 10BASE2

- Máxima distancia de un segmento de 185 m.
- El cable típicamente usado es el RG-58A/U (cable delgado que usa conectores BNC).
- Máximo de 5 segmentos (con 4 repetidores) entre dos estaciones; 3 segmentos coaxiales y 2 segmentos de unión.
- El cable debe ser terminado, con una impedancia característica de  $50 \pm 2$  ohms y conectado a masa en sólo un punto dentro del segmento.
- Conexiones a través de adaptadores BNC en "T".
- Un total de 30 estaciones o nodos por segmento.

### ETHERNET 802.3 10BASE5

- Máxima longitud de segmento de 500 m.
- Máximo de 5 segmentos (con 4 repetidores) en cualquier trayectoria de red entre 2 estaciones. La distancia máxima de un segmento se especifica por el retraso de propagación de una señal, no por su distancia.
- Cuando no existe un segmento de enlace, pueden coexistir en cualquier trayectoria hasta 3 segmentos.
- El cable debe ser terminado con una impedancia de  $50 \pm 2$  ohms y conectado a masa en sólo un punto dentro del segmento.
- Máximo de 100 MAU's en un segmento de cable.
- Los MAU's deben ser conectados en el cable a una distancia entre ellos de 2,5 m.

### CARACTERÍSTICAS DE ETHERNET

- Topología: de canal (bus)
- Velocidad: 10 Mbits/s sobre el cable en banda base.
- Medio: Cable coaxial.
- Tipo de enlace: Half-duplex.
- Acceso al medio: CSMA/CD.
- Aplicación: Sistemas de oficinas.
- Estándar: ISO 8802/3 (estándar internacional), IEEE 802.3.
- Distancia máxima: 2,500 m.
- No. de Estaciones: 1024 como máximo.
- Repetidores: Cada 500 m.

### LIMITACIONES DE ETHERNET

- La máxima longitud de un segmento de cable coaxial es de 500 m.
- El cable debe ser terminado, con una impedancia característica de  $50 \pm 2$  ohms y conectado a masa sólo un punto.
- Dos repetidores como máximo en una trayectoria entre dos estaciones de trabajo.
- Máximo 50 m. entre un transceptor conectado al segmento del cable coaxial y su estación de trabajo (cable AUI).
- Máximo de 100 transceptores por segmento.
- Ethernet es costoso en la instalación, dado que el cable coaxial de difícil manejo.

### TOKEN BUS (IEEE 802.4)

El estándar Token bus IEEE 802.4 es muy conocido principalmente por ser adoptado por General Motors en su especificación MAP (Manufacturers Automation Protocol). En éste estándar el acceso a la red es controlado por la posesión de un token. La red usa un cable de banda ancha (ancho de banda de 10 M bits/s), muy parecido al que se usa en cable de TV. para transportar señales en diferentes frecuencias.

De acuerdo con este estandar, todas las estaciones sobre el canal reciben la superposición de las transmisiones de las demás estaciones, así como la propia. El control es pasado via un token. Una estación debe recibir el preambulo de un mensaje completo, antes de que esta pueda pasar el token a la siguiente estación o transmitir su propio mensaje/paquete. El intervalo equivalente al tiempo de propagación de energía de un punto del bus a otro punto, más el tiempo que la estación se toma para el procesamiento de la señal requerida, se le denomina "interfase de retardo".

#### **CARACTERISTICAS DE TOKEN BUS (IEEE 802.4)**

En general, el estandar token bus IEEE 802.4 se puede resumir de la siguiente manera:

- Velocidad: 10 Mbits/s
- Medio: Cable de banda ancha.
- Acceso al medio: Por posesión de un token.
- Aplicación: Automatización de fábricas.
- Estandar: ISO 8802/4 (estandar internacional)

#### **TOKEN RING (IEEE 802.5)**

El estandar token ring fue diseñado por IBM como una estrategia para establecer comunicaciones en redes de área local. La red token ring se adhiere al estandar IEEE 802.5 y está actualmente soportada por un número importante de fabricantes (IBM, Proteon, 3 COM, etc).

Como su nombre lo indica, se emplea la topología de anillo y el acceso a la red es controlada por un token. La eficiencia del estandar token ring es determinado principalmente por el tiempo de propagación, el cual define el tiempo en que la energía se propaga desde una estación a través de las demás y el regreso de la misma. Cada unidad de interfase de estación es caracterizado por el tiempo que toma en detectar el token y decidir cuando pasarlo a la próxima estación o transmitir un paquete/mensaje.

El elemento básico de la red en anillo es un dispositivo denominado MAU (Multistation Accas Unit, Unidad de Acceso a Multiestaciones) o WVC (Wire Concentrador, Concentrador de Cableado). Este dispositivo es el punto central de conexión de los nodos a la red. Entre MAUs pueden añadirse repetidores para incrementar la distancia de transmisión. Una de las funciones principales de un MAU es la reconfiguración dinámica del anillo cuando un nodo deja de estar conectado a la red.

#### **CARACTERISTICAS DE TOKEN RING (IEEE 802.5)**

- El estandar IEEE 802.5 tiene físicamente una topología de estrella y electricamente topología de anillo.
- Cada nodo actua como un repetidor a nivel de bits.
- El reparto de propagación a nivel de bits por cada nodo es muy pequeño.
- El token es simplemente una secuencia especifica de bits que circula a través de los nodos, dando a estos permiso para transmitir.
- Velocidad: 10 M bits/s.
- Medio: Cable par trenzado.
- Acceso al medio: Por posesión de un token.
- Aplicación: Sistemas de oficinas.
- Estandar: ISO 8802/5 (estandar internacional).

#### **ARCNET**

ARCNET es uno de los sistemas de redes mas antiguos y fue desarrollado por la corporación DATAPOINT. La topología estandar para ARCNET es una estrella, aunque también puede usarse en bus o en una combinación de ambas a la que se le denomina estrella distribuida, donde las estaciones ARCNET estan conectadas a un dispositivo central llamado "Hub", a través de módulos de interfase de recursos (RIM's, Resource Interfase Modules). Existe 2 tipos de hubs para ARCNET: activos y pasivos. Cada Hub proporciona conexión (via un anillo interno) a los RIM's, sirviendo PC's, servidores de archivos, impresoras, etc. de acuerdo con la topología de estrella física.

ARCNET transmite todas las señales a través de toda la red simultáneamente. Esto se debe a que usa una topología de bus lógico, donde todos los dispositivos comparten el mismo cable. ARCNET utiliza un método de token-passing lógico para controlar el acceso al cable y reconfigura dinámica y automáticamente una estación cada vez que es añadida o removida de la red.

Para que ARCNET pueda comunicarse con una red SNA/SDLC, HDLC, X.25 u otra ARCNET, requiere del uso de bridges (puentes) o gateways (pasarelas).

#### **HUB ACTIVO (ARCNET)**

Un hub activo es una unidad con su propia alimentación que actua como un dispositivo de distribución y amplificador de señal.

#### **HUB PASIVO (ARCNET)**

Un hub pasivo sólo distribuye la señal sobre distancias pequeñas (sin amplificar).

#### **CARACTERISTICAS DE ARCNET**

- Es fácil de instalar.
- El diagnóstico de fallas es muy sencillo.
- Soporta cable coaxial y par trenzado.
- Su velocidad de transmisión es de 2.5 Mbps (no es rápida).
- Posee una gran flexibilidad en el cableado de una instalación.
- Proporciona un grado de estandarización muy alto.
- Aumenta su velocidad a 20 Mbps con el uso de ARCnet plus que es compatible con los productos de ARCnet anteriores, y aumenta el paquete de datos a 4,096 bytes.

## FDDI (Fiber Distributed Data Interface)

De la misma forma de que las LAN's surgen por la necesidad de compartir recursos como impresoras, modems, discos, etc. en un área relativamente pequeña; asimismo surge la necesidad de que estas redes sean rápidas. Este es el motivo por el que surge el estándar FDDI (Interfase de datos distribuidos por fibra), que gracias al uso de fibra óptica como medio de transmisión, es posible compartir desde sistemas de archivos hasta, cualquier dispositivo gráfico (plotters, impresoras gráficas, etc) que requiera de buena resolución. FDDI se compone de las siguientes capas:

- Control de acceso al medio (MAC)
- Parte física (PHY)
- Dependencias del medio físico (PMD)
- Administración de estación (SMT)

Cada estación FDDI está compuesta de entidades lógicas que conforman al estándar FDDI. Las redes con diferentes topologías físicas pueden ser construidas, dependiendo de las conexiones con los diferentes tipos de estaciones usadas, que pueden ser:

- Estación con conexión dual (DAS)
- Concentrador con conexión dual (DAC)
- Estación con conexión simple (SAS)
- Concentrador con conexión simple (SAC)

### CONTROL DE ACCESO AL MEDIO (MAC) EN FDDI

Como en los estándares 802, la capa MAC del estándar FDDI, es la porción de la capa de enlace de datos que regula el acceso al medio de la LAN. El protocolo de acceso al medio es token ring, similar al estándar IEEE 802.5, con la diferencia de que en FDDI se usa:

- Fibra óptica que alcanza velocidad de 100 Mbps
- Señalización NRZI.
- Sistemas distribuidos
- Token nuevo después de transmitir, etc.
- Especificación rentable

Gracias al uso de esta técnica de control de acceso al medio, FDDI proporciona diversos servicios con acceso determinístico, los cuales no se degradaran con el incremento del número de estaciones a la red.

### PARTE FÍSICA (PHY) EN FDDI

La especificación de la capa física para FDDI incluye una parte independiente del medio y una parte específica del medio. La porción independiente del medio, de la capa física, incluye la codificación de datos digitales.

La capa física maneja todas las funciones basadas en símbolos. Un símbolo es la secuencia básica de bits que representa datos e información de control. La capa física codifica y decodifica los datos y los símbolos de control.

### DEPENDENCIAS DEL MEDIO FÍSICO (PMD) EN FDDI

Caracteriza los aspectos dependientes del medio de la capa física. El estándar FDDI especifica un anillo de fibra óptica con una tasa de datos de 100 Mbps, usando la codificación NRZI-4B/5B (de 4 a 5 bits), con longitud de onda de 1300 nm, especificada para la transmisión de datos. Dado el medio de transmisión empleado, se requiere de láser o fuentes de luz como es el caso de los LED's.

El receptáculo conector del cable definido por el estándar FDDI es llamado MIC (Media Interface Conector, Conector Interfase del Medio).

### ADMINISTRACION DE ESTACION (SMT) EN FDDI

Proporciona el control necesario en el nivel de estación para manejar procesos en las diferentes capas de FDDI, como el monitoreo, detección de fallas y recuperación de errores. SMT es subdividido dentro de tres áreas:

- Administración basada en frame.- Recoge y controla la información circulante en toda la red FDDI.
- Administración de conexión (CMT).- Maneja los componentes físicos y sus interconexiones.
- Administración de anillo (RMT).- Maneja los componentes de MAC conectados lógicamente al anillo.

### ESTACION CON CONEXION DUAL (DAS) EN FDDI

Tiene dos pares de entidades PHY y PDM, y una o más entidades MAC que participan en el tronco del anillo dual. Los DAS se usan cuando hay un número limitado de usuarios. También pueden ser usados para interconectar LAN's departamentales. Con cada estación FDDI, funcionan como un bridge.

### CONCENTRADOR CON CONEXION DUAL (DAC) EN FDDI

Consisten de un DAS con PHY y PDM, más allá de los requerimientos de conexión de un anillo dual. Las entidades adicionales permiten la conexión de las estaciones adicionales que son lógicamente parte del anillo pero están físicamente aisladas.

Puede ser usado para conectar múltiples dispositivos de alta potencia en un grupo de trabajo o LAN's múltiples. Con cada estación FDDI funcionan como un bridge.

### ESTACION CON CONEXION SIMPLE (SAS) EN FDDI

Constan de entidades PHY, PMD y MAC, y no pueden ser conectadas en el tronco del anillo, deben ser conectadas por medio de un concentrador.

Esta estación puede ser conectada a un anillo sencillo como una topología de estrella. Un SAS puede ser conectado a un DAC y mantener la rentabilidad de un anillo dual. Esto significa que las conexiones con SAS pueden estar aisladas por un concentrador para no causar demanda al resto del anillo.

**CONCENTRADOR CON CONEXION SIMPLE (SAC) EN FDDI**

Consta de un SAS con entidades PHY y PDM que van más allá de los requerimientos para conectarse a un concentrador. Los SAC's pueden ser conectados a otros DAC's u otros SAC's para formar una topología de árbol jerárquica estructurada.

Los SAC's proporcionan gran flexibilidad para agregar y remover concentradores y estaciones o cambiar su localización sin romper con la estructura de la LAN.

**TOPOLOGIAS DE FDDI**

La definición de los cuatro tipos de estaciones permiten la creación de una variedad de topologías, entre ellas estan:

- Múltiples estaciones conectadas a un sólo concentrador.
- Anillo dual. Es usada sólo con número limitado de usuarios. El anillo secundario es usado sólo si el anillo primario falla.
- Concentradores en árbol. Esta topología proporciona gran flexibilidad para el movimiento de estaciones o concentradores.
- Anillo dual de árbol. Esta es la más elaborada y flexible de las topologías de FDDI.

**FDDI-II**

FDDI-II es una extensión de FDDI, que agrega la habilidad de soportar tráfico conmutado por circuito, mientras que mantiene el servicio de conmutación por paquetes proporcionado por FDDI.

En FDDI-II la capa física (PHY) y la administración de estación (SMT) son las mismas que en FDDI. En el nivel de MAC, son referidos dos nuevos componentes: un multiplexor híbrido (HMUX) y el MAC isosíncrono (IMAC).

Durante la operación normal, la actividad sobre FDDI-II consiste de una secuencia de ciclos, generados por un ciclo maestro. La comunicación entre estaciones usan conmutación por circuito para compartir el uso del canal isosíncrono dedicado. Las estaciones se comunican usando conmutación por paquetes sobre el canal de paquetes de datos, observando las reglas impuestas por el protocolo de token ring.

**RECOMENDACION CCITT X.25**

La recomendación CCITT X.25 es diseñada para trabajar en las tres capas inferiores de una red conmutada por paquetes. La recomendación X.25 se encarga de realizar el enlace entre un DTE y un DCE. Cuando un equipo anuncia su compatibilidad con X.25, está indicando que puede utilizarse para la realización de un sistema distribuido, utilizando como elemento de transmisión de datos una red pública de comunicación de paquetes que se ajusta a esta recomendación. Los niveles que contiene la norma X.25 son:

- 1) Nivel físico
- 2) Nivel de enlace
- 3) Nivel de red

**CARACTERISTICAS DE CCITT X.25**

Generalmente la recomendación X.25 se puede resumir como sigue:

- Velocidad: Depende de la implementación.
- Medio: Combinación de líneas digitales y líneas análogas con modems.
- Acceso al medio: Los paquetes de datos son multiplexados juntos; un PAD es requerido para equipo asíncrono.
- Aplicación: Comunicación a largas distancias.
- Estándar: ISO 7776/8208 (estándar Internacional)

**NIVEL FISICO DE CCITT X.25**

En este nivel se definen las características físicas, eléctricas, funcionales y de procedimiento para establecer, mantener y desconectar el enlace físico entre el DTE y el DCE.

Los elementos que constituyen este nivel están contemplados en la norma X.21.

Dentro de un X.25 se contempla la posibilidad de interconexión conforme a la recomendación X.21

**NIVEL DE ENLACE DE CCITT X.25**

En este nivel se define el procedimiento de acceso al enlace para el intercambio de datos entre el DTE y el DCE. El procedimiento utiliza el principio y la terminología del procedimiento de control, para enlaces de datos de alto nivel (HDLC) especificado por ISO.

El procedimiento HDLC permite intercambiar paquetes entre un DTE y un DCE; corregir los errores que hubieran podido detectarse, mediando la retransmisión de los paquetes afectados; controlar el flujo de paquetes en el enlace DTE-DCE, así como confirmar la recepción correcta de los paquetes transmitidos.

**NIVEL DE RED DE CCITT X.25**

En este nivel se definen los procedimientos de control para el intercambio de paquetes que contengan información de control y datos del usuario, entre el DTE y el DCE. Para llevar a cabo la comunicación se requiere de la realización de una llamada virtual entre ambos, es decir, el establecimiento de un circuito virtual entre los dos DTEs implicados.

El intercambio de información entre dos sistemas informáticos se realizará a través de un circuito virtual, operación que se realizará en niveles superiores. Así pues, en el nivel de red sólo se realizarán las operaciones de establecimiento, mantenimiento y desconexión de circuitos virtuales.

### RECOMENDACION CCITT X.21

La recomendación X.21 del CCITT, incluye un protocolo para enviar y recibir llamadas, y para enviar y recibir datos mediante transmisión síncrona dúplex. La recomendación X.21 proporciona la interfase física hacia la red. Aunque X.21 fue diseñada para redes conmutadas por circuitos, parte de la recomendación cubre conexiones físicas de todas clases de redes: conmutadas por paquetes o conmutadas por circuitos. La interfase X.21 es diseñada para dispositivos que accesan la red sobre líneas digitales.

La mayor ventaja de X.21 sobre el RS-232C y el RS-449 es que las señales de la X.21 van codificadas en forma serial digital. En contraposición a RS-449, X.21 utiliza únicamente 6 señales.

Aunque X.21 se define como el nivel más bajo del protocolo internacional de conmutación de paquetes X.25, aún no es posible la conexión digital directa a las redes telefónicas públicas. Por esta razón, el CCITT ofrece la recomendación X.21 bis como medida provisional para conectar los equipos informáticos existentes con los servicios de conmutación de paquetes.

### SEÑALES DE LA RECOMENDACION CCITT X.21

Circuito	Nombre
G	Tierra de señalización
Ga	Retorno común del DTE
T	Transmisión de datos
R	Recepción de datos
C	Control
I	Indicador
S	Reloj de bit
B	Reloj de byte

### ISDN

La red digital de servicios integrados (ISDN, Integrated Digital Services Networks), nace de la necesidad de proveer a los usuarios de la red telefónica con un rango de aplicaciones más amplio (voz, video, etc.), soportadas por una red única y con acceso normalizado. La característica más importante de ISDN es el uso de señalización fuera del ancho de banda, es decir, es capaz de enviar información de control en paralelo a los datos en un canal separado.

ISDN ofrece conexiones por conmutación de circuitos a 64 Kbits/s; no obstante, también permite conexiones por conmutación de paquetes. La estructura básica de ISDN es la siguiente:

- Estructura de acceso
- Estructuras básicas
- Puntos de referencia
- Servicios ofrecidos

### ESTRUCTURAS DE ACCESO (ISDN)

El acceso de los usuarios a ISDN se realiza por medio de una línea digital multiservicio de características funcionales y eléctricas muy diferentes a las utilizadas en las redes telefónicas convencionales. Las agrupaciones funcionales definidas para ISDN son:

- Equipo terminal tipo 1 (ET1): Equipo terminal de usuario que presenta directamente las características de la interfase usuario/red propias de ISDN (Interfase "S").
- Equipo terminal tipo 2 (ET2): Equipo terminal de usuario que presenta otro tipo de interfase diferente de "S".
- Adaptador de terminal (AT): Permite la conexión de terminales que no son de ISDN.
- Equipo de terminación de red (TR): Este equipo incluye funciones como: conmutación, mantenimiento y terminación de la interfase física.
- Terminación de línea (TL): Equipo que adapta la línea de transmisión procedente de la instalación del usuario a la central de conmutación.
- Terminación de central (TC): Es el elemento de la central de conmutación que conecta la línea de acceso del usuario a las etapas de conmutación internas.

### ESTRUCTURAS BASICAS (ISDN)

El acceso de usuarios a ISDN, está compuesto por varios tipos de canales de transferencia de información:

- Canal "B": Canal de 64 Kbit/s, destinado al transporte de la información del usuario, tanto voz, datos digitales y combinación de ambos.
- Canal "D": Se utiliza para el intercambio de información de control entre el usuario y la red, es necesario para establecer las comunicaciones en los canales "B" o "H".
- Canal "H": Se destina al transporte de información de usuario con velocidades más elevadas.

Las configuraciones de acceso al usuario a ISDN son de dos tipos:

- 1) Acceso básico: Consiste de dos canales de transmisión de datos de 64 Kbits/s ("B" y "D") y un canal de 16 Kbits/s. Este acceso es soportado por la mayoría de líneas telefónicas de dos hilos.
- 2) Acceso primario: Consiste de 23 canales y un canal de 64 Kbits/s.

### PUNTOS DE REFERENCIA (ISDN)

Con objeto de definir el acceso a ISDN, y al mismo tiempo para dar cierta flexibilidad y modularidad, se ha definido una serie de puntos de referencia:

- Punto "R": Representa las interfaces físicas para terminales convencionales, tales como el RS-232.
- Punto "S": Interfase normalizada de conexión física, de terminales a la red ISDN.
- Punto "T": Define la separación entre el equipo de transmisión de la línea digital y las instalaciones propias del usuario.
- Punto "U": Se localiza en la propia línea de transmisión digital, entre el usuario y la central local.
- Punto "V": Separación entre las funciones de transmisión y de conmutación, en la central local.

### SERVICIOS DE ISDN

Dentro de los servicios ofrecidos por ISDN se consideran:

#### TELESERVICIOS

- Telefonía: Servicio básico de voz ofrecido a través de teléfonos digitales con codificación de señal vocal.
- Teletex: Comunicación entre terminales del servicio teletex.
- Facsímil: Servicio ofrecido con terminales FAX.
- Videotex: Servicio de acceso normalizado a bases de datos, para el intercambio de información tanto de texto como de imágenes fijas.

#### SERVICIOS SUPLEMENTARIOS

- Identificación del llamante
- Desvío de llamadas
- Información de tráfico
- Grupo de captura

### CARACTERISTICAS DE ISDN

Resumiendo las características principales del estándar ISDN se tiene:

- Velocidad: Dos canales para transporte de datos de 64 Kbits/s.
- Medio: Líneas digitales de alta capacidad.
- Control de acceso: Un subconjunto de HDLC.
- Aplicación: Comunicación local de datos y voz a largas distancias.
- Estándar: Recomendación CCITT Q.920/921/930/931.

### APLICACIONES DE ISDN

- Comunicaciones entre micros y LAN
- Comunicaciones de micros remotos con el ordenador principal
- Conectividad de LAN's
- Soporte de tráfico intenso de imágenes digitales
- Capacidad de compartir pantallas

## ESTANDARES PARA LA CAPA DE TRANSPORTE

La capa de transporte tiene tres fases:

- Fase de establecimiento de llamada.- Se relaciona con: el inicio del tráfico, el servicio de red, la cantidad de información a transmitir, el tamaño de la unidad de datos, etc.
- Modo de transferencia de datos.- Se relaciona con: segmentación/encapsulación, multiplexaje de conexiones, control de flujo de datos de fin-a-fin, identificación (ID) de la unidad de datos, detección de errores, etc.
- Fase de terminación.- Se relaciona con la terminación de la conexión.

El escoger el tipo de transporte depende sobre que nivel de servicio es requerido por las capas superiores y sobre que nivel de servicio puede ser esperado de las capas inferiores. Los tipos de servicio proporcionados por las tres capas más bajas son divididos de acuerdo con la tasa de error residual y la tasa de error señalada en tres tipos:

1. Tipo A
2. Tipo B
3. Tipo C

Existen 5 clases de servicios en la capa de transporte:

1. Clase 0
2. Clase 1
3. Clase 2
4. Clase 3
5. Clase 4

Los servicios de la capa de transporte están definidos en ISO 8072 o en la recomendación CCITT X.214 y en los protocolos que usan. El modo sin conexión está definido en ISO 8602. Uno de los protocolos ampliamente usado en la capa de red y en la capa de transporte es TCP/IP.

#### TASA DE ERROR RESIDUAL (Capa de transporte)

El número de errores que ocurren, en las capas inferiores, sin que exista alguna notificación del mismo.

#### TASA DE ERROR SEÑALADA (Capa de transporte)

El número de errores que son notados por las capas inferiores pero no corregidos.

#### TIPO A (Capa de transporte)

Es de calidad aceptable tanto en tasa de error residual como en tasa de error señalada.

#### TIPO B (Capa de transporte)

Tiene aceptabilidad baja en la tasa de error residual, pero alta aceptabilidad en la tasa de error señalada.

#### TIPO C (Capa de transporte)

Tiene una tasa de error residual inaceptable.

**CLASE 0 (Capa de transporte)**

Es la más simple clase del protocolo de transporte, y asume que un tipo A de red está disponible y no realiza algún esfuerzo para recuperar errores.

**CLASE 1 (Capa de transporte)**

Esta clase incluye la recuperación de errores, los cuales son señalados por la capa de red. Esto asume un tipo B de red, el cual señalará todos los errores que no se puedan corregir desde las capas inferiores.

**CLASE 2 (Capa de transporte)**

Esta clase permite el multiplexaje y también asume un tipo A de red. Si la capa de red es reiniciada, la conexión de transporte debe ser truncada y realizada de nuevo.

**CLASE 3 (Capa de transporte)**

Esta clase permite el multiplexaje y asume un tipo B de red, el cual señalará todos los errores que no se puedan corregir desde las capas inferiores.

**CLASE 4 (Capa de transporte)**

Esta clase es diseñada para el tipo C de red. Esta clase puede detectar errores que no puedan ser detectados o señalados por la capa de red.

**ESTANDARES PARA LA CAPA DE SESION**

La capa de sesión maneja las siguientes tareas:

- Conexión de sesión.
- Establecimiento de sesión y liberación/sincronización de sesión.
- Intercambio de datos.

Los estándares de la capa de sesión organizan el flujo de la información entre los sistemas de red y realizan tareas como el asegurarse que dos sistemas no intenten transmitir datos simultáneamente. El propósito principal de los servicios de sesión es permitir diálogo entre sistemas, al estructurarlo como actividades. La capa de sesión también marca puntos de referencia dentro de las cadenas de datos, dado que una actividad puede ser resincronizada si es necesario. El permiso para el uso de un servicio particular es regulado por la capa de sesión, a través de la distribución de tokens.

Los servicios de la capa de sesión para redes orientadas a conexión están definidas en ISO 8326, ISO 8327 o en CCITT X.215 y CCITT X.225.

**ESTANDARES PARA LA CAPA DE PRESENTACION**

La capa de presentación cumple con las siguientes capacidades:

- Transformación de datos requerida para procesar información.
- Formateo de datos en código ASCII, binario, EBCDIC, numérico o gráfico.
- Selección de sintaxis.

La capa de presentación es la que maneja el lenguaje que dos sistemas usarán para su comunicación. La importancia de los estándares de la capa de presentación consiste en permitir a los sistemas abiertos el uso de cualquier sintaxis en la capa de aplicación. En el caso de que dos sistemas usen el mismo lenguaje de comunicación no existe alguna dificultad, en cambio si los sistemas manejan sintaxis diferentes, se requiere el uso de una tercera sintaxis denominada sintaxis de transferencia, que permita la comunicación entre esos sistemas. La capa de presentación sigue la guía de los principios de los sistemas abiertos.

Desde que la capa de presentación es estándar, se dice que todos los sistemas han aprendido el mismo lenguaje de comunicación. El estándar internacional proporcionado es ISO 8822.

**ESTANDARES PARA LA CAPA DE APLICACION**

La capa de aplicación proporciona los siguientes servicios necesariamente para una aplicación:

- Identificación de usuarios participantes.
- Establecimiento de permisos para comunicación.
- Prueba de mecanismos de privacidad.
- Pruebas de calidad de servicios.
- Recuperación de errores y asignación de responsabilidades.
- Selección de una disciplina de diálogo.
- Transferencia de información actual entre terminales semejantes.

La capa de aplicación maneja el significado de la comunicación. Esta capa se sirve de varios elementos llamados ASEs (Application Service Elements, Elementos de Servicios de Aplicación) los cuales proporcionan los servicios de aplicación requeridos. Los ASEs originalmente estaban divididos en elementos que eran comunes a muchas aplicaciones (CASE, Common Application Services Elements) y aquellos que eran específicos a aplicaciones particulares. (SASE, Specific Application Services Elements). La intención de ISO es que cada capa de aplicación use una combinación de elementos tanto de CASE como de SASE, pero esta intención ha sido opacada.

Sin embargo, existen algunos elementos de las capas superiores que son ampliamente usados, en medio de los diferentes estándares:

Bloques de construcción de la capa de aplicación:	ACSE	CCR	ROSE	RTSE		
Estándares específicos de aplicación:	FTAM	X.400	ODA	EDI	VTP	JTM

**ACSE (Capa de aplicación)**

ACSE (Association Control Service Element, Elemento de servicio de control de asociación), es el que permite ligar y comunicar a las entidades de la capa de aplicación. El servicio ACSE, y el protocolo que éste usa, están definidos en ISO 8649 y ISO 8650.

**CCR (Capa de aplicación)**

CCR (Commitment Concurrency and Recovery) asegura que el trabajo de los sistemas abiertos sea completado, en caso de que uno o más sistemas sobre el mismo enlace se traslapen. El servicio CCR está definido en ISO DIS 9804, y los protocolos que usa en DIS 9805.

**ROSE (Capa de aplicación)**

ROSE (Remote Operations Service Element, Elemento de servicio para operaciones remotas) permite a una aplicación solicitar una operación de un sistema remoto. El sistema remoto luego reporta el resultado: ya sea que la operación sea satisfactoria, sea fallida o rechazada. ROSE está definido en X.218 y X.219.

**RTSE (Capa de aplicación)**

RTSE (Reliable Transfer Service Element, Elemento de servicio de transferencia rentable) permite la transferencia rentable de información entre sistemas abiertos. RTSE está incluido en X.228 y X.229.

**FTAM (Capa de aplicación)**

El propósito principal de FTAM (File Transfer Access and Management, Acceso y administración de transferencia de archivos) es garantizar la transferencia rentable de los archivos de datos, o parte de ellos, desde un sistema a otro, en caso de que la estructura de los archivos y los métodos de manejo de datos de los dos sistemas sean diferentes. El servicio FTAM está definido en ISO 8571.

**X.400 (Capa de aplicación)**

El servicio X.400 (Correo electrónico) se refiere al correo de información que puede ser enviado a diferentes direcciones individuales sobre la red, almacenándola electrónicamente. X.400 es una recomendación producida por CCITT para sistemas de correo electrónico, basados en el modelo OSI.

**ODA (Capa de aplicación)**

El estándar ODA (Office Document Architecture, Arquitectura de documentos de oficio) está fuera del modelo de las 7 capas de ISO. ODA define la estructura de un documento el cual puede ser enviado por sistemas OSI. ODA permite 3 tipos de contenido:

- 1) Carácteres de texto
- 2) Gráficos por vector
- 3) Gráficos rastreados

ODA está definido en ISO 8613

**EDI (Capa de aplicación)**

El servicio de EDI (Electronic Data Interchange, Intercambio de datos electrónicos) es direccionar la misma clase de asuntos que ODA, pero adoptando toda la información almacenada por el departamento de procesamiento de datos tradicional. El estándar EDI está definido en ISO 7372.

**VTP (Capa de aplicación)**

El VTP (Virtual Terminal Protocol, Protocolo de terminal virtual) es un servicio que habilita a las estaciones comunicarse con controladores terminales sobre una red OSI. VTP está definido en DIS 9040/9041

**JTM (Capa de aplicación)**

El servicio de JTM (Job Transfer Manipulation, Manipulación y transferencia de trabajo) consiste en permitir a un sistema, o a un usuario, someter las tareas de procesamiento a otros sistemas. JTM está definido en ISO DIS 8831.

**BIBLIOGRAFIA: DISEÑO DE REDES**

- Data communications & teleprocessing systems.  
Trevar Housley, 1987. (2ª edición).
- Data Network Design.  
Darren L. Sphon, 1993.
- Decnet phase V an OSI implementation.  
James Martin, 1992.
- Guía práctica de comunicaciones y redes locales.  
Antonio Cebrian Ruz, Eduardo Borraz F., 1993.
- Local and metropolitan area networks.  
William Stallings, 1993. (4ª edición).
- Network topology optimization.  
Roshan L. Sharma, 1990.

- Open systems.  
Peter Judge, 1988.
- Sistemas de comunicaciones.  
José Manuel Huidobro, 1993.
- System architecture and system design.  
Dimitris N. Chorafas, 1989.
- Telecommunications, protocols and design.  
John D. Spraging, 1991.

## **BIBLIOGRAFIA**

## BIBLIOGRAFIA DE REDES DE COMPUTADORAS

- ❑ Business data communications.  
David A. Stamper, 1991. (3ª edición).
- ❑ Computer communications.  
K.G. Beauchamp, 1990. (2ª edición).
- ❑ Computer mediated communications.  
Matthew Rapaport, 1991.
- ❑ Comunicaciones y redes de procesamiento de datos.  
Nestor Gonzalez.
- ❑ Communications wiring and interconnection.  
Fred J. Mc Climans.
- ❑ Contemporary data communications.  
Emilio Ramos, AL Schroeder, 1994.
- ❑ Data and computer communications.  
William Stallings, 1988. (2ª edición).
- ❑ Data communication, computer networks and open systems.  
Fred Halsall, 1992.
- ❑ Data communications.  
Larry Hughes, 1992.
- ❑ Data communication.  
Lynn A. De Noia, 1987.
- ❑ Data communications & teleprocessing systems.  
Trevar Housley, 1987. (2ª edición).
- ❑ Data communications networking devices.  
Gilbert Held, 1992. (3ª edición).
- ❑ Data Network Design.  
Darren L. Sphon, 1993.
- ❑ Decnet phase V an OSI implementation.  
James Martin, 1992.
- ❑ Design and validation of computer protocols.  
Gerard J. Holzmann, 1991.
- ❑ Diseño de redes locales.  
Hopper/Temple/Williamson, 1989.
- ❑ Guia práctica de comunicaciones y redes locales.  
Antonio Cebrian Ruz, Eduardo Borraz F., 1993.
- ❑ Local and metropolitan area networks.  
William Stallings, 1993. (4ª edición).
- ❑ Microcomputer LAN's.  
Michael Hordeski, 1991. (2ª edición).
- ❑ Network topology optimization.  
Roshan L. Sharma, 1990.
- ❑ Open systems.  
Peter Judge, 1988.
- ❑ Sistemas de comunicaciones.  
Jose Manuel Huidobro, 1993.
- ❑ SNA architecture, protocols and implementation.  
Atul Kapoor, 1992.
- ❑ System architecture and system design.  
Dimitris N. Chorafas, 1989.
- ❑ Telecommunications and computer.  
James Martin, 1976. (2ª edición).
- ❑ Telecommunications protocols and design.  
John D. Spragon, 1991.

## BIBLIOGRAFIA DE APOYO PARA EL SISTEMA TUTORIAL (REDTUT).

- ❑ Borland C++ 3.0 programming.  
Ben Ezzell. (3ª edición).
- ❑ C for Windows Animation programming.  
Lee Adams.
- ❑ Designing the user interface.  
Ben Sheiderman, 1992.
- ❑ Developing online help for Windows.  
Scott Bogan, David Farkas, Joe Welinske, 1993.
- ❑ Developing Windows 3 applications.  
Brent Rector, 1992.
- ❑ Developing Windows applications with Borland  
C++ 3.1.  
James W. McCord, 1992
- ❑ Engineering the human-computer interface.  
Andy Downton, 1991.
- ❑ Gran enciclopedia informática.  
Ediciones Nevalente, 1986.
- ❑ Hypertext in context.  
Cliff McKnight, Andrew Dillon, 1991.
- ❑ Ilustrating computer documentation.  
William Horton, 1991.
- ❑ Imaging and animation for Windows.  
Nabajyoti Barkakati, ph. D., 1993
- ❑ Introducing computers.  
Roberth Blissmer, 1993.
- ❑ Introduction to computers and information systems.  
Szymanski Robert A. 1988.
- ❑ Opening Windows.  
Joseph F. Fournier II, 1993.
- ❑ Programación en Windows 3.1.  
William H. Murray y Chlrís H. Pappas, 1993.
- ❑ Programming the user interface principles and  
examples.  
Judith R. Brown, 1989.
- ❑ Windows 3.1 guide to programming.  
Microsoft.
- ❑ Windows 3.1 programmer's reference.  
James W. McCord, 1992.
- ❑ Windows API bible.  
James L. Conger.
- ❑ Windows bitmapped graphics.  
Steve Rimmer.