

78  
291



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

METODOLOGIA PARA EL DISEÑO Y DESARROLLO  
DE PAGINAS EN INTERNET

T E S I S  
QUE PARA OBTENER EL TITULO DE  
INGENIERO EN COMPUTACION  
P R E S E N T A N  
RENE PEREZ ESPINOSA  
EFRAIN RODRIGUEZ CERVANTES  
ISAAC VIVAS ESCOBEDO

DIRECTOR DE TESIS:  
M. EN C. SERGIO CASTRO RESINES



México, D. F.

1997

TESIS CON  
FALLA DE ORIGEN



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**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.

**A DIOS, que me ha permitido llegar hasta donde estoy y que me ha ayudado en cada acción de mi vida incondicionalmente.**

**A mis PADRES que me han dado valores y educación así como todo su amor.**

**A mis HERMANOS por todo su apoyo comprensión y cariño que me han brindado.**

**A mis AMIGOS que de alguna manera han contribuido para la realización de este trabajo, por su tiempo y por su ayuda.**

**Al departamento de Análisis Desarrollo y Mantenimiento de Sistemas de la Subdirección de Informática de la Dirección General de Bibliotecas de la UNAM, por el apoyo para la realización de este trabajo.**

**René Pérez Espinosa**

**A Dios :**

Por haberme dado las fuerzas para llegar hasta aquí y por darme la oportunidad de vivir tan intensamente.

**A mis Padres :**

Por todo el apoyo que me han dado en todo momento, por darme la oportunidad de recibir una educación , por haberme guiado por un camino que me ha dado grandes satisfacciones y por todo su amor para mí. GRACIAS.

**A mis Hermanos :**

Alberto, Raúl, Jaquelinn y Griselda,

Por todo el apoyo, motivación y cariño que me han dado, para terminar mi carrera; gracias y les deseo lo mejor de la vida a todos.

**A todos mis Familiares :**

Por haberme apoyado siempre y motivarme para continuar con mis Estudios y por todo el aprecio y afecto que siempre me han demostrado.

**A M.C. Sergio Castro Resines :**

Por haber aceptado dirigir este trabajo sin conocernos, por apoyarnos en la elaboración y guiarnos hasta la conclusión del mismo y principalmente por brindarnos tu amistad.

**A los Amiguitos de la DGB :**

A prácticamente todos los amigos en la DGB (Sol ,Dany, Pedro, Arcadio, Laura, Isaac, Sergio, Alex, Daniel, etc., y mil perdones por no mencionarlos a todos) por el apoyo que siempre tuvimos de ellos y de la dependencia para la realización de este trabajo y por la amistad que nos brindaron.

**A mis Amigos (as) y personas que de alguna forma me han motivado y apoyado para llegar hasta aquí; ya que a algunas de ellas las recordare siempre...**

Gracias a todos ....

**Efrain Rodríguez Cervántes**

**A MIS PADRES** por su  
gran esfuerzo y estímulos  
brindados para cumplir  
esta meta.

**A MIS HERMANOS JOSÉ Y  
GABRIEL** por sus consejos,  
apoyo y ánimo.

**A MARTHA** por su  
amor y confianza que  
son la luz que guía mi  
vida.

**Isaac Vivas Escobedo**

## AGRADECIMIENTOS

**Agradecemos al Maestro en Ciencias Sergio Castro Resines por la dirección, entusiasmo y constante apoyo para la realización de este trabajo.**

**Al Maestro Adolfo Rodríguez Gallardo y al Matemático Alejandro Ramírez Nieto por su apoyo, confianza y facilidades brindadas para el desarrollo del mismo.**

**A la Lic. Leticia López Anguiano por su apoyo y facilidades brindadas para la recopilación de la información bibliográfica y hemerográfica.**

**Nuestro reconocimiento a la Lic. Isabel Rodríguez y la Lic. Maira Camacho por sus asesorías en algunos aspectos de este trabajo.**

**A la Maestra en Ciencias Martha Olvera García por sus valiosas observaciones y sugerencias que ayudaron en gran medida al desarrollo de la tesis.**

**Este trabajo se realizó en el departamento de Análisis, Desarrollo y Mantenimiento de la Dirección General de Bibliotecas de la Universidad Nacional Autónoma de México.**

**Isaac Vivas Escobedo**

# ÍNDICE

<b>INTRODUCCIÓN</b>	<b>1</b>
<b>Capítulo 1</b>	
<b>EL WORLD WIDE WEB</b>	
1.1 Antecedentes	5
1.2 El WWW bajo el modelo Cliente/Servidor	7
1.3 Elementos del WWW	9
1.4 El Servidor	10
1.5 El Cliente	11
1.6 El Lenguaje de intercambio de información	11
<b>Capítulo 2</b>	
<b>INTERNET Y TCP/IP</b>	
2.1 Origen de <i>Internet</i> y TCP/IP	13
2.2 TCP/IP y la conmutación de paquetes	14
2.2.1 TCP/IP como norma abierta	16
2.2.2 TCP/IP como norma modular	16
2.3 Relación de TCP/IP y el modelo OSI	17
2.4 Descripción de TCP/IP	18
2.4.1 Nivel de Protocolo <i>Internet</i> (IP)	19
2.4.2 Nivel de transporte	19
2.4.3 Protocolos específicos del nivel de aplicación	20
2.5 Redes y Subredes	22
2.5.1 Los <i>hosts</i> y sus diversos nombres	22
2.5.2 Direcciones IP y su origen	23
2.5.3 Direcciones IP especiales	23
2.5.4 Nombres de <i>host</i> y dominios	24
2.5.5 Nombres <i>Ethernet</i> y su utilización	25
2.5.6 Ruteadores y rutas	25
2.5.7 Máscara de red y difusión	26
2.5.8 Funciones de seguridad de TCP/IP	26
2.6 TCP/IP y DOS	27
2.6.1 Pilas de protocolos dobles	28
2.7 TCP/IP en Macintosh	28

### Capítulo 3 EL SERVIDOR

3.1 Selección del <i>hardware</i>	32
3.2 Selección del <i>software</i> HTTP	32
3.3 Instalación del <i>software</i> HTTP	40
3.4 Compilación del archivo <i>httpd</i>	56
3.5 Configuración del HTTP	57
3.6 Ejecución del archivo <i>httpd</i>	65
3.7 Verificación de la puesta en marcha del servidor <i>Web</i>	67

### Capítulo 4 EL CLIENTE

4.1 <i>Hardware</i> para el cliente	69
4.1.1 MÓDEM	70
4.1.2 Tarjeta de interfaz de red	70
4.2 <i>Software</i> auxiliar de enlace para el cliente	71
4.2.1 <i>Software</i> de interfaz de red	71
4.3 <i>Socket</i> para Windows	72
4.4 Características del <i>browser</i>	76
4.5 Acceso al WWW y despliegue de elementos multimedia	77
4.6 Acceso a los recursos de <i>Internet</i> a través de URL, URI, URN y URC	77
4.6.1 Construcción y sintaxis del URL	79
4.6.2 Especificaciones de los URL	81
4.7 Herramientas de búsqueda en el WWW	88

### Capítulo 5 DISEÑO DE PÁGINAS EN *INTERNET*

5.1 Origen del HTML	97
5.2 Presentación del lenguaje HTML	99
5.3 Estructura del HTML	100
5.3.1 Cómo se utilizan las etiquetas HTML	101
5.4 Etiquetas HTML de estructura	101
5.5 Etiquetas HTML de formato de párrafos	103
5.6 Etiquetas HTML de formato de carácter	104
5.7 Etiquetas HTML de formato de listas	105
5.7.1 Lista no numerada	106
5.7.2 Lista numerada	106
5.7.3 Lista de glosario (listas de definición)	106
5.8 Etiquetas HTML de hiperenlaces	108



5.9 Etiquetas HTML de integración (multimedia)	110
5.10 Editores HTML	115
5.10.1 HotMetal y HotMetal Pro	116
5.10.2 HotDog y HotDog Pro	118
5.10.3 HTML Writer	119
5.10.4 HTML Assistant	120
5.10.5 Microsoft Front Page Editor	120
5.10.6 HTMLed	121
5.11 Herramientas gráficas	123
5.11.1 PaintShop Pro (para Windows)	123
5.11.2 Corel Draw 5.0	125

## **Capítulo 6**

### **CGI**

6.1 Programación de los CGI	130
6.2 Etiquetas HTML vinculadas con las formas	137
6.3 Método POST y GET	139
6.3.1 Método POST	139
6.3.2 Método GET	140
6.4 Variable de ambiente del CGI	140

## **Capítulo 7**

### **TENDENCIAS EN EL WWW**

7.1 Animación de imágenes en formato GIF	147
7.2 Aplicaciones JAVA	148
7.3 Perl	149
7.4 Realidad Virtual (VR)	150

## **Capítulo 8**

### **El WEB DE LA DGB**

8.1 Creación de las páginas	154
8.2 Aplicación de un CGI	159

<b>CONCLUSIONES</b>	<b>164</b>
---------------------	------------

<b>BIBLIOGRAFÍA</b>	<b>166</b>
---------------------	------------

<b>GLOSARIO</b>	<b>169</b>
-----------------	------------

## INTRODUCCIÓN

Debido al gran avance tecnológico de las telecomunicaciones, actualmente se ha logrado la integración de las redes de computadoras, permitiendo que la capacidad de intercambiar información sea mejor y mayor.

La red de redes llamada *Internet*, ha logrado que la información obtenida de las computadoras pueda viajar de un lugar a otro disminuyendo en gran medida las barreras de la distancia y el tiempo, llegando en algunos casos a crear la ilusión de desaparecerlas. Por ejemplo, la asistencia a un curso que se imparte en un lugar que se encuentra al otro lado del mundo a través de la llamada "Educación virtual".

El incremento de la popularidad de *Internet* se ha debido en gran parte al *World Wide Web*, también conocido como *Web*, *WWW* o *W<sup>3</sup>*, que es un sistema distribuido de hipertexto-multimedia. Podemos encontrar una gran variedad de tópicos y personas que desarrollan y/o explotan estos conocimientos.

Anteriormente, sólo las universidades e instituciones gubernamentales ocupaban *Internet*, pero con la aparición del *Web* muchas empresas, han buscado tener su página en *Internet*, descubriendo que resulta ser un gran medio para publicitar sus productos o servicios con un relativo bajo costo a un gran número de usuarios.

¿Por qué actualmente el *Web* tiene éxito?. Debido a que es posible obtener información de audio, video, imágenes y por supuesto texto, además de otros servicios propios de *Internet* como el correo electrónico, el *ftp* y el *telnet*, por mencionar algunos.

A diferencia de otras aplicaciones que también permiten obtener información (como el *gopher*, *WAIS*, *Archie*, etc.), el *Web* tiene dos ventajas principales: a) el usuario no requiere de una clave y un *password* en particular para acceder al servidor del *Web*, pues basta con conocer la dirección para comenzar a visualizar la información; b) el impacto visual que recibe el usuario, aunado a la facilidad de uso, resultando mucho más atractivo.

Sin embargo, cuando uno ha tomado la decisión de tener una página en *Internet*, surge una cascada de preguntas: ¿qué elementos de software requiero para instalar una página

en Internet?, ¿de dónde los obtengo?, ¿en dónde instalo mis archivos?, ¿que *software* es mejor?, ¿qué *hardware* se requiere?, ¿quién me puede asesorar al respecto?, etcétera.


Es indudable que para la instalación, el diseño y la construcción de un *Web* se requieren de ciertos conocimientos en el área de cómputo, sin embargo, ¿cuánto tiempo tomaría adquirir estos conocimientos y traducirlos en la creación de un servicio de este tipo? Seguramente el tiempo estimado no estaría acorde con los tiempos para resolver este problema. Es por ello que, palpando esta necesidad de digerir la mayor cantidad de conocimientos asociados con el *Web*, surge el presente trabajo, cuyo objetivo es proporcionar una guía con los conocimientos básicos y necesarios para aquellas personas interesadas en la instalación, diseño y construcción de un *Web*. Al realizar este trabajo señalamos una metodología para su instalación, así como una breve descripción de estos elementos y cómo se pueden emplear considerando la experiencia obtenida al implantar el *Web* de la Dirección General de Bibliotecas (DGB) de la Universidad Nacional Autónoma de México (UNAM).

Para una mejor identificación de los elementos del *Web*, el presente se ha dividido en cuatro partes. La primera de ellas engloba los conceptos básicos que consideramos importantes, así como la metodología para la instalación del servidor *Web*. Cabe aclarar que debido a que actualmente existe una gran cantidad de *software* para este tipo de servidores, consideramos el caso particular de uno de ellos conocido como "Apache versión 0.8.8", previo un análisis, y a partir de éste desarrollamos la citada metodología, además de otras consideraciones que se mencionan en el capítulo correspondiente al servidor. Esto abarca desde el capítulo 1 hasta el capítulo 3.

La segunda parte se refiere al cliente, es decir, hace una descripción de los elementos requeridos tanto a nivel de *hardware* como de *software*, además de describir las características del cliente. Esto se encuentra en el capítulo 4.

Los capítulos 5 y 6, que corresponden a la tercera parte, describen la metodología para la creación de una página en Internet y el uso básico de herramientas auxiliares, y el capítulo 7 muestra una breve explicación de lo que es hoy por hoy una de las herramientas más poderosas para desarrollar aplicaciones interactivas en el *Web*, el Java.

Por último, el capítulo 8 es el resultado de la realización de los capítulos anteriores, es decir, menciona algunos de los servicios que se proporcionan en el *Web* de la Dirección General de Bibliotecas y que hacen uso de una interface llamada CGI, que permite hacer más dinámico el uso de las páginas.



Como notas importantes para el resto de los capítulos de este trabajo mencionaremos que algunas referencias se tomaron de documentos electrónicos que se encontraron en su momento en algún servidor o en alguna dirección de *Web* específica, por lo que no se tiene la certeza de que este servidor o dirección se encuentren aún vigentes o bien su información no haya sido modificada o eliminada.

Por otra parte, decidimos respetar los términos en inglés, como se mencionan en la bibliografía consultada y no traducirlos; éstos se encuentran marcados en letra cursiva.

.....



# Capítulo 1



.....

## EL WORLD WIDE WEB

Internet debe en gran parte su popularidad a la capacidad que tiene de integrar diferentes plataformas, ambientes, protocolos, tecnologías y topologías de redes de computadoras, además de los avances tecnológicos en el campo de las telecomunicaciones. Por otro lado, la flexibilidad y eficiencia de TCP/IP hacen posible que se lleve a cabo la comunicación entre dos computadoras de la misma o diferente tecnología, de tal forma que no se requiere conocer las características de éstas para que se realice un intercambio de información. Esta flexibilidad proporciona un cierto grado de interoperabilidad o sea, "la capacidad de que diferentes sistemas de cómputo, redes, sistemas operativos y aplicaciones puedan trabajar conjuntamente y compartir información"<sup>1</sup>.

En el caso de Internet, aun cuando a través de TCP/IP se pueden intercambiar archivos y establecer una sesión remota con otra computadora, o enviar correo electrónico, no se puede compartir la información con cualquier usuario, ya que es necesario tener una cuenta, en alguna máquina que esté conectada a Internet, para tener acceso a información que se encuentre disponible en otras computadoras.

Debido a esto, y aunque existe un gran número de usuarios de Internet que tienen esa posibilidad, no se compara con el número de personas que actualmente pueden utilizar el *World Wide Web*, que ha popularizado en mayor medida a Internet y al que en la actualidad un mayor número de usuarios está incorporándose. Con el *World Wide Web* no es indispensable tener una cuenta, no se requiere utilizar alguna computadora con sistema operativo UNIX y se puede tener una cierta interacción en la información que se desea obtener.

### 1.1 Antecedentes

El *World Wide Web* es llamado a menudo W3 o W<sup>3</sup> debido a la W inicial en cada palabra; o nombrado *Web* porque se puede homologar en cierta medida a Internet con un diccionario inmenso (*Web* son las letras iniciales de *Webster*, uno de los diccionarios más famosos); o WWW por las iniciales, como se manejará en adelante; es un sistema distribuido de hipertexto multimedia: es *distribuido* porque la

<sup>1</sup> Lan times: guide to Interoperability. Edit. México: McGraw Hill Página xxiii

información sobre el WWW se localiza en sistemas de computadoras alrededor del mundo; multimedia, ya que la información que se encuentra sobre el WWW puede incluir texto, gráficos sonido y video de forma integrada, e *hipertexto*, dado que el acceso a la información está disponible usando técnicas en las que típicamente se utiliza un ratón (*mouse*) para seleccionar frases subrayadas o imágenes. Una vez que se seleccionó una frase o imagen, puede ser información que se recuperó de ese mismo documento o de otro documento que puede estar en la misma computadora o en otra en cualquier parte del mundo.

El WWW que toma su nombre telaraña por la manera en que está estructurada la información en él, fue diseñado en 1989 por Tim Berners-Lee y Robert Cailliau del Laboratorio Europeo de Física de Partículas (CERN), con el fin de proveer una infraestructura para los físicos y poder compartir la información en Europa, ya que los investigadores se encontraban en diferentes organizaciones y utilizaban una gran variedad de sistemas de cómputo y de *software* de aplicación (incluyendo varios procesadores de texto para producir sus reportes).

La forma de intercambio de información se realizaba a través de una sesión *Telnet*, por lo que solamente se podía desplegar información en modo carácter y, si existían opciones o ligas, éstas se seleccionaban por medio de números que se encontraban al final de cada una de las ligas, las cuales enlazaban a otra información. Por ejemplo, uno de los primeros programas que se desarrolló fue el de comando en línea del CERN. Este podía ser accedido usando la orden:

```
telnet telnet.w3.org
```

Un ejemplo del uso del programa de comando en línea del CERN se muestra en la figura 1.1

En sus inicios, el WWW no contaba con todos los elementos multimedia como los que existen en la actualidad como son texto, gráficas, sonido o inclusive animaciones y video. Fue hasta 1993 cuando el WWW alcanzó gran popularidad una vez que apareció Mosaic, teniendo versiones para tres plataformas: X Windows de UNIX, MS Windows y Mac. Mosaic es un *software* conocido como *browser* o visualizador para el WWW fácil de utilizar y que trabajan en un entorno gráfico, el cual permite desplegar texto con formato y elementos multimedia; desarrollado por el Centro Nacional para Aplicaciones de Supercómputo (NCSA) de la Universidad de Illinois.

La importancia de Mosaic es que, desde su aparición en *Internet*, cambió radicalmente la proporción de usuarios y servidores de WWW.

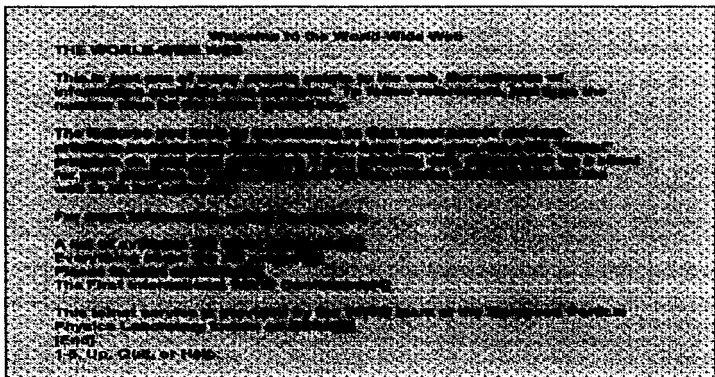


Figura 1.1 Sesión de telnet para entrar al WWW

## 1.2 El WWW bajo el modelo Cliente/Servidor

El avance en los medios de comunicación tanto en enlaces vía satélite, como enlaces terrestres, líneas telefónicas de alta velocidad de transmisión, fibra óptica, dan la posibilidad de que cada vez más sitios se puedan integrar a Internet y a su vez Internet también evolucione, sin embargo, no sería posible hablar del WWW si no existieran avances en la forma de compartir la información, de distribuirla y almacenarla, además de contar con mecanismos que den soporte a estas necesidades y que sean transparentes tanto para los desarrolladores como para los usuarios finales, para lo cual se requiere de una tecnología.

En el caso del WWW se basa en el modelo Cliente/Servidor, el cual se refiere a una relación entre dos sistemas o procesos. El cliente es el sistema que hace una petición de trabajo para ser realizada a encargo de éste por el sistema servidor. En muchas



situaciones, cual es el cliente y cual es el servidor es determinado por las relaciones de petición (cliente) al servidor.

La tecnología Cliente/Servidor surgió para aplicaciones de bases de datos, sin embargo, en la actualidad su utilización no se restringe solamente a ese campo.

Las características principales de un modelo cliente servidor son:

- **Procesamiento distribuido.** Significa poder ejecutar un proceso que se subdivide en varias computadoras, todas ellas conectadas en red, y en el cual cada una realizará parte de ese proceso de manera independiente; en conjunto forman una sola entidad para la finalización del proceso.
- **Cómputo distribuido.** Este término significa que se requiere de varias computadoras para llevar a cabo un proceso distribuido, las cuales pueden o no ser de la misma tecnología tener diferentes sistemas operativos, están conectadas en red, no necesariamente se encuentran en la misma área geográfica y disponen de procedimientos y protocolos para intercambio de información, de manera que en conjunto forman una entidad de cómputo.
- **Integridad de las transacciones.** Una transacción es definida como una serie de acciones que necesitan ser completadas en conjunto. La integridad de las transacciones son los mecanismos necesarios para asegurar la integridad de los datos en un proceso distribuido, con la capacidad de detectar la si se ha corrompido en la información y poder ejecutar procesos para su corrección y/o control.
- **Seguridad del sistema.** Se deben contar con mecanismos eficientes para poder restringir el acceso al sistema, a un proceso o a datos, tanto del cliente como del servidor.
- **Cliente.** El cliente como es quien realiza peticiones a un servidor para que realice cierto proceso; está conformado de dos partes: de un *hardware*, una computadora, la cual dispone de un *software* que permite hacer peticiones e interpretar las respuestas provenientes del servidor.
- **Servidor.** El servidor, al igual que el cliente, se compone de un *hardware* y un *software*, pero en general es de mayor capacidad en los dos aspectos; en la mayoría de los casos, son computadoras que sean capaces de realizar multitarea y multiproceso, debido a que realizan la parte más compleja de todo el proceso;

además, se tiene la opción de que varios clientes hagan peticiones a simultáneamente a un mismo servidor.

Los elementos que contempla el modelo Cliente/Servidor son aplicados en el WWW y en conjunto proporcionan un sistema que es fácil de utilizar y varios aspectos del modelo son transparentes para el usuario (tales elementos se describirán tomando en cuenta su función en el WWW).

### 1.3 Elementos del WWW

El usuario, ingresa al WWW a través de una computadora cliente (porque realiza una petición), la cual puede ser una PC, Macintosh o inclusive otro servidor, que dispone del *software* necesario para recibir e interpretar la información. El cliente desplegará la información que el servidor de WWW le envía, en respuesta a la petición del cliente. Lo que el servidor envía es un archivo que contiene información en formato HTML, éste documento HTML puede contener elementos multimedia además de hipertexto, que son ligas a otros documentos HTML las cuales pueden estar en frases subrayadas o imágenes. Una vez seleccionada una liga (al pulsar el botón del ratón, si se utiliza un *browser* gráfico) se envía una petición por la red (que podría ser una red local, una red nacional o sobre una red global como Internet) al servidor que almacena dicho documento. La petición es enviada a un servidor de WWW, que típicamente corre sobre un sistema poderoso de computadora. El servidor recuperará el archivo que se ha solicitado y lo entrega al cliente.

Una vez que el cliente ha comenzado a recuperar el archivo, podrá comenzar a desplegarlo a través del *software* instalado sobre la máquina local. Si el cliente no puede desplegar el archivo (muchos clientes, por ejemplo, no pueden ver videos), puede pasar el archivo a un *software* auxiliar para procesarlo.

Considerando lo anteriormente descrito se pueden distinguir tres elementos que conforman al WWW como sistema, los cuales son:

- Un servidor de WWW conocido como servidor HTTP.
- Un cliente que disponga del *software* necesario para enviar o recibir peticiones al servidor .
- El lenguaje de intercambio de información, HTML.

La figura 1.2 muestra esquemáticamente la relación de los elementos del WWW

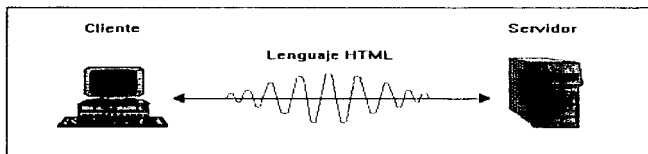


Figura 1.2 Elementos del WWW

La facilidad y transparencia que se proporciona con estos elementos y la incorporación de información de universidades, compañías y toda clase de personas con elementos multimedia hacen al WWW una atractiva manera de comunicación. En seguida se describen detalladamente cada uno de los elementos mencionados, los cuales se basan en el modelo Cliente/Servidor.

## 1.4 El Servidor

El servidor es una computadora que sea capaz de realizar multiproceso, multitarea y que disponga de la *software* requerido para tales funciones; además en esta máquina, se debe tener el *software* requerido para el manejo del protocolo HTTP (*HyperText Transfer Protocol*), que es el protocolo de consulta de hipertexto y acceso al WWW, por lo que se le conoce como servidor HTTP. Es *software* que se encarga de manejar la información que se publica en el WWW; además, tiene la función de responder a las peticiones hechas por los clientes y también se encarga de la seguridad de la computadora en donde está instalado, ya que debe tener la capacidad de restringir el acceso a archivos y directorios que no son públicos.

Por lo general, las computadoras en las cuales se instalan los servidores HTTP son estaciones de trabajo con sistema operativo UNIX, sin embargo, actualmente existen computadoras PC de gran velocidad y con gran cantidad de memoria, que ejecutan LINUX (sistema operativo equiparable a UNIX que se instala en PC), las cuales también pueden ser servidores HTTP.

La mayoría del *software* para que una computadora sea servidor HTTP es desarrollado para ejecutarse en sistema operativo UNIX o VMS, pero existe, aunque en menor medida, *software* para sistemas como Windows NT o Windows 95.

## 1.5 El Cliente

El cliente es una computadora que dispone del *software* necesario para realizar y atender peticiones. Entre este *software* se encuentra el llamado *browser*, que permite desplegar los archivos que sean enviados por algún servidor HTTP y navegar por *Internet* en forma sencilla. En este caso, la computadora puede ser de cualquier tipo, ya que lo único que se requiere es que tenga el *software* para visualizar el documento HTML y el *hardware* para el despliegue de elementos multimedia. Como ya se mencionó, Mosaic fue el primer *browser* que trabajó en ambiente gráfico. Existen actualmente una gran variedad de *browsers*, para diversas plataformas y diversos sistemas operativos. La principal diferencia entre los diferentes tipos de clientes son los servicios, también conocidos como URLs, que pueda utilizar.

Los URL (*Uniform Resource Locator*) permiten la conexión entre un cliente y un servidor por medio de un protocolo que no necesariamente debe ser HTTP, teniéndose *ftp*, *Gopher*, *News*, *Telnet*, *file*, *mail*, *WAIS*.

## 1.6 El Lenguaje de intercambio de información

El lenguaje de intercambio de información entre el cliente y el servidor en el WWW es el lenguaje HTML (*HyperText Markup Language*). Se caracteriza por ser un lenguaje para edición de documentos electrónicos que trabaja con base a etiquetas, las cuales modificarán la forma en que se desplegará la información, pero que solamente sirven para indicar cómo se realizará ese despliegue, aunque no son las encargadas de hacerlo.

El conjunto de etiquetas de que dispone el lenguaje HTML son instrucciones para que el *browser* tenga parámetros de cómo desplegar el documento recibido; si el documento contiene elementos multimedia, el servidor envía los archivos requeridos y el *browser* con las instrucciones contenidas en el documento HTML, que sirven para que esos archivos sean desplegados en forma correcta.



# Capítulo 2

## Capítulo 2

# INTERNET Y TCP/IP

## 2.1 Origen de Internet y TCP/IP

A mediados de 1960, un hombre llamado J. Licklid er propuso establecer comunicación entre computadoras para transmitir los mensajes del gobierno de los Estados Unidos de Norteamérica. En 1967, la Universidad de Michigan tomó el proyecto denominado ARPANET (*Advanced Research Projects Agency Network*). En 1968, el Departamento de Defensa de los Estados Unidos concesionó el proyecto a una compañía llamada BBN (Bolt, Beranek, and Newman, Inc.); este proyecto constaba de dos objetivos: el militar y el de investigaciones.

El objetivo militar era el de mantener en curso toda la información del gobierno mientras los Estados Unidos estuvieran en guerra, por ser éste un sistema descentralizado para resistir ataques, operable al cien por ciento desde cualquier punto de la red y con la capacidad de funcionar correctamente bajo las condiciones ideales, aun con la pérdida de una parte física de la red.

El objetivo de investigación tenía las siguientes características: era un vehículo de investigación para compartir la información en lugar de duplicarla, minimizar tiempos de desarrollo y motivar la cooperación entre instituciones localizadas en diferentes sectores. El resultado fue una red muy robusta. Finalmente, el proyecto constaba de múltiples líneas telefónicas conectadas a un *hardware* especializado denominado nodo, en el que los nodos podían aceptar llamadas de las computadoras conectadas a él. Esto fue el inicio del ahora denominado correo electrónico, concebido con las mismas características que las actuales.

El primer nodo de ARPANET fue instalado en la Universidad de California en los Ángeles, en 1969; tres nodos más fueron instalados en universidades de California (Santa Barbara y Stanford Research Institute ) y en la Universidad de Utah, conformando así el total de cuatro nodos. En 1970, se instalaron tres nodos en distintas universidades ARPANET quedó integrado por siete nodos trabajando con el protocolo NCP (*Network Control Protocol*) y *Host to Host Protocol*.

Alrededor de 1973. ARPANET había crecido a 25 nodos en los Estados Unidos y continuó trabajando con el primer objetivo planeado originalmente por el gobierno,

pero más tarde se comenzó a difundir su funcionalidad hasta llegar a un mayor número de investigadores y científicos, quienes tenían que absorber el mantenimiento de su conexión por un costo aproximado de 250000 dólares.

A finales de los años setenta, ARPANET se estaba acercando a su máximo soporte con 256 máquinas conectadas; el protocolo NCP no podía soportar ya el tráfico de la red y se necesitaba un sucesor. Alrededor de 1974 se hizo necesario un esquema de trabajo en red que pudiera manejar más nodos. DARPA (*Defense Advanced Research Projects Agency*) también estaba interesada en expandir el trabajo en red, desde las líneas arrendadas hasta la conmutación de paquetes por radio y satélite; éste se convirtió en el segundo criterio de diseño del nuevo esquema de trabajo en red. La conversión completa de NCP de ARPANET a TCP/IP se completó en 1983 y en 1985 la NSF (*National Science Foundation*) propuso extender la red para dar acceso fácil a investigadores y científicos, instalando cinco supercomputadoras más a lo largo de Estados Unidos.

Una vez puesto en marcha este proyecto, la cantidad de conexiones excedía la capacidad de la red, aun con el nuevo protocolo IP. En 1987 la NSF anunció una sociedad estratégica para resolver los problemas de la red, con las compañías MERIT Inc. -operadora de redes-, IBM -manufacturera de computadoras- y MCI -servicios de telefonía-. Esta sociedad trajo como consecuencia en 1988, la instalación de NSFNET. Finalmente, en 1990 ARPANET original se unió a NSFNET, además de otras pequeñas redes que se habían formado, dando como resultado la unión de redes que actualmente se conoce como *Internet*.

TCP/IP no se desarrolló para proporcionar soluciones a las redes modernas, fue creada para resolver los problemas de red de finales de los años setenta y principios de los ochenta, no obstante, ha superado la prueba del tiempo y evolucionado para cubrir las necesidades de su comunidad de usuarios. Muchas de las funciones de TCP/IP se incorporaron en las especificaciones de diseño de OSI (*Open Systems Interconnection*) de la organización internacional de normalización.

## 2.2 TCP/IP y la conmutación de paquetes

Muchas de las funciones de TCP/IP se incorporaron en casi todas las redes que se utilizan, esto no cambia el hecho de que muchos de estos conceptos se crearon con TCP/IP y su ancestro, NCP. TCP/IP puede caracterizarse como una red que puede

recuperarse frente a fallas en los componentes del *Host*, debido a que utiliza conmutación de paquetes.

La conmutación de circuitos se probó en el experimento NCP original, estableciendo un enlace orientado a la conexión o circuito entre dos nodos; éstos tendrían una conversación y luego se desconectarían. Un problema de esta configuración es que es necesario tener un circuito específico para conectar cada nodo con los demás. Este método sirve para dos o tres nodos, pero cuando se pasa a 10 o 100 nodos, el resultado es una mezcla de cables. Otro problema se produce cuando se rompe la conexión entre dos nodos, que pueden "hablar" con todos los demás, pero no entre sí.

Debido a que el gobierno quería una red que pudiera manejar conexiones indisponibles, se eligió la conmutación de paquetes para crear una red más coherente y fiable. Con la conmutación de paquetes las transmisiones sin conexión pueden seguir muchas rutas entre el nodo de origen y el de destino, los datos que se van a intercambiar se ponen en un paquete y se etiquetan con la dirección del nodo destino. Este paquete de datos se transmite entonces al siguiente sistema de la red; si este sistema no es el indicado en la dirección, transmite el paquete al siguiente destino más probable.

En la figura 2.1, si la computadora "A" envía un mensaje a la computadora "E", el mensaje podría ir por "B", a continuación por "D" y finalmente llegaría a "E". Si la conexión entre "B" y "D" no estuviera disponible por alguna razón, el mensaje podría reencaminarse desde "A" hasta "C", "D" y finalmente "E".

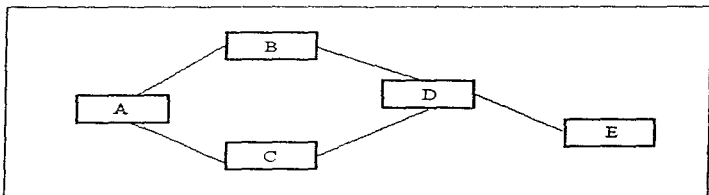


Figura 2.1 Esquema de conexión entre dos computadoras



En esta configuración la conexión con mayor riesgo es la que se encuentra entre "D" y "E". Si se rompiera dicha conexión, "E" quedaría aislado. Para evitar esta clase de riesgo, la mayoría de las redes en la actualidad tienen al menos dos conexiones principales entre anfitriones.

### 2.2.1 TCP/IP como norma abierta

Los protocolos TCP/IP no sólo se normalizaron, también se hicieron disponibles públicamente. Esta fue, y sigue siendo, una variación significativa en la política del gobierno de los Estados Unidos sobre la nueva tecnología; además, se propuso incorporar el conjunto de protocolos TCP/IP a la variante de Distribución Normalizada de Berkeley (BSD, *Berkeley Standard Distribution*) del sistema operativo UNIX. En el momento de incorporar TCP/IP a BSD, éste se encontraba ampliamente distribuido por colegios y universidades de todo el país en formato de código fuente. La posibilidad de ver este código aceleró el desarrollo de protocolos adicionales para uso específico. Al empezar a conectarse las PC a las redes, se utilizaron estas normas abiertas para integrar los sistemas en redes más grandes.

### 2.2.2 TCP/IP como norma modular

La característica más significativa de TCP/IP es la modularidad del conjunto de protocolos. Suponiendo que se tiene una configuración de red con interfaz de datos distribuidos por fibra óptica (FDDI, *Fiber Distributed Data Interface*) entre edificios de algún campus local, o conexión *Ethernet* con par trenzado entre los sistemas de cada edificio y un enlace T1 con *Internet*, el TCP/IP tiene la capacidad para manejar todos estos medios. TCP/IP es una pila de protocolos, y no sólo es un único protocolo, cada nivel de la pila cumple una función diferente en la tarea de llevar los datos desde el origen hasta el destino.

Con TCP/IP puede utilizarse cable de fibra óptica con *Ethernet*, o cable de par trenzado con la interfaz de datos distribuidos (CDDI, *Copper Distributed Data Interface*), o pueden utilizarse módems con protocolo punto a punto (PPP, *Point-to-Point Protocol*) o SLIP (*Serial Line IP*); esta independencia significa que puede elegirse el mejor método para cada parte de la red.

## 2.3 Relación de TCP/IP y el modelo OSI

TCP/IP se desarrolló como arquitectura en niveles o capas y se diseñó para poder intercambiar las partes de cada nivel según el tipo de conexión que se llegara a utilizar. Como se concibió para funcionar con varias tecnologías distintas, las partes tenían que estar estructuradas de forma que sólo fuera necesario cambiar una pieza para utilizarlo en un medio diferente.

La *International Standard Organization* (ISO) ha establecido el modelo OSI (*Open System Interconnection*) que es el estándar utilizado para describir la tecnología de trabajo en red por niveles. Este modelo se desarrolló varios años después de TCP/IP. En la figura 2.2 se muestra la relación entre los niveles de TCP/IP y OSI.

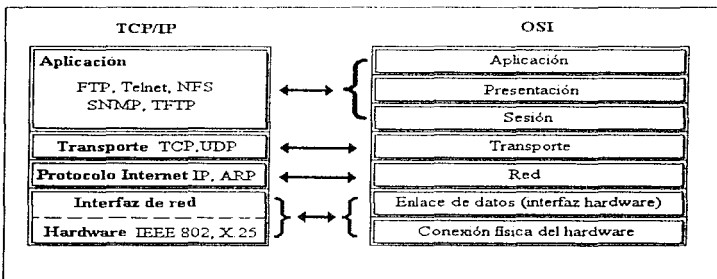


Figura 2.2 Comparación entre TCP/IP y el modelo OSI

En el modelo OSI, la capa **Física** asegura que la información llegue a las computadoras independientemente del tipo de enlace (cable coaxial, módem, fibra óptica, enlace satelital).

La capa de **Enlace de datos** se encarga de llevar a cabo un intercambio de mensajes entre lo que ISO llama IMP (*Interface Message Processor*) y DCE (*Data Communication Equipment*).

La capa de **Red** tiene una dirección lógica IP (*Internet address*) e identifica la dirección de los equipos (PCs, *gateway*, etc.); es la encargada de llevar a cabo el encaminamiento de los paquetes de información.

La capa de **Transporte** establece comunicación entre las diferentes aplicaciones, asegurándose que los paquetes de información sean entregados en orden y sin pérdida o duplicación.

La capa de **Sesión** se conoce también como capa de conexión y su función es permitir a los usuarios identificarse cuando desean ingresar a la red, por ejemplo, *Telnet*.

La capa de **Presentación** tiene dos funciones principales: el encriptamiento de la información y la compresión de los datos.

La capa de **Aplicación** maneja el correo electrónico, el WWW, etcétera.

## 2.4 Descripción de TCP/IP

En la parte inferior de cada una de las pilas de la figura anterior se encuentra el nivel físico de *hardware*, el cual puede ser cable coaxial, cable de par trenzado, fibra óptica o una línea telefónica. En el diseño del modelo OSI se incluye esta parte del trabajo en la capa de red.

El modelo TCP/IP no considera el medio *hardware* como componente específico e independiente en su modelo. TCP/IP tiende a agrupar la interfaz *hardware* con el nivel de interfaz de red.

Independientemente del medio *hardware* que se utilice, necesitará una tarjeta de interfaz de red específica. Estos dispositivos de interfaz de red son específicos del medio *hardware* por el que se transmiten las señales. Cada uno de estos servicios necesita un componente *software* llamado controlador de dispositivo. En la mayoría de los sistemas operativos de red, el controlador de dispositivo debe incluirse con el sistema operativo de base o proporcionado por el fabricante del *hardware*.

Existen protocolos específicos para los niveles de *hardware* y de transmisión, por ejemplo ARP, RARP, SLIP y PPP:

- *ARP y RARP*. Son protocolos específicos de *Ethernet* que se encargan de traducir las direcciones de *Ethernet* a direcciones de IP (Protocolo *Internet*); también se utilizan para arrancar sistemas por la red en lugar de iniciar a partir del disco local.
- *SLIP y PPP*. El IP por línea serie (*SLIP, Serial Line IP*) y el Protocolo punto a punto (*PPP, Point-to-Point Protocol*) se utilizan por líneas serie o telefónicas. Estos protocolos permiten un abanico más amplio de conectividad entre sistemas. Existen algunos proveedores de *Internet* asociados con vendedores de TCP/IP para DOS y Macintosh para generar productos de conectividad directa con *Internet*.

### 2.4.1 Nivel de Protocolo *Internet* (IP)

El siguiente nivel hacia arriba es nivel de Protocolo *Internet* (IP), también conocido como nivel *Internet*; el nivel equivalente al nivel IP de TCP/IP en el modelo OSI se llama nivel de red.

Este nivel se encarga de dirigir y encaminar los datos hasta el sistema de destino. En este nivel se realizaría la conmutación de paquetes en caso de utilizar un puente para cambiar de TCP/IP a DECnet, que es un grupo de productos de comunicaciones (incluyendo protocolos) desarrollados y mantenidos por Digital Equipment Corporation. Una pasarela (término de *Internet* para referirse a los ruteadores) que encamina datos desde una subred a otra también trabaja en este nivel. Todos los demás protocolos se montan sobre el protocolo *Internet*. IP es como el sobre que lleva la dirección, contiene el número IP del sistema al que se dirige el paquete. En este nivel también se maneja el protocolo de mensaje de control de *Internet* (ICMP, *Internet Control Message Protocol*), que es un protocolo de mantenimiento/gestión de red que ayuda a supervisar la red.

El modelo OSI se diseñó para que fuera genérico, en lugar de específico para el TCP/IP o cualquier metodología de red conocida.

### 2.4.2 Nivel de transporte

Ambos modelos concuerdan en llamar a este nivel el nivel de transporte. El método de transporte de los datos determina la rapidez y fiabilidad con que los datos viajan desde el origen hasta el destino. TCP/IP proporciona dos protocolos de transporte, uno de ellos se llama *TCP*. Originalmente sólo había *TCP* y proporcionaba un mecanismo de transporte fiable desde un extremo al otro, ya que envía un paquete o un bloque de paquetes y espera hasta que el sistema de destino confirme la recepción de los datos y

que los datos sean correctos antes de que el sistema emisor envíe más datos. En aquellos tiempos todavía no se disponía de la tecnología de corrección de errores que actualmente se tiene. El *hardware* actual es mucho más fiable que las conexiones originales por satélite y teléfono de larga distancia. La comunicación WAN es mejor que hace una década; no obstante, todavía no se ha llegado a los niveles de rapidez de la comunicación LAN. Debido a la necesidad de conseguir mayores velocidades de transferencia, se desarrolló el Protocolo de datagramas de usuario (UDP, *User Datagram Protocol*). Este no espera una confirmación, envía los datos de forma continua siempre que puede transmitir por la red y responde a los problemas de los que informa el sistema receptor; para este protocolo la falta de noticias es una buena noticia.

### 2.4.3 Protocolos específicos del nivel de aplicación

Éste es el punto en el que el modelo OSI y el modelo TCP/IP empiezan a ir por caminos separados. TCP/IP reconoce cualquier cosa a partir de este punto como un protocolo de aplicación, mientras que el modelo OSI descompone aun más las descripciones.

En la actualidad se utilizan cientos de protocolos de aplicación, algunos todavía están en fase experimental, otros están en espera de un reconocimiento formal. A continuación se tratan algunos de los protocolos más comunes y su utilización.

**FTP (File Transfer Protocol).** FTP junto con *Telnet* son los protocolos de aplicación más antiguos que existen. FTP es un programa para transferencia de archivos en ambientes TCP/IP, tanto en formato ASCII como binario. Típicamente, un usuario en una computadora cliente descarga archivos de un servidor remoto, por lo que se requiere de un nombre de usuario y clave de acceso. FTP sigue generando la mayor parte del tráfico de *Internet*.

**Telnet (Terminal Emulation).** Se utiliza para establecer una sesión o ejecutar programas en un sistema remoto; se requiere de nombre de usuario y clave de acceso. En todas las variantes de TCP/IP se ofrece FTP y *Telnet* en el lado del cliente, independientemente de la plataforma.

**HTTP (HyperText Transfer Protocol).** Es uno de los protocolos más recientes. Se utiliza para manejar la consulta de hipertexto y el acceso a datos en el World Wide Web (WWW o W3). El tráfico de HTTP ha pasado a ser recientemente una parte significativa del tráfico de *Internet* debido al crecimiento exponencial de los servidores

y usuarios de WWW. El tráfico producido por HTTP no es mayor que el producido por FTP. El problema de HTTP es que los programas que lo utilizan pueden solicitar datos de cualquier lado de forma casi instantánea. En el mismo tiempo que un usuario de FTP puede recibir 100K de datos, un usuario con un *browser* como Mosaic puede solicitar cerca de 2MB.

**NNTP (Net News Transfer Protocol).** Se creó para transferir noticias de red por *Internet* sin el engorroso método de transferencia UUCP (*UNIX to UNIX Copy*). Las noticias de red no siempre han estado en *Internet*; al principio se utilizaba el programa UUCP para crear un tablón de anuncios que se empezó a conocer como Noticias de red. Al expandirse *Internet* se desarrolló el protocolo NNTP para transmitir estos comunicados en lugar de utilizar conexiones telefónicas caras.

**NTP (Network Time Protocol).** Permite que todos los sistemas sincronicen su hora con un sistema designado como servidor de "hora".

**SNMP (Simple Network Management Protocol).** Sirve para administrar los sistemas de forma remota. Se ha utilizado mucho anteriormente para supervisar el tráfico de la red y la disponibilidad del sistema. No obstante, este protocolo robusto puede utilizarse para realizar muchas tareas de administración rutinarias, como desconectar una impresora problemática.

**SMTP, X.400, UUCP y POP3.** El correo se ha transmitido con UUCP desde su creación. No obstante, UUCP tiene una sintaxis robusta y críptica. X.400, SMTP y POP3 son protocolos de transferencia de correo más recientes, con muchas más funciones que el UUCP original.

**RIP, EGP, OSPF, IGP y HELLO.** La transferencia de datos del punto "A" al punto "B" es una de las funciones primarias de TCP/IP. El encaminamiento ha evolucionado hasta convertirse en un proceso muy dinámico que maneja la complejidad de la propia *Internet*. Éstos son algunos de los protocolos que difunden información y realizan un seguimiento de la ruta posterior.

**SUN-NFS y SUN-RPC.** Sun Microsystems fue el pionero en la utilización de discos virtuales en TCP/IP cuando desarrolló NFS, con el que se tiene un sistema de archivos distribuido que permite a múltiples usuarios compartir datos en una red, sin tener en cuenta el tipo de procesados, sistema operativo, arquitectura de red o protocolo. Sun descubrió que necesitaban un protocolo específicamente para ello; más tarde desarrolló las llamadas a procedimientos remotos (RPC, *Remote Procedure Call*). La mayoría del tráfico NFS es ahora un caso especial del protocolo RPC.

**NVP-II (Network Voice Protocol)** y **PVP (Packet Video Protocol)**. El Protocolo de voz de red y el Protocolo de video empaquetado todavía están en fase experimental. Serán la base de la expansión de la videoconferencia por *Internet*.

**Domain.** Este protocolo es la base del DNS (*Domain Name System*), el principal servicio de nombres para localizar dominios y sistemas en *Internet*. La información de las direcciones es almacenada en varias localidades en estructura jerárquica, cada sitio tiene un servidor DNS que mantiene la información de los nodos locales.

## 2.5 Redes y subredes

Las redes TCP/IP son entidades sencillas. La unidad más pequeña reconocible es el dispositivo de interfaz de red. Para PC y Macintosh, el dispositivo de interfaz de red es, de hecho, una tarjeta que se añade al sistema; muchos de los sistemas UNIX suelen venir con una tarjeta de red en la tarjeta madre.

El siguiente componente en tamaño es el propio *host*, que es la computadora central que puede tener uno o más dispositivos de interfaz de red. Un *host* tiene un nombre, una dirección IP única y, si este es una red *Ethernet*, tendrá una dirección *Ethernet* o también conocida como MAC (*Media Access Control*), que es la dirección que asigna el proveedor a las tarjetas de este tipo. La dirección *Ethernet* no es un número de *Ethernet* por sistema, es una dirección para cada dispositivo de red que tiene un sistema. Si un sistema dispone de dos tarjetas de red, se le conocerá por dos direcciones *Ethernet* diferentes.

Varios *hosts* conectados entre sí forman una pequeña red llamada subred. Éste es el componente más pequeño de una red TCP/IP. Las subredes se conectan entre ellas mediante pasarelas para formar redes. Las redes se conectan entre ellas para formar *interredes*, de las cuales la más conocida es *Internet*.

### 2.5.1 Los *hosts* y sus diversos nombres

Debido a que se puede tener un conjunto de redes, y de alguna forma se tienen que identificar, TCP/IP se diseñó para utilizar nombres de *hosts* que fueran palabras normales; se hizo así porque no es fácil recordar una larga cadena de números para cada sistema al que se tiene que acceder; no obstante, los propios sistemas no utilizan el nombre del *host* al comunicarse con otro sistema, sino números en lugar de nombres

porque son más fáciles de manejar. El primer número, y el único que se utiliza siempre, es el número IP, que es el identificador del nivel de IP.

### 2.5.2 Direcciones IP y su origen

La dirección de IP consta de cuatro campos de números separados por puntos. También se conoce como notación decimal puntuada. El número de cada campo puede valer hasta 255. Esto se debe a que cada uno de estos números utiliza un byte (8 bits) para cada sección de la dirección. Cada uno de estos campos también se llama *octeto*. Un ejemplo de dirección de IP es 132.248.67.2.

Una parte de cada dirección está dedicada al número de ID (identificación) de la red, y el resto se ocupa de identificar al anfitrión específico dentro de esa red. Los rangos de las direcciones se dividen en clases, en función de qué parte de la dirección se refiere a la ID de la red y cuál está reservada para identificar anfitriones específicos. Por ejemplo, es habitual oír una referencia a la "red 132.248.67". En este caso, 132.248.67 es la parte de red. La parte de ID del anfitrión no se menciona al referirse a la red como un todo. Una red de clase B se referiría como la "red 129.144".

Rangos de direcciones del protocolo *Internet*.

Tipo	Rango	ID de red
Reservado	0.xx.xx.xx	
Clase A	1.0.0.0 - 126.255.255.255	Primer octeto.
Reservado	127.0.0.xx	Bucle de retorno o dirección del anfitrión local
Clase B	128.0.0.0 - 191.255.255.255	Dos primeros octetos
Clase C	192.0.0.0 - 223.255.255.255	Tres primeros octetos
Multidifusión	224.0.0.0 y superiores	Estas direcciones se encuentran reservadas para un tipo especial de difusión

El rango está en el formato *nnn.hhh.hhh.hhh*; *nnn* es el ID de la red y *hhh* es el ID del *host*.

### 2.5.3 Direcciones IP especiales

A la red como un todo se accede poniendo a 0 toda la parte de ID del *host* de la dirección de IP. Un ejemplo de ello es 132.248.0.0. Esto se utiliza al definir información de encaminamiento.



Existe un segundo tipo especial de direcciones de IP llamado difusión, que envía un mensaje a un grupo de sistemas en lugar de a un solo sistema. En el convenio BSD, el direccionamiento con todos ceros también se utilizaba para la difusión. El Sistema V y las versiones más recientes del sistema operativo UNIX utilizan la ID de anfitrión 255 para realizar la difusión a toda la red. Cada sistema lo utiliza al recibir información de encaminamiento y de nombre de *Ethernet*. Los números de ID del *host* 0 y 255 están reservados para la difusión en función de las normas de red. Esto suele ser configurable al instalar el *software* de la red. Idealmente, cada sistema debería atender a los mensajes de difusión tanto 0 como 255. En realidad, si un sistema está cargado, puede que nunca responda a una difusión distinta de aquella para la que está configurado.

Ahora también se utiliza la multidifusión, que sirve para realizar una difusión a sistemas seleccionados en diversas redes. *Internet* utiliza la Red MBONE (*Multicast Backbone*) para transmitir varios tipos de datos de multidifusión. Una uso importante de la multidifusión incluye el video y el sonido de varias misiones del transbordador espacial retransmitidas en directo. De forma local, varias compañías están experimentando con videoconferencia por medio de la multidifusión. Esta tecnología ocupa gran parte del ancho de banda de una red. En una red normalizada *Ethernet* a 10 Mbps se recomienda que cada utilidad de tráfico de multidifusión mantenga sus tasas de refresco y transmisión y recepción alrededor de 100 Kbps para evitar la degradación de la red.

La parte de red de la dirección se utiliza principalmente para el encaminamiento de los datos a la subred específica en la que se encuentra el *host*..

### 2.5.4 Nombres de *host* y dominios

Las redes suelen tener nombres para que los usuarios no se vean forzados a utilizar sistemas de números largos y desagradables. Cada sistema se conoce por un nombre del *host* normal, muchos son conocidos por este nombre. Una red siempre tendrá un nombre de un *host* primario, y podrá tener nombres secundarios si:

- Dispone de múltiples interfaces. Si un sistema está conectado a dos o más redes, tiene un nombre para la interfaz de red específica que conecta con cada red.
- Utiliza nombres de alias. Estos nombres pueden utilizarse para indicar los servicios que proporcionan, por ejemplo, *sendmail*, el procesador de correo estándar del UNIX, utiliza el nombre "*mailhost*" como nombre por defecto para el concentrador

de encaminamiento del correo. Puede que quiera que la papelera del sistema sea el concentrador de correo. Al darle a este sistema el nombre secundario de *mailhost*, no es necesario modificar *sendmail* en cada sistema cliente

Una red también puede tener un nombre de dominio; si la red está conectada a *Internet*, *siempre* tendrá un nombre de dominio. Un nombre de dominio es un nombre para describir de forma exclusiva el sistema de redes en los vastos dominios de *Internet*. Su descripción puede ser Serpiente, Librunam, Cóndor. En *Internet*, los sistemas se identifican de forma parecida. Un sistema puede tener el nombre *tesiuam.dgbiblio.unam.mx*. Este nombre le indicará a todo el mundo que el nombre de anfitrión del sistema es *tesiuam*, que está en el dominio *dgbiblio* y que *dgbiblio* pertenece a una institución educativa. Los nombres de dominios son jerárquicos, por ejemplo, *unam.mx* es un dominio y *dgbiblio.unam.mx* es un subdominio de *unam.mx*. Los dominios pueden anidarse a una profundidad de varios niveles.

### 2.5.5 Nombres de *Ethernet* y su utilización

El sistema emisor necesita ponerse en contacto con el sistema receptor y obtener su dirección *Ethernet* para enviar los datos. El sistema receptor no estará esperando su número de IP, sino que espera su dirección *Ethernet*. Esto puede llevarse a cabo en el dispositivo de interfaz de red. Cuando llegan datos que concuerdan con la dirección *Ethernet*, se transmitirán al CPU para su descodificación y procesado.

### 2.5.6 Ruteadores y rutas

La *ruta* es el desplazamiento de los datos de una subred a otra, hasta que estén próximos a su destino. La mayoría de los ruteadores no conocen la localización de todos los sistemas de red. Cada uno de éstos sólo conoce las subredes específicas con las que puede hablar y las subredes accesibles de los ruteadores conectados directamente con él. Dispondrá de una tabla de números de red y de trayectos para llegar a cada red. Si el número de red específico no concuerda con la dirección a la que van destinados los datos, éstos se enviarán al ruteador por defecto.

Los ruteadores realizan un seguimiento de las rutas de los demás mediante la difusión de sus propias rutas de forma periódica. Cuando "oye" la difusión de otra ruta, actualiza sus tablas de ruteo con esta información. Cuando se conoce el número de saltos hasta una red específica, se incluye en esta difusión. Generalmente la ruta más rápida es la más corta o con menos saltos. Cuando hay mas de una ruta, la mayoría del ruteo se realiza con la menor cantidad de saltos. Existe un protocolo de ruteo, el OSPF

(*Open Shortest Path First*), que realiza un seguimiento del tiempo además del número de saltos. En los casos en que el tiempo sea menor en una ruta que tiene más saltos, se tomará la ruta con un tiempo menor.

El método que se acaba de describir se denomina *ruteo dinámico*. Al principio era estático, es decir, una persona codificaba de forma permanente las rutas a las redes. Cuando los sistemas se averiaban, las rutas estáticas fallaban. Un ruteo dinámico que pueda actualizar de forma periódica las rutas que están disponibles es el método más eficiente para *Internet* y para la mayoría de las redes con ruteo redundante. En la mayor parte de los sistemas DOS y Macintosh es necesario codificar de forma permanente una pasarela por defecto.

### 2.5.7 Máscara de red y difusión

Sólo una computadora debería entablar una comunicación en una red en cada instante. En *Ethernet*, si hay más de un sistema haciendo esta operación, se produce una colisión y cada sistema debe esperar un poco y volver a empezar. Cuanto más sistemas haya en la red, más tiempo tendrá que esperar un sistema su turno. La cantidad de tiempo que la red está ocupada transmitiendo datos es el nivel de tráfico del sistema. Para controlar la cantidad de tráfico que se produce, las redes pueden descomponerse en subredes. Al descomponer una red en partes más pequeñas, o distribución en subredes, el tráfico de una subred puede permanecer dentro de ella. Sólo los datos que tienen que ir a un sistema de otra subred saldrán de la subred actual.

La pasarela o sistema encaminador examinará cada paquete de datos y decidirá a qué subred debe ir, en función de la parte de ID de red de la dirección de IP. La máscara de red tiene una parte de la dirección que debe concordar si los datos deben ir a esa red. Una máscara tiene el mismo aspecto que una dirección de IP y define cuál es la parte de anfitrión de la dirección mediante la utilización de un procedimiento de enmascaramiento.

### 2.5.8 Funciones de seguridad de TCP/IP

La seguridad es un aspecto importante cuando se enlaza la red con el mundo exterior. Debe establecerse un equilibrio entre la seguridad de la red y dejar que los usuarios trabajen. El peor problema que se encuentra es que pocas aplicaciones utilizan algún método de encriptado. El *software* básico de TCP/IP no encripta las cosas por sí mismo; debe hacerlo la aplicación. Existe un protocolo llamado *kerberos* para transportar las transacciones encriptadas, pero no para codificarlas específicamente. Sin

el encriptado, si se realiza un Telnet o rsh con otro sistema y se conecta con él, la contraseña que se escribe se envía en texto puro. Cualquier persona con conocimientos y capacidad para ver los paquetes reales de la red puede obtener la contraseña.

## 2.6 TCP/IP y DOS

Existen varias versiones de TCP/IP para el sistema operativo DOS, pero el mayor problema es que DOS no permite utilizar todas las funciones de TCP/IP, debido principalmente a que es un sistema monousuario, carece de las capacidades de multitarea que necesitaría para establecer comunicación con la red y realizar tareas locales a la vez.

Los productos TSR (*Terminate-and-Stay-Resident*) pueden manejar parte de esta responsabilidad, pero utilizan una buena cantidad de recursos (por ejemplo, memoria) en un sistema que está limitado en éstos, y no puede sustituir a un sistema operativo multitarea real.

La falta de funcionalidad en forma completa de TCP/IP se resuelve con las últimas aplicaciones TCP/IP para MS Windows, que pueden realizar un trabajo razonablemente bueno en la gestión de múltiples tareas. Este *software* de TCP/IP para DOS y MS Windows ofrecen una amplia gama de funciones, que se indican a continuación:

- Capacidades de cliente de NFS. Se ofrecen entre 75% y 80 % de los paquetes de TCP/IP para DOS. De hecho, el motivo original de que el TCP/IP entrara en el mundo DOS fue por las capacidades de compartir archivos.
- Capacidades de servidor de NFS. Suelen ofrecerse sólo en paquetes de TCP/IP basados en MS Windows. Disponer de un sistema que actúe como servidor y ejecute MS Windows es un buen entorno para un grupo de trabajo pequeño, no obstante, la combinación de MS Windows y el servicio de archivos no es muy eficiente cuando se trata de atender a muchos usuarios o de manejar grandes cantidades de datos.
- Capacidades de cliente de *FTP* y *Telnet*. Todos los paquetes de TCP/IP lo ofrecen.
- Capacidades de servidor de *FTP*. Existen algunos paquetes basados en MS-Windows que ofrecen la posibilidad de utilizar *FTP*.

- *Software X Window*. Es la función más reciente de TCP/IP en los sistemas MS-Windows y DOS con QEMM.
- *Software WWW y Gopher*. Proporciona acceso a la mayor parte de los datos de *Internet* y ofrece una interfaz intuitiva y fácil de usar. Suele ser *shareware* que puede obtenerse en la propia red *Internet*.

### 2.6.1 Pilas de protocolos dobles

TCP/IP coexiste pacíficamente con otros protocolos de red. Existe *software* que ofrece capacidades de doble o múltiple pila de protocolos. Una pila de protocolos múltiple se utiliza principalmente en sistemas que se conectan a *Ethernet*. Estos datos se pasan entonces al protocolo primario. Si éste determina que los datos no se encuentran en un formato que pueda leer, los pasa al protocolo secundario.

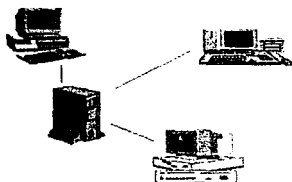
TCP/IP es capaz de compartir el cable con otros Sistemas operativos de red (NOS, *Network Operating Systems*). En muchos casos, los anfitriones pueden comunicarse por medio tanto de TCP/IP como de los otros protocolos utilizados. Algunas de las combinaciones pueden ser:

- *TCP/IP y SPX/IPX de Novell*.
- *TCP/IP y NetBIOS*.

## 2.7 TCP/IP en Macintosh

El Centro nacional para actividades de supercómputo (NCSA, *National Center for Supercomputing Activities*) fue el encargado de desarrollar el primer producto TCP/IP para Mac. También es la única versión *shareware* de TCP/IP disponible para él. Una de las mayores dificultades de los sistemas Macintosh es que el usuario tiene que conmutar manualmente entre las redes *Ethernet* y *AppleTalk* cuando utiliza ambas. Una solución a este problema consiste en utilizar un puente entre *Ethernet* y *AppleTalk*. El software TCP/IP para Macintosh ofrece funciones similares a los productos para DOS. Entre ellas se incluyen las siguientes:

- Capacidades de cliente de NFS.
- Capacidades de cliente de *FTP* y *Telnet*.
- *Software X Window*.
- *Software WWW y Gopher*.



# Capítulo 3

## Capítulo 3

## EL SERVIDOR

Como se mencionó en el capítulo uno, el servidor debe reunir ciertas características que le permitan cumplir las demandas de sus clientes. Existen muchos tipos de servidores, pero básicamente todos ellos reúnen ciertas propiedades tanto en el nivel de *software* como en el de *hardware* que permiten al cliente satisfacer las peticiones.

Las características mínimas o básicas a tomar en cuenta para el servidor HTTP que atenderá a varios usuarios en forma simultánea, establecerá conexiones en el nivel LAN o WAN y almacenará volúmenes de información considerables son:

- Empleo de un sistema operativo con capacidad de multiprocesamiento y multiusuario, como es el caso de UNIX. Aunque actualmente existen sistemas operativos de red soportados en sistemas operativos monousuarios -como es el caso de Windows NT o Novell, que resultan una buena opción-, no tienen la capacidad de de un sistema multiusuario como UNIX.
- Compilador en lenguaje C.
- Soporte del protocolo TCP/IP.
- Relación proporcional entre memoria RAM y clientes, es decir, el número de clientes simultáneos que podrá atender el servidor estará en función de la cantidad de memoria RAM. Por ejemplo, si un cliente consume alrededor de 1MB. y se disponen de 32 MB. en RAM, sin contar la cantidad de memoria que consume el sistema operativo, entonces se podrán atender alrededor de 32 usuarios en forma simultánea.
- Capacidad en disco duro según el tipo de información que se quiera ofrecer a los clientes.
- Tarjeta de comunicación.

Las tareas del servidor HTTP o servidor *Web*, también conocido como servidor HTTPD (la "D" proviene de la palabra en inglés *daemon*, que es un programa que se

aloja en memoria esperando peticiones; cuando alguna de éstas es solicitada, se levanta y procesa la petición para después regresar y esperar a la siguiente petición) serán el almacenamiento y manipulación de la información que es solicitada por el cliente mediante el empleo del protocolo denominado HTTP que significa *HyperText Transfer Protocol*, el cual además de hacer posible el envío y recepción de información en formatos de multimedia- tales como hipertextos, imágenes, video, sonido y realidad virtual, además de permitir el acceso a otras aplicaciones que emplean otros protocolos de Internet como el *telnet*, *gopher*, *e-mail*, *Wide-Area Information Servers* o *WAIS*, *ftp* etcétera.

El empleo del protocolo HTTP implica el uso de los denominados *stateless protocols*, los cuales tienen características muy diferentes de los otros protocolos de comunicación tradicionales que emplea el modelo Cliente-Servidor, conocidos como *state protocols*.

Los *state protocols* tienen las siguientes características:

- El cliente permanece conectado hasta el fin de la sesión.
- El servidor mantiene la información acerca del estado de la sesión.
- El servidor puede finalizar la sesión para evitar clientes "colgados", es decir, sesiones abiertas.

Como ejemplo de este tipo de protocolo tenemos al *ftp*, *telnet* y *Z39.50*.

En cambio los *stateless protocols* se caracterizan por:

- Toda la información es mantenida por el cliente.
- El cliente se reconecta cada vez que hace un nuevo requerimiento.
- Diferentes servidores pueden "servir" a los clientes en cada petición.

Cuando un servidor envía alguna información al cliente o *browser* solicitante, además de remitirle ésta, también le transmite la clase o tipo de archivo, es decir, si se trata de una imagen, un archivo ASCII, etcétera.

Para llevar a cabo la instalación de un servidor HTTP o servidor *Web*, es necesario tener una metodología que permita realizar este proceso, para lo cual se deberá



considerar desde la selección del *software* y *hardware* hasta la puesta en marcha del servidor. A continuación se propone una metodología para tal acción.

- Selección del *hardware*.
- Selección del *software* HTTP.
- Instalación del *software* HTTP.
- Compilación del archivo httpd.
- Configuración del HTTP.
- Ejecución del archivo httpd.
- Verificación de la puesta en marcha del servidor *Web*.

### 3.1 Selección del *hardware*

Basados en las características que debe tener un servidor y considerando el equipo que estuvo disponible en su momento en la Dirección General de Bibliotecas de la UNAM, se utilizó un equipo SUN Sparc 10 con procesador RISC, 32 Megabytes en RAM, tarjeta de comunicación, 1.2 Gygabyte en disco duro, sistema operativo UNIX Solaris 2.3 y compilador de lenguaje C versión 2.0.

### 3.2 Selección del *software* HTTP

Actualmente existe una gran cantidad de *software* HTTP, cada uno con diferentes versiones, por lo que la elección de uno resulta difícil. Hoy en día es posible consultar "WebCompare" (<http://webcompare.iworld.com>), que es una tabla comparativa no solamente entre el *software* para HTTP, sino también para el *software* del cliente, también conocido como cliente *web* o *browser*; sin embargo, alrededor de 1995, este tipo de tablas comparativas no existían, por lo que era necesario considerar algunos parámetros o indicadores que permitieran una buena elección. La figura 3.1 muestra la pantalla correspondiente al WebCompare; allí aparece un conjunto de ligas opcionales (palabras o frases subrayadas); al hacer *click* en alguna de éstas se mostrará más información. Es importante mencionar que este tipo de servidores *Web* puede cambiar tanto su dirección como su contenido en forma periódica o incluso desaparecer, sin embargo, considerando la importancia de la información que almacena será difícil que tienda a desaparecer.

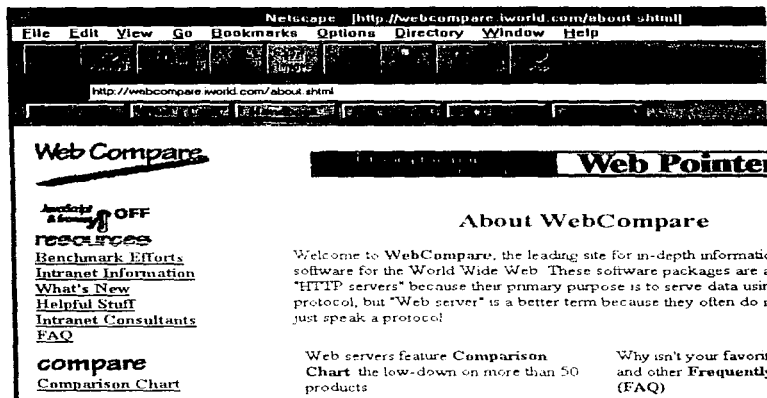


Figura 3.1 Pantalla del WebCompare

Entre los factores que se consideraron para nuestra selección se encontraban la tendencia al uso de los diferentes sistemas operativos y sus diferentes versiones, bajo los cuales era posible ejecutar este *software*, marcas de equipos de cómputo sobre las cuales era soportado, documentación descriptiva, tanto de la instalación como de sus características, costo y restricciones en la licencia, entre otros.

Para dar una idea al lector de la cantidad de *software* que en la actualidad se tiene para instalar un servidor HTTP, a continuación se muestran dos tablas: la primera de ellas contiene los diferentes tipos de *software* correspondiente al HTTP (tabla 3.1) y la segunda muestra los sistemas operativos que los soportan (tabla 3.2). Esta información se obtuvo en noviembre de 1996 a través del WebCompare.

Alibaba	Amiga Web Server	Apache	Apache/SSL: Stronghold
Basis Web	Boulevard	Cheetah	Common Lisp Hypermedia Server (CL-HTTP)
Commerce Builder	COSMOS Web Server	CyberPresence	DEChreads server
EasyServe	EMWAC HTTPS	EnterpriseWeb	Esplanade
Esplanade Secure Edition	ExpressO HTTP Server	FTP Web Server	Fjord Server
FolkWeb	GLACI-HTTPD	GLACI-Secure Serv	GN
GNNserver	GoServe for OS/2	httpd4Mac	Hypelt
IBM Internet Connection Server	IBM Internet Connection Secure Server	InterServer Publisher	INTRANet Jazz Server
Jigsaw	LANWeb	MacHTTP	Microsoft Internet Information Server
NCSA HTTPd	NetBasic for Internet	NetManage IntraNet server	NetPresenz
Netscape Enterprise Server	Netscape FastTrack Server	NetWare Web Server	N.E.T. Web Server
NovaServer	OmniHTTPd	Open Market Secure ebServer	Open Market WebServer
Oracle WebServer	Power MachTen	PowerWeb Server	Purveyor WebServer
Purveyor Encrypt WebServer	Quarterdeck WebServer	SAIC-HTTP	SCO Open Server
Sitebuilder	Sioux	SoftNet WebServer Enterprise	Sonic Web Server
Spinnaker Web Server	Spinner	Spry SafetyWeb Server	Spry Web Server
Squid	Stairways Web Server	SU/httpd	SuperWeb Server
TECWeb Server	thttpd	VBServer	W4-Server
WebCenter	Web Commander	Web for One	Web WebQuest
Web Server/400	Web Server 4D	Webshare	WebSite
WebSite Pro	WebSTAR 95/NT	WebSTAR Mac	Webware Commercial Edition
WN	YAWN	ZBServer	Zeus Server

Tabla 3.1 Tabla de servidores HTTP o servidores Web

Unix	Apache, ApacheSSL, CL-HTTP, COSMOS, ExpressO, FTP Web Server, GN, M Connection, IBM Connection Secure, Jazz, Jigsaw, NCSA, N.E.T, Web pe Enterprise, Netscape FastTrack, , Open Market WebServer, Oracle, SCO ioux, Spinner, SU/httpd, TECWeb, thhttpd, WN, Zeus
Windows NT	Alibaba, Cheeta, Commerce Builder, EMWAC, Esplanade, Esplanade Secure, ExpressO, Fnord, FolkWeb, GNNserver, IBM Connection, Jigsaw, Microsoft IIS, N.E.T. Web Server, NetManage IntraNetServer, Netscape Enterprise, Netscape FastTrack, Open Market Secure, Open Market WebServer, Oracle, Purveyor, Purveyor Encrypt WebServer, Quarterdeck WebServer, SAIC-HTTP, Spinnaker, SPRY SafetyWeb, SPRY Web, SuperWeb, VBServer, W4, Web Commander, WebSite, WebSite Pro, WebSTAR 95/NT, ZBServer
Windows 95	Alibaba, Cheeta, Commerce Builder, ExpressO, Fnord, FolkWeb, Jigsaw, OmniHTTPd, Purveyor, Quarterdeck WebServer, SAIC-HTTP, Spinnaker, VBServer, W4, Web Commander, WebQuest, WebSite, WebSTAR 95/NT, ZBServer
Windows 3.1	Quarterdeck WebServer, VBServer, ZBServer
OS/2	Windows (Version not specified) Basis Web, CyberPresence
Macintosh OS	AB Software HTTPd, Apache, GoServe, IBM Connection, IBM Connection Secure, Jigsaw, PowerWeb Server
Novell Netware	Boulevard, CL-HTTP, EasyServe, ExpressO, Httpd4Mac, InterServer Publisher, Jigsaw, MacHTTP, Power MachTen, NetPresenz, NovaServer, Sonic, WebCenter, Web for One, Web Server 4D, WebSTAR Mac, WebSTAR Mac/SSL
VMS	GLACI-HTTPD, GLACI-Secure Serv, LanWeb, NetBasic, NetWare Web, Purveyor, Sitebuilder, SoftNet WebServ Enterprise, Webware Comm, YAWN
Amiga OS	Cheeta, DECthreads, Purveyor
AS/400	AmigaWeb
VM/CMS	WebServer400
MVS	EnterpriseWeb, Webshare
MS-DOS	IBM Connection
Be OS, Starways	Httpelt

#### Variantes de Unix

AIX	IBM Connection, IBM Connection Secure
Digital Unix 3.2, HP/UX 9, HP/UX 10, Intel Solaris 2.4, Linux 1.2.x, SGI Irix 5.3, Sparc Solaris 2.4, SunOS 4.1.4	GNNserver

Tabla 3.2 Tabla relacional entre los sistemas operativos y el HTTP

Most	Apache, ApacheSSL, COSMOS, GN, NCSA, N.I.E.T. Web Server, SU/Httpd, Spinner, WN
Requires Java runtime	ExpressO, Jigsaw
SCO, AIX, HP-UX	Jazz
Solaris (for Intel and Sparc), SunOS, HP-UX, AIX, IRIX, Digital UNIX, BSD/OS, SCO UNIX, NEC UNIX, SINIX, Sony UNIX	Netscape Enterprise, Netscape FastTrack
Solaris/Sparc	TECWeb
Solaris/Sparc, SunOS 4.1, IBM AIX, BSD/386/1.1, HP-UX, HP MPE, SGI, Digital Unix, Tandem NSK	Open Market Secure, Open Market WebServer
Solaris/Sparc, SunOS, IRIX, Linux, BSDI, FreeBSD, OSF/1, HP-UX	Zeus
Sun SPARC Solaris 2.4	Oracle
SunOS, OSF, Solaris, SGI	CL-HTTP
SunOS, Solaris, FreeBSD, NetBSD, Linux, Irix, HP/UX, OSF/1, BSDI	thttpd
SunOS4, Solaris 2.4, Solaris 2.5, Linux, FreeBSD, Irix 5.3	Sioux

#### Windows NT CPUs

Alpha, PowerPC	Commerce Builder
Intel	Alibaba, Esplanade, Esplanade Secure, Fnord, FolkWeb, IBM Connection Secure, Quarterdeck WebServer, SPRY SafetyWeb, SPRYWeb, Spinnaker, SuperWeb, VBServer, W4, Web Commander, WebSTAR 95/NT, WebSite, WebSite Pro
Intel, Alpha	GNNserver, Oracle, Purveyor, SAIC-HTTP
Intel, Alpha, MIPS	Netscape Enterprise, Netscape FastTrack
Intel, Alpha, MIPS, PowerPC	EMWAC, Microsoft IIS
Requires Java runtime	ExpressO, Jigsaw

Tabla 3.2 Continuación

La investigación realizada para la selección del *software* del HTTP a finales de 1995, considerando los elementos antes mencionados, arrojó como resultado la preferencia por Apache en su versión más estable en esos momentos, la ver. 0.8.8. Por ejemplo, respecto de la popularidad, alrededor del 60% del mercado era repartido entre NCSA y Apache, según lo reportado por las estadísticas del Web Compare (documentos electrónicos con fecha del 4 de agosto de 1996). El estudio consistió en una búsqueda de servidores HTTP a través de Yahoo (enero de 1996), obteniéndose, una lista de 45 000 dominios únicos de éstos, de los cuales se seleccionó un rango de 2 000 lugares en forma aleatoria y, de éstos, 1 709 contestaron, obteniéndose los siguientes resultados (tablas 3.3 y 3.4).

<b>Servidor</b>	<b>Núm. de servidores</b>	<b>Porcentaje</b>
NCSA	732	41%
Apache	305	17%
Netscape	237	13%
CERN	198	11%
WebSTAR/MacHTTP	101	6%
WebSite	73	4%
BESTWWWWD	37	2%
OSU	14	1%
Purveyor	12	1%

Tabla 3.3 Tabla comparativa entre servidores *Web* considerando su popularidad

La tabla 3.4 refleja el incremento de Apache y Netscape respecto de los otros servidores en un periodo de un año.

<b>Servidor</b>	<b>Enero/96</b>	<b>Enero/95</b>
NCSA	41%	54%
Apache	17%	7%
Netscape	13%	8%
CERN	11%	17%
WebSTAR/MacHTTP	6%	5%
WebSite	4%	1%
BESTWWWWD	2%	<1%
OSU	1%	<1%
Purveyor	1%	<1%

Tabla 3.4 Tabla comparativa de porcentajes entre los servidores *Web*

En algunas direcciones como [http://www.apache.org/info/apache\\_users.html](http://www.apache.org/info/apache_users.html) se muestra una lista de instituciones en donde se corre Apache y en <http://www.netcraft.com/cgi-bin/Survey/> podrán consultar estadísticas respecto del *software* de los servidores *Web*, así como averiguar quiénes utilizan este *software*. La siguiente ilustración muestra un ejemplo de quién esta usando Apache, además del *Web* de la Dirección General de Bibliotecas de la UNAM.

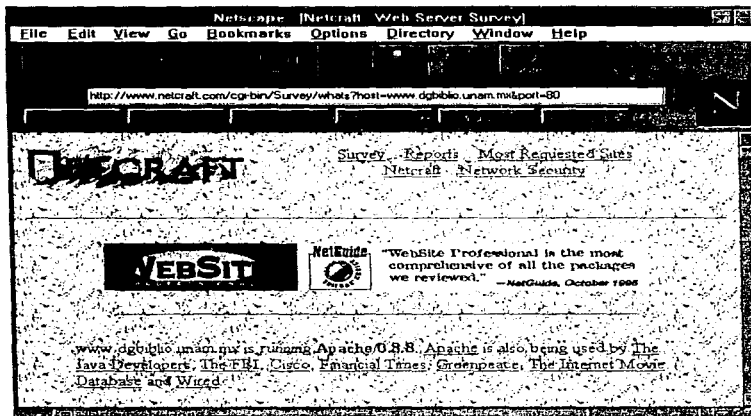


Figura 3.2 Pantalla de resultados de Netcraft

Al revisar las comparaciones documentales referentes a las ventajas de Apache respecto a la versión 1.3 de NCSA, se encontró que ofrecía :

- Formas de optimizar las respuestas erróneas, esto es, se tiene la capacidad configurar los mensajes de error y además redireccionarlos de la siguiente forma:

- a) Respuestas de salida de tipo texto.
  - b) Redireccionamiento al URL local.
  - c) Redireccionamiento al URL remoto.
- Un servidor puede soportar en la misma computadora múltiples direcciones IP (direcciones virtuales) que tienen un *HomePage* diferente para cada una de estas direcciones IP.
  - Manejo de variables adicionales para el CGI (*Common Gateway Interface*).
  - Empleo del concepto *Content Negotiation*, en el que los clientes pueden elegir (negociar) el tipo de información que prefieran, por ejemplo, trabajar con archivo de extensión jpeg en lugar de archivos con extensión gif, configurar el idioma deseado (alemán) en lugar del preestablecido (ingles), etcétera

Para aquella época, una ventaja adicional del *software* de Apache ver. 0.8.8 sobre el de NCSA ver. 1.3 es que podía emplear servidores *proxy*; esto significa que el servidor HTTP tiene la capacidad de manipular las peticiones que salen de una red interna y enviarlas a una computadora denominada servidor *proxy*, este servidor realizará el enlace de *Internet* con los servidores remotos, enviando y recibiendo respuestas respectivamente de todas las computadoras de la red interna. Los servidores *proxy* tienen el propósito de prevenir o minimizar los accesos no autorizados a la red interna aumentando con ello la seguridad de la red. La figura 3.3 ilustra la función del servidor *proxy*.

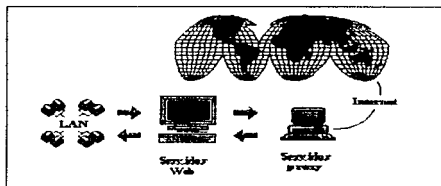


Figura 3.3 Esquema del servidor proxy



Finalmente, el costo fue otro de los factores que se consideraron para su empleo, ya que esta versión no cuesta, por lo que a este tipo de *software* se le conoce como *freeware*.

Apache es el resultado de la programación de un grupo de personas que se separaron de NCSA y que habían desarrollado la versión 1.3 de ésta a principios de 1995. Este grupo proporcionó los archivos "patch" para la versión 1.3 de NCSA y, después de una serie de combicciones con esta palabra, el resultado fue APAtCH.

Apache intenta ofrecer todas las características y opciones de configuración de HTTP 1.3, así como muchas de las particularidades encontradas en las versiones 1.4 y 1.5, todas ellas de NCSA. La filosofía de Apache es "¿Qué es lo requerido y cuándo es requerido?"

### 3.3 Instalación del *software* HTTP

El proceso de la instalación tiene como antecedente tres pasos que son la localización, la obtención y la descompresión del *software* HTTP.

Algunos servicios en *Internet* que permiten la localización del *software* del HTTP son:

*Archie*. Ayuda a encontrar los archivos almacenados en los servidores de *FTP*, que son servidores con información puestos a disposición de la comunidad; los sitios de *FTP* son regularmente indexados por títulos y/o palabras clave y *Archie* busca en estos índices.

*Servidores FTP*. Si ya se conoce alguna dirección IP de la cual se puedan extraer los archivos necesarios para trabajar, deberá conectarse a esta computadora con algún *software* de comunicación como Minuet, Lan Work Place, Super TCP, etc., que le permita emplear *FTP*, y deberá proporcionar como clave de usuario la palabra *anonymous* y como *password* su dirección de correo electrónico, es decir, su *e-mail*. El formato en que deberán recuperarse los archivos debe ser binario. Cuando recupere el o los archivos deseados se guardarán en la clave que se utiliza para correo electrónico.

*Gopher*. El *gopher* es otra herramienta que permite la localización de *Archie* y/o servidores *FTP*, ya que a través de un menú jerárquico muestra las opciones disponibles, dentro de las cuales es posible localizar alguna información. A través del *gopher* de la Dirección General de Servicios de Cómputo Académico (DGSCA) de la

Universidad Nacional Autónoma de México, es posible localizar servidores de *FTP* para la obtención de Apache. A continuación se muestra un ejemplo del procedimiento empleado para acceder a este *gopher*:

- Conexión al *gopher* de la DGSCA a través del comando de *telnet* con la dirección IP:

132.248.10.3                    ó  
condor.dgsc.unam.mx

- Teclar la clave: info

La siguiente secuencia de pantallas (figuras 3.4 a 3.12) se extrajeron de los menús de este *gopher*.

Servicio de INFO [GOPHER] (UNAM DGSCA)

Gopher en: condor.dgsc.unam.mx

1. Acerca de estos servicios de información/
2. Información sobre la UNAM/
3. Revistas y boletines electrónicos/
4. Noticias de México y el mundo/
5. Consulta a bases de datos de la UNAM/
6. Consulta a Catálogos en línea de bibliotecas/
7. Otros servicios de información en la UNAM/
8. Otros servicios de información en México y el mundo/
9. Presencia de otras instituciones en este Gopher/
- > 10. Herramientas de búsqueda en Internet/
11. Directorio de servicios por tema/
12. Centro de información de la RedUNAM/
13. Centro de operación de la RedUNAM/

Presione ? para Ayuda, q Salir

Página: 1/1

Figura 3.4 Pantalla principal del *gopher* de la DGSCA

## Servicio de INFO [GOPHER] (UNAM DGSCA)

## Herramientas de Búsqueda en Internet

1. Acerca de herramientas de búsqueda en Internet
2. Búsqueda en los títulos de Gopher a través de Jughead/
3. Búsqueda en los títulos de Gopher en Internet (Veronica)/
4. Búsqueda de archivos a través de Archie/
- > 5. Obtención de archivos a través de FTP/
6. Búsqueda de información basada en WAIS/

Presione ? para Ayuda, q Salir, u Menú anterior

Página: 1/1

Figura 3.5 Resultado de la opción "Herramientas de búsqueda en Internet"

## Servicio de INFO [GOPHER] (UNAM DGSCA)

## Obtención de Archivos a través de FTP

1. ¿Qué es FTP ?
2. Manual de referencia sobre el uso de FTP
3. Acceso a ftp anónimo de ftp.unam.mx/
4. Acceso a ftp anónimo de Supercómputo (UNAM)/
5. Acceso a FTP anónimo de la Dir. Gral. de Telecomunicaciones (UNAM)/
- > 6. Acceso a servidores FTP mediante interfaz Gopher/

Si Ud. desea anunciar un servidor de FTP Anónimo de su institución en esta sección, por favor envíe un correo electrónico a:

nicadm@servidor.unam.mx

Incluya en el campo "Subject" de su mensaje la leyenda:

Subject: FTP-ANONIMO <Direccion de su servidor>

Presione ? para Ayuda, q Salir, u Menú anterior

Página: 1/1

Figura 3.6 Resultado de la opción "Obtención de archivos a través de FTP"

Servicio de INFO [GOPHER] (UNAM DGSCA)

Acceso a servidores FTP mediante interfaz Gopher

1. Instrucciones

--> 2. Acceso a un host FTP específico <?>

Presione ? para Ayuda, q Salir, u Menú anterior

Página: 1/1

Figura 3.7 Resultado de la opción "Acceso a servidores FTP mediante interfaz Gopher"

Servicio de INFO [GOPHER] (UNAM DGSCA)

Acceso a servidores FTP mediante interfaz Gopher

1. Instrucciones

--> 2. Acceso a un host FTP específico <?>

```
+-----Acceso a un host FTP específico-----+
| Palabras a buscar                               |
| www.apache.org                                 |
| [Ayuda: ^_] [Cancel: ^G]                       |
+-----+-----+-----+-----+-----+-----+
```

Presione ? para Ayuda, q Salir, u Menú anterior

Página: 1/1

Figura 4.8 Resultado de la opción "Acceso a servidores FTP mediante el Gopher"

Servicio de INFO [GOPHER] (UNAM DGSCA)

Acceso a un host FTP específico: [www.apache.org](http://www.apache.org)

--> 1. [Link to ftp server www.apache.org/](http://www.apache.org/)

Presione ? para Ayuda, q Salir, u Menú anterior

Página: 1/1

Figura 3.9 Resultado de la petición

Servicio de INFO [GOPHER] (UNAM DGSCA)

[Link to ftp server www.apache.org](http://www.apache.org/)

1. 404.html
2. README-FTP
3. abducted/
4. ambient/
- > 5. [apache/](http://www.apache/)
6. bin/
7. bridgestone/
8. csp/
9. drugs/
10. epsilon/
11. etc/
12. feedback/
13. frontimages/
14. httpd/
15. incoming/
16. index-black.html
17. index.html
18. info/

Presione ? para Ayuda, q Salir, u Menú anterior

Página: 1/3

Figura 3.10 Resultado del acceso al Host especificado

## Servicio de INFO [GOPHER] (UNAM DGSCA)

apache

1. 404.html
2. announcelist.html
3. bug\_report.html
4. bugdb.cgi
5. bugs.cgi
6. cgi-bin/
7. comsupport.html
8. contributors/
- > 9. dist/
10. docs/
11. from-cvs
12. images/
13. in\_the\_news.html
14. in\_the\_news.html~
15. includes/

Presione ? para Ayuda, q Salir, u Menú anterior

Página: 1/2

Figura 3.11 Algunas opciones de los directorios del servidor *FTP* de Apache

## Servicio de INFO [GOPHER] (UNAM DGSCA)

dist

1. CHANGES
2. Old/
3. README.html
- 4. apache\_1.1.1.tar.Z <Bin>
5. apache\_1.1.1.tar.gz
6. binaries/
7. contrib/

Presione ? para Ayuda, q Salir, u Menú anterior

Página: 1/1

Figura 3.12 Resultado de la opción dist/

En algunos de estos directorios se podrán encontrar los archivos de las diferentes versiones de Apache por ejemplo, en el directorio Old/ se encuentran:

## Servicio de INFO [GOPHER] (UNAM DGSCA)

Old

- > 1. 0.6/  
2. 0.8/  
3. 1.0/  
4. 1.1/  
5. apache\_1.0.5.tar.Z <Bin>  
6. apache\_1.0.5.tar.gz  
7. apache\_1.1.0.tar.Z <Bin>  
8. apache\_1.1.0.tar.gz

Presione ? para Ayuda, q Salir, u Menú anterior

Página: 1/1

Figura 3.13 Lista de opciones del directorio old/

**Browser.** Si se dispone de alguna computadora que ya tenga instalado y configurado algún *browser*, también conocido como visualizador, Mosaic, Netscape, etc, entonces podrá emplearse alguna página de búsqueda o alguna opción *searcher* que le permita buscar a través de *Internet* lo relacionado con las palabras clave que se hayan escrito; en nuestro caso, las palabras clave a buscar son *servers http* o *servers Web*. El resultado de la búsqueda dará información relacionada con las palabras claves, además de una dirección IP o la dirección lógica de la computadora a la cual podrá entrar de una manera sencilla a partir de que haga *click* en la liga o palabra/frase subrayada o de color seleccionada.

La siguiente pantalla es un ejemplo del empleo de un *browser* (Netscape) proporcionándole al *searcher* la palabra clave *server Web*.

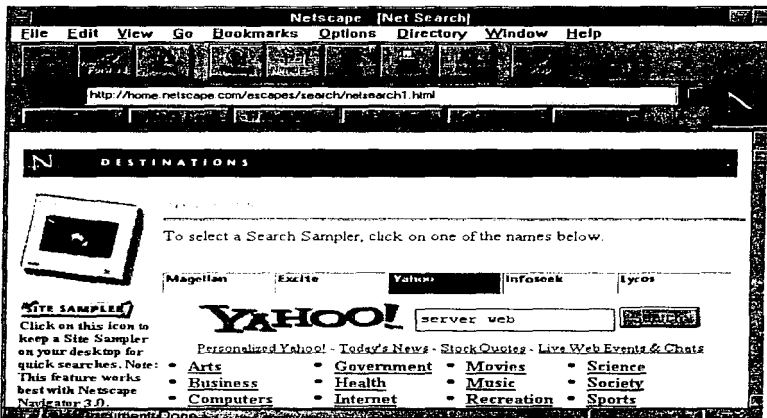


Figura 3.14 Pantalla de presentación del *searcher* Yahoo



El resultado de la petición realizada al *searcher* Yahoo se encuentra ilustrado en las siguientes 3 pantallas (figuras 3.15 a 3.17) debido a que la presentación de éste es amplia. De las opciones que están subrayadas es posible elegir alguna de éstas y acceder a la información.



Figura 3.15 Pantalla de resultados de la búsqueda *server Web* (Primera parte)

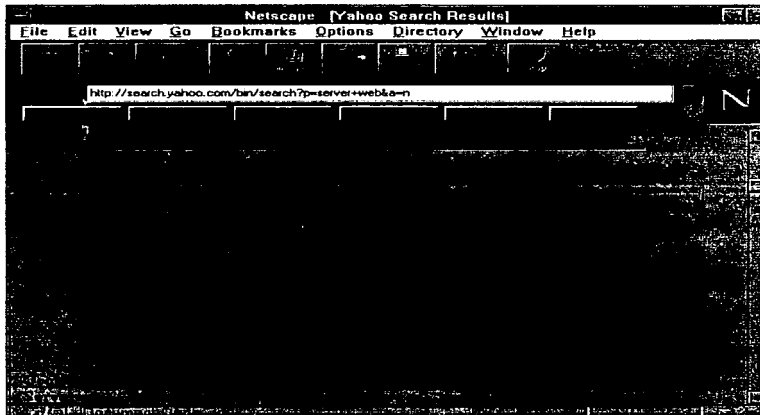


Figura 3.16 Segunda parte

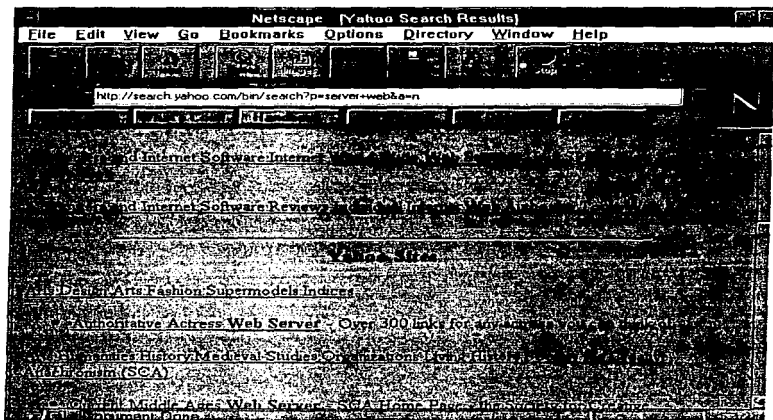


Figura 3.17 Tercera parte

Antes de iniciar la obtención o recuperación del *software* HTTP, será necesario observar las condiciones de la licencia, con el fin de no incurrir en algún delito de piratería o desconocer las limitaciones o fallas del *software*.

En lo sucesivo, se hablará del caso particular del servidor *Web Apache* ver 0.8.8, considerando el equipo de cómputo antes descrito, además del sistema operativo *Solaris* 2.3.

La información de algunas versiones de Apache puede ser localizada en el servidor de Apache (<http://www.apache.org/>). Este listará la versión común o cualquier versión beta probada más reciente y puesta en algunos servidores de *FTP anonymous*, como lo muestra la secuencia de pantallas (figura 3.18 a figura 3.20).

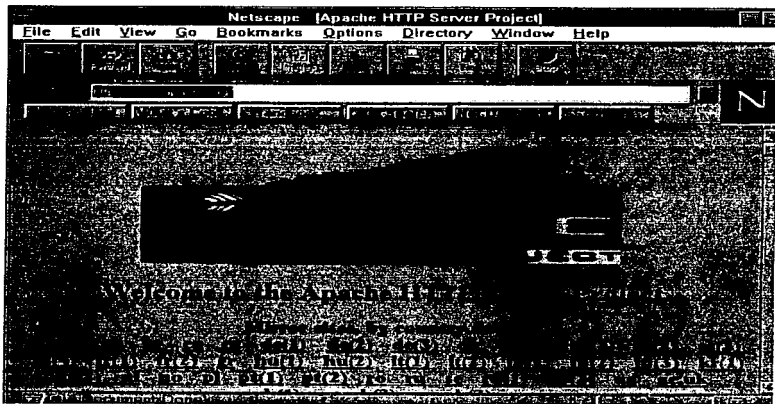


Figura 3.18 Pantalla de presentación de Apache (Primera parte)

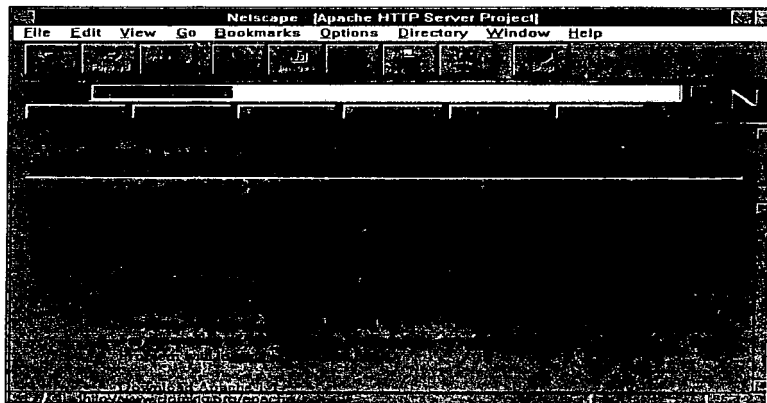


Figura 3.19 Segunda parte

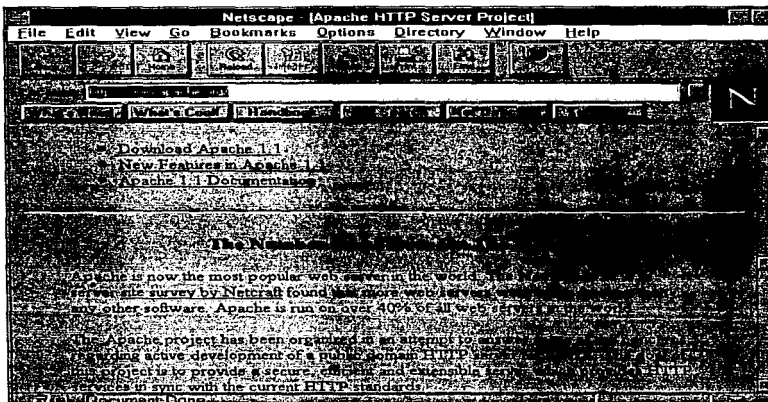


Figura 3 20 Tercera parte

Una vez que ya se ha localizado la dirección IP de la computadora y la ruta de donde se obtendrán los archivos, es factible encontrar una lista de las diferentes versiones, por ejemplo, en el caso de Apache (<http://www.apache.org/apache/dist>) se encontraron los siguientes archivos de instalación:

```
apache_0.6.5.tar.Z
apache_0.8.8.tar.Z
apache_0.8.14.tar.Z
```

Como se observa existe una extensión Z (mayúscula): esto indica que el archivo se encuentra comprimido por el comando de Unix llamado *compress*, por lo que se procede de descomprimirlo con el comando *uncompress* con la siguiente sintaxis para el Solaris 2.3:

```
uncompress -v apache_0.8.8.tar.Z
```

Una vez que se ha aplicado este comando, inmediatamente dejará el mismo nombre del archivo pero sin la extensión `.Z` y se podrá observar que el tamaño del archivo es diferente.

El siguiente paso es recuperar la información que se encuentra agrupada y comprimida en el archivo con extensión `.tar`, para ello se deberá emplear la siguiente sintaxis para el Solaris 2.3:

```
tar xvf apache_0.8.8.tar
```

A partir de la ruta donde se encuentre localizado el archivo con extensión `tar` se creará un directorio llamado `apache_0.8.8` y, dentro de éste, se descompactarán todos los directorios y archivos necesarios para la compilación y configuración.

Los directorios que podemos encontrar después de aplicar el comando `tar` con parámetros `xvf` son el directorio `conf/` (archivos de configuración), `logs/` (archivos de acceso y errores), `support` (programas de soporte), `src/` (código fuente del HTTP), `cgi-bin/` (*scripts* para el servidor) y el directorio `cgi-src/` (archivos fuente de los `cgi`). Las figuras 3.21 y 3.22 muestran la lista de directorios y archivos que corresponden a la versión 0.8.8 después de haber sido descompactada.

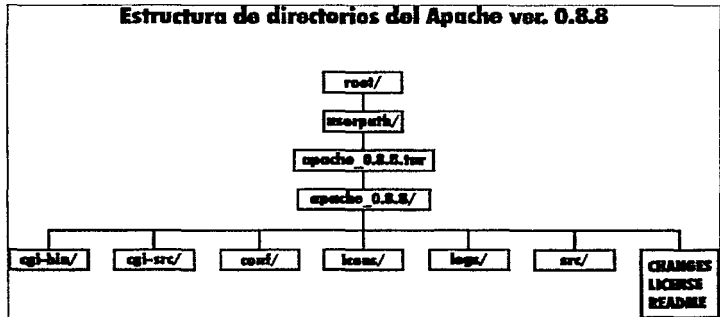


Figura 3.21 Lista de directorios de Apache ver. 0.8.8

cp-lin/	cp-arc/	conf/	html/	img/	src/	scripts/
artic	MkEfcE	awssarkfct	hckgf	WUmf	MkEfcE	
ukandr	drtggsrswdc	hpkckrfdt	hnyqf	CIUNCTS	drtggsrswdc	
kngr	mggrmpc	nntcLpM	hbrhgf	CIrgation	dmrmap	
knuz	jac	smuarkdt	hbrkqf	CIrgare	hymwlc	
qyBss-qs	pfE		hckngf	ISSI/LL	inc2hrnc	
tesqj	psp-qsyc		hrcgf	MkEfcE	inc2hrncdrt	
tespaj	qsyE		amrkqf	MkEfcEtrpl	kgrwfcuE	
ytne	ullc		amrkqf	RE/OME	unscqpc	
wesqf			cmrpswqjif	ICID		
			drqf	dkuc		
			de2qf	dkcl		
			kymqgf	ardf		
			gwnqgf	hfp amf jk hskh		
			gwnckqf	hfp arf qc		
			gwnckqf	hfp arf qch		
			mgqjif	hfp arnc		
			krwckqf	hfp arwh		
			krbcqf	hfp kge		
			mraqgf	hfp kgh		
			narkwqjif	hfp junc		
			sarkgf	hfp jmrh		
			sarkckqf	hfp jmwlc		
			wqjif	hfp jmkwh		
			wesqf	hfp jwqtc		
			twesqf	hfp jwqtlh		
			vkrwqjif	hqrh		
			wqjif	mtj aawsc		
				mtj a bakscmpc		
				mtj absc		
				mtj ascc		
				mtj aubc		
				mtj awh cbruc		
				mtj que		
				mtj qwscsc		
				mtj drc		
				mtj dck		
				mtj jnrc		
				mtj jruubc		
				mtj jrgqrkc		
				mtj jg strmrc		
				mtj jrg rkr c		
				mtj jmrcc		
				mtj jqrjckanc		
				mtj qmckc		
				mtj ksc		
				msB/c		
				srwhrdc		
				ullc		
				ull srqtc		
				ull srqth		

Figura 3.22 Lista de archivos correspondiente a cada directorio de Apache ver. 0.8.8



### 3.4 Compilación del archivo `httpd`

Una vez que se han obtenido los archivos del HTTP, es necesario compilarlo debido a que no hay un programa ejecutable que permita poner en funcionamiento al *Web*.

Para comenzar con la compilación se debe estar ubicado en el directorio `src/`, el cual se encuentra en el mismo nivel de `conf/`. Dentro del directorio `src/` se encuentran todos los archivos que permiten crear el archivo ejecutable `httpd` al compilarlo, ya que, en conjunto, estos archivos son librerías.

Esta versión de Apache soporta la alternativa de usar módulos opcionales; sin embargo, es necesario indicarle cuáles serán éstos para ser compilados, así como el orden. Para esto se requiere de la generación de un código corto (`modules.c`) con una simple lista de ellos.

Si se está de acuerdo con el conjunto de módulos estándar de Apache para continuar, sólo se necesitará editar el archivo `Makefile` y compilarlo; de lo contrario, si se desea seleccionar módulos opcionales, se necesitará correr el *script* de configuración. Para hacer esto, se ejecutan las siguientes acciones:

- Editar el archivo llamado *Configuration*, que contiene algunas configuraciones para diversos tipos de máquina del archivo `Makefile` y también una sección adicional en la parte inferior con las listas de los módulos que han sido compilados, así como los nombres de los archivos que los contienen. Si desea conocer el resto de los parámetros, existe una breve documentación en el mismo archivo.

El primer parámetro se refiere al tipo y nombre del compilador. Hay dos opciones para éste, `cc` y `gcc`, además de agregarse la ruta en donde se encuentra este compilador, si es que no está definida en el archivo `.login` de `root`.

Ejemplo:

```
CC -/usr/uecb/cc
```

en este caso estamos indicando que el compilador se llama `cc` (compilador de C para el sistema operativo Solaris 2.3) y se encuentra instalado en la ruta `/usr/uecb/`.

Eliminar el comentario en el segundo parámetro llamado `CFLAGS`, que son las banderas de compilación, en nuestro caso tendrá un valor de `-O2`, es decir,

CFLAGS--O2

Eliminar el comentario a las banderas auxiliares que están ligadas con el sistema operativo. Por ejemplo, si el sistema operativo es Solaris 2.x, entonces

AUX\_CFLAGS=-DSOLARIS2

Por último, eliminar los comentarios de aquellas líneas correspondientes a los módulos opcionales que se quieran incluir o agregar (las líneas se ubican en la parte inferior del archivo).

- Correr el *script Configure* y utilizar el archivo Configuration como el archivo de configuración. Para ello empleé la siguiente sintaxis:

```
sh Configure Configuration
```

Esto generará nuevas versiones del archivo Makefile y del *modules.c*. Si se quieren mantener múltiples configuraciones, se puede dar otro nombre como archivo de configuración, por ejemplo:

```
sh Configure -f Configuration.ai
```

- Por último ejecutar la instrucción *make*, esto creará el archivo binario llamado *httpd* de modo ejecutable.

### 3.5 Configuración del HTTP

Como se mencionó anteriormente, los archivos de configuración se encuentran ubicados en el directorio llamado *conf*, y ellos almacenarán los parámetros necesarios para que el servidor del *Web* funcione correctamente. Los nombres de estos archivos son:

- *httpd.conf-dist* (configuración del servidor)
- *srmd.conf-dist* (configuración de los recursos)

- `access.conf-dist` (control de accesos)
- `mime.types` (tipos de formatos reconocibles)

Es necesario realizar una copia de los tres primeros archivos antes de comenzar con la configuración, por dos razones: se requiere que éstos no tengan el sobrante del nombre "-dist" (`httpd.conf`, `srn.conf`, `access.conf`) para ser utilizados y, por seguridad. El único archivo que no se configurará será `mime.types`.

Para configurar los archivos, en algunos casos basta con eliminar el comentario de los parámetros (eliminar el signo de #) para que queden listos y, en otros, habrá que teclear la condición. Se debe editar en el orden en que se muestra arriba.

#### Archivo `httpd.conf`

Es el archivo principal de configuración y sus parámetros son:

**ServerType:** Sirve para indicar de qué tipo es el servidor, es decir, *standalone* o *inetd*; la primera opción significa que el servidor tendrá al menos un disco en el cual podrá almacenar sus propios archivos de sistema (*file system*), esto es, no dependerá de otra computadora para arrancar su sistema, mientras que en *inetd* sí se tiene esta dependencia. Para nuestro caso se eligió la opción *standalone*.

**Port:** Así como existe un número de puerto para *ftp*, *telnet*, *e-mail*, etc., existe el número del puerto a través del cual se podrá realizar la comunicación con el *Web*; la convención es que sea el puerto 80, que ofrece la ventaja de que ya no será necesario teclear este número de puerto en el momento de que se quiera entrar en un *Web*. Sin embargo, utilizar este puerto o cualquiera menor al 1023 requiere que el archivo `httpd` sea ejecutado con la clave de *root* o superusuario; para puertos mayores al 1023, se puede utilizar cualquier clave. Puede observar los puertos disponibles en el archivo llamado `etc/services`.

**User y Group:** Se refiere al nombre o número del usuario y grupo al que pertenece éste con el fin de asignarle los privilegios suficientes, una vez que previamente se ha corrido el `httpd`, ya que se compilará con la clave de *root* por utilizar un puerto menor al 1024. Esto se hace para obtener mayor seguridad. Para saber estos datos se puede consultar al administrador de la red o visualizarlo en el archivo `etc/passwd`.

**ServerAdmin:** Este parámetro se usa para designar la dirección de correo electrónico (*e-mail*) al cual las personas pueden enviar mensajes al administrador del *Web*, también conocido como *Webmaster*.

**ServerRoot:** Aquí se designa la ruta a partir de la cual se referencian los archivos de configuración (*conf*), error y acceso (*logs*). Algunos tienen definida la ruta */usr/local/httpd* o */usr/local/etc/httpd*.

Ejemplo:

```
/usr/mi_directorio/apache_0.8.8/htdocs
```

Obviamente, en esta ruta es donde se instaló el *software* del servidor del *Web*.

**BindAddress:** Tiene como objetivo "escuchar" cualquier dirección IP de un servidor o sólo una dirección en particular; esta opción puede ser empleada en la configuración de *hosts* virtuales. Este parámetro tiene tres opciones:

```
** * Escucha cualquier dirección IP (preestablecida)
Una dirección IP específica
El nombre completo de dominio de Internet válido
```

La tercera opción puede ser empleada, por ejemplo, en el caso de que se quiera construir un *Intranet*.

**ErrorLog:** Es el archivo bitácora (incluyendo la ruta) que registra los errores que detecta el HTTP; puede ser cualquier ruta o nombre de archivo; si la ruta no comienza con una diagonal (*/*), entonces se tomará la ruta que se definió en *ServerRoot*. El valor preestablecido sería:

```
logs/error_log
```

**TransferLog.** Es el archivo bitácora (incluyendo la ruta) que contiene todas las peticiones que se han hecho al servidor del *Web*; algunos de los datos que se guardan son la dirección IP que hace la petición, fecha y hora y archivo solicitado; si la ruta no comienza con una diagonal (*/*), entonces se tomará la ruta que se definió en *ServerRoot*. Por ejemplo:

logs/access\_log

*PidFile*: Este parámetro se usa en modo *standalone*. Es un archivo que maneja los procesos de identificación del *daemon* de Apache enviando una señal (SIGHUP o kill -1) e indican a los archivos log (*access\_log* y *error\_log*) que deben cerrarse y después reabrirse para registrar los movimientos. Tendrá la misma ruta que los archivos log, por ejemplo:

logs/httpd.pid

*ServerName*: Es el nombre del servidor del *Web*. Una computadora puede tener varios nombres y a través de este parámetro se puede hacer alusión al nombre del *Web*. Se recomienda emplear "www" en lugar del nombre real de la máquina. El nombre de la máquina no puede ser inventado, debe ser un nombre registrado en el DNS. Por ejemplo:

ServerName www.dgbiblio.unam.mx

*CacheNegotiatedDocs*: Si se elimina el comentario de esta opción, los documentos estarán ocultos, por lo que será necesario preguntar a los servidores *proxy*.

*Timeout*: Es el número de segundos que deberán transcurrir para cancelar una sesión y su valor. En Apache, su valor inicial es de 400 segundos.

*Server-pool size regulation (MinSpareServers, MaxSpareServers)*: Son parámetros que se dan con el fin de regular la cantidad de procesos que se encuentran en espera en los servidores. Los valores predeterminados son cinco y diez respectivamente. Este parámetro hace que Apache adapte su carga de peticiones automáticamente.

MinSpareServers 5  
MaxSpareServers 10

*StartServers*: Es el número de servidores para comenzar; su valor inicial es de cinco.

*MaxClients*: Número máximo de clientes que pueden conectarse simultáneamente al servidor; si se sobrepasa este número, el cliente será bloqueado y sacado de sesión. La cantidad es de 150.

*MaxRequestsPerChild*: Número máximo de preguntas o peticiones que a cada proceso sencillo (*child*) le es permitido antes de que éste concluya. Su valor predeterminado es 30.

Si se requiere instalar un *Host* virtual, entonces será necesario definir los siguientes parámetros que son muy similares a los descritos anteriormente y que de alguna forma se consideran de los más importantes. Para el caso de HTTP de la DGB no utilizamos este tipo de *hosts*.

```
<VirtualHost>
ServerAdmin
DocumentRoot
ServerName
ErrorLog
TransferLog
.</VirtualHost>
```

### Archivo *srm.conf*

Contiene los parámetros necesarios que permiten definir las siguientes asignaciones que afectan a las peticiones solicitadas, así como su formateo.

*DocumentRoot*. Es usado para definir el directorio raíz a partir del cual Apache deberá buscar los documentos. Por omisión, todos los requerimientos son tomados desde este directorio, pero las ligas y los alias pueden ser usados en otros lugares. Por ejemplo:

```
DocumentRoot /usr/mi_directorio/apache_0.8.8/htdocs
```

*UserDir*. Es el nombre del directorio que se encuentra anexo dentro del directorio de llegada del usuario; si un requerimiento de éste es recibido, es decir, si algún usuario solicita algo de algún servidor, el o los archivos resultantes de la petición serán almacenados en este directorio.

Ejemplo:

UserDir public\_html

*DirectoryIndex.* Este parámetro no está contemplado en la versión 1.3 de NCSA. Permite listar todos aquellos archivos que previamente han sido escritos en este parámetro, con el fin de que si un usuario escribió mal el nombre de un archivo en el URL, puede ver la posible opción más parecida al nombre que tecléo. Por predefinición se pone únicamente el *index.html*, pero se pueden agregar otros archivos mismo que deben estar separados por espacios.  
Ejemplo:

DirectoryIndex index.html /cgi-bin/conteo

*FancyIndexing:* Este parámetro es como un interruptor entre lo que se conoce como *fancy indexing* y *standard indexing*; es mejor usar el *fancyIndexing*, porque para activarlo bastará ponerlo en *on* y, para desactivarlo en *off*.

*AddIcon* y *AddIconByType:* Estos dos parámetros son empleados junto con *FancyIndexing*. Definen los iconos utilizados para representar los tipos de archivos, que son definidos vía los archivos de tipo MIME (*Multi-purpose Internet Mail Extensions file*). Puede haber archivos de tipo texto, imágenes, audio, video, etcétera.

*DefaultIcon:* Es el nombre del archivo que se muestra para aquellos archivos en los que no tienen un icono establecido explícitamente.

/usr/mi\_directorio/apache\_0.8.8/icons/logo.gif

*AddDescription:* Permite escribir una breve descripción antes del nombre del archivo; el empleo de este parámetro no es muy necesario. Su sintaxis es:

Format AddDescription "*descripción*" nombre del archivo

*ReadmeName* y *HeadName:* Estos parámetros no son muy relevantes; el primero se emplea para describir el nombre del archivo que muestra su contenido y, el segundo, contendrá un índice de directorios.

*IndexIgnore:* Se emplea para listar los archivos a ser ignorados. Por ejemplo:

IndexIgnore \*/?/\* \*# \*/HEADER\*/README\*

**Access File Name:** Contiene el nombre del archivo que espera en cada directorio para el control de acceso a la información. El nombre preestablecido es `.htaccess`, aunque puede ser cualquier nombre.

AccessFileName .htaccess

**DefaultType:** Este parámetro se emplea cuando un archivo no puede desplegarse por Apache y entonces toma el valor definido, tratando de interpretarlo de esa forma.

**Ejemplo:**

```
DefaultType text/plain
o
DefaultType image/gif
```

**AddEncoding:** Este parámetro permite al cliente descomprimir un archivo que tiene su propio MIME. De esta forma, el cliente puede descomprimir y manejar el documento en el formato deseado, en lugar de un tipo predeterminado. Se puede eliminar el comentario de este parámetro según sea necesario. No todos los *browsers* soportan el AddEncoding (Mosaic\_ /x 2.1+ sí lo soporta).

**AddLanguage:** Permite especificar el idioma de un documento, pudiendo entonces usar el contenido de negociaciones al dar al *browser* un archivo en un idioma que puede entender. Nótese que el sufijo no tiene que ser el mismo como el idioma clave. Por ejemplo:

```
AddLanguage en .en
AddLanguage pl .pl
AddLanguage pl .po
AddLanguage fr .fr
```

**LanguagePriority:** Permite dar prioridad a algunos idiomas en caso de un enlace durante el contenido de negociaciones. Sólo lista los idiomas en orden descendente de preferencia.

LanguagePriority en fr de



*Redirect:* Mapea de un URL antiguo a uno nuevo. Por ejemplo, si el cliente quiere redireccionar a `http://www.dgbiblio.unam.mx/servidgb`, habiéndose definido el parámetro, se preguntará por `servidgb/`

`Redirect /servidgb/ http://www.dgbiblio.unam.mx/servidgb`

*Alias:* Se emplea para renombrar URLs antiguos y para acortar los URLs largos. Ejemplo:

`Alias /icons/ usr/local/httpd/htdocs/htdocs/fred/long/path/to/some/icons/`

*ScriptAlias:* Es una opción similar al Alias, sólo que es aplicada a los *scripts*. Por ejemplo:

`ScriptAlias /cgi-bin/ /usr/mi_directorio/apache_0.8.8/cgi-bin/`

*ErrorDocument:* Esto es algo nuevo en Apache. En caso de que haya un problema o un error, Apache puede ser configurado para uno de cuatro casos:

1. Comportarse como un httpd 1.3 de NCSA.
2. Dar como salida un mensaje acostumbrado.
3. Redireccionar a un URL local para manejar el problema/error.
4. Redireccionar a un URL externo para manejar el problema/error.

Del uno al tres son configurados usando `ErrorDocument`, seguido por el código respuesta HTTP y un mensaje o URL. Ejemplos:

```
ErrorDocument 500 /cgi-bin/prueba
ErrorDocument 404 /cgi-bin/mal_url.pl
ErrorDocument 401 /http://www2.foo.bar/subscription_info.html
ErrorDocument 403 "No puedo permitir tu acceso hoy"
```

Las comillas en el último mensaje no son parte de éste.

### Archivo *access.conf*

Cada directorio al cual Apache tiene acceso debe ser configurado respecto de los servicios o características que le son permitidos. Los parámetros más importantes a ser cambiados son:

*ServerRoot*: Aquí se define la ruta en donde se encuentra el directorio *cgi-bin*, que podría ser utilizado para correr las aplicaciones de CGI. Son necesarios los signos "< y >"

Ejemplo:

```
</usr/mi_directorio/apache_0.8.8/cgi-bin>
```

*DocumentRoot*: Se define la ruta en la cual se encuentran los archivos que desplegarán al *HomePage* o pantalla de bienvenida al visitante. Ejemplo:

```
</usr/mi_directorio/apache_0.8.8/htdocs>
```

Los parámetros que corresponden a *Options*, *AllowOverride* y *Limit* se pueden dejar como estaban, es decir:

<i>Options</i>	FollowsymbLinks
<i>AllowOverride</i>	All
<i>Limit</i>	1

En el archivo *mime.types* se definen los formatos de archivos que el *Web* es capaz de interpretar; para nuestro caso no fue necesario renombrarlo, copiarlo y/o realizar alguna modificación.

## 3.6 Ejecución del archivo *httpd*

Cuando se haya realizado los procesos de compilación y configuración de parámetros, se ejecutará el programa que se encargará de activar al servidor del *Web*. Hacer una llamada desde el directorio *src/* al archivo *httpd* con el parámetro *-f* para proporcionar la ruta completa al *httpd.conf* y tomar estos valores.

Ejemplo:

```
/usr/mi_directorio/apache_0.8.8/src/httpd -f
/usr/mi_directorio/apache_0.8.8/conf/httpd.conf
```

El programa `httpd` es invocado cada vez que una conexión es hecha en el HTTP. Las siguientes opciones de sintaxis son reconocidas por el comando `httpd`:

`-d` (*directory* o *serverroot*)

Asigna el valor inicial de la variable de `ServerRoot` al `serverroot`. Esto puede ser pasado por el comando `ServerRoot` en el archivo de configuración.

El nombre predeterminado es:

```
/usr/local/etc/httpd.
```

`-f` (*file* o *config*)

Ejecuta los comandos en el archivo `config`. Si `config` no comienza con `/`, entonces es tomado con la ruta designada en `ServerRoot`.

El predeterminado es:

```
conf/httpd.conf
```

`-v`

Imprime la versión del `httpd` y después sale.

```
httpd -v
```

Para algunas otras versiones de Apache, existen estas opciones:

`-X`

Corre en modo monousuario, solamente para propósitos de depuración interna; el `daemon` no se separa de la terminal. No se debe usar este modo para proporcionar el servicio ordinario del *Web*.

`-?`

Imprime una lista de las opciones del `httpd` y después aparece.

```
httpd -?
```

Si se desconoce el estado de activación del `httpd`, entonces se puede utilizar el comando (Solaris 2.3):

```
ps -fea |grep httpd
```

Éste indicará si se ha ejecutado previamente el `httpd` en caso de que aparezca una lista con los números de los procesos.

### 3.7 Verificación de la puesta en marcha del servidor *Web*

Una forma muy sencilla de probar si el servidor *Web* está listo para ofrecer servicios es copiar en un directorio (que se haya creado en el nivel de los directorios principales como `src/`, `/conf/` `cgi-bin/`, etc. y que coincida con el nombre definido en el parámetro `ServerRoot` en el archivo `httpd.conf` ) un archivo con formato `.gif` y, después, en el *browser*, dar la ruta, puerto (si es diferente al 80) y nombre de este archivo con extensión `.gif`, debiéndose mostrar la imagen.

Por ejemplo, si en la ruta `usr/mi_directorio/apache_0.8.8/htdocs` está el archivo llamado `logo.gif` entonces, para visualizar esta imagen, se debe escribir:

```
http://www.personal.escuela.mx/logo.gif
```

Donde `http` es el protocolo a emplear y la palabra `www.personal.escuela.mx` es el nombre del servidor *Web* y desde luego `logo.gif` es el archivo que se va a visualizar, en este caso, por tratarse de una imagen.

.....



# Capítulo 4

## Capítulo 44



.....

## EL CLIENTE

Cliente es el elemento dentro del modelo Cliente/Servidor que realizará peticiones al servidor y desplegará los resultados que éste envíe. Es la computadora que cuenta con un *hardware* y un *software* para mostrar la información del WWW. Entre el *software* necesario se encuentra el *browser*, el programa encargado de realizar peticiones a alguno de los servidores de WWW para que éste le envíe un documento en lenguaje HTML; el *browser* también es el encargado de enlazar alguno de los recursos disponibles en el WWW, de interpretar ese documento y de desplegarlo. Debido a que un documento HTML puede contener texto, imágenes, sonido, vídeo (estos tres últimos en varios formatos) e incluso imágenes tridimensionales, animación o realidad virtual (VR), el *software* en el cliente debe ser capaz de interpretar toda esa información y si no es así, solamente mostrará la información que pueda manejar. Debido a lo anterior se requiere también de *hardware* para manejar elementos multimedia y de comunicación para poder enlazarse a *Internet*. Las restricciones que tengan los *browsers* en cuanto al tipo de información que pueden manejar depende de varios elementos:

- El Sistema Operativo bajo el cual están trabajando.
- El entorno en el que se ejecute el *software* (ambiente en modo gráfico o carácter).
- La computadora que se esté utilizando y los componentes con que cuenta ésta (que tenga los dispositivos periféricos necesarios para poder manejar, por ejemplo, sonido o vídeo).
- Restricciones inherentes al *software* cliente.

Antes de hablar acerca de características de los *browsers*, es importante tomar en cuenta el *software* y *hardware* adicional que requieren éstos; para poder conectarse a *Internet* y para ingresar al WWW, se considera el caso particular de computadoras personales compatibles (basadas en DOS, Windows y Windows 95).

### 4.1 Hardware para el cliente

Para que un cliente pueda conectarse a *Internet* a través del *browser*, se requiere contar con un dispositivo en la computadora que permita realizar un enlace a una red, la cual a

su vez esté conectada a *Internet*; independientemente del tipo de computadora (monousuario o multiusuario), se requiere de ese dispositivo, el cual puede ser:

- Un MÓDEM
- Una tarjeta de interfaz para red.

#### 4.1.1 MÓDEM

Un MÓDEM es un dispositivo que MODula y DEModula una señal, esto significa que para que el *browser* o alguna otra aplicación que requiera acceso a red, a través de una computadora pueda conectarse a la red, necesita enviar cierta información (petición), la cual se tiene que modular (convertir la señal o información, de manera que pueda ser transferida a grandes distancias sin distorsionarse, ya sea a través de cables o algún otro medio físico); si se consigue establecer comunicación con la red, ésta a su vez para poder interpretar la información, tiene que demodular esa señal (convertir la señal a un formato binario que puede ser manejado por una computadora), de la red se regresa información (respuesta a la petición) para establecer un enlace con la computadora solicitante.

El MÓDEM se utiliza en computadoras aisladas, esto quiere decir que no forman parte de una red; la conexión a alguna red se puede realizar a través de algún proveedor de servicio de conexión a *Internet* o en algunos casos a través de universidades.

Existen varios tipos de MÓDEM, los cuales difieren principalmente en la velocidad de transmisión, la cual es importante, ya que, entre mayor sea dicha velocidad, se obtendrán mejores resultados al establecer comunicación con una computadora remota.

#### 4.1.2 Tarjeta de interfaz de red

Una tarjeta de interfaz de red es un dispositivo que permite que una computadora forme parte de una red LAN. Esta tarjeta va a permitir la comunicación con toda la red, y si esta red, a su vez, está conectada a *Internet*, también se podrá establecer comunicación con *Internet*.

También hay varios tipos de tarjetas de red, sin embargo, éstas dependen de la topología de la red y de las características de la misma.

Independientemente de cual de los dos dispositivos se esté utilizando, es indispensable un *software* conocido como controlador, que va a permitir a la computadora utilizarlo correctamente y que no existan conflictos con otros dispositivos.

## 4.2 Software auxiliar de enlace para el cliente

Ya sea que en la computadora cliente se disponga de un MÓDEM o la computadora forme parte de una red LAN, es indispensable en ambos casos contar con un *software* adicional a los controladores que permita utilizar el protocolo de comunicación TCP/IP; el *software* encargado de ejecutar lo anterior es conocido como *socket* (enchufe). El *socket* conectará al *browser* a través de la red de manera directa si la computadora forma parte de una red LAN o, de manera indirecta, a través de la conexión por medio de un MÓDEM, para esto se requiere que la computadora local y la remota estén ejecutando uno de los protocolos seriales de comunicación SLIP (*Serial Line Interface Protocol*) o PPP (*Point to Point Protocol*), con los que se consigue el mayor provecho de TCP/IP, sin embargo, como ya se mencionó, es limitado en función de la velocidad a la que trabaje el MÓDEM o de las velocidades de transmisión en la red LAN según sea el caso.

Para el caso de las computadoras personales que ejecutan Windows, existen varios *socket* con las características mencionadas, en el caso de DOS, el propio *software* de aplicación es el encargado de realizar el enlace; los usuarios de Windows 95 no requieren de un *software* auxiliar, ya que este sistema operativo cuenta con *software* propio para realizar lo anterior, aunque también existen *sockets* especiales para esta plataforma.

### 4.2.1 Software de interfaz de red

En el caso de plataforma Windows, existe un Programa de Interfaz de Aplicación (API por sus siglas en inglés), conocido como *winsock* API, con el cual se tiene a un *software* interactuando con otro. Un API es una especificación pública que define un conjunto de funciones de *software* estándares que están disponibles para un determinado recurso. *Winsock* API, por ejemplo, define las funciones que estarán disponibles para la red TCP/IP. Esto permite que aplicaciones de alto nivel sean diseñadas (por diferentes compañías y tecnologías de *software*) para una sola y genérica interfaz de nivel-medio, en vez de varias interfaces de aplicación de bajo nivel y de diferentes desarrolladores o diferentes manejadores de *hardware*.

*Winsock* API define la capa que ocupa la parte más alta de el núcleo de TCP/IP y el final de cualquier aplicación TCP/IP, y es siempre ejecutada a través de un programa llamado WINSOCK.DLL, que es una Librería de Enlace Dinámico (DLL, por sus siglas en inglés) y consiste en un módulo que contiene un conjunto de rutinas que pueden ser accesadas por cualquier aplicación de Windows. Los DLL son cargados en memoria cuando son requeridos por cualquier programa y descargados y desocupan



cualquier recurso cuando no son utilizados. Los DLL pueden ser utilizados por varios y diferentes programas simultáneamente.

Para ambiente DOS, existen programas TSR (*Terminate and Stay Ready*), los cuales se cargan en memoria y se ejecutan sólo cuando son requeridos (un antecedente de los DLL), pero están presentes en todo instante, permiten realizar el enlace con TCP/IP y simulan la multitarea, sin embargo en algunos lapsos consumen memoria innecesariamente. En cambio Windows 95, dispone de controladores propios para diferentes protocolos, entre los que se encuentra TCP/IP, por lo que no requiere de programas o librerías auxiliares.

### 4.3 Socket para Windows

En el caso de plataforma windows existe un *socket* llamado Trumpet Winsock (no es el único, pero es el más utilizado para Windows, por lo que se explicará con detalle), que es el intermediario entre el *software* de aplicación (que no necesariamente es el *browser*, puede ser un *software* para correo electrónico o para transferencia de archivos *ftp*, por ejemplo) y la red, da soporte para el manejo de algunos de los elementos de la pila de protocolos TCP/IP. Es indispensable que se ejecute, ya sea antes o al momento de iniciar alguna aplicación que utilice la red.

Trumpet Winsock es *software shareware* que puede conseguirse en *Internet*. La figura 4.1 muestra el aspecto de Trumpet Winsock una vez que se ejecutó alguna aplicación que lo requería, mostrando información acerca de las direcciones en donde hace enlace para llegar al destino final.

Trumpet Winsock opera inicialmente como un DLL a través de un programa llamado TCPMAN.EXE, que controla y maneja la pila de protocolos. Trumpet incorpora prácticamente todas las rutinas TCP/IP en el mismo WINSOCK.DLL.

Para que este *software* se ejecute correctamente, es necesario establecer algunos parámetros como la dirección IP de la máquina, la red a la que se pertenece si es el caso, un servidor de nombres y otros parámetros si se utiliza un módem, todo lo anterior se debe establecer en la configuración del *software*; a continuación se explicarán estos parámetros, ya que es esencial para el correcto funcionamiento del *socket* y, por tanto, de las aplicaciones que lo utilicen; en la figura 4.2 se muestra la ventana de configuración.

En seguida se describen los elementos que aparecen en la pantalla de configuración para Trumpet Winsock

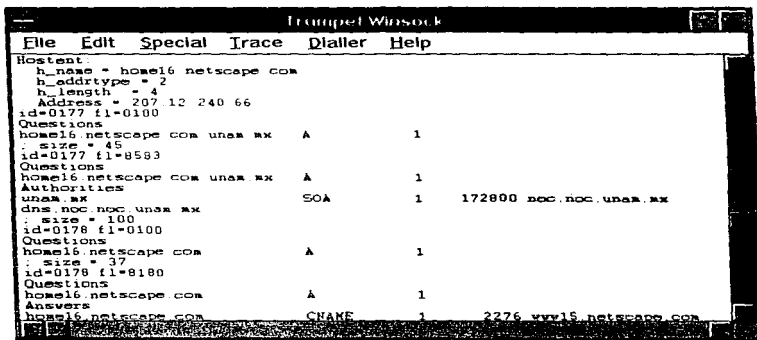


Figura 4.1 Ventana de inicio de Trumpet Winsock

**IP address**

Es la dirección IP de la máquina sobre la cual se está ejecutando la aplicación.

**Netmask**

Fija la máscara de red, que es usada para determinar si la dirección IP destino pertenece a la misma red.

**Domain Suffix**

Es el nombre lógico del dominio al cual se pertenece.

**Default Gateway**

La dirección de un *gateway* por medio del cual se tiene acceso a otra red (*Internet*) y que será el que se utilice por defecto.

**Time Server**

Esta opción es utilizada para tomar como referencia la hora de algún servidor.

### Packet Vector

Fija el número del vector que utiliza el controlador del dispositivo de comunicación (módem o tarjeta de red), el controlador utiliza ese número (también conocido como *packet driver*) con el sufijo "0x", en este caso, ese sufijo se omite: en el caso de dejarlo con el valor 00, winsock buscará si existe algún *packet driver*.

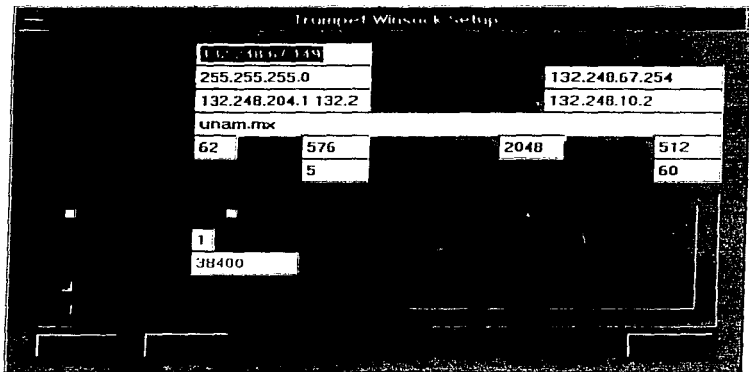


Figura 4.2 Ventana de configuración de Trumpet Winsock

### MTU

MTU (*Maximum Transmission Unit*), es la unidad máxima de transmisión de una red es la cantidad más grande de datos que puede ser transferida en una trama física en la red.

Puede ocurrir una fragmentación si un paquete de información es enviado a través de la red que tiene un MTU menor que la longitud de la trama. Esto repercutirá en una ejecución lenta ya que los fragmentos tienen que ensamblarse.

**TCP RWIN**

TCP RWIN (TCP *Receive Window*) determina la cantidad de datos que puede recibir la computadora. Usualmente de tres a cuatro veces el tamaño de TCP MSS.

**TCP MSS**

TCP MSS (TCP *Maximum Segment Size*) define el segmento más grande de datos TCP que Winsock está preparado para recibir.

**TCP RTO MAX**

TCP RTO MAX (TCP *Retransmission Time Out Maximun*), establece un límite superior para la retransmisión después de que ésta ha sido cancelada.

**Demand load timeout**

Los segundos que tepman (el nombre del archivo que ejecuta winsock) permanecerá cargado después de que todas las aplicaciones han terminado con él. Se aplica solamente cuando el controlador winsock.dll ha sido cargado antes que tepman, lo que sucede cuando otra aplicación iniciará a winsock.

**Internal SLIP y PPP**

Estas opciones se seleccionan cuando se utiliza un módem

Algunos de estos parámetros pueden obtenerse, consultando a los administradores de red, si es el caso (como en el caso del *default gateway y name server*). Otros de los parámetros, como en las opciones SLIP o PPP, dependen del módem que se utilice.

Aunque Trumpet Winsock no dispone de *software* de soporte para los servicios de TCP/IP, permite un aprovechamiento eficiente de memoria y además es un programa genérico utilizado por varias aplicaciones, que sí dan soporte a uno o varios de los servicios TCP/IP; algunos de ellos son *WS-ftp*, *WS-Finger*, *WS-Ping*, *H-Gopher* etc., que son programas para los servicios *ftp*, *finger*, *ping*, *gopher*, respectivamente.

Existe otro conjunto de programas que no requieren de Trumpet Winsock necesariamente, pues utilizan su propia interfaz, sin embargo requieren de la librería winsock.dll y en adición a ésta añaden otros módulos, algunos de ellos son:

**LAN WorkPlace** de NOVELL, *software* que da soporte para varios servicios TCP/IP, que se ejecuta bajo DOS o Windows; cada servicio utiliza un módulo (programa) independiente.

**Chamelon NFS** de NetManage, que provee soporte para la mayoría de los servicios, pero se ejecuta en Windows.

PC-NFS de Sunsoft, el cual se puede utilizar tanto en DOS como en windows, además da soporte para NFS y NetBIOS, para ello utiliza su propio API.

En la figura 4.3 se muestra un esquema general de funcionamiento de todos los elementos descritos hasta ahora: *hardware*, *software* y medios de enlace.

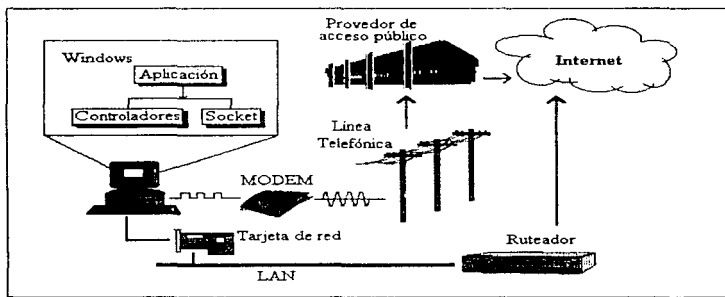


Figura 4.3 Esquema de enlace del cliente a Internet

#### 4.4 Características del *browser*

Una vez que se tienen instalados el *hardware* y el *software* necesarios, además de configurarlo correctamente según sea el caso, se puede realizar el enlace a Internet y al WWW con ayuda de un *browser*.

El *browser* es el *software* que hace posible el acceso al WWW a través de la interpretación del formato HTML. Sin embargo existen otros servicios de Internet a los que se puede tener acceso por medio de los *browsers*. Las características principales de éstos son:

- Acceso al WWW y despliegue de elementos multimedia.
- Acceso a los recursos de Internet a través de URL, URI, URN, URC.
- Capacidad de realizar búsquedas por medio de *searchers*.

## 4.5 Acceso al WWW y despliegue de elementos multimedia

La principal función de un *browser* es proporcionar acceso al WWW y mostrar los elementos multimedia que puede contener un archivo HTML, así como soportar las diferentes versiones de HTML. No todos los *browser* tienen la capacidad de manejar por sí mismos los elementos multimedia los cuales pueden ser:

- Archivos de sonido con los siguientes formatos:
  - WAV
  - AU
  - MIDI
- Archivos de video con los siguientes formatos:
  - MPEG
  - MOV
  - AVI
- Archivos de imágenes con los siguientes formatos:
  - GIF
  - JPG

En algunos casos no se despliega ese tipo de elementos o solamente alguno de ellos o utilizan programas externos que lo soporten, sin embargo, de ésta forma no se procesan junto con el documento HTML, sino como una aplicación aparte.

## 4.6 Acceso a los recursos de *Internet* a través de URL, URI, URN y URC



Figura 4.4 Esquema general de los mecanismos de localización de recursos en *Internet*

El término URL es más comúnmente usado en el WWW, pero no es el único; a continuación se describen los nombres y algunas de las características de los términos propuestos para localizar recursos en *Internet*.

La figura 4.4 describe un esquema muy general de cómo se encuentran estructurados los principales mecanismos para la localización de recursos en *Internet*, entre los que se encuentran términos como URN, URI, URL y URC; cada uno de éstos describe diferentes mecanismos para localizar recursos dentro de *Internet*; actualmente sólo el URL es el mecanismo que se encuentra activo en el *Internet*, ya que en los demás aun no se han definido por completo sus especificaciones.

A continuación se hace una breve descripción de los mecanismos de localización de recursos más comunes (ya definidos o en sus propuestas) en *Internet*.

**Uniform Resource Name** o URN está diseñado para localizar, independientemente del tipo de referencia, un objeto; así, un URN no determina la localización específica del recurso deseado, pero debe señalar un nombre genérico. El *software* procesa el nombre y entonces localizará el objeto más cercano a la localización más accesible; usando este nombre buscará el servicio. Las especificaciones del URN están incompletas y actualmente se encuentran en desarrollo por la IETF (*Internet Engineering Task Force*).

**Uniform Resource Locator** o URL son protocolos y términos de localización específica para referenciar recursos en *Internet*. Actualmente éste es el único mecanismo implantado para realizar este fin.

**Uniform Resource Identifier** o URI, generalmente representa cualquier esquema de nombres usado para referenciar recursos en *Internet*. Así, tanto los URL como los URN caen en la categoría de URI. Los nombres URI y URL generalmente son usados como sinónimos, pero esto sólo es exacto en ausencia de un esquema de nombres definido por URN.

**Uniform Resource Citation** o URC, son diseñados para hacer colecciones de valores/atributos conjuntados que describen un objeto en particular (usando una referencia a URI). Algunos valores en esta conjunción pueden también ser URI.

Un URC puede actuar de muchas maneras, por ejemplo, como un recurso de intersección-indexada para una colección de recursos largos o simplemente como una

colección de referencias para datos relacionados. Las especificaciones para URC no están completas, así que actualmente no se encuentra en uso.

#### 4.6.1 Construcción y sintaxis del URL

*Uniform Resource Locator* o URL, son un esquema para especificar recursos en *Internet* usando una línea de caracteres imprimibles ASCII. Este esquema comprende todos los mejores protocolos de *Internet* incluyendo *ftp*, *gopher*, HTTP y WAIS.

URL es una de las herramientas fundamentales en el WWW y son usadas dentro del los documentos HTML para referencia de etiquetas de una liga de hipertexto; sin embargo, los URL no están restringidos para el WWW, y pueden ser usados para comunicar información acerca de los recursos de *Internet* en correo electrónico (*e-mail*) y otros.

Un URL contiene la siguiente información:

- El protocolo que usará cuando accede al servidor (por ejemplo HTTP, *gopher*, WAIS)
- El nombre del dominio (*domain name*) en *Internet* del sitio sobre el cual el servidor está corriendo, junto con la información requerida, identificador *username* y *password*.
- El número de puerto del servidor. Si éste es omitido, el *browser* asume un valor de puerto por omisión, comúnmente establecido, indicado por el protocolo. (el valor por omisión del protocolo HTTP 80 y el valor del *gopher* es 70)
- La localización del recurso en la estructura jerárquica del servidor.

Cada URL debe ser escrito usando caracteres imprimibles ASCII y no usando completamente el conjunto de caracteres ISO Latin-1. Esta restricción asegura que el URL pueda ser enviado por correo electrónico; muchos programas de correo electrónico no manejan caracteres de la parte superior del conjunto de caracteres ISO Latin-1.

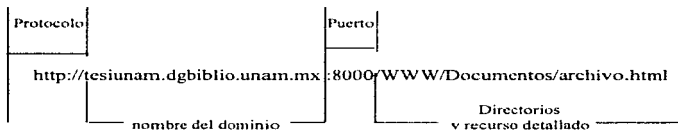
Por otro lado, un URL permite caracteres ASCII que tienen significado especial; los caracteres especiales más comunes son:



- El signo de porcentaje (%). Este es el carácter de escape para los caracteres codificados y es especial en todos los URL.
- El *hash* (#). Este separa el URL de un recurso del fragmento identificado por ese recurso; un fragmento identificado (*fragment identifier*) hace referencia a una localización particular dentro de un recurso. Este carácter es especial en todos los URL.
- El *slash* (/). Es usado para indicar la estructura jerárquica y los directorios.

#### Ejemplo de un URL:

La siguiente figura ilustra un típico URL (en el caso del protocolo HTTP) mostrando sus diferentes partes.



1. **Protocolo.** La primera cadena en el URL es el protocolo de *Internet* usado para entrar en el recurso. El protocolo es indicado con la abreviatura del nombre en minúsculas antes de los dos puntos en el URL.
2. **Nombre del dominio y número de puerto (dirección).** La segunda parte del URL es la dirección de *Internet* del servidor; esta información se encuentra entre un doble *slash* (//) y con una terminación de *slash* (/). El ejemplo tiene el nombre del dominio y el número de puerto y esa es la sintaxis que se debe utilizar en un caso así. Omitir el número de puerto implica tomar el valor por omisión que se tiene para ese protocolo. Las direcciones IP numéricas pueden ser usadas en lugar del nombre del dominio, por ejemplo: `//132.248.67.112/`
3. **Localización del recurso.** El siguiente *slash* después del campo del *host* y el número de puerto indica el comienzo de la ruta, información requerida para localizar el recurso en el servidor. Este campo varía considerablemente dependiendo del servicio que comience a ser requerido.

## 4.6.2 Especificaciones de los URL

Los protocolos comúnmente más referenciados a los URL son :

*ftp:*  
*gopher:*  
*http:*  
*mailto:*  
*news:*  
*telnet:* (rlogin, tn3270:)  
*wais:*  
*file:* (acceso a archivo local)

### El URL *ftp*

El URL *ftp* es designado para entrar en archivos y directorios usando el protocolo *ftp*, en la ausencia de información de cualquier identificador *username* y *password*, *ftp* se entra como *anonymous*, este se conecta al servidor designado como un usuario anónimo y usa la dirección de correo de *Internet* como *password*. El siguiente es un típico ejemplo :

```
ftp://Internet.adress.edu/ruta_acceso/archivo.txt
```

También se puede especificar un acceso a un directorio usando el URL *ftp*:

```
ftp://Internet.adress.edu/ruta_acceso/
```

En este caso, el servidor *ftp* provee la lista del contenido del directorio. Los *browsers* WWW despliegan esta información como un menú, permitiendo al usuario navegar dentro del sistema de archivos o seleccionar un archivo en particular para descargarlo.

El protocolo *ftp* permite varias formas de transferencia de archivos, el más importante es imagen o modo binario, el cual hace una copia byte por byte del archivo; éste es un modo que se utiliza cuando se transfieren programas, datos comprimidos o archivos de imágenes.

La figura 4.5 muestra el acceso a un servidor *FTP* a través de Netscape; en este caso se trata del servidor de NCSA en el subdirectorio Web.

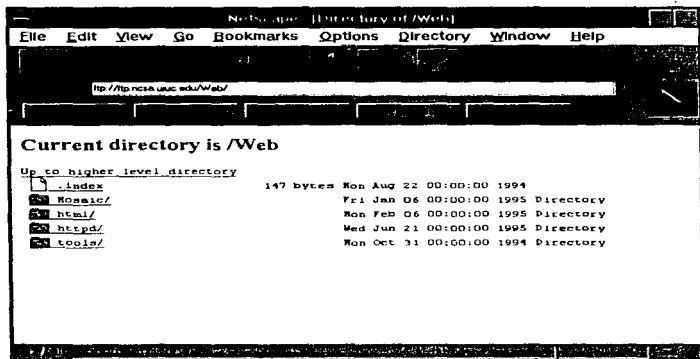


Figura 4.5 Acceso a un servidor FTP con Netscape 2.01

### El URL *gopher*

A los servidores *Gopher* se puede entrar vía un URL de manera similar a como se hace para los servidores FTP o HTTP, sin embargo, es algo diferente, porque los recursos del *gopher* son referenciados usando una combinación de códigos identificadores de recursos y una cadena de selección, por lo que un URL *gopher* es un poco diferente de un URL *ftp* o un HTTP.

Los identificadores de recursos son códigos de dígitos que identifican el tipo de recurso del *gopher*, por ejemplo, si es un archivo de texto, un directorio o una búsqueda indexada.

La cadena de selección es sólo un nombre asociado con el recurso y puede ser el nombre de un archivo o el redireccionamiento a un procedimiento de búsqueda en una base de datos o una sesión *telnet*. La forma básica del URL *gopher* es:

**`gopher://nombre_dominio/Tcadena_seleccion`**

donde *T* es tipo de código del *gopher* y *cadena\_seleccion* es la cadena de selección.

La tabla 4.1 muestra los códigos identificadores de recursos.

0	Archivo de texto
1	Directorio
2	* CSO nombre/teléfono
3	error
4	Archivo binhexed de Macintosh
5	Archivo binario de algún tipo DOS
6	Archivo UNIX no-encryptado
7	Búsqueda indexada
8	Sesión telnet
9	Archivo binario

Tabla 4.1 Códigos identificadores de recursos

El acceso al servidor *gopher* está disponible con el siguiente URL:

**`gopher://panoramx.dgbiblio.unam.mx/`**

También son posibles las relaciones jerárquicas:

**`gopher://panoramx.dgbiblio.unam.mx/1info`**

que indican que *info* es un directorio y regresará el contenido de *info* con este URL:

**`gopher://panoramx.dgbiblio.unam.mx/7info/indice`**

el acceso indica una búsqueda de *indice* dentro de *info*.

La figura 4.6 muestra el acceso al *gopher* de la UNAM a través de Netscape; la sintaxis del URL utilizado para esto, es: **`gopher://gopher.unam.mx/`**.

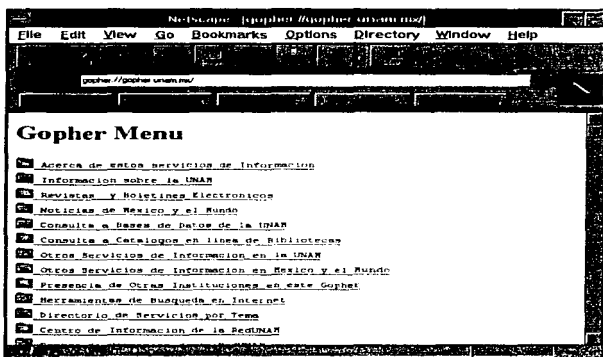


Figura 4.6 Acceso a un gopher con Netscape 2.01

## El URL HTTP

El URL `http` designa archivos directorios o accesos a programas en el servidor utilizando el protocolo HTTP, este URL permite apuntar a un archivo o a un directorio; un directorio es indicado al terminar el nombre del directorio con un *slash*. Los ejemplos más comunes de este URL, son los siguientes:

**`http://www.site.utoronto.ca/HTMLdocs/informacion.html`**

Para indicar un directorio:

**`http:// www.site.utoronto.ca/HTMLdocs/`**

En la figura 4.6 se puede ver un ejemplo del URL HTTP a través de Netscape, esta es la hoja principal de Microsoft el 15/1/1996 (`http://www.microsoft.com`).

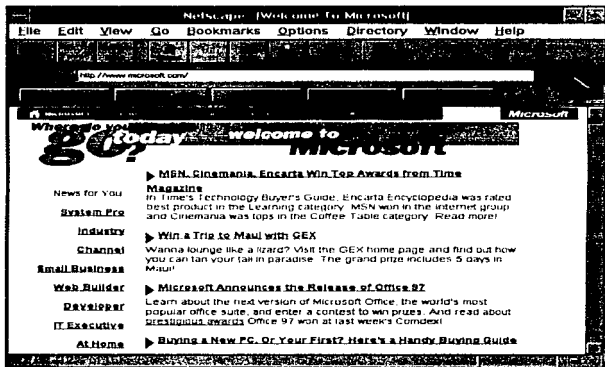


Figura 4.7. Acceso al Home Page de Microsoft con Netscape 2.01

## El URL *mailto*

El URL *mailto* envía correo a la dirección de *e-mail* especificada, el formato de *mailto* es:

`mailto: direccion_correo`

donde *direccion\_correo* es una dirección de correo de *Internet* a la cual el mensaje deberá ser enviado; típicamente esto se envía desde `nombre@host`, donde el nombre es el identificador de la persona (*username*) y el *host* es el nombre de la máquina de donde se envía el correo.

Algunos *browser* como MacWeb y algunas versiones de Mosaic, no soportan el URL *mailto*.

Un URL *mailto* no indica como debe ser enviado el correo, en general, el programa de

correo del cliente deberá contactar a un servidor de correo (*mailserver*), un programa que valida el correo y lo envía a el destino especificado. el URL *mailto* no especifica un *mailserver*, así que esta información debe ser configurada por separado dentro del *browser*.

### EL URL *news*

Este URL hace referencia a los *newsgroups* de USENET o individualmente a los nuevos artículos en USENET. Dentro de Internet, existen más de 4,000 grupos de discusión de tópicos llamados *newsgroups* y USENET, en ocasiones conocida como *netnews*, esta basado en mensajes enviados desde una maquina local, que involucra o habla de cualquier tema, este mensaje es distribuido a todas las demás maquinas sobre la red (*newsgroup*), entonces todos los usuarios de las demás maquinas podrán ver este mensaje y al mismo tiempo todos los mensajes de las demás personas.

Los *newsgroups* son especificados vía URL de la siguiente manera:

**news:news.group.name**

donde *news.group.name* es el nombre en particular de un *newsgroup*.

La manera especial es: **news:\***  
usada para referirse a todos los *newsgroup* disponibles.

### El URL *telnet*

El URL's *telnet* se usa para referenciar una sesión *telnet* a una maquina remota, un ejemplo es:

**telnet://flobor.rodent.edu**

Más generalmente, se puede usar el URL de la siguiente manera:

**telnet://username:password@flobor.rodent.edu/**

aquí se indican el *username* y el *password* que el usuario utilizara para conectarse, los caracteres *(:)* y *(@)* son utilizados para designar estos campos; dado que esta información no hace y completa la conexión, el *username* y el *password* no se debe de incluir en el URL por razones de seguridad.

Algunos *browsers* soportan una variedad de URL *telnet*, llamando a un URL `tn3270`, este indica una conexión que requiere un emulador de terminal IBM 3270 sobre la parte del cliente; en general, este solo es usado si la computadora cliente soporta un programa cliente que emule una terminal `tn3270` y que debe de ser configurado en el *browser*.

### El URL WAIS

Los servidores WAIS pueden ser accedidos de manera similar a un servidor HTTP; la mayor diferencia esta en la especificación del archivo; aquí, se pasan correctamente las instrucciones de búsqueda sobre el servidor WAIS, es decir la información del tópico que se va ha buscar.

La manera estándar de acceder a un servidor WAIS es :

`wais://wais.server.edu/database?search`

donde *wais.server.edu* es el nombre del dominio en *Internet* del *host* donde esta corriendo el servidor WAIS (incluyendo el número de puerto, si es requerido; el valor por omisión para este protocolo es 210); *database* es el nombre de la base de datos WAIS donde se realizara la búsqueda; *search* es una lista de instrucciones de búsqueda que se pasan a la base de datos.

### El URL *file*

Este URL es específico para un sistema local y no debe ser usado sobre documentos para hacer accesos públicos desde el *Internet*; un URL *file* representa accesos de archivos por computadoras de una red, particularmente de computadoras locales, pero no se especifica el protocolo usado para acceder estos archivos.

La forma general de este URL es:

`file://hostname/ruta/archivo`

donde *hostname* es el nombre del dominio y *ruta/archivo* es la localización del archivo, el URL *file* es comúnmente usado para representar accesos a archivos locales, el archivo puede estar sobre un disco local o comúnmente en un sistema de archivos en cualquier parte.



## 4.7 Herramientas de búsqueda en el WWW

Como ya se ha visto, los *browsers* es una parte esencial para poder ingresar en el WWW, las capacidades de que disponga nos permitirá visualizar determinada información, sin embargo, el buscar información en el WWW es independiente de el *browser* que se este utilizando, además debido a la inmensa cantidad de información y la gran cantidad de lugares que actualmente existen en *Internet*, resulta difícil navegar, sin alguna herramienta que nos permita seleccionar lugares, dependiendo de la información que estos contengan.

Existen varias compañías que se dedican a mantener actualizar bases de datos de direcciones de *webs*, organizándolos en base al tipo de información que proporcionen, estableciendo una clasificación y jerarquía en áreas y temas, los cuales inicialmente son muy generales (educación, ciencia, computación, entretenimiento, etc.), pero que se subdividen para limitar los resultados, este texto no tiene la finalidad de evaluarlos o ponderar entre los diferentes *searchers* (como se les conoce a las páginas de búsqueda de información) existentes, sino solamente mostrar los más importantes y hablar un poco acerca de ellos.

Los *webs* de los *searchers* realizan búsquedas de diferentes formas que son las siguientes:

- Búsqueda restringida. Navegando entre las diferentes áreas ya definidas y refinando la búsqueda en cada selección
- Búsqueda directa. Ingresando texto relacionado con lo que se pretende buscar sin una área específica.
- Combinando los dos métodos anteriores.

La mayoría de los *browsers* cuentan con un botón en el cual se puede acceder a un *web* en donde se encuentran varios *searchers*, seleccionando alguno de ellos, se accede al web del *searcher* elegido y se puede iniciar la búsqueda, como ya se mencionó, cada *searcher* dispone de una clasificación de áreas de interés que son muy generales, las cuales son ligas a otra página, que a su vez contiene otras clasificaciones, que son más específicas, dependiendo del área inicial, en donde se puede refinar la búsqueda, seleccionando alguna de las subareas, el resultado es otra página con temas relacionados a la jerarquía de áreas previamente seleccionadas o en otros casos existen varios subniveles de áreas, antes de llegar a un despliegue de temas o tópicos relacionados, siendo esta una de las alternativas de búsqueda.

Otra alternativa es iniciar la búsqueda, proporcionando palabras relacionadas con el tema o área, esta petición es enviada al servidor y se regresa, si se tuvo éxito, las correspondencias que se obtuvieron y en que dirección electrónica se ubican, de esta forma se obtienen resultados de diversos temas y áreas, que contengan alguna coincidencia con respecto al texto ingresado.

Una alternativa más es combinando las dos anteriores, se puede seleccionar un área de interés para limitar la búsqueda inicial y una vez estando en esa área, proporcionar texto relacionado con la búsqueda, de esta forma se tendrán resultados más específicos al tema buscado.

La rapidez y efectividad de los resultados, muchas veces es determinado por el tipo de información proporcionada, cada *searcher* tiene su propia metodología de búsqueda de información, sin embargo el que no se obtengan los mismos resultados para una misma entrada de información en los *searchers* no significa que alguno sea mejor, sino que simplemente contienen diferente información. La eficiencia en la obtención de los resultados del servidor también depende del tráfico en la red, ya que a determinadas horas existe una mayor afluencia a estos *webs*.

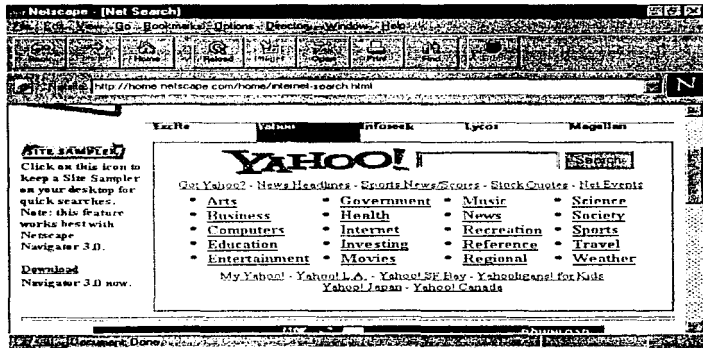


Figura 4.8 Pantalla de búsqueda en Netscape

La figura 4.8 muestra la página que proporciona netscape con el botón search, se muestran los *searchers* que se pueden escoger y la opción de iniciar una búsqueda. en seguida se muestran los *webs* de los *searchers* más importantes.

Un ejemplo de una búsqueda directa es proporcionar información de lo que se quiere buscar, por ejemplo: en el *searcher* yahoo si se ingresa el texto "UNAM" como se muestra en la figura 4.9, se obtendrá como resultado, los sitios en donde exista información acerca de la UNAM, como se muestra en la figura 4.10, sin importar al tipo de información que pueda contener (educación investigación etc.).

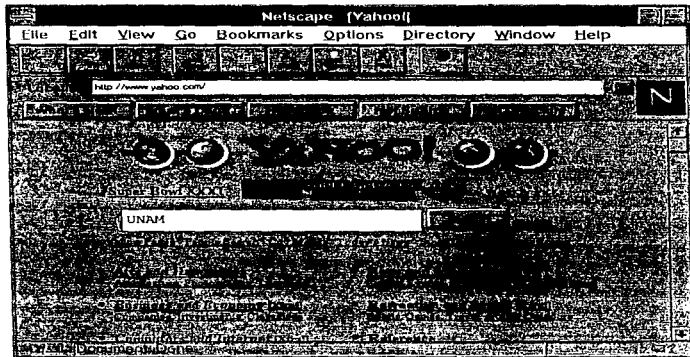


Figura 4.9 Pantalla inicial de búsqueda de yahoo

Cuando se realiza una búsqueda directa, resulta difícil en ocasiones, seleccionar alguna de las direcciones resultantes, ya que no están organizadas por categorías.

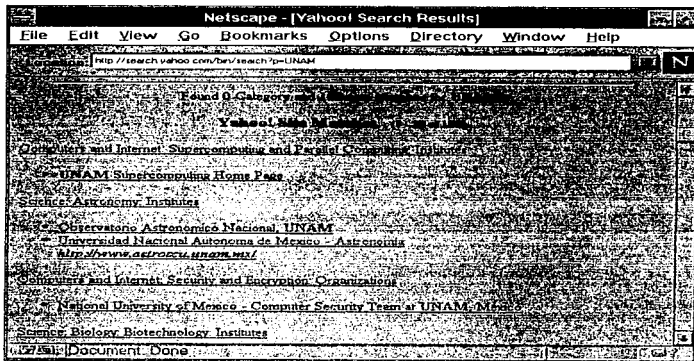


Figura 4.10 Resultados de una búsqueda en yahoo

Para realizar una búsqueda restringida, se selecciona primero alguna de las categorías a las que pudiera pertenecer el tipo de información que se está buscando, por ejemplo: si se busca información acerca de *Internet*, estando en la página de búsqueda de Excite figura 4.11, se puede seleccionar la liga *computing*, la cual se enlaza a otra página, en donde se encuentran tópicos relacionados con la computación, si se selecciona la liga *Internet*, se muestra otra página con *Internet*, dependiendo de lo que se quiera consultar, se selecciona la liga correspondiente, en este caso se selecciona *guides to the Net*, en donde se puede encontrar información de la historia de *Internet*, el WWW, recursos etc., la secuencia anterior se muestra en las figuras 4.11, 4.12, 4.13 y 4.14 respectivamente.

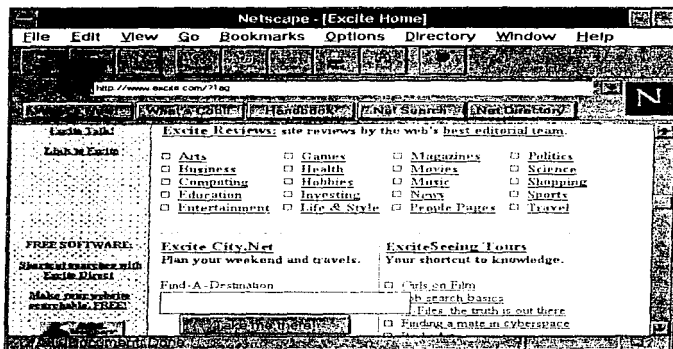


Figura 4.11 Página de búsqueda de Excite

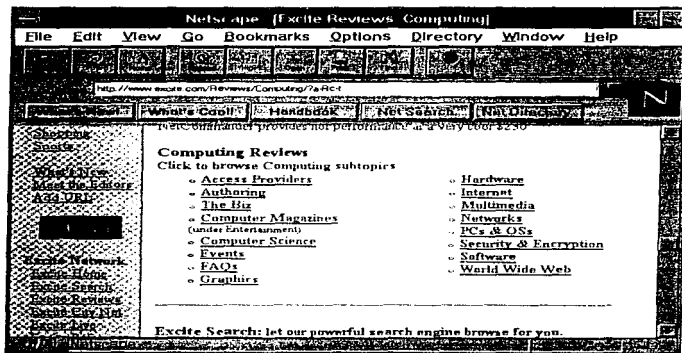


Figura 4.12 Enlace a la liga computing

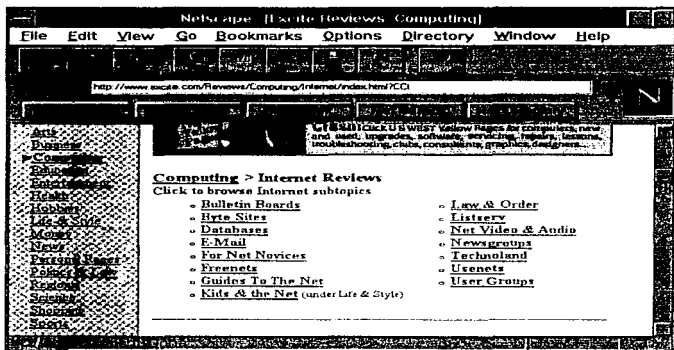


Figura 4.13 Enlace a la liga *Internet*

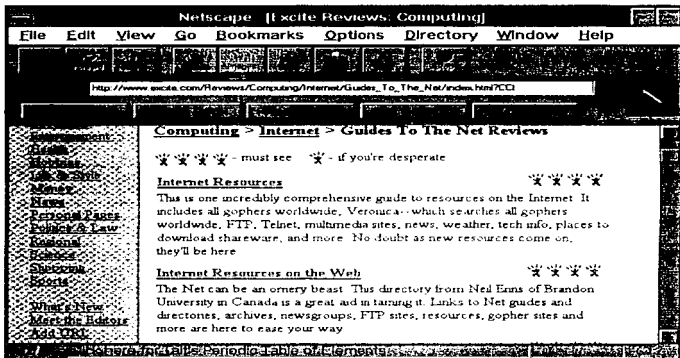


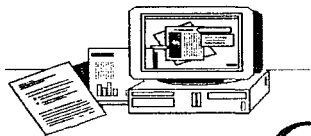
Figura 4.14 Resultado final de la búsqueda

Una búsqueda combinada se puede realizar, seleccionando una categoría o tema en la página inicial del searcher, una vez estando dentro de ese tópico, se puede realizar una búsqueda ingresando texto, con lo cual se obtienen resultados en base al texto y a la categoría seleccionada, cada uno de los *searchers* dispone de una sintaxis, para ingresar texto, o para utilizar operadores booleanos.

Las direcciones de los *searchers* más utilizados son las siguientes:

Yahoo	www.yahoo.com
Excite	www.excite.com
Infoseek	www.infoseek.com
Webcrawler	webcrawler.com
Altavista	www.altavista.com





# Capítulo 5

## Capítulo 5

## DISEÑO DE PÁGINAS EN *INTERNET*

*Internet*, era un medio lleno de comandos y sistemas operativos diversos que no resultaban nada atractivos. Existía la necesidad de desarrollar algún sistema que facilitara la tarea del intercambio de información, y la manera de realizarlo, fue utilizando un sistema multimedia ya que la información proporcionada o la información solicitada en muchas ocasiones contenía archivos de imágenes, texto, etc., y documentos con enlaces de hipertexto (término conocido para referir un texto que contiene conexión a otros documentos con información adicional sobre un tópico relacionado).

Pero por otro lado se requería de un lenguaje común, un método estándar para estructurar y escribir la información de tal forma que fuera legible para todos los *browsers* que circulaban por *Internet*. A ese lenguaje se le denominó: *Hypertext Markup Language* o HTML, que sirve para desarrollar documentos de hipertexto en el WWW.

### 5.1 Origen del HTML

HTML no se creó de la nada, es un derivado de un lenguaje de marcas conocido como SGML. SGML (*Standard Generalizer Markup Language*) es el estándar para describir el lenguaje de marcas y es el resultado de un trabajo de más de 10 años, cuando el uso de el SGML estaba limitado a aplicaciones militares y de gran escala (ISO 8879 ). La meta fue producir un juego de reglas para describir y manejar el contenido de cualquier documento digital.

SGML no es una aplicación en el sentido tradicional ; no puede comprarse y no es la propiedad de ningún fabricante o gobierno; es un estándar abierto, un juego de reglas sobre cómo organizar los documentos y una tecnología compartida. La diferencia clave entre SGML y otras aplicaciones de edición es que SGML está preocupado por la estructura y por el contenido de los datos, pero no por su apariencia o presentación.

Los documentos SGML tienen tres partes:

Primero la declaración. Se trata de un archivo de encabezado que contiene la información específica del sistema, necesaria para correr el documento en un sistema.

El segundo, es la definición de tipo de documento, o DTD (*“Document Type Definition”*; este es un lenguaje específico de etiquetas). La DTD es una estructura de árbol para el documento y el propósito principal de DTD es declarar la jerarquía de los elementos del documento, cada elemento a su vez puede ser modificado por atributos, los cuales le permiten que un elemento tome un valor específico.

En un documento producido en SGML está etiquetado con los nombres del elemento y los atributos. Las etiquetas generalmente usan delimitadores de corchete angular, como en `<nombre_de_etiqueta>`.

La parte final es la instancia del documento. La instancia del texto en sí y sus etiquetas acompañantes, conformando a las especificaciones y restricciones en la DTD. Es importante comprender que todo elemento y etiqueta de atributo están definidos específicamente en la DTD, existen diferentes nombres de etiqueta y relaciones estructurales para cada DTD. En SGML, no existe un sistema de etiquetación estándar para documentos.

Por otro lado, se puede decir que HTML (inventado en el CERN por Tim Berners-Lee) es un SGML DTD; es decir HTML es un subconjunto o derivado del SGML y en términos prácticos, HTML es una colección de estilos (indicados por etiquetas) que define la variedad de componentes de un documento del WWW.

Aunque SGML ofrece mucho más variedad para la creación de documentos complejos, también es cierto que se requiere de un mayor compromiso de tiempo y recursos, HTML es mucho más sencillo que SGML pero menos extenso.

Y mientras que para SGML se requiere adquirir un sistema de visualización, así como un técnico de SGML de tiempo completo; con HTML solo se necesita una cuenta en *Internet*, los servicios de alguien que administre las páginas de presentación e interés para aprender HTML.

## 5.2 Presentación del lenguaje HTML

Como resultado del aumento de servidores *Web*, la comunidad de usuarios de *Internet* decidió definir un estándar general para HTML. La versión original del HTML, recibió el nombre de Versión 1.0; La versión 2.0 es la mas utilizada, pero en la actualidad, se encuentra en uso la versión 3.0 de HTML, a la que, en ocasiones, se conoce como HTML+.

HTML es un método que permite representar el aspecto y el contenido de un documento, utilizando un archivo de texto ASCII. HTML consta de una serie de instrucciones especiales, o etiquetas, que servirán para marcar el texto incluido en el documento.

En este caso, se entiende como documento, la base de una página y cuando hablamos de elementos de un documento nos estamos refiriendo a títulos, cabeceras, párrafos, listas, hiperenlaces, imágenes, etc.

La gran ventaja del HTML es que se pueden generar documentos en este lenguaje con la ayuda de un simple editor de texto.

HTML es un lenguaje fuertemente estructurado o, mejor aun, requiere que el usuario desarrolle sus documentos utilizando una estructura específica, ya que el propósito es transmitir la estructura de los documentos entre todos los usuarios que deseen consultarlo.

Sin embargo, el aspecto de este documento puede ser distinto cuando llegue a la pantalla del usuario final, dependiendo del *browser* que se este utilizando, de la configuración que se haya dado al mismo y del hardware en el que este instalado el *browser* (resolución del monitor, que cuente con tarjeta de sonido o vídeo, etc.).

No importa que un hiperenlace se muestre en color azul o verde, ni que cada elemento de una lista se muestren precedidos por un asterisco o por un pequeño cuadrado, lo que realmente importa es que, tanto el usuario como el *browser*, interpreten cada elemento de los documentos WWW como tales.

Otra de las ventajas con las que cuenta HTML es que los códigos de control introducidos en los documentos ocupan poco espacio, por lo que los archivos HTML suelen ser de pequeño tamaño, sin embargo los elementos multimedia que pueda

contener la página pueden ser de mayor tamaño (si contienen complicados gráficos, sonido o vídeo). De esta forma, la velocidad de transferencia por la red dependerá del tamaño y cantidad de los elementos multimedia que contenga.

Por otra parte, HTML cuenta con el inconveniente de que es un lenguaje un poco limitado, ya que los códigos de control (o etiquetas) que se pueden utilizar no son muy numerosos y tampoco se pueden crear otros nuevos a riesgo de hacer incomprensible los documentos para la mayoría de los *browsers* existentes.

### 5.3 Estructura del HTML

HTML tiene una estructura que se define a partir de comandos básicos, estos comandos se pueden clasificar en los siguientes grupos:

- **Comandos estructurales.** Estos comandos identifican a un archivo como un documento HTML y proporcionan cierta información relacionada con los datos contenidos en él.
- **Comandos de formato de párrafos.** Estos comandos especifican operaciones con los párrafos (saltos de línea, introducción de rayas separadoras) así como diferentes tipos de títulos (seis niveles distintos).
- **Comandos de formato de caracteres.** Estos comandos le permitirán aplicar diferentes estilos (tales como negrita, cursiva o subrayado) a los caracteres contenidos en el documento.
- **Comandos de especificación de listas.** HTML puede trabajar con varios tipos de listas, incluyendo listas numeradas y tablas.
- **Comandos de hipervínculo.** Estos comandos le permitirán introducir enlaces que transportaran a los usuarios a otra parte del mismo documento, a otro documento incluido en el servidor o a otro servidor de cualquier parte del mundo.
- **Comandos de integración.** Estos comandos permiten acceder a información almacenada en formato multimedia. Estos comandos muestran imágenes, acceden a archivos de sonido y visualizan películas digitales.

Los documentos HTML son archivos de texto en formato ASCII estándar, que tendrán la extensión .HTM en MS-DOS y .HTML en UNIX. En Macintosh, como los archivos carecen de extensión, bastara con definirlos del tipo ASCII.

### 5.3.1 Cómo se utilizan las etiquetas HTML

Las etiquetas (es decir, instrucciones que le dirán al *browser* como debe presentar el texto en pantalla) se representaran por nombres encerrados entre los caracteres menor que(<) y mayor que (>), lo que separara los caracteres de control del resto del texto. Por ejemplo, <Nombre\_etiqueta> será la representación estándar de cualquier etiqueta. Existen dos tipos de etiquetas, el primer tipo esta formado por las etiquetas constituidas por un único elemento.

El segundo tipo son etiquetas que cuentan con un código de apertura y otro de cierre.

El texto que esta contenido entre ambas etiquetas será el que se vera afectado. En este tipo de etiquetas el comando de finalización será el mismo que el de inicio pero precedido por una barra inclinada, por ejemplo, </comando>.

Del mismo modo, muchas etiquetas aceptan diferentes parámetros, que analizaremos más adelante.

## 5.4 Etiquetas HTML de estructura

HTML cuenta con tres etiquetas se utilizan para describir la estructura global de todos los documentos. Este tipo de etiquetas no afectan al aspecto que presenta el documento cuando se visualiza en un *browser* simplemente, se incluyen para ayudar a los *browsers* y a las herramientas de búsqueda a reconocer con facilidad los documentos HTML. Las primeras etiquetas de estructuración que todo documento HTML debe contener son las dos siguientes:

<HTML> . . . </HTML>, que indicarán que el contenido de un documento esta escrito en formato HTML.

El siguiente paso es dividir el contenido del archivo HTML en dos áreas bien diferenciadas: cabecera y cuerpo. Para identificar la cabecera del documento se utilizan las etiquetas: <HEAD>...</HEAD>.

La cabecera incluirá únicamente información que no afectara al contenido de lo mostrado en el *browser* del usuario.

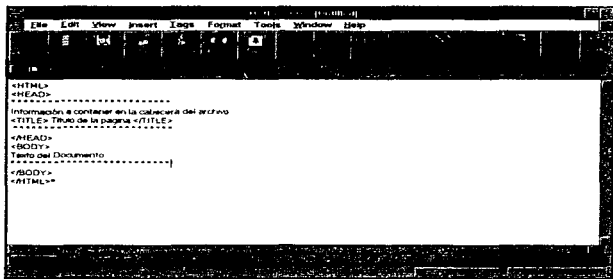
Por ejemplo, uno de los datos a incluir dentro de la cabecera será el título de la página. El título deberá estar incluido entre las etiquetas: <TITLE>...</TITLE>.

La posición en que se mostrara el título dentro de la página *Web* dependerá del *browser* que este utilizando. Así, por ejemplo, Netscape muestra el título en la barra de títulos de la ventana; Mosaic, muestra el título en la barra de herramientas justo debajo de la barra de menús.

Finalmente, el texto que desca mostrar en la pantalla de los *browsers* estará contenido en el "cuerpo" del archivo HTML.

Para especificar el cuerpo deberá utilizar las etiquetas: <BODY>...</BODY>.

Una vez introducidos las etiquetas estructurales, el documento HTML tendrá el aspecto que se muestra en la siguiente figura 5.1 .



```

<HTML>
<HEAD>
.....
Información a contener en la cabecera del archivo
<TITLE> Título de la página.</TITLE>
.....
</HEAD>
<BODY>
.....
Texto del Documento
.....
</BODY>
</HTML>
  
```

Figura 5.1 Etiquetas estructurales en un editor de HTML

## 5.5 Etiquetas HTML de formato de párrafos.

El elemento básico de texto de un documento es el párrafo, todo el texto introducido en un documento formará párrafos, si no se incluye ninguna etiqueta de control el texto del documento se mostrará en cualquier *browser* como texto corrido, sin ningún salto de línea (los *browsers* ignorarán los puntos y aparte y cualquier otro salto de línea manual que se introduzca pulsando la tecla enter).

Los saltos de línea que se visualizan en el *browser* utilizado por el usuario, dependerán del tamaño de la propia ventana del *browser*, del tamaño de letra utilizado y de las siguientes etiquetas HTML.

Si desea especificar un punto y aparte introduzca la etiqueta `<P>` detrás del punto: esta etiqueta es única y no cuenta con su respectiva etiqueta de cierre (`</P>`).

Si desea introducir un salto de línea, deberá incluir la etiqueta `<BR>`.

Si quiere introducir una línea horizontal de separación entre diferentes partes de un texto que cruce la pantalla del *browser* de parte a parte deberá utilizar la etiqueta: `<HR>`

Otra manera de dar formato a los párrafos contenidos en los archivos HTML es especificar diferentes niveles de títulos. HTML cuenta con seis tipos de títulos predefinidos y todos ellos se marcarán utilizando la siguiente instrucción :

`<Hn> Texto de título de nivel "n" </Hn>`

La letra "n" especifica el nivel del título que puede ir desde `<H1>...</H1>` hasta `<H6>...</H6>`. Las cabeceras se pueden mostrar en el *browser* con diferente tipo de letra, diferente tamaño, color, atributos, etc.; pero el aspecto final vendrá marcado por el *browser* utilizado y la configuración del mismo realizada por el usuario.

Si desea mostrar texto con un tipo de letra que se escriba con una anchura fija (monoespaciado), por ejemplo Courier, deberá introducir ese texto dentro de las etiquetas `<PRE>...</PRE>`. Este método es ideal para mostrar por pantalla cosas tales como código fuente de un programa, tablas o diagramas construidos utilizando exclusivamente caracteres.



## 5.6 Etiquetas HTML de formato de carácter

Con estas etiquetas de formato podrá resaltar los caracteres o palabras que usted elija. Las etiquetas de formato de caracteres más empleadas son: `<B>...</B>` para activar negrita, `<I>...</I>` para cursiva, `<U>...</U>` para subrayado.

El texto al que desea aplicar estos atributos deberá ir encerrado entre las etiquetas de formato anteriores. Por supuesto, podrá combinar estas etiquetas de formato de carácter para combinar distintos atributos.

Un ejemplo del uso de estas etiquetas se muestra a continuación:

```
<HTML>
<HEAD>
<TITLE>Etiquetas de formato de carácter</TITLE>
</HEAD>
<BODY>
<B>Este es un texto en negrita</B>
<I>Este es un texto en cursiva</I>
<U>Este es un texto subrayado</U>
<B><I>Este es un texto en negrita y cursiva</I></B>
</BODY>
</HTML>
```

Por otro lado, en algún momento tendrá la necesidad o el deseo de visualizar en sus documentos ciertos caracteres especiales (por ejemplo, el carácter "&" o extendidos (por ejemplo, vocales acentuadas en mayúsculas); por lo que deberá utilizar una serie de cadenas especiales de caracteres, ya que estos caracteres son tratados de una manera especial en HTML.

En el siguiente ejemplo se verá el manejo de estos caracteres especiales:

```
<HTML>
<HEAD>
<TITLE>Etiquetas de formato de carácter</TITLE>
</HEAD>
<BODY>
<H1>ESTA ES UNA P&Aacute;GINA </H1>
Puedes tener p&aacute;rrafos de texto, im&aacute;genes y
v&iacute;deos<P>
</BODY>
</HTML>
```

El ejemplo anterior muestra el manejo de un caracter especial para HTML (el acento agudo) y la instrucción que se debe utilizar para su uso.

En HTML los símbolos del menor que "<" y mayor que ">" son utilizados para indicar el inicio y fin de una etiqueta; para los caracteres especiales el símbolo "&" indica el inicio de una secuencia de escape que el *browser* interpreta y el punto y coma ";" indica el final de la secuencia de escape.

En la siguiente tabla 5.1 se muestran solo algunos de los caracteres especiales.

Comando	Caracter
&lt;	<
&gt;	>
&amp;	&
&quot;	"
&Aacute;	Á
&ntilde;	ñ
&Ccedil;	Ç
&AElig;	Æ
&ecirc;	ê
&Ouml;	Ö

Tabla 5.1

## 5.7 Etiquetas HTML de formato de listas

Con HTML podrá crear diferentes tipos de lista, entre las que se pueden incluir:

- Listas no numeradas o tabuladas.
- Listas numeradas.
- Listas de glosario (también denominadas listas de definición) en el que cada elemento cuanta con un termino y una definición.

### 5.7.1 Lista no numerada

Para hacer una lista no numerada:

1. Comenzar con la etiqueta de apertura de la lista <UL>.
2. Introducir la etiqueta <LI> y enseguida el elemento de la lista ( no es necesario poner al final la etiqueta de cierre del elemento de la lista </LI> ).
3. Finalizar con la etiqueta de cierre de la lista </UL>.

En el siguiente ejemplo se muestra el código de una lista no numerada con dos elementos.

```
<UL>
<LI> Manzanas
<LI> Uvas
</UL>
```

Cada elemento <LI> puede contener múltiples párrafos, solo se necesita separar cada párrafo con la etiqueta <P>.

### 5.7.2 Lista numerada

Una lista numerada (o también llamada lista ordenada) es idéntica a una lista no numerada, excepto por la etiqueta <OL> que se utiliza en lugar de <UL> para definir a esta clase de listas. Los elementos son etiquetados usando la misma etiqueta <LI>. El siguiente ejemplo muestra una lista numerada.

```
<OL>
<LI> Manzanas
<LI> Naranjas
<LI> Limones
</OL>
```

### 5.7.3 Lista de glosario (listas de definición)

Las listas de definición (también conocidas como listas discursivas o lista de glosario) son usadas para crear listas donde cada elemento o entidad de la lista también tiene un párrafo descriptivo, estas listas son usadas típicamente para glosarios y sus discños.

Una lista de definición comienza con la etiqueta DL, cada entidad de la lista es creado con la línea que define al término con la etiqueta DT y su definición indicada por la etiqueta DD; la definición puede contener más de una línea. El siguiente ejemplo muestra una lista de definición:

```
<DL>
<DT>UNAM
<DD>La Universidad Nacional Autónoma de México, ubicada dentro
de la Ciudad Universitaria en México, D.F., cuenta actualmente
con un acervo bibliográfico de aproximadamente 56,000 títulos.
<DT>DGB
<DD>La Dirección General de Bibliotecas, ubicada en el edificio
de la biblioteca central de la UNAM, es la encargada de
coordinar el buen funcionamiento de todas la bibliotecas de la
UNAM.
</DL>
```

La salida de los ejemplos de listas numeradas, no numeradas y de glosario se vería como se muestra en la figura 5.2 :

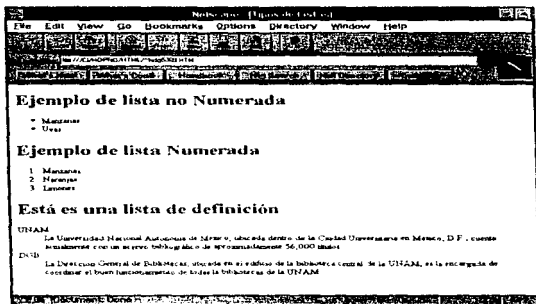


Figura 5.2 Ejemplo de listas visualizado en Netscape 2.01

Se pueden encadenar listas aunque sean de diferentes tipos.

## 5.8 Etiquetas HTML de hiperenlaces

Existen tres formas de hacer ligas o hiperenlaces con este tipo de etiquetas, las ligas dentro de una página, las ligas dentro del sistema y las liga entre sistemas. Una liga dentro de una página permite hacer un enlace a otra zona dentro de la misma página; una liga dentro del sistema permite el enlace de un documento con otro dentro del servidor actual y una liga entre sistemas permite el enlace de un documento con otro dentro del servidor, e incluso, con un documento contenido en otro servidor.

Los puntos de partida y de destino de un enlace o liga de hipertexto son conocidos como anclas y están marcados con la etiqueta `<A>`. Para crear una liga se necesita definir claramente dos datos:

- El nombre del archivo (o URL) con el que se desea enlazar.
- El texto que va a ser resaltado en su documento y, por lo tanto, va a servir como punto de unión.

La etiqueta `<A>` acepta varios parámetros, sin embargo, los mas utilizados son solo dos, el primero es el parámetro `HREF`. Con este parámetro podrá indicar el nombre tanto del servidor como del archivo destinos al que saltara el *browser* cuando el usuario pulse sobre la liga. Este parámetro permite el empleo de cualquier servicio *Internet* soportado por el *browser* por lo que la liga podrá apuntar a: *ftp*, *Gopher* y *USENET*.

A continuación se muestra un caso de hiperenlace o liga entre sistemas; la dirección URL a la que se dirige el enlace se halla encerrada entre comillas y se incluye como argumento del parámetro `HREF`; el servicio que se solicita en este ejemplo es referido a un archivo dentro de un servidor `HTTP`.

Como en otras etiquetas HTML, al finalizar el comando de ancla habrá que introducir la etiqueta de cierre, es decir, `</A>`. Todo el texto contenido entre `<A>` y `</A>` se convertirá en un área de la pantalla sobre la que el usuario podrá pulsar con el ratón para desplegar otra pagina, el usuario solo vera como texto del hiperenlace el rotulo contenido entre las etiquetas `<A>` y `</A>` (este texto es conocido como hipertexto; se suele mostrar en otro color y subrayado).

La dirección a la que apunta el hiperenlace solo se mostrará en la barra de estado del *browser* cuando el usuario sitúe el puntero del ratón sobre el hiperenlace.

El siguiente comando del hipertexto, es el texto que se mostrará resaltado en el *browser* (hipertexto), en este caso será: "Pulse Aquí" y el servidor *Web* al que se conectará será el que tiene por dirección: "http://www.inta.es", mientras que el documento al que se hará referencia lleva por título: "presenta.html".

```
<A HREF="http://www.inta.es/presenta.html"> Pulse aqui </A>
```

El siguiente ejemplo muestra una liga dentro del sistema, es decir una liga que hará referencia a un documento dentro del servidor actual, esta instrucción es parecida a la liga entre sistemas; aquí se debe de indicar la ruta y nombre del archivo al que se hará referencia, sin necesidad de indicar que servicio es el que se esta solicitando :

```
<A HREF="/Internet/www/presenta.html"> Pulse aqui </A>
```

Por otro lado, el segundo parámetro más utilizado dentro de <A> es NAME, con este parámetro se crean las liga dentro de una misma página, es decir, un enlace que conectara dos zonas de texto dentro de la misma página.

En primer lugar, deberá introducir el comando <A> y utilizar el parámetro HREF en el lugar donde desea introducir el hipertexto; para indicar el lugar de destino se debe volver a utilizar el comando <A> pero esta vez con el parámetro NAME, como argumento de este parámetro se introduce un nombre, el mismo al que hará referencia el parámetro HREF del hipertexto y para el parámetro HREF, este nombre ira precedido por el caracter "#": un ejemplo de este tipo de hipertexto se muestra a continuación:

```
<A HREF="#FinalPagina">Salta al fin de la página</A>
resto de la página
<A NAME=FinalPagina>;Este es el final de la Página!</A>
```

También es posible saltar a un área específica incluida en otra página, sin más que añadir el nombre de la zona a la dirección URL; como se muestra en el siguiente ejemplo :

```
<A HREF="http://www.ziff.com/demo.html#dirección">Pulsa aqui</A>
```

La única precaución a adoptar en este caso, es que en la página destino exista un ancla de destino con ese nombre (dirección).

## 5.9 Etiquetas HTML de integración (multimedia)

Los documentos HTML podrán contener gráficos, imágenes, videos plenamente dinámicos e, incluso, archivo de audio. En todos estos casos esta integración se lleva a cabo utilizando un hiperenlace, que en lugar de apuntar a otra parte del documento apunte, precisamente, a un gráfico o imagen, archivo de vídeo o de audio. En general, el *browser* utilizado no podrá mostrar o reproducir el contenido del archivo al que se esta enlazando (vídeo o audio), pero podrá ejecutar automáticamente una aplicación que lo podrá hacer; si así esta indicado en la configuración.

Los *browsers* suelen tener la capacidad de discernir el tipo de archivo al que se hace referencia y, según el caso, poner en marcha un visualizador de gráficos (para el caso de que el archivo tenga extensión .GIF o .JPG), un reproductor de sonido (si la extensión del archivo es AIFF, SND o WAV) o un reproductor de vídeo (si la extensión del archivo es AVI, MOV o MPG).

En los siguientes ejemplos se muestra la manera de hacer enlaces con estos tipos de archivos:

```
<A HREF="/www/imagen/biblio.gif"> Pulse aquí </A>
<A HREF="/www/audio/audio.wav"> Pulse aquí </A>
<A HREF="/www/video/video.avi"> Pulse aquí </A>
```

En estos tres casos, cuando el usuario haga clic sobre las palabras "Pulse aquí" el *browser* mostrara o reproducirá un fichero gráfico, de audio o de vídeo (respectivamente) siempre que pueda ejecutar el programa reproductor necesario para ello.

Existe otro método para introducir directamente imágenes en la propia pagina. Este procedimiento tiene el inconveniente de que aumenta mucho el tamaño del archivo HTML y, por lo tanto, el tiempo de transferencia de este archivo se incrementa en gran medida. Sin embargo, es el método mas utilizado en el WWW ya que las paginas conseguidas así son mucho mas vistosas, al incluir gráficos de gran calidad.

La etiqueta HTML que permite incluir gráficos en las páginas es <IMG>, la sintaxis es la siguiente:

```
<IMG SRC="nombre.ext">
```

Dentro del parámetro SRC se escribe la dirección completa del archivo gráfico al que se hace referencia. Un ejemplo mas explícito podría ser:

```
<IMG SRC= "/imágenes/dibujo.gif">
```

En el ejemplo anterior, se insertara el archivo gráfico DIBUJO.GIF en la posición donde se encuentre el comando dentro de la pagina.

Para permitir que los *browsers* basados en texto puedan mostrar documentos HTML que contengan gráficos se puede introducir el parámetro ALT (alternativo) antes del caracter > del final de la etiqueta <IMG>. ALT indica al *browser* que muestre el texto introducido detrás de ALT= en lugar de mostrar el propio gráfico.

En el siguiente ejemplo los *browsers* HTML visualizaran la frase "Imagen Biblioteca Central" en lugar de la imagen contenida en el archivo BIBLIO.GIF.

```
<IMG SRC="BIBLIO.GIF" ALT="Imagen Biblioteca Central">
```

Es posible hacer que un gráfico se convierta en hipervínculo, lo único que se requiere es encadenar la etiqueta <A> con la etiqueta <IMG>, tal y como se muestra a continuación:

```
<A HREF= "file://www.meteor.es/clima/hoy/meteosat.mpg">
<IMG SRC="mapa_espana.gif">Pulse en la imagen para ver una
película meteorológica</A>
```

Como se puede ver en el ejemplo, se comienza introduciendo la etiqueta de hipervínculo, a continuación se incluye la etiqueta de gráfico en línea y, por último, se termina con la etiqueta de cierre del hipervínculo. Cuando el usuario pulse sobre la imagen, hará un enlace con el archivo "meteosat.mpg"; el *browser* ejecutara un programa de aplicación para poder ver el archivo de video. Observe que los elementos que hacen referencia a imágenes no necesitan una etiqueta de finalización.

Aunque el uso del lenguaje HTML es sencillo, cuenta con los suficientes comandos como para poder hacer que el usuario tenga sus dudas razonables a la hora de crear sus propias paginas. Además, solo se han tratado algunas de las instrucciones básicas de este lenguaje; es por eso que en las siguientes tabla mostramos algunas de las etiquetas más comunes, su uso y la versión HTML a la que pertenecen.



Nivel	Explicación
2	parte de las especificaciones de HTML 2.0
3	parte de las especificaciones de HTML 3.0
N	extensión HTML para el browser Netscape

Tabla 5.2

## Estructura del Documento

Nivel	Declaración HTML	Explicación
2	<HTML></HTML>	Inicio y fin de del documento HTML
2	<HEAD></HEAD>	Inicio y fin de la cabecera
2	<BODY></BODY>	Contenido del documento que será desplegado por el browser

Tabla 5.3

## Meta Etiquetas del Documento

Nivel	Declaración HTML	Explicación
2	<TITLE></TITLE>	Título del documento, va en la cabecera.

Tabla 5.4

## Texto

Nivel	Declaración HTML	Explicación
2	&#code	Caracter especial con código numérico o nombre ISO latin-1
2	<!-- comentario -->	Comentario, no desplegado por el browser

Tabla 5.5

## Anclas

Nivel	Declaración HTML	Explicación
2	<A HREF="URL">hipertexto</A>	Ancla, liga el texto "hipertexto" a el documento URL.
2	<A NAME="salto">texto</A>	Ancla con el nombre "salto" asociada con "texto".
2	<A NAME="URL#salto">hipertexto</A>	Ancla con salto de "hipertexto" a la ancla llamada "salto" en el documento URL.

Tabla 5.6

## Listas y Bloques de Texto

Nivel	Declaración HTML	Explicación
2	<UL></UL>	Lista no numerada (los elementos son marcados con <LI>).
2	<OL></OL>	Lista numerada (los elementos son marcados con <LI>).
2	<DL></DL>	Lista definida ( los términos son marcados con <DT>, las definiciones son marcadas con <DD>).

Tabla 5.7

## Separadores

Nivel	Declaración HTML	Explicación
2	<H1></H1> .... <H6></H6>	Cabeceras de nivel 1(mayor) a la 6 (menor).
2	<HR>	Regla horizontal.
N	<HR Size="espesor">	Regla horizontal, espesor es la en pixeles.
2	 	Línea de quiebra o rompimiento.
2	<P>	Comienza párrafo.
3	<P Align="left   center   right"></P>	Inicio y fin de párrafo con texto alineado a la izquierda, centro y derecha.

Tabla 5.8

## Formato Físico de Caracteres

	Declaración HTML	Explicación
2	<B></B>	Letra de tipo Bold.
2	<I></I>	Letra de tipo Itálica.
2	<PRE></PRE>	Texto preformateado.
3	<SUP></SUP>	Letra de tipo superscript
3	<SUB></SUB>	Letra de tipo subscript.
N	<CENTER></CENTER>	Texto centrado.
N	<BLINK></BLINK>	Lucirá como texto parpadeante.
N	<FONT Size="tamaño"></FONT>	Conjunto de letras donde "tamaño" ira desde 1 a 7.
N	<FONT Color="#RRGGBB"></FONT>	Conjunto de letras, el color esta dado por el valor en hexadecimal del rojo, verde y azul.

Tabla 5.9

### Formato Lógico de Caracteres

Nivel	Declaración HTML	Explicación
2	<CITE></CITE>	Desplegar una cita (citación).
2	<CODE></CODE>	Desplegar código fuente.
2	<EM></EM>	Despliega letra subrayada.
2	<STRONG></STRONG>	Despliega letra negrita.
2	<VAR></VAR>	Nombre de variable.

Tabla 5.10

### Imágenes.

Nivel	Declaración HTML	Explicación
2	<IMG SRC= "URL">	Inserta una imagen de "URL" dentro del documento.
2	<IMG SRC= "URL" ALT= "cadena">	Despliega "cadena" en lugar de la imagen para los browser no gráficos.
2	<IMG SRC= "URL" ALIGN="top bottom   middle">	Alienaciones de un texto después de un a imagen.
N,3	<IMG SRC= "URL" width="ancho" height="alto">	Despliega una imagen con un "ancho" y "alto" en pixeles.

Tabla 5.11

### Tablas

Nivel	Declaración HTML	Explicación
3	<TABLE BORDER></TABLE>	Inicio y fin de una tabla con borde
3	<TR></TR>	Inicio y fin de un renglón.
3	<TH colspan="C" rowspan="R"></TH>	Inicio y fin de una celda de cabecera expandida C columnas y R renglones.
3	<TD colspan="C" rowspan="R"></TD>	Inicio y fin de una celda de datos expandida C columnas y R renglones.
3	<TD TH ALIGN=" left   right   center"></TH TD>	Alineación horizontal de un elemento en una celda de datos o cabecera
3	<TD TH VALING="top middle  bottom"></TH TD>	Alineación vertical de un elemento en una celda de datos o cabecera
N	<TD   TH WIDTH="ancho"></TH TD>	Ancho horizontal de una celda de datos o cabecera donde "ancho" esta en pixeles.

Tabla 5.12

## Extensiones de Netscape en BODY.

Nivel	Declaración HTML	Explicación
N	< BODY BACKGROUND="URL">	La textura del fondo es la imagen "URL".
N	< BODY BGCOLOR="#RRGGBB">	El color del fondo esta dado por el valor en hexadecimal del rojo, verde y azul.
N	< BODY TEXT="#RRGGBB">	Conjunto de colores para el texto.
N	< BODY LINK="#RRGGBB">	Conjunto de colores para una liga que no ha sido visitada.
N	< BODY VLINK="#RRGGBB">	Conjunto de colores para una liga visitada.

Tabla 5.13

## 5.10 Editores HTML

Como los documentos HTML son simplemente archivos de texto ASCII estándar, es posible utilizar el bloc de notas de Windows, el EDIT del MS-DOS o cualquier otro editor de texto para desarrollar los documentos HTML. Sin embargo, esta tarea se facilita si se emplean editores especiales adaptados a la sintaxis HTML.

Actualmente existen muchos en el mercado y la gran ventaja es que la mayoría de ellos son programas *shareware* y *freeware* que se pueden obtener directamente a través de *Internet*.

La lista de editores en comandos HTML es larga y sería extremadamente laborioso describir en detalle todas y cada una de sus funciones. En resumen, se puede decir que todos ellos permiten introducir las etiquetas de control que se utilizan para escribir un documento HTML con solo realizar una pulsación de ratón sobre un determinado icono.

Incluso, algunos de estos editores permiten verificar la estructura del documento para ver si es correcta la sintaxis HTML utilizada. Muchos de estos editores cuentan con algunas plantillas predesarrolladas que facilitan al máximo el desarrollo de páginas tipo, como formularios, presentaciones, listados, etc.

Básicamente existen dos tipos de software de editores HTML disponibles, los editores *stand-alone* y los procesadores de palabras de plantilla; la mayoría de los editores *stand-alone* no requieren soporte de otro software para trabajar, aunque algunos están

escritos en Visual Basic y requieren de un archivo (librería dinámica de liga) llamado VBRUN300.DLL.

Los editores HTML y los procesadores de plantilla caen dentro de dos categorías, los editores HTML basados en reglas y los editores HTML sin comprobación.

En los editores HTML basados en reglas, requieren que al escribir un documento HTML, se debe de ser muy cuidadoso en las convenciones del formato de los documentos HTML; estos editores no permiten utilizar una etiqueta que no sea estándar o insertar una etiqueta en un lugar incorrecto. Por otro lado, los editores sin comprobación no realizan la validez del documento, si se insertan etiquetas incorrectamente el editor no se lo hará saber; el beneficio de usar un editor de este tipo es que se podrán utilizar etiquetas no estandarizadas.

Una característica que se puede notar en la mayoría de los editores HTML para Windows es que se podrá ver la etiqueta actual de HTML introducida, esto es lo que se conoce como WYSIWYG (*What You See Is What You Get*) dentro de las publicaciones en el WWW; esto quiere decir que conforme se escriba el documento HTML se puede ver en pantalla la forma en que aparecerá en el *browser*, esto no siempre es confiable dado que cada *browser* tiene su propia manera de desplegar la página; aunque WYSIWYG le da al usuario el control sobre el tamaño de la página y el tipo de letra para que los *browser* lo desplieguen de la misma manera lo mejor es previsualizar el documento en un *browser* antes de ser publicada. Algunos de los editores HTML *stand-alone* más conocidos son los siguientes:

### 5.10.1 HotMetal y HotMetal Pro

Uno de los editores de comandos HTML más potentes que existen hoy en día es HoTMetal de SoftQuad (la figura 5.3 muestra la interfaz de HoTMetal 2.0).

Este programa está disponible para una amplia variedad de plataformas entre las que se incluyen: Windows, Macintosh y UNIX, contando en todas ellas con un interfaz de usuario muy similar. También existe una versión comercial denominada HoTMetal Pro.

HoTMetal muestra los documentos en modo WYSIWYG y con esto se asegura que las etiquetas HTML introducidas en el documento cumplen correctamente las especificaciones del lenguaje HTML. Además, el programa solo insertará las etiquetas en aquellos puntos en que tenga sentido hacerlo. Un inconveniente de HoTMetal es

que no se podrá abrir documentos que no se ajusten perfectamente a la especificación del lenguaje HTML..

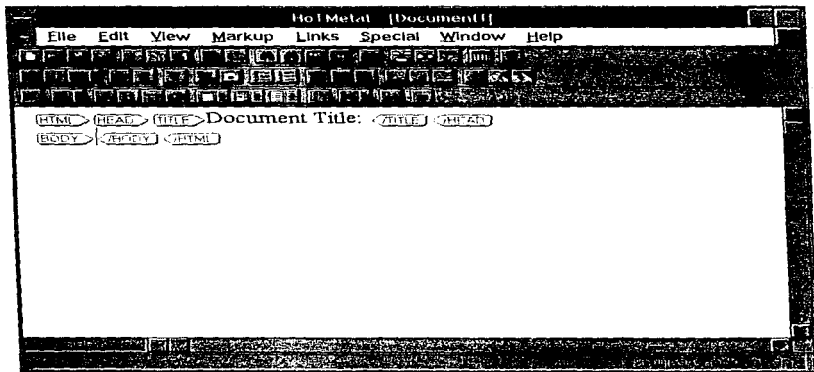


Figura 5.3 . Pantalla principal de HoTMetal 2.0

HotMetal Pro es un editor HTML basado en reglas que incluye diversas características, que no están disponibles en la versión *freeware*, estas son usadas para trabajos duros de edición HTML, algunas de estas son :

- Importación de archivos a través de un filtro: esta característica permite importar archivos HTML que no están correctos.
- Flexibilidad de edición adicional: en la versión Pro, se tiene control sobre la edición y esto hace el proceso de creación más fácil.
- tablas : Hot Metal Pro permite insertar tablas de HTML 3 dentro del documento.
- Documentación en línea : HotMetal Pro provee de un extenso sistema de ayuda en línea .

que no se podrá abrir documentos que no se ajusten perfectamente a la especificación del lenguaje HTML.

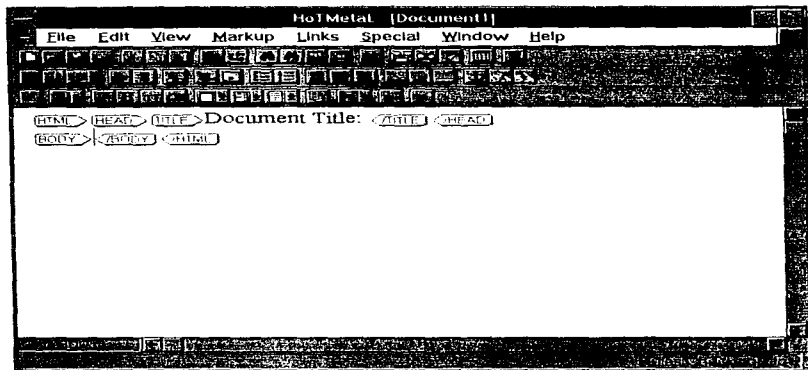


Figura 5.3 . Pantalla principal de HoTMetaL 2.0

HotMetal Pro es un editor HTML basado en reglas que incluye diversas características, que no están disponibles en la versión *freeware*, estas son usadas para trabajos duros de edición HTML, algunas de estas son :

- Importación de archivos a través de un filtro: esta característica permite importar archivos HTML que no están correctos.
- Flexibilidad de edición adicional: en la versión Pro, se tiene control sobre la edición y esto hace el proceso de creación más fácil.
- tablas : Hot Metal Pro permite insertar tablas de HTML 3 dentro del documento.
- Documentación en línea : HotMetal Pro provee de un extenso sistema de ayuda en línea .

- Tutorial en línea : Un documento HTML que lo guíara para la construcción de un documento HTML.
- Velocidad: HotMetal Pro 2.0 usa mucho menos memoria y es mucha más rápido.
- Barras de herramientas flotantes : tres barras de herramientas que permiten insertar elementos al hacer una pulsación del ratón en un icono.
- Mejor soporte previsual : es posible previsualizar el documento HTML con múltiples browsers.
- Rápido acceso a archivos : el menú FILE muestra los últimos cuatro archivos que fueron abiertos.
- Mejor soporte para HTML 3 y Netscape : HotMetal Pro provee de soporte para etiquetas de HTML3 y Netscape como son los *backgrounds* (fondo de la pagina) y las tablas.
- Mejor soporte para caracteres especiales : los caracteres especiales pueden ser ingresados desde el teclado o desde un ventana.

Otras características de HotMetal Pro es que se puede ver la estructura de marcas del HTML, complementar la estructura HTML, abrir un archivo HTML existente, editar archivos largos y previsualizar una imagen.

También podrá abrir páginas creadas en otros editores, y como HotMetal Pro es un editor basado en reglas, este verificará que la sintaxis HTML sea correcta.

### 5.10.2 HotDog y HotDog Pro

HotDog es uno de los editores de más fácil uso y más completo que se puede usar; HotDog soporta extensiones de Netscape y etiquetas HTML 3. HotDog puede correr sobre un 386, pero un 486 con 8 MB de memoria es lo recomendado.

Este programa necesita de aproximadamente 2 MB de espacio en disco. HotDog también soporta las operaciones de marca y arrastre para insertar ligas imágenes y archivos de texto, la figura 5.4 muestra la pantalla principal del HotDog Pro.

Un característica de este programa es que cuando se crea un nuevo documento HTML partiendo de cero el editor insertara automáticamente algunas de las etiquetas de



estructura global, tales como `<HTML>...</HTML>`, `<HEAD>...</HEAD>`, `<TITLE>...</TITLE>`, y `<BODY>... </BODY>`.

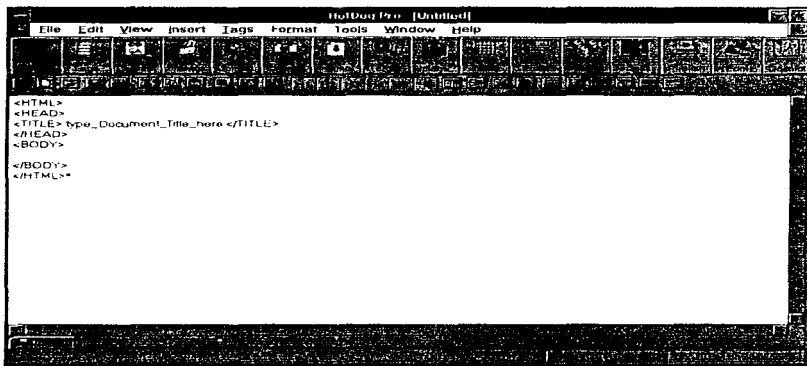


Figura 5.4 Pantalla principal del HoTDog Pro.

### 5.10.3 HTML Writer

HTML Writer es un programa *shareware* que permite utilizar una amplia variedad de métodos para insertar etiquetas de formato HTML, incluyendo menús, barras de herramientas y atajos con el teclado. El programa incluye un completo sistema de ayuda y también puede trabajar con archivos de gran tamaño.

Todas las funciones del programa se pueden obtener a través de los comandos contenidos en los menús y utilizando los iconos de la barra de herramientas.

La barra de herramientas contiene iconos que permiten probar un documento; cortar, copiar o pegar zonas del mismo; deshacer operaciones realizadas erróneamente; insertar etiquetas de negrita, cursiva, subrayado y texto preformateado; aplicar los seis

niveles de cabecera; crear listas (ordenadas y no ordenadas, y glosarios); añadir elementos a las listas, insertar saltos de párrafos, saltos de línea y líneas horizontales; añadir imágenes gráficas y crear hiperenlaces.

Este editor al igual que HotDog cuando crea un nuevo documento HTML inserta automáticamente las etiquetas de estructura global:

```
<HTML>...</HTML>, <HEAD>...</HEAD>, <TITLE>...</TITLE>, y <BODY>...</BODY>.
```

#### 5.10.4 HTML Assistant

Este es uno de los editores de comandos más populares, HTML Assistant es, en esencia, un editor de texto especial que reduce los errores en la edición de documentos HTML; la barra de herramientas incluye botones para introducir casi cualquier tipo de etiqueta HTML.

HTML Assistant es freeware, sin embargo, existe una versión comercial denominada HTML Assistant Pro que proporciona funciones adicionales de edición de documentos HTML. En este editor es posible introducir todas las etiquetas que desee en los documentos HTML utilizando exclusivamente la barra de iconos, sin necesidad de seleccionar ninguna opción del menú. Una limitación del HTML Assistant es el hecho de que no es posible editar archivos con una longitud superior a 32 Kbytes.

#### 5.10.5 Microsoft Front Page Editor

La gran ventaja de este editor con respecto de los demás editores es que prácticamente no se escribe un documento HTML, sino que este programa permite crear y visualizar la página tal y como se verá en el *browser*; es decir, este programa no inserta etiquetas HTML sino que permite insertar el elemento que se desee dentro de la página ya sea una imagen o un enunciado de cabecera.

Así también el programa es capaz de crear ligas con solo definir los elementos necesarios para esto (definir el hipertexto y el documento HTML al que se hace referencia), la facilidad que ofrece este software para crear un página es muy grande ya que no se requiere del completo conocimiento de la sintaxis del lenguaje HTML, el programa genera el documento HTML dependiendo de los elementos que se inserten a la página. La figura 5.5 muestra la pantalla principal del editor Front Page.

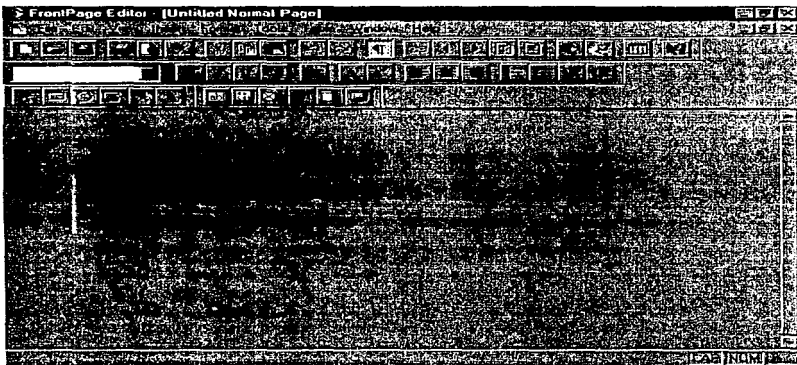


Figura 5.5 Pantalla principal de Microsoft Front Page

Este software se ejecuta sobre Windows 95 y con esto las características que debe tener la PC aumentan considerablemente, ya que el editor presenta problemas para desplegar una página cuando no se cuentan con los suficientes recursos en la PC.

### 5.10.6 HTMLed

HTMLed es un programa *shareware*. HTMLed es similar a HTML Assistant y HTML Writer ya que contiene iconos en la barra de herramientas que permiten insertar etiquetas HTML en los documentos HTML. HTMLed cuenta con una serie de barras de herramientas muy completas; cuenta con la opción de mostrar simultáneamente en pantalla hasta cuatro barras de herramientas, incluyendo una barra con opciones personalizadas. Por último, HTMLed parece trabajar a una mayor velocidad que los editores HTML Assistant y HTML Writer.

Estos son solo algunos de los editores más populares, actualmente existe una gran variedad de editores que se ejecutan en diferentes plataformas como pueden ser UNIX, Macintosh, Windows 3. x o Windows 95.

En la siguiente tabla se muestran varios editores HTML que existen en para las diferentes plataformas :

EDITOR	PLATAFORMA	DIRECCIÓN DE LOCALIZACIÓN
4W Publisher	WINDOWS	<a href="http://www.infoanaitic.com/webbldr/index.html">http://www.infoanaitic.com/webbldr/index.html</a>
AmiWeb (requiere Ami Pro)	WINDOWS	<a href="http://www.es.nott.ac.uk/~sbx/amiweb.html">http://www.es.nott.ac.uk/~sbx/amiweb.html</a>
CU_HTML.DOF (con Word 2.0 y 6.0)	WINDOWS	<a href="http://www.cuhk.hk/cu_html/cu_html.html">http://www.cuhk.hk/cu_html/cu_html.html</a>
GT_HTML.DOT (con Word 2.0 y 6.0)	WINDOWS	<a href="http://www.gatech.edu/word_html/release.html">http://www.gatech.edu/word_html/release.html</a>
HotDog Web Editor	WINDOWS	<a href="http://www.sausage.com">http://www.sausage.com</a>
HoTMetal y HoTMetal Pro	WINDOWS	<a href="http://www.sq.com/products/hotmetal/hm-ftp.html">http://www.sq.com/products/hotmetal/hm-ftp.html</a>
HTML Assistant	WINDOWS	<a href="http://cs.dal.ca/ftp/htmlasst/htmlfaq.html">http://cs.dal.ca/ftp/htmlasst/htmlfaq.html</a>
HTML Author	WINDOWS	<a href="http://www.salford.ac.uk/depts/iti/homepage.html">http://www.salford.ac.uk/depts/iti/homepage.html</a>
HTML Easy Pro!	WINDOWS	<a href="http://www.seed.net.tw/~mikylin/htmlasy.html">http://www.seed.net.tw/~mikylin/htmlasy.html</a>
HTML Writer	WINDOWS	<a href="http://lal.cs.byu.edu/people/nosack/index.html">http://lal.cs.byu.edu/people/nosack/index.html</a>
HTMLed	WINDOWS	<a href="http://www.ist.ca/htmlcd/">http://www.ist.ca/htmlcd/</a>
Internet Assistant para Word (requiere Word 6.0)	WINDOWS	<a href="http://www.microsoft.com/MSOffice/Word/ia/default.html">http://www.microsoft.com/MSOffice/Word/ia/default.html</a>
Web Publisher	WINDOWS	<a href="http://www.skisoft.com/skisoft/">http://www.skisoft.com/skisoft/</a>
WebAuthor (requiere Word 6.0)	WINDOWS	<a href="http://www.qdeck.com/webauthor/fact.html">http://www.qdeck.com/webauthor/fact.html</a>
Alpha	MACINTOSH	<a href="http://www.cs.umd.edu/~keleher/alpha.html">http://www.cs.umd.edu/~keleher/alpha.html</a>
BBEdit HTML Extensions	MACINTOSH	<a href="http://northshore.shore.net/~quantum/BBEdit.html">http://northshore.shore.net/~quantum/BBEdit.html</a>
HTML edit	MACINTOSH	<a href="http://ogopogo.nttc.edu/tools/HTMLedit/HTMLedit.html">http://ogopogo.nttc.edu/tools/HTMLedit/HTMLedit.html</a>
HTML Editor	MACINTOSH	<a href="http://dragon.acadiau.ca/~giles/HTML-Editor/Documentation.html">http://dragon.acadiau.ca/~giles/HTML-Editor/Documentation.html</a>
HTML HyperEditor	MACINTOSH	<a href="http://balder.syo.lu.se/Editor/HTML-HyperEditor.html">http://balder.syo.lu.se/Editor/HTML-HyperEditor.html</a>
HTML Grinder	MACINTOSH	<a href="http://www.matterform.com/mfgrinder/htmlgrinder.html">http://www.matterform.com/mfgrinder/htmlgrinder.html</a>
HTML Web Weaver	MACINTOSH	<a href="http://137.143.111.3/web.weaver/about.html">http://137.143.111.3/web.weaver/about.html</a>
HTML Writer	MACINTOSH	<a href="http://www.uwic.washington.edu/JonWiederspan/HTMLEditor.html">http://www.uwic.washington.edu/JonWiederspan/HTMLEditor.html</a>
Navipress	MACINTOSH	<a href="http://www.navisoft.com/products/press/press.html">http://www.navisoft.com/products/press/press.html</a>
S.H.E.	MACINTOSH	<a href="http://dewey.lib.ncsu.edu/staff/morgan/simple.html">http://dewey.lib.ncsu.edu/staff/morgan/simple.html</a>
A.S.H.E.	UNIX	<a href="http://www.cs.rpi.edu/~puninj/TALK/head.html">http://www.cs.rpi.edu/~puninj/TALK/head.html</a>
Emacs	UNIX	<a href="http://www.santafe.edu/~nelson/bhm-beta/">http://www.santafe.edu/~nelson/bhm-beta/</a>
HoTMetal	UNIX	<a href="http://www.sq.com">http://www.sq.com</a>
Navipress	UNIX	<a href="http://www.navisoft.com/products/press/press.html">http://www.navisoft.com/products/press/press.html</a>
tkHTML	UNIX	<a href="http://www.ssc.com/~roland/tkHTML/tkHTML.html">http://www.ssc.com/~roland/tkHTML/tkHTML.html</a>

tkWWW		UNIX	<a href="http://www.osf.org/ri/hci_pepers/tkwww.html">http://www.osf.org/ri/hci_pepers/tkwww.html</a>
WebMagic Author		UNIX	<a href="http://www.sgi.com/Products/WebFORCE/WebFORCESoft.html">http://www.sgi.com/Products/WebFORCE/WebFORCESoft.html</a>
WYSIWYG Editor	HTML	UNIX	<a href="http://www.cs.city.ac.uk/homes/njw/htmltext/htmltext.html">http://www.cs.city.ac.uk/homes/njw/htmltext/htmltext.html</a>

Tabla 5.14

Este trabajo no pretende hacer una evaluación de los editores para HTML, solo intenta mostrar algunos de los más utilizados.

## 5.11 Herramientas gráficas

Para la realización de los gráficos de las páginas, se debe de contar con las herramientas necesarias para obtener imágenes de buena calidad; estas herramientas pueden comprender desde un software de edición gráfica hasta una *scanner*, si no se desea dibujar para crear las imágenes, entonces dependiendo de las características del *scanner* que se utilice se obtendrán imágenes con una mayor o menor resolución, o también es posible obtener las imágenes ya digitalizadas a través de una cámara de vídeo, aunque para esto se requiere de un hardware especial para realizar esta tarea.

Es posible obtener las imágenes con la ayuda de software de edición gráfica que sea capaz de trabajar con varios formatos de archivos de gráficos, principalmente el formato .GIF y el formato .JPG que son los más comunes en el WWW, este software debe de contar con suficientes herramientas tanto de edición como de diseño de imágenes que permitan realizar esta tarea de una manera sencilla y obteniendo buenos resultados. Por otra parte, si no se cuenta con un software que pueda manejar los formatos .GIF o .JPG es necesario un software adicional que convierta los archivos a los formatos especificados anteriormente ya que otros formatos no son desplegados dentro de la página. Aquí exponemos algunas de las herramientas más comunes que pueden ser utilizadas para la creación y modificación de las imágenes.

### 5.11.1 PaintShop Pro (para Windows)

Este es un programa *shareware* que se puede obtener en *Internet*; este software es un poderoso editor y visualizador de imágenes, soporta alrededor de 20 diferentes formatos de archivos de gráficos incluyendo los formatos .gif y .jpg encontrados comúnmente en el WWW. Cuenta con características para la edición y manipulación de gráficos, cumple con los paquetes comerciales con un número de variedad de

filtros y efectos especiales; este software no requiere que la PC cuente con características especiales, para correr este paquete, puede ofrecer buenos resultados con un monitor y tarjeta VGA ; puede trabajar con 256 colores y obtener una buena definición en las imágenes; pero para obtener mejores resultados se puede tener un monitor SVGA y mejorar la definición de las imágenes.

Una de las tantas características de este paquete es la posibilidad de presentar el fondo de la imagen como una transparencia, esto es de mucha utilidad ya que la imagen en la página aparecerá sin el fondo o recuadro y dar la impresión de estar flotando en la página.

Esta herramienta gráfica también permite sumar efectos especiales al texto, que puede estar dentro de la imagen, efectos como texto en capas, texto sombreado, texto realzado, texto luminoso y texto translucido.

La figura 5.6 muestra la pantalla principal del PaintShop Pro.

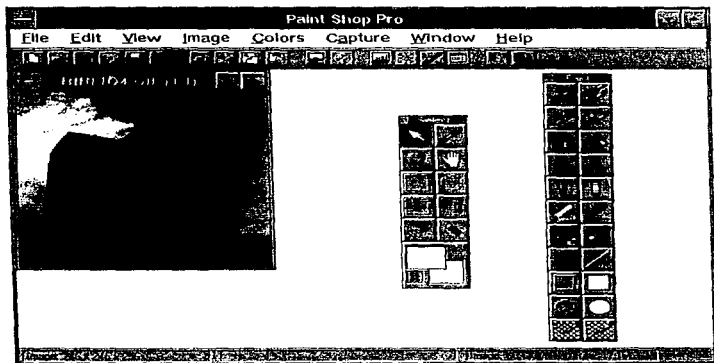


Figura 5.6 Pantalla principal de PaintShop Pro.

### 5.11.2 Corel Draw 5.0

Este programa al igual que PaintShop Pro ofrece grandes posibilidades en cuanto a la edición gráfica maneja la mayoría de los formatos de archivos gráficos existentes; este paquete cuenta con una mayor cantidad de herramientas para el diseño de imágenes a través del método vectorizado, aunque también cuenta con un programa (Corel Photo Paint) que trabaja las imágenes como mapas de bits, dispone de un programa de captura y otro con el cual es posible realizar animaciones. La figura 5.7 muestra la pantalla principal de Corel Draw 5.0.



Figura 5.7 Pantalla Principal de Corel Draw 5.0

Algunas de las herramientas que ofrece Corel Draw 5.0 es la capacidad de definir texturas a los dibujos, añadir texto a las imágenes así como el color, tipo de letra y

tamaño de la misma, es capaz de escalar una imagen y redefinir su tamaño, dentro de los elementos de edición puede copiar, cortar y pegar texto, también cuenta con tablas de símbolos y gráficos ya definidos que pueden ser incrustados en la imagen y con esto se acelera el diseño de la imagen.

Con Corel Photo Paint se pueden sumar efectos especiales a las imágenes como son la perspectiva, degradación al combinar colores, y un envolvimiento de la imagen; Corel Draw tiene la posibilidad de importar y exportar imágenes en la mayoría de los formatos existentes; también cuenta con un tutorial y la opción de la ayuda a través de una ventana que da la opción de realizar una búsqueda de algún tópico en especial.

Estos son solo algunos de los paquetes de edición gráfico que se pueden utilizar para el desarrollo de imágenes, ya que actualmente existe una gran cantidad de software de diseño gráfico disponible, que puede implicar herramientas de diseño más flexible en comparación con los paquetes aquí mencionados, por lo que la elección del software de diseño gráfico es decisión de cada persona, basándose en las necesidades gráficas que desee cubrir, así como los recursos con los que se cuente para adquirir alguno de los paquetes de diseño y las características mínimas de la maquina donde se montara este software.

Es claro que en la medida que se desee una mejor calidad de gráficos los recursos, tanto de software como de hardware, dentro de la maquina (PC) también aumentan, es decir con los recursos mínimos es posible obtener gráficos con una calidad aceptable, correr algún paquete de edición gráfica y visualizarlos, pero dentro del WWW el impacto visual es una parte muy importante, es por esto que los gráficos aquí tienen una mayor importancia, pues gráficos de gran calidad darán una mayor presentación e interés a las páginas; y para lograrlo es necesario tener una mayor cantidad de recursos, hardware y software, que brinden la posibilidad de tener este tipo de gráficos.



.....



# Capítulo 6

## Capítulo 66



.....

## CGI

Hasta ahora se han descrito los elementos para que se pueda navegar en el WWW, así como la forma en que se pueden hacer dichos documentos, sin embargo existe otra característica para poder obtener resultados en base a datos que se introduzcan por medio del cliente y que sean procesados en el servidor HTTP y una vez terminado, el resultado sea enviado al cliente. El lenguaje HTML cuenta con elementos que permiten desplegar formas, que son documentos HTML, los cuales contienen campos que pueden ser completados con información (numérica o de texto) por los usuarios, además de tener listas desplegables y casillas de selección.

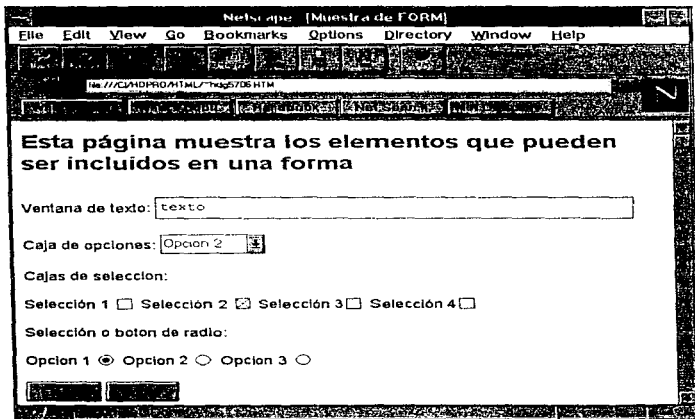


Figura 6.1 Elementos de una forma

Cuando la información es captada por el browser, ésta es enviada a un servidor HTTP que es especificado en la forma, el servidor inicia un programa, también especificado en la forma, que puede procesar la información obtenida, tales programas son conocidos como programas CGI (Common Gateway Interface), scripts CGI o simplemente CGI como se manejará en adelante. Estos programas se ejecutan en el servidor y son independientes al software instalado para dicho servidor, aunque es necesario que dichos CGI interactúen con algunos parámetros que proporciona el servidor, en la figura 6.2 se muestra un esquema de funcionamiento.

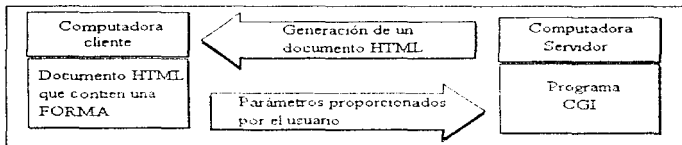


figura 6.2 Esquema de funcionamiento del cgi

Los CGI son un ejemplo de la aplicación del modelo Cliente/Servidor, más palpable; por que se aplican todos los elementos que lo caracterizan, además que esta aplicación permite en el WWW realizar aplicaciones más elaboradas y de mayor utilidad. Con los CGI se puede tener una mayor interacción entre los propietarios de los servidores de WWW y los usuarios que acceden a las páginas. Aunque esta interacción sea a un nivel muy bajo y no se realice en tiempo real, si permite una comunicación más directa, ya que se pueden diseñar páginas con elementos que puedan ser completados por los usuarios, de tal forma que esta información sirva para describir alguna situación, requerimiento o servicio, dependiendo del objetivo de la aplicación y una vez captada dicha información, se procese en el servidor, en donde dependiendo de los datos obtenidos del cliente se llevaran a cabo determinadas acciones, las cuales permitirán generar un documento HTML en respuesta a la petición del cliente. Con los CGI se introduce el concepto de documentos HTML dinámicos, lo que significa que se generan dependiendo de que datos sean introducidos. Una vez determinado que datos se introdujeron, se procede a fabricar el documento HTML, lo que significa que éste no existe físicamente, sino que se genera una instancia en función de los datos, que solamente podrá ser desplegada por el cliente que realizó la petición o por otro cliente que proporcione la misma información.

La utilización de los CGI es variada, se pueden utilizar para vender productos, para realizar consultas a bases de datos o como ya se menciono anteriormente, para realizar búsquedas; en algunos casos se utilizan para realizar cálculos o simplemente para conocer algunos aspectos de los usuarios que utilicen ese CGI.

## 6.1 Programación de los CGI

Los CGI se pueden programar en varios lenguajes como Pascal, C, C++, Fortran o lenguajes de tipo script como Perl, TCL, Shell de Unix (por lo que también se les llama scripts CGI). Lo importante es que la aplicación CGI que se genere en cualquiera de los lenguajes, sea soportada en la computadora o computadoras en la que se encuentra tal aplicación, ya que cuando el cliente realice una petición, para que se ejecute dicho CGI, éste ejecutara los procesos requeridos en la computadora servidor, por lo que dicha aplicación debe considerar las características de ese equipo.

Los CGI por lo general son almacenados en un subdirectorio llamado cgi-bin, sin embargo ello depende de como se realizó la instalación del servidor HTTP.

Para poder generar una aplicación CGI es necesario considerar ciertos parámetros además de los que el usuario proporciona, anteriormente se describió el funcionamiento e interacción del cliente y el servidor en el WWW, sin embargo no se profundizó en la información que se transfiere entre ambos, la cual es la siguiente.

Cuando el cliente le envía una petición al servidor para obtener un documento HTML, el cliente envía la siguiente información:

```
GET /docu2.html HTTP/1.0
Accept: www/source
Accept: text/html
.
Accept: image/gif
User-Agent: Lynx/2.2 libwww/2.14
From: Isaac@panoramx.dgbiblio.unam.mx
      * línea en blanco *
```

La petición GET indica cual archivo quiere el cliente y anuncia que esta usando HTTP versión 1.0 para comunicarse. El cliente también lista los tipos de MIME

(Multipurpose Internet Mail Extension) que aceptará en el regreso (en este caso la lista "Accept" ha sido truncada por brevedad), y se identifica a sí mismo como cliente Lynx, el cliente también identifica a su usuario en el campo "From". Finalmente el cliente envía una línea en blanco indicando que se ha completado la petición.

El servidor entonces responde enviando lo siguiente:

```
HTTP/1.0 200 OK
Date: Wednesday, 02-Feb-94 23:04:12 GMT
Server: Apache/0.8.8
MIME-version: 1.0
Last-modified: Monday, 15-Nov-93 23:33:16 GMT
Content-type: text/html
Content-length: 2345
  * línea en blanco *
<HTML><HEAD><TITLE>...
```

En este mensaje el servidor está de acuerdo en utilizar HTTP versión 1.0 para comunicación y envía el estado 200 indicando que ha sido procesada exitosamente la petición del cliente. Entonces envía la fecha y se identifica a sí mismo como un servidor HTTP Apache. También indica que está utilizando MIME versión 1.0 para describir la información que está enviando e incluye información MIME-tipo acerca de lo que se va a enviar en la cabecera "Content-type". Finalmente, envía el número de caracteres que va a mandar, seguido de una línea en blanco y de los mismos datos.

Tanto la cabecera del cliente como del servidor son RFC 822 obedeciendo a mail headers

Un cliente puede enviar cualquier número de campos Accept : el servidor y headers está en espera para convertir los datos en una forma que el cliente pueda aceptar.

Estos datos son los que hacen posible establecer una comunicación entre el cliente y el servidor para que se pueda ejecutar correctamente la aplicación CGI. Los datos que son enviados por el cliente son pasados por medio del servidor HTTP al programa ejecutable que procesará los datos de la forma, este realiza los procedimientos establecidos y genera un documento HTML con el formato descrito. Como ya se mencionó se puede realizar aplicaciones en diferentes lenguajes, en este caso se utiliza C como lenguaje de desarrollo, a continuación se muestran rutinas para poder captar y

manipular los datos descritos hasta ahora, las cuales son proporcionadas por NCSA, se puede consultar la siguiente dirección <http://www.ncsa.uiuc.edu>, la función principal esta modificada con respecto a la original, para desplegar los datos de otra forma (estas rutinas se utilizaron en estaciones de trabajo con sistema operativo solaris 2.x y compilados con C versión 2.0).

El siguiente programa es un programa de propósito general, que independiente de la forma y los elementos que contenga, despliega el nombre de cada elemento, así como el valor que el usuario haya dado a cada elemento de la forma.

El programa procesa formas de tipo *POST*, pero se puede modificar para procesar formas enviadas con el método *GET*.

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

#define LF 10
#define CR 13

#define MAX_ENTRIES 10000

typedef struct {
    char *name;
    char *val;
} entry;

char *makeword(char *line, char stop);
char *fmakeword(FILE *f, char stop, int *len);
char x2c(char *what);
void unescape_url(char *url);
void plusospace(char *str);

/* Función principal */

main(int argc, char *argv[]) {
    entry entries[MAX_ENTRIES];
    register int x,m=0;
    int cl;

    printf("Content-type: text/html%c%c",10,10);

    if(strcmp(getenv("REQUEST_METHOD"),"POST")) {
        printf("Este script debe ser referenciado con el METODO POST \n");
        exit(1);
    }
}
```

```

if(strcmp(getenv("CONTENT_TYPE"),"application/x-www-form-urlencoded")) {
    printf("Este script se utiliza solo para decodificar datos de formas \n");
    exit(1);
}
cl = atoi(getenv("CONTENT_LENGTH"));

for(x=0;cl && (!feof(stdin));x++) {
    m=x;
    entries[x].val = fmakeword(stdin,'&",&cl);
    plusospace(entries[x].val);
    unescape_url(entries[x].val);
    entries[x].name = makeword(entries[x].val,'=');
}

printf("<H1>Resultados de la FORMA</H1>");
printf("Usted introdujo la siguiente informaci&oacute;n:<p>%c",10);

printf("<ul>%c",10);

for(x=0; x <= m; x++)
    printf("<li> <code>%s = %s</code>%c",entries[x].name,
        entries[x].val,10);
printf("</ul>%c",10);
}

/-----/
/*
/* Rutinas de soporte para la manipulaci&oacute;n de informaci&oacute;n de una forma en lenguaje C, */
/* desarrolladas por NCSA (National Center for Supercomputing Applications). */
/*
/-----/

void getword(char *word, char *line, char stop) {
    int x = 0,y;

    for(x=0;((line[x]) && (line[x] != stop));x++)
        word[x] = line[x];

    word[x] = '\0';
    if((line[x]) ++x;
    y=0;

    while(line[y++] = line[x++]);
}

```

```

char *makeword(char *line, char stop) {
    int x = 0, y;
    char *word = (char *) malloc(sizeof(char) * (strlen(line) + 1));

    for(x=0; ((line[x]) && (line[x] != stop)); x++)
        word[x] = line[x];

    word[x] = '\0';
    if(!line[x]) ++x;
    y=0;

    while(line[y++] = line[x++]);
    return word;
}

char *fmakeword(FILE *f, char stop, int *cl) {
    int wsize;
    char *word;
    int ll;

    wsize = 102400;
    ll=0;
    word = (char *) malloc(sizeof(char) * (wsize + 1));

    while(1) {
        word[ll] = (char)fgetc(f);
        if(ll==wsize) {
            word[ll+1] = '\0';
            wsize+=102400;
            word = (char *)realloc(word, sizeof(char)*(wsize+1));
        }
        --(*cl);
        if((word[ll] == stop) || (feof(f) || !(*cl))) {
            if(word[ll] != stop) ll++;
            word[ll] = '\0';
            return word;
        }
        ++ll;
    }
}

char x2c(char *what) {
    register char digit;

    digit = (what[0] >= 'A' ? ((what[0] & 0xdf) - 'A') + 10 : (what[0] - '0'));
}

```



```

digit *= 16;
digit += (what[1] >= 'A' ? ((what[1] & 0xdf) - 'A') + 10 : (what[1] - '0'));
return(digit);
}
void unescape_url(char *url) {
register int x,y;

for(x=0,y=0;url[y];++x,++y) {
if((url[x] = url[y]) == '%') {
url[x] = x2c(&url[y+1]);
y+=2;
}
}
url[x] = '\0';
}

void plusospace(char *str) {
register int x;

for(x=0;str[x];x++) if(str[x] == '+') str[x] = ' ';
}

int rind(char *s, char c) {
register int x;

for(x=strlen(s) - 1; x != -1; x--)
if(s[x] == c) return x;
return -1;
}

int getline(char *s, int n, FILE *f) {
register int i=0;

while(1) {
s[i] = (char)fgetc(f);

if(s[i] == CR)
s[i] = fgetc(f);

if((s[i] == 0x4) || (s[i] == LF) || (i == (n-1))) {
s[i] = '\0';
return (feof(f) ? 1 : 0);
}
}
}
}

```

```

void send_fd(FILE *f, FILE *fd)
{
    int num_chars=0;
    char c;

    while (1) {
        c = fgetc(f);
        if(!feof(f))
            return;
        fputc(c,fd);
    }
}

int ind(char *s, char c) {
    register int x;

    for(x=0;s[x];x++)
        if(s[x] == c) return x;

    return -1;
}

void escape_shell_cmd(char *cmd) {
    register int x,y,l;

    l=strlen(cmd);
    for(x=0;cmd[x];x++) {
        if(ind("&","|\"?~<>^()[]{}$\\",cmd[x]) != -1){
            for(y=l+1;y>x;y--)
                cmd[y] = cmd[y-1];
            l++; /* length has been increased */
            cmd[x] = '\\';
            x++; /* skip the character */
        }
    }
}

```

Este programa se debe compilar en la computadora donde esta instalado el servidor HTTP, una vez generado el archivo ejecutable, éste debe estar en el área definida para que residan las aplicaciones CGI en muchos casos en el directorio *cgi-bin*.

El funcionamiento de este programa se basa en los datos que el servidor HTTP pasa a éste por medio de la entrada estándar (stdin), debido a que la información que recibe el servidor viene en la estructura antes descrita (versión HTTP, que MIME se utiliza, los

parámetros de la forma etc.), se tiene que realizar un barrido de toda esta información para poder conocer los parámetros de la forma así como sus valores, esto lo realizan en conjunto las funciones `makeword`, `fmakeword`, `x2c`, `unescape_url`, `plustospace`, las cuales se encarga de recibir la cadena completa de los datos de la forma y separar cada elemento (con su respectivo nombre) y el valor de cada uno de ellos, una vez que se tienen estos datos, lo que realiza la función principal es elaborar un documento HTML con su estructura y enviarlo por la salida estándar (`stdout`), el cual será desplegado por el browser cliente.

Este procedimiento solo muestra la forma en que se pueden recibir y enviar datos con una aplicación CGI, sin embargo se pueden realizar aplicaciones más complejas con estas rutinas, ya que una vez obtenidos los datos, sobre éstos o a partir de estos, se pueden ejecutar procesos mas complejos (como la consulta de una base de datos, realizar algún cálculo, modificar la visualización de alguna página, realizar algún proceso estadístico etc.).

## 6.2 Etiquetas HTML vinculadas con la formas

Un conjunto de etiquetas agregadas al lenguaje HTML sirven para dirigir un cliente WWW para desplegar una forma para que sea llenada por el usuario y entonces enviar la colección de datos a un servidor HTTP especificado en la forma.

Los servidores fueron modificados para que pudieran iniciar un programa CGI especificado en la forma y pasar la colección de datos al programa, el cual podrá realizar algún proceso, preparar una respuesta (siendo lo mas probable por consulta de una base de datos existente) y regresar un documento HTML al usuario

Las etiquetas agregadas al HTML, para las formas HTML son:

`<FORM> ... </FORM>`

*Define una forma de entrada*

*Atributos: ACTION, METHOD, ENCTYPE*

`<INPUT>`

*Define un campo de entrada*

*Atributos: NAME, TYPE, VALUE, CHECKED, SIZE, MAXLENGTH*

**<SELECT>... </SELECT>**

*Define una lista de selección*

*Atributos: NAME, MULTIPLE, SIZE*

**<OPTION>**

*Define las selecciones de una lista de selección (dentro de un SELECT)*

*Atributos: SELECT*

**<TEXTAREA>... </TEXTAREA>**

*Define una ventana para entrada de texto*

*Atributos NAME, ROWS, COLS*

**Tipos de datos de entrada**

**TEXT**

*Para ingresar una sola línea de texto. El atributo tamaño puede ser utilizado para especificar el área visible del campo. El atributo MAX es usa para especificar el máximo número de caracteres que pueden aceptarse.*

**INT**

*Para ingresar enteros. El máximo número de dígitos se especifica con el atributo SIZE.*

**CHECKBOX**

*Para variables booleanas, o variables que pueden tomar múltiples valores al mismo tiempo, cuando una opción es seleccionada, el valor especificado en el atributo VALUE es asignado a la variable especificada con el atributo NAME.*

**RADIO**

*Para variables que sólo pueden tomar un valor de un conjunto de alternativas. Si varios radio botones tienen el mismo nombre (atributo NAME), seleccionando uno de los botones, desactivará cualquier otro que anteriormente halla sido seleccionado.*

### **SUBMIT**

*Crea un botón con un texto, seleccionándolo envía la forma al servidor.*

### **RESET**

*Genera un botón con algún texto, su función es la de regresar a los valores iniciales de la forma cuando es seleccionado.*

### **HIDDEN**

*Se utiliza para pasar información de una forma a otra o de un script a otro. Un dato de este tipo no se visualiza en la forma per el valor especificado en el atributo *VALUE* será pasado como un parámetro mas.*

## **6.3 Método POST y GET**

Existen dos formas de enviar los datos de la forma al servidor, especificándola a través del atributo *ACTION* de la etiqueta *FORM* son el método *POST* Y *GET* :

### **6.3.1 Método POST**

El método *POST* envía los datos de la forma con la siguiente estructura:

Tipo de método utilizado URL y programa a ejecutar
Elementos MIME aceptados
Parámetros de identificación del Cliente
Línea en blanco
Datos de la Forma

Utilizando el método *POST* envía los datos de la forma en último lugar y a su vez son enviados en el orden que fueron declarados en el documento HTML y van en parejas, cada par por separado, primero el nombre de la variable y en seguida su valor separados por un signo igual (=) los caracteres que no son alfanuméricos son remplazados por su equivalente valor en hexadecimal.

Este método es utilizado cuando los datos que el cliente recibe son dinámicos (que su valor cambia), por ejemplo cuando se requieren obtener parámetros para ejecutar un proceso en base a esos datos.

### 6.3.2 Método GET

El método *GET* tiene la siguiente estructura :

Tipo de método utilizado URL, programa a ejecutar y Datos de la Forma
Elementos MIME aceptados
Parámetros de identificación del Cliente
Línea en blanco

Con el método *GET* los datos son enviados como parte del URL al que se envía la forma. Los datos también se pasan por parejas, de la misma forma que en el método *POST* (variable=valor), pero todos se envían juntos en una sola línea.

El método *GET* se utiliza cuando los datos de la forma son estáticos (los valores de las variables no se modifican), o cuando los datos que se van a enviar se mandan junto con el URL, es muy utilizado cuando se requiere ejecutar un proceso que no requiere de parámetros que sean proporcionados por el usuarios, como programas que despliegan la hora o fecha actual, o cuando se hacer referencia a algún URL con algunos parámetros.

## 6.4 Variables de ambiente del CGI

SERVER\_SOFTWARE=nombre\_servidor/versión. El nombre y versión del software servidor que ejecutará el CGI. El formato es nombre/versión.

SERVER\_NAME= domain.name. El nombre del dominio Internet del servidor . Si el nombre de dominio no esta disponible, éste será la dirección IP.

GATEWAY\_INTERFACE=CGI/versión. La versión de CGI utilizada por el servidor HTTP.

**SERVER\_PROTOCOL**=HTTP/versión. El protocolo HTTP utilizado y la versión. Esto permite que los programas que soporten diferentes versiones de protocolo HTTP, o que soporten múltiples protocolos (por ejemplo Gopher y HTTP), ejecuten el proceso adecuado para cada protocolo.

**SERVER\_PORT** = número. El número del puerto llamado por el cliente.

**REQUEST\_METHOD**=HTTP\_método. El método asociado con la petición del cliente, por ejemplo GET o POST. Se puede diseñar una sola aplicación CGI, que soporte diferentes métodos y ejecutando determinado proceso, dependiendo del valor de ésta variable.

**PATH\_INFO**=extra\_path. Contiene cualquier información extra encontrada en el URL.

**PATH\_TRANSLATED**=/transl/extra\_path. La parte de **PATH\_INFO** como una ruta absoluta en el sistema local. Por ejemplo si la máquina local almacena los archivos HTML en el directorio /usr/servidor/docu\_html y **PATH\_INFO** contiene /dir/archivo. **PATH\_TRANSLATED** contendrá : /usr/servidor/docu\_html/dir/archivo. La ruta no se refiere a la ubicación del CGI.

**SCRIPT\_NAME**=/path/script\_name. La ruta y el nombre del script que será ejecutado que se referenció a través de un URL. Esto puede ser utilizado para construir un URL del servidor actual, que haga referencia a otra aplicación CGI, a partir de una anterior

**QUERY\_STRING**=query\_string. La cadena de petición que sigue después del signo de interrogación en un URL, los espacios son sustituidos por un signo mas "+", los valores de las variables de la forma son separados por un ampersand "&" y se encuentran en pares (nombre=valor)

**REMOTE\_HOST**=client.domain.name. el nombre y dominio Internet del cliente que realiza la petición. Si el nombre del dominio no está disponible, éste queda en blanco.

**REMOTE\_ADDR**=xxx.xxx.xxx.xxx. La dirección numérica IP del cliente que accesa al servidor.

**AUTH\_TYPE**=tipo. El método de autenticación requerido para verificar la autenticidad del cliente que intenta acceder al servidor. Solamente es utilizado con scripts que son protegidos. Si la autenticación no se requiere la variable esta vacía.

**REMOTE\_USR**=nombre . El nombre de autenticidad del cliente. Esta variable es fijada, solo cuando la autenticación es requerida . Se usa para registro del nombre de usuario y para control de acceso dependiendo de la identidad del cliente. La variable esta vacía si no se requiere método de autenticación.

**REMOTE\_IDENT**=nombre. El nombre del cliente obtenido por el servidor usando el demonio de identificación **identd**.

**CONTENT\_TYPE**=MIME\_tipo. Si el cliente realiza una petición por el método POST, esta variable contiene el tipo MIME de los datos. Si no se envían datos, la variable esta vacía. Los datos mismos están disponibles para la aplicación CGI a través de la entrada estándar. El único tipo ampliamente utilizado es `application/x-www-form-urlencoded`.

**CONTENT\_LENGTH**=longitud. Si el cliente envía los datos por el método POST, esta variable contiene la longitud de los datos. Si no se envía ningún dato, la variable está vacía. La aplicación CGI no tiene que leer todos los datos necesariamente, antes de regresar alguna respuesta al cliente, o antes de salir.

Un ejemplo de una forma que realice una petición a un servidor para que ejecute una aplicación CGI es el siguiente, se utilizó el programa en C antes descrito el código HTML de la forma es el siguiente.

```
<html>
<head>
<title>Muestra de CGI</title>
</head>
<body text="#000000" link="#0000FF" vlink="#FF00FF" alink="#800000">
<h2>Ejemplo de CGI con el m&ecacute;todo <b>POST</b></h2>
<form action="http://tesiuam.dgbiblio.unam.mx:8000/cgi-bin/cgi" method="POST">
<p>Ingresar texto: <input type="text" size=50 maxlength=256 name="Texto"></P>
<P>Caja de opciones: <select name="caja_opciones" size=1>
    <option>Opción A</option>
    <option>Opción B</option>
    <option>Opción C</option>
</select></p>
```



<p>Cajas de seleccion:

```
Selecci&oacute;n 1 <input type=checkbox name="Caja_1"> Selecci&oacute;n 2
<input type=checkbox name="Caja_2"></p>
```

<p>Selecci&oacute;n o boton de radio: </p>

```
Opci&oacute;n 1 <input type=radio name="boton_1"> Opci&oacute;n 2 <input
type=radio name="boton_2"> Opci&oacute;n 3 <input type=radio name="boton_3">
```

```
<p><input type=submit value="Enviar datos"> <input type=reset value="Borrar
datos"></p>
```

```
</form>
```

```
</body>
```

```
</html>
```

El aspecto de este HTML se muestra en la siguiente figura 6.3

En la línea de código `<form action="http://tesiunam.dgbiblio.unam.mx:8000/cgi-bin/cgi" method="POST">`, se especifica el parámetro del URL y el programa CGI que se ejecutará y el parámetro del método que se utiliza, en este caso el método POST. El resultado del CGI una vez que se selecciona el botón Enviar se muestra en la figura 6.4.

Lo que el programa CGI realizó, es reflejar los datos ingresados por el usuario, no les aplico ningún proceso, sin embargo los decodificó y obtuvo tanto su valor, como el nombre de cada elemento.

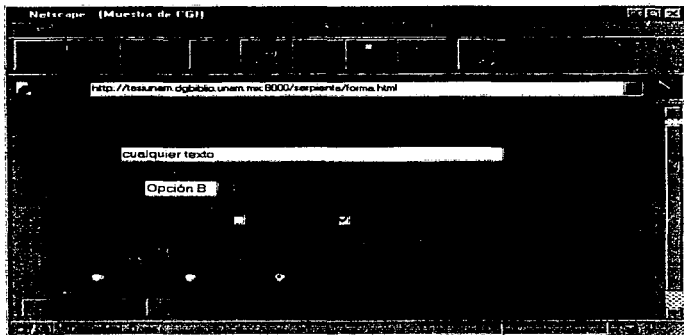


Figura 6.3 Campos de una forma

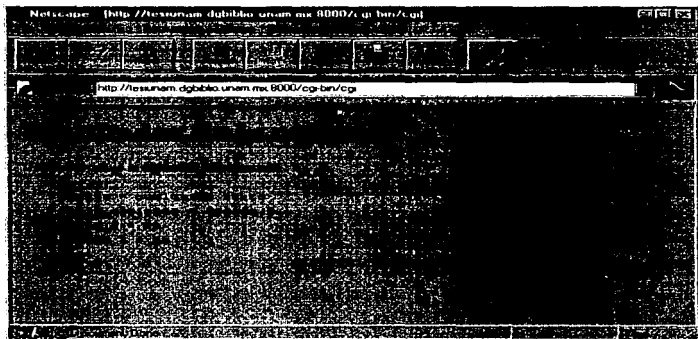


Figura 6.4 Resultados de la forma enviada



# Capítulo 77



## TENDENCIAS EN EL WWW

Este capítulo describe las tendencias que empezaron a surgir, durante el transcurso de la realización de este trabajo, no se detallará a fondo, solamente se presentan los aspectos más importantes de dichas tendencias y los antecedentes de las mismas, en algunos casos.

### 7.1 Animación de imágenes en formato GIF

Una de las restricciones que se tienen en el formato HTML, es que aunque permite mostrar imágenes, no me permite presentar una animación en tiempo real en la misma página, esto se puede conseguir con la inclusión de algún vídeo, pero se tiene la desventaja de que se presenta como una aplicación separada del documento HTML, sin embargo, el formato GIF, tiene la posibilidad de almacenar una secuencia de imágenes en un solo archivo y presentarlas consecutivamente en un *browser*, las imágenes pueden ser secuencias de alguna animación, además dentro del archivo gráfico, se almacena información acerca del tiempo en que se irán cambiando las imágenes así como el número de veces que se repetirá la presentación de todas las imágenes.

Con lo anterior se consigue presentar animaciones en una página, sin la necesidad de incluir algún vídeo, además la calidad de las imágenes, depende de la resolución que se les halla dado y el software con que se realizaron y en el documento HTML, sólo se incluye dicha imagen con la sintaxis ordinaria, el efecto visual en la página será el de una animación, la transparencia de ésta también depende de la cantidad de imágenes que se tengan.

Existe software en Internet, que se distribuye como *shareware*, que realiza lo anterior algunos de ellos se encuentra en las siguientes direcciones electrónicas

⇒ <http://www.mindworkshop.com/alchemy/gifcon.html>

⇒ <http://www.fastlane.net/~samici>

## 7.2 Aplicaciones JAVA

Podría decirse que después del impacto que tuvo Mosaic en sus inicios, Java actualmente tiene un impacto equiparable. Java es un lenguaje de programación, con el cual se pueden diseñar aplicaciones que pueden ejecutarse en varias plataformas (Windows 3.x, Windows NT, UNIX con X-Windows o Motif, Windows 95), dichas aplicaciones están orientadas al WWW y el HTML, algunas de las características de las aplicaciones Java son las siguientes:

- Ejecutar sonidos en línea en tiempo real, cuando algún usuario acceda a la página que lo contiene.
- Ejecutar música de fondo durante la consulta de una página
- Animaciones
- Video en tiempo real
- Ejecutar operaciones que afectan el estado de la página en tiempo real (como cambiar el tipo, tamaño o color del texto; modificar colores de fondo, etc.)
- Dividir el área de visualización en el *browser* en varias áreas (frames)
- Ejecutar validación de información en tiempo real.

Las características descritas, son ejecutadas por los *browsers* sólo si estos soportan dichas aplicaciones tales como HotJava, Internet Explorer o Netscape (a partir de la versión 2.0).

Independientemente de lo anterior, Java es un lenguaje de programación para aplicaciones distribuidas. No solamente permite agregar nuevos tipos de contenido a las páginas, como lo hacen Netscape e Internet Explorer (con sus extensiones de etiquetas al HTML estándar), permite agregar tanto el contenido, como el código necesario para interactuar con dicho contenido. La ventaja que se proporciona con esta forma de programación es la de tener independencia de las aplicaciones respecto del *browser*, ya que no importa la versión del *browser* que se este utilizando o el protocolo, ya que se envía tanto el contenido como el código para que la aplicación pueda ejecutarse.

- Otra de las ventajas de Java es en cuanto a interacción, si se tiene una base de datos, la cual se quiere poner en el WWW, se requiere de una aplicación CGI que se ejecute por lo general en ambiente UNIX, para poder interpretar los datos que nos proporciona el usuario y de esa forma a través del CGI realizar algún proceso para

poder regresar al usuario la información de la base de datos, sin embargo si la base de datos se encuentra en un *mainframe*, que no interprete HTTP, resulta más difícil el acceso a los datos, ya que se requiere un proceso intermedio para poder utilizar el protocolo HTTP. Con Java se evita lo anterior, enviando el código, para que el usuario pueda interactuar directamente con la base de datos, y no necesariamente se tenga que utilizar el puerto 80 para tener acceso a la información.

Además una aplicación Java se ejecuta de igual forma en cualquier arquitectura en la que se encuentre un *browser* que soporte Java, por ejemplo Netscape 2.0 el cual esta disponible para Windows 95, Windows NT, MacOS, Sun OS 4.1.3, SGI-IRIX, OSF/1, HP-UX.

Java no solo es para crear aplicaciones WWW, es un lenguaje de programación de propósito general como Fortran o C++, pero sin algunos de los problemas que estos lenguajes tiene. Las características de Java como lenguaje son las siguientes:

- Es simple en su estructura sintáctica, de modo que no se tiene que agregar una gran conjunto de características adicionales.
- Es Orientado a Objetos. Casi todo en Java es una clase, un método o un objeto. Solamente los elementos básicos no lo son (operadores y tipos de datos).
- Es Independiente de la Plataforma. Los programas en Java son compilados a un formato de código-byte, que puede ser leído y ejecutado en diversas plataformas.
- Es Seguro. Puede ejecutarse en ambientes que son propicios para introducción de virus.
- Alto desempeño. Java puede ser compilado a velocidades cercanas a las de C++.
- Multitarea . un solo programa en Java puede tener varios y diferentes procesos ejecutándose independiente y continuamente.

La dirección electrónica donde se puede obtener más información sobre esta herramienta es : <http://www.javasoft.com/nav/whatis/index.html>

### 7.3 Perl

Perl (*Practical Extraction and Report Language*) es un lenguaje que abigarras varias características de otros programas, por UNIX, tales como: awk, sed, tr, BASIC y C, también es conocido como un lenguaje de tipo *script*. Perl no compila los programas,

ya que un programa escrito en este lenguaje, no genera un ejecutable. Perl en realidad es un intérprete, el procedimiento que realiza con un programa es: leer el programa, verificarlo sintácticamente y ejecutar cada una de las instrucciones, las velocidades que alcanza en la ejecución de programas es cercana a las de C, el procedimiento anterior es repetido cada vez que se ejecute el programa.

La sintaxis de Perl es muy parecida a la sintaxis que se utiliza en los shell de UNIX, otra característica de Perl es la facilidad que provee para utilizar bases de datos de tipo xbase, dispone de rutinas de nivel de software 4GL..

La popularidad que actualmente tiene Perl en Internet es debido a la capacidad que se tiene de aplicarlo en el diseño de formas para las páginas de *Web*, obtener estadísticas, manejar catálogos y bases de datos en los servidores de WWW.

Para lo anterior, actualmente se han diseñado librerías, las cuales permiten tener un mejor control y facilidad de uso en las aplicaciones CGI por medio de Perl, estas librerías ofrecen funciones altamente especializadas en la descodificación de datos enviados por los usuarios a través de formas, con estas rutinas el manejo de esa información así como su procesamiento se facilita en gran medida. Perl actualmente existe para plataformas UNIX y para DOS.

Si se desea tener más información sobre este lenguaje de programación se recomienda consultar la siguiente dirección electrónica: <http://www.perl.com/perl/index.html>

## 7.4 Realidad Virtual (VR)

Otra tecnología que está surgiendo actualmente es la de la Realidad Virtual conocido como VR en el WWW, es una aplicación compleja, ya que intenta simular ambientes, lugares, de manera que el usuario tenga una sensación de espacio, de dimensionalidad, sin embargo actualmente no se tiene un estándar para estas aplicaciones, además que se requiere un software adicional para el *browser* de manera que este pueda interpretar la aplicación VR, sin embargo actualmente no existe compatibilidad entre los diferentes estándares de VR.

La característica de estas aplicaciones es la de presentar en el *browser* (con su respectivo software para VR) imágenes tridimensionales, las cuales pueden ser vistas desde diferentes perspectivas, de manera que el usuario puede llegar, por ejemplo a una



ciudad y poder "caminar" a través de sus calles y entrar en alguno de los edificios, cambiando de imágenes.

Desde luego estas aplicaciones aún cuando tienen gran calidad gráfica, todavía no son lo suficientemente asequibles como para que se integren como un estándar, ya que el tipo y cantidad de imágenes, así como la complejidad que se tiene para realizar un determinado ambiente, actualmente requieren de software y equipo de cómputo sofisticado para poder realizarlas, además que la cantidad de información que se requiere transmitir del servidor al cliente en el WWW es grande, las aplicaciones de VR son lentas en su ejecución, ya que con cada cambio de escenario, se requiere de un nuevo conjunto de imágenes, así como de detectar las diferentes perspectivas y opciones válidas para ese nuevo escenario.

Sin embargo con el avance en cuanto a la transferencia de información, la aparición de nuevas características en los *browsers* y el surgimiento de nuevas técnicas para la tecnología de VR en el WWW, se están teniendo cada vez más aplicaciones de este tipo.

Para obtener mayor información sobre el software, las compañías, los proyectos y las organizaciones que desarrollan la VR en el WWW, se recomienda consultar la siguiente dirección electrónica:

[http://www.yahoo.com/Computers\\_and\\_Internet/Multimedia/Virtual\\_Reality/](http://www.yahoo.com/Computers_and_Internet/Multimedia/Virtual_Reality/)



# Capítulo 8

## Capítulo 8

## El *Web* de la DGB

En este capítulo se hará una breve descripción del trabajo realizado en el desarrollo e implantación del *Web* de la Dirección General de Bibliotecas de la UNAM, a la cual en lo sucesivo nos referiremos como la DGB, se encuentra ubicada en el edificio de la Biblioteca Central, y entre sus objetivos y funciones principales se encuentra la de "utilizar la tecnología adecuada en cada caso"<sup>1</sup>.

Tomando en consideración este objetivo a inicios del año de 1995 se dio a la tarea de investigar algún tipo de tecnología que fuese capaz de dar a conocer a la comunidad universitaria algunos de los servicios que se proporcionan en la DGB, así como en la Biblioteca Central en una forma sencilla.

El resultado de esta investigación fue el empleo del *gopher*, que a través de su estructura de menús jerárquicos permitía mostrar en forma relativamente sencilla algunos de los servicios como por ejemplo la consulta de un catálogo de acceso público en línea (OPAC).

A pesar de que el *gopher* en su momento resolvía algunas de las necesidades, resultaba poco atractivo para dar a conocer algunos nuevos servicios, que requerían simplificar los procedimientos de captura de la información. En el segundo semestre de 1995 se decidió crear el *Web* de la DGB (<http://www.dgbiblio.unam.mx>) con el objetivo de mostrar una nueva imagen de estos servicios, además de seguir considerando el objetivo principal anteriormente mencionado; para ello se dispuso de una computadora SUN Spare server 10 con 32 MB en RAM, 1GB en disco duro y sistema operativo Solaris 2.4 y a finales del mismo año el *Web* de la DGB en su primera versión ya se encontraba en funcionamiento con los servicios de:

- Consulta al catálogo bibliográfico de la UNAM.
- Directorio interno de servicios de la DGB.

<sup>1</sup> Universidad Nacional Autónoma de México. Dirección General de Bibliotecas. In Dirección General de Bibliotecas y la Biblioteca Central de la UNAM: historia, organización y servicios. p.4

- Alerta hemerográfica (consulta de la tabla de contenidos de las publicaciones periódicas de reciente adquisición).

En seguida se explicará en forma breve el archivo `index.html`, que es el archivo que contiene la estructura de la página principal, así como el empleo del CGI, que permite el empleo de la programación para obtener resultados de las búsquedas en forma dinámica..

## 8.1 Creación de las páginas

El archivo `index.html` contiene el código de la página principal, las figuras 8.1 y 8.2 muestran la entrada a dicho *Web*. Contiene archivos gráficos digitalizados y otros diseñados con Corel Draw; además de contener dos formas de hacer una liga a otro documento, es decir, con hipertexto y a través de una imagen.

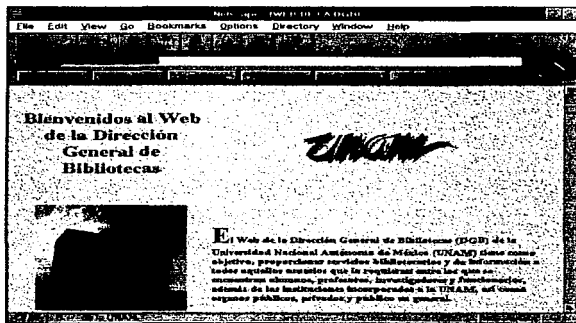


Figura 8.1 Primera pantalla de la Página principal del *Web* de la DGB.

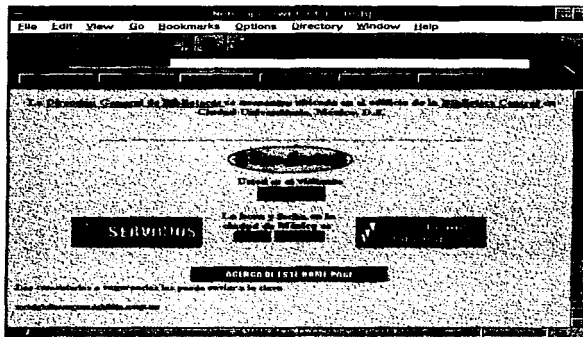


Figura 8.2 Segunda Pantalla de la Página principal del Web de la DGB.

El código fuente del archivo index.html describe lo mostrado en las figuras 8.1 y 8.2.

```
<HTML>
<HEAD><TITLE>WEB DE LA DGB</TITLE>
<SCRIPT LANGUAGE="JavaScript" *
<!-- Beginning of JavaScript Applet -----
function scrollit_r2l(seed)
{
  var m1 = "Bienvenidos al Web de la D. G. B. ";
  var m2 = " de la UNAM ";

  var msg=m1+m2;
  var out = " ";
  var c = 1;

  if(seed > 100) {
    seed--;
    var cmd="scrollit_r2l(" + seed + ")";
    timerTwo=window.setTimeout(cmd,100);
  }
  else if (seed <= 100 && seed > 0) {
```

```

    for (c=0 ; c < seed ; c++) {
        out+=" ";
    }
    out+=msg;
    seed--;
    var cmd="scrollit_r2l(" + seed + ")";
    window.status=out;
    timerTwo=window.setTimeout(cmd,100);
}
else if (seed <= 0) {
    if (-seed < msg.length) {
        out+=msg.substring(-seed,msg.length);
        seed--;
        var cmd="scrollit_r2l(" + seed + ")";
        window.status=out;
        timerTwo=window.setTimeout(cmd,100);
    }
    else {
        window.status=" ";
        timerTwo=window.setTimeout("scrollit_r2l(100)",75);
    }
}
}
// -- End of JavaScript code ----- -->
</SCRIPT>
</HEAD>
<BODY onLoad="timerONE=window.setTimeout('scrollit_r2l(100)',500);"
BACKGROUND="uff.jpg" TEXT="#00000 VLINK=#700050 ALINK=#00FFFF"
<TABLE BORDER=0 cellspacing=1 cellpadding=1 >
<TR><th colspan=3>
<H1>Bienvenidos al Web de la Dirección General de Bibliotecas</H1>
</th><th colspan=1></th><th colspan=3>
<IMG SRC="unam2.gif" align=center alt="" hspace=5 vspace=25 border=0>
</th><th colspan=3></th></TR>
<TR><th colspan=10><BR><BR></th></TR>
<TR><th colspan=4>
<CENTER><IMG SRC="biblio4.gif" width=230 height=230> </CENTER> </th>
<th colspan=3>
<BR><BR>
<H3 align="left"><Font size="+6">E</font>1 Web de la Dirección General de Bibliotecas (DGB) de la
Universidad Nacional Autónoma de México (UNAM) tiene como objetivo, proporcionar servicios
bibliotecarios y de información a todos aquellos usuarios que la requieran entre los que se encuentran
alumnos, profesores, investigadores y funcionarios, además de las instituciones incorporadas a la UNAM,
así como órganos públicos, privados y público en general. </H3></th></TR>
<BR><BR>
<TR><th colspan=10>
<BR>
<BR>
<H3 align="center">La <A HREF="seconhp.html">Dirección General de Bibliotecas </A> se encuentra
ubicada en el edificio de la <A HREF="histbib.html">Biblioteca Central </A> en Ciudad Universitaria,
México, D.F.</H3></P><BR></th></TR>

```

```

<BR>
<TR><th colspan=10>
<th size=7 width=550>
<A HREF="/publicdgb/catalres.html"><IMG SRC="news.gif" border=0 height=56 width=180></A>
<BR>
</th></TR>
<TR><th colspan=10>
<CENTER><H3>Usted es el visitante
<BR><IMG SRC="/cgi-bin/Count.cgi?dd=Djft-6&frgb=255;215;0(df=sample.dat"
ALIGN=absmiddle></H3></CENTER>
</th></TR>
<TR><th colspan=3>
<A HREF="servicios.html"><IMG SRC="esc.gif" align=right alt="" hspace=20 vspace=10
border=0></A></th>
<th colspan=3>
<CENTER><H3>La hora y fecha en la ciudad de M&eacute;xico es
<BR><IMG SRC="/cgi-bin/Count.cgi?dd=Djft-6&frgb=255;215;0(df=sample.dat;display=clock"
ALIGN=absmidd
le>
<IMG SRC="/cgi-bin/Count.cgi?dd=Djft-6&frgb=255;215;0(df=sample.dat;display=date"
ALIGN=absmidd
le><H3></CENTER>
</th>
<th colspan=3>
<A HREF="trihp.html"><IMG SRC="cat.gif" align=left alt="" hspace=20 vspace=10
border=0></A></th></TR>
<TR><th colspan=3></th>
<th colspan=4></th></TR>
<TR><th colspan=2><BR></th></TR>
<TR><th colspan=10>
<CENTER><A HREF="credit.html"><IMG SRC="aboutweb.gif"></A></CENTER>
</th></TR>
<TR><th colspan=10>
<H4 align="left">Los comentarios o sugerencias los puede enviar a la clave</H4>
</th></TR>
<TR><th colspan=10>
<AHREF="mailto:wwwdgb@panoramx.dgbiblio.unam.mx"><H5
align="left">wwwdgb@panoramx.dgbiblio.unam.mx</H5></A>
</th></TR>
<applet code="Dgclock.class" width=100 height=30>
<param name="TZ" value="GMT-0555">
</applet>
</TABLE>
</BODY>
</HTML>

```

Como se puede ver en la parte superior del listado existen las etiquetas de estructura ( <HTML> , <HEAD> y <TITLE>) que son parte fundamental de un archivo HTML; posteriormente aparece un código que es adicional a este lenguaje,

localizado entre las etiquetas `<SCRIPT LANGUAGE="JavaScript">` y `</SCRIPT>`. Este código es una aplicación Java, que sale de los alcances de este trabajo y con lo descrito en el capítulo anterior sobre las tendencias en el WWW, nos limitamos a decir que la salida de esta aplicación se observa en la parte del estatus de la ventana del visualizador y despliega el mensaje "Bienvenidos al Web de la D.G.B. de la UNAM" .

Posteriormente se encuentran las etiquetas que definen el cuerpo del documento, las extensiones de etiquetas introducidas por Netscape y las etiquetas que dan formato a las tablas (`<TABLE BORDER=#0 cellspacing=#1 cellpadding=#1 >`).

La etiqueta `<BACKGROUND="ruff.jpg" TEXT=#00000 VLINK=#700050 ALINK=#00FFFF>` define la imagen que servirá como fondo de la página y los atributos que tendrá el texto, así como los atributos de las ligas. Posteriormente aparece la etiqueta que colocará el título de la página con el tipo de letra más grande definida entre las etiquetas `<H1>` y `</H1>`.

La etiqueta `<IMG SRC="biblio4.gif" width=230 height=230 >` desplegará el archivo gráfico llamado `biblio4.gif` que corresponde a la imagen de la Biblioteca Central con los parámetros de ancho y alto.

Un ejemplo de una liga dentro del archivo `index.html` se encuentra en la siguiente línea:

La `<A HREF="seconhnp.html">Dirección General de Bibliotecas </A>` se encuentra... donde se pueden apreciar las etiquetas de hipertexto hacia el archivo "seconhnp.html".

En las siguientes líneas de código se hace una llamada a una aplicación CGI, la cual mostrará el número de accesos registrados hasta el momento por medio de una imagen:

```
<CENTER><H3>Usted es el visitante
<BR><IMG SRC="..cgi-bin/Count.cgi?dd=Bjft=6&frgb* 255;215;0)df=sample.dat"
ALIGN="absmiddle"></H3></CENTER>
```

En la parte final del código se encuentra ejemplificado el uso del URL *mailto*, como se puede ver en las siguientes líneas de código:

```
<A HREF="mailto:wwwdgb@panoramx.dgbiblio.unam.mx"><H5
align="left">wwwdgb@panoramx.dgbiblio.unam.mx</H5></A>
```



los correos electrónicos serán recibidos en la clave **www.dgb** dentro del servidor **panoramx.dgbiblio.unam.mx**.

Al final del código se encuentran las etiquetas de cierre, tanto del cuerpo del documento como del archivo HTML.

## 8.2 Aplicación de un CGI

El departamento de publicaciones periódicas de la Biblioteca Central, como un servicio más, edita una alerta hemerográfica llamada HEMERUNAM. Su finalidad es mostrar las listas de contenido de las publicaciones periódicas de reciente ingreso al departamento, en las áreas de: Computación, Bibliotecología y Ciencias de la Información.

Lo que se pretende a través del *Web* es proporcionar esta lista, además de integrar en cada uno de los títulos de las revistas, la imagen del índice, la figura 8.3, muestra la página del *Web* de la dicha alerta.

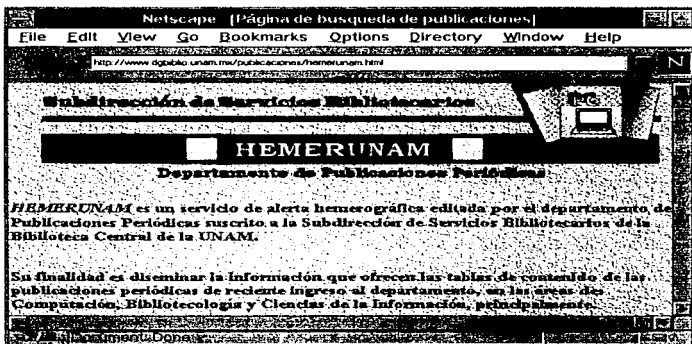


Figura 8.3 Página de Hemerunam (Alerta hemerográfica).

La forma de realizar la consulta a esa información, es proporcionar el mes en que se publicó la revista y/o seleccionar en una lista desplegable la publicación que se quiere consultar, la figura 8.4, muestra lo anterior. Con el botón buscar se envían estos datos al servidor, en donde se realizará una búsqueda en una base de datos la cual contiene los siguientes campos:

- Nombre de la revista.
- Nombre y ruta del archivo del índice de la revista.
- Mes de la publicación.
- Nombre y ruta del archivo que contiene el HTML para esa revista.

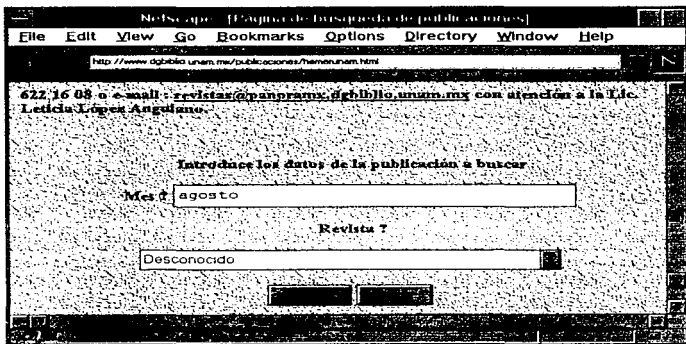


Figura 8.4 Pantalla de captura de la información.

Una vez encontrada la información que proporcionó el usuario, si es el caso, se despliega la información en base a las siguientes situaciones:

- Si el usuario proporciona tanto el mes como la publicación, se regresa esa revista en específico, si esta se encuentra en la base o si la información proporcionada es válida (ya que se pudo haber ingresado un mes no válido para esa publicación).
- Si el usuario ingresa solamente el mes y no selecciona alguna revista en especial, el resultado de la búsqueda es, la información de todas las revistas publicadas en ese mes.
- En el caso de seleccionar solamente la publicación y no proporcionar algún mes, se regresan, todas las publicaciones de la revista seleccionada de todos los meses que contenga la base de datos.

Una vez que el usuario reciba los resultados de la petición realizada, puede seleccionar la o las opciones proporcionadas, según sea el caso. La figura 8.5 muestra la pantalla de búsqueda de la alerta hemerográfica.

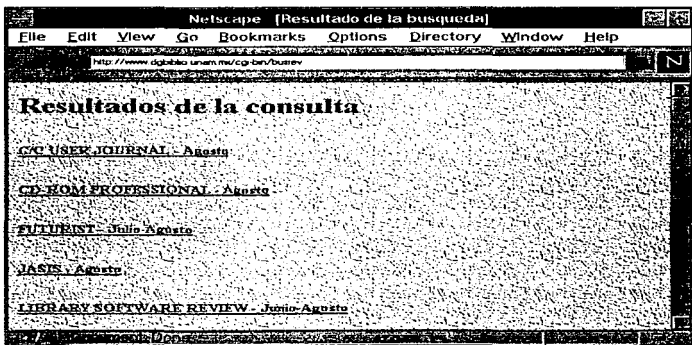


Figura 8.5 Pantalla de búsqueda de la alerta hemerográfica.

En el caso de que alguna revista contenga el índice en más de una página, se proporciona cada una de esas páginas, agregando botones de desplazamiento hacia la página siguiente si es el caso, la figura 8.6 muestra el resultado de haber seleccionado alguna de las ligas proporcionada en el resultado de la búsqueda.

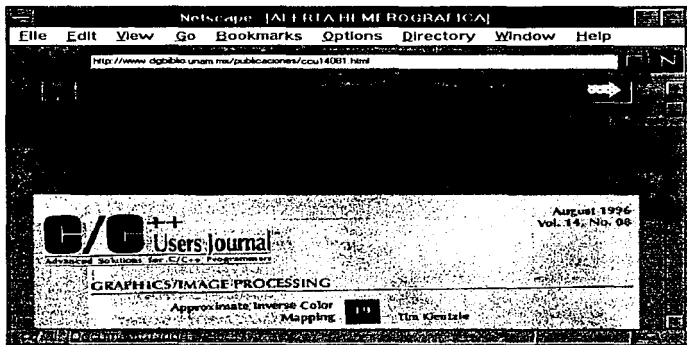


Figura 8.6 Resultado de la búsqueda.

.....



# Conclusiones



.....

## CONCLUSIONES

Se concluye que el presente trabajo aporta los conocimientos básicos y necesarios para guiar al lector desde la instalación del servidor HTTP hasta la creación de páginas en *Internet*, con lo cual el lector no requerirá de complicados y grandes volúmenes de información.

La metodología propuesta en este estudio consideró como hardware un equipo Sun Sparc, con sistema operativo Solaris 2.x y el software Apache versión 0.8.8, permitió la creación del *Web* de la Dirección General de Bibliotecas, el cual hasta el momento ofrece varios servicios de interés para la comunidad universitaria y público en general, ampliando la difusión de los acervos de libros, tesis y publicaciones periódicas de la UNAM.

El glosario que se incluye en esta trabajo recopila una serie de términos útiles al lector, reunidos en una sola obra, facilitándole la consulta rápida de estos.

Por último, cabe señalar que las herramientas que ofrece este trabajo solo contemplan la instalación y desarrollo de un *Web*, por lo que también deberá considerarse como aspecto importante del mismo, la presentación visual de la página, así como la calidad de la información, porque el producto final es la página en sí, que es lo que el usuario recibe, consideramos que para lograr una página de calidad se requiere de diferentes disciplinas tales como el diseño gráfico, la bibliotecología y otras ciencias de la comunicación.



# Bibliografía

# Bibliografía



## BIBLIOGRAFÍA

La Dirección General de Bibliotecas y la Biblioteca Central de la UNAM: historia, organización y servicios. México : UNAM, Dirección General de Bibliotecas, 1993. 48 p.

Lan times : guide to interoperability. 1a ed. México : McGraw-Hill, 1995.

Solaris 2.3 advanced users guide. 1a ed. California : Sun Soft, 1993. 196 p.

BREEDING, Marshall. TCP/ IP for the Internet : the complete buyer's guide to micro-based TCP/ IP software. 1a ed. Westport : Mecklermedia, c1995. 304 p.

GRAHAM, Ian S. The html sourcebook : a complete guide to html 3.0. 2a ed. Nueva York : John Wiley & Sons, 1996. 688 p.

GUNDAVARAM, Shishir. CGI programming on the world wide web. 1a ed. Estados Unidos: O'Rally & Associates, 1996. 433 p.

HESLOP, Brent D. HTML Publishing on the Internet for Windows : create great-looking documents on line: home pages, newsletters, catalogs, ads & forums. -- 1a ed. Chapel Hill, NC : Ventana Press, c1995. 480 p.

KARNEY, James. "SGML y HTML maestros de la etiqueta." p. 85-96. En PC Magazine en español. Vol. 6 no. 6 (jun. 1995).


MATUK, Javier. "Diseño su página en el web de Internet." p. 30-39. En Personal computing México. no. 89 (oct. 1995).

RODRÍGUEZ VEGA, Jorge. "Introducción al HTML. El lenguaje del Web". p. 21-28. En Revista profesional para programadores. no. 17.

RODRÍGUEZ VEGA, Jorge. "Internet al alcance de todos. Editores .HTML". p. 29-30. En Revista profesional para programadores. no. 17.

ROSEN, Kenneth H. Unix sistema V versión 4. 1a ed. Madrid; México : Osborne McGraw-Hill, c1991. 1046 p.





TANENBAUM, Andrew S. Organización de computadoras : un enfoque estructurado. 1a ed.\_ México : Prentice Hall, 1985. 507 p.

VELAZCO DÍAZ, Enrique. "Los lenguajes de Internet. Una panorámica". p.13-20. En Revista profesional para programadores. - no. 17.



# Glosario



## GLOSARIO

**Agent.** Programa que puede viajar en el *Internet* y acceder a recursos remotos que son de interés. Un *agent* propio puede estar disponible para correr en una máquina remota y viajar libremente desde una máquina hacia otra.

**Apache.** Software para poder convertir una computadora con sistema operativo UNIX en servidor HTTP.

**Applet.** Son programas escritos en lenguaje Java que se alojan en el *WWW* y se ejecutan mediante un *browser*. Para ejecutarse, dependen de que dicho *browser* este habilitado para aplicaciones Java.

**Archie.** Un sistema que automáticamente genera y mantiene el contenido de las bases de datos para los servidores FTP *anonymous*. Un servidor *archie* usa información de los servidores FTP y archiva en una lista los directorios de éste. Un cliente *archie* puede acceder a estas bases de datos y buscar programas o archivos asociados a un nombre particular.

**ASCII.** *American Standard Code for Information Interchange*, un código de 8 bits que representa un conjunto de 256 caracteres. Entre los cuales se encuentran los alfanuméricos, de letras con acentuación y de control.

**Browser.** Cualquier programa usado para visualizar material contenido en el *World Wide Web*; Mosaic, MacWeb, Linx, Netscape son ejemplos de *browsers*. Tienen la capacidad de interpretar URL y el lenguaje HTML.

**CGI (Common Gateway Interface).** Interfaz por la cual se puede intercambiar información entre servidores en el *WWW* y los usuarios del mismo y además de poder ejecutar procesos para aplicaciones mas complejas: como la consulta de bases de datos, registro de información o ejecución de cálculos.

**Compilador.** Se encarga de reconocer las estructuras sintácticas y semánticas de un lenguaje de programación, creando un código objeto.

**DNS (Domain Name Server).** Es un servidor, el cual contiene los nombres de dominio y direcciones IP asociadas a estos, de tal manera que cuando se utiliza el nombre de dominio para acceder a otra máquina, primero se consulta a este servidor para obtener su dirección IP y poder realizar un enlace.

**Domain name.** Nombre con el cual se asocia una dirección IP de una red, cada elemento de esa red, se identifica por medio de este nombre y con el identificador propio de cada máquina que forma dicha red.

**E-mail o Electronic mail.** Correo electrónico, uno de los servicios con los que dispone *Internet*, con el cual se pueden transmitir mensajes y/o archivos entre los usuarios de *Internet*.

**Firewall.** Es usado para separar una red local del mundo exterior. En general una red local es conectada al mundo exterior por una computadora llamada "gateway". Esta máquina puede ser convertida en una *firewall*, mediante la instalación de un software especial que no permite que los paquetes de TCP/IP no autorizados fluyan desde el interior al exterior y viceversa.

**Freeware.** Software que puede ser encontrado en *Internet*, el cual puede ser descargado y utilizar sin ningún costo.

**FTP (File Transfer Protocol).** Es un protocolo cliente-servidor de *Internet* para transmitir archivos entre computadoras.

**GIF (Graphic Interchange Format).** Es un formato para almacenar archivo de imágenes. Es uno de los tres formatos que pueden desplegarse en un documento HTML, los otros dos son X-Bitmaps y X-Pixelmaps.

**Gopher.** Un protocolo para distribuir información liberada comúnmente usado en sistemas de información distribuida. Los clientes *gopher* permiten el acceso a esta información. El *gopher* es un sistema basado en menús jerárquicos y no tiene capacidad para hipertextos.

**Home Page o Página principal.** Página introductoria para un sitio o lugar del *World Wide Web*. Una página principal, usualmente proporciona una introducción al sitio con ligas de hipertexto al recurso local.

**DNS (Domain Name Server).** Es un servidor, el cual contiene los nombres de dominio y direcciones IP asociadas a estos, de tal manera que cuando se utiliza el nombre de dominio para acceder a otra máquina, primero se consulta a este servidor para obtener su dirección IP y poder realizar un enlace.

**Domain name.** Nombre con el cual se asocia una dirección IP de una red, cada elemento de esa red, se identifica por medio de este nombre y con el identificador propio de cada máquina que forma dicha red.

**E-mail o Electronic mail.** Correo electrónico, uno de los servicios con los que dispone *Internet*, con el cual se pueden transmitir mensajes y/o archivos entre los usuarios de *Internet*.

**Firewall.** Es usado para separar una red local del mundo exterior. En general una red local es conectada al mundo exterior por una computadora llamada "gateway". Esta máquina puede ser convertida en un *firewall*, mediante la instalación de un software especial que no permite que los paquetes de TCP/IP no autorizados fluyan desde el interior al exterior y viceversa.


**Freeware.** Software que puede ser encontrado en *Internet*, el cual puede ser descargado y utilizar sin ningún costo.

**FTP (File Transfer Protocol).** Es un protocolo cliente-servidor de *Internet* para transmitir archivos entre computadoras.

**GIF (Graphic Interchange Format).** Es un formato para almacenar archivo de imágenes. Es uno de los tres formatos que pueden desplegarse en un documento HTML, los otros dos son X-Bitmaps y X-Pixelmaps.

**Gopher.** Un protocolo para distribuir información liberada comúnmente usado en sistemas de información distribuida. Los clientes *gopher* permiten el acceso a esta información. El *gopher* es un sistema basado en menús jerárquicos y no tiene capacidad para hipertextos.

**Home Page o Página principal.** Página introductoria para un sitio o lugar del *World Wide Web*. Una página principal, usualmente proporciona una introducción al sitio con ligas de hipertexto al recurso local.



**HTML (*Hypertext Markup Language*)**. Lenguaje a base de etiquetas, las cuales indican la forma y los elementos multimedia que contiene un documento en formato HTML. Dicho lenguaje es la base de WWW.

**HTTP (*HyperText Transfer Protocol*)**. Protocolo por medio del cual se transmite la información en el WWW.

**Hypertext o hipertexto**. Es un documento que contiene texto dentro del cual existen frases o palabras las cuales permiten el enlace a otros documentos. Un hipertexto puede ligarse con otra parte del mismo documento, con otro documento que esta en el mismo servidor o con un documento en un servidor remoto.

**Hyperlinks o hiperligas**. Son enlaces a documentos a través de hipertextos o imágenes habilitadas para esta tarea.

**Internet**. Es la red más amplia de computadoras comunicándose a través del protocolo TCP/IP, a esta red están conectadas diferentes arquitecturas de computadoras y diferentes plataformas de sistemas operativos.

**IP address o dirección IP**. Dirección numérica de una computadora o un segmento de una red en *Internet*. Cada computadora conectada a *Internet* tiene una dirección numérica única.

**JAVA**. Lenguaje de programación desarrollado por la compañía Sun Microsystems. La programación en *Java* esta totalmente orientada a objetos solamente puede correr bajo un intérprete *Java*, ya que no se genera un archivo ejecutable con este lenguaje.

**LAN (*Local Area Network*)**. Un conjunto de computadoras conectadas dentro de un edificio o un área limitada.

**MIME (*Multipurpose Internet Mail Extensions*)**. Esquema que permite a mensajes de correo electrónico que contiene elementos multimedia mezclados (sonido, video, imágenes y texto), ser reconocidos.

**Mosaic**. Es un *browser* gráfico, para el *World Wide Web*, desarrollado en el *Nacional Center of Supercomputing Applications (NCSA)*.

**Multitask o Multitarea.** Método en el que el procesador central de una computadora es capaz de trabajar simultáneamente en varias tareas diferentes. Ejecución de dos o más programas en un mismo CPU.

**Multiusuario.** Sistema en el que varios usuarios utilizan, desde diferentes terminales, a un mismo procesador central. Sistema por medio del cual varios usuarios comparten un solo procesador.

**Patch.** Secciones de software que sirven para corregir errores en programas de aplicación

**Proxy server o servidor proxy.** Es un servidor que desempeña el papel de intermediario entre una computadora (local ) y la computadora a la cual se quiere conectar (remota). Esto es, si se hace una petición desde una computadora "A" en la red local hacia una computadora remota "B", ésta se redirecciona a un servidor *proxy*, el cual la toma, envía la petición y recibe la respuesta de la computadora "B" y solo hasta haberla analizado, la transfiere a la computadora "A". Los servidores *proxy* sirven para acceder las peticiones del *World Wide Web* desde un *firewall*.

**Protocols o Protocolos.** Definen el "lenguaje" que las computadoras de una red "hablan".

**RISC (*Reduced Instruction Set Code*).** Tecnología aplicada a los microprocesadores, de manera que contengan un conjunto reducidos de instrucciones, para realizar operaciones más rápida y eficientemente, pero con la limitante de utilizar software especializado para este tipo de arquitecturas.

**Searcher.** Herramienta de búsqueda de información en el WWW.

**Server o Servidor.** Es un *software* instalado en una computadora, que esta en una red y responde a las peticiones de programas clientes ya sea locales o remotos.

**Shareware** Terminio utilizado para el software que se encuentra disponible dentro de *Internet*, que puede ser descargado y utilizado con la restricción de ser registrado antes de un periodo de tiempo, si al evaluarlo durante este periodo se decide adquirirlo.

**SGML (Standard Generalizer Markup Language).** Conjunto de reglas para describir la estructura y manejar el contenido de cualquier documento digital. Es un estándar abierto

**Tag (HTML) o Etiqueta.** Los documentos HTML usan etiquetas. Una etiqueta es simplemente un texto rodeado por los signos de mayor menor que "<" y mayor que ">", por ejemplo <TAG>. El final de una etiqueta tiene un diagonal y después el nombre de dicha etiqueta, por ejemplo </TAG>.

**Telnet.** Es un protocolo de emulación de una terminal que permite hacer una conexión de remota a otra computadora en *Internet*.

**TCP/IP (Transmission Control Protocol/Internet Protocol).** Es el protocolo de comunicación base que es el fundamento de *Internet*. Todos los otros protocolos como el HTTP, FTP y *gopher* son ejecutados sobre TCP/IP.

**UNIX.** Sistema operativo con capacidades de multitarea y multiusuarios, comúnmente usado en los servidores conectados a *Internet*.

**URL (Uniform Resource Locator).** Es el esquema usado para direccionar en *Internet* las peticiones del *World Wide Web*. Un URL especifica el protocolo para acceder un recurso de una máquina en particular.

**WAN (Wide Area Network).** Red de área amplia que enlaza varias redes LAN, ubicadas en diferentes lugares ya sea nacional o internacionalmente.

**WWW.** El *World Wide Web*, también llamado *Web*, W<sup>3</sup> o WWW es un sistema con características multimedia, que permite el intercambio de información en *Internet* utilizando como herramienta principal el hipertexto y las hiperligas.