



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
ARAGON

“DESARROLLO DE CGI'S (COMMON GATEWAY
INTERFACE) EN UN LENGUAJE DE
PROGRAMACION VISUAL”

295841

T E S I S
QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION
P R E S E N T A:
NOEMI JACQUELINE PARRA ARROYO

ASESOR: ING. DAVID MOISES TERAN PEREZ



SAN JUAN DE ARAGON, EDO. DE MEX. 2001



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

AGRADECIMIENTOS

A mi familia por todo su apoyo y comprensión.

**A mis
maestros** por compartir conmigo sus
conocimientos y experiencia.

**A mis
amigos** porque de una u otra forma
colaboraron con este trabajo.

INDICE

OBJETIVO	i
INTRODUCCION	ii
CAPITULO 1. Introducción a redes	1
Antecedentes	2
Clasificación de las redes	6
Componentes de una red	11
CAPITULO 2. Internet	13
Historia	13
Protocolo TCP/IP	17
Servicios proporcionados por Internet	25
Correo electrónico	25
Conexión remota	28
USENET	29
FTP	30
ARCHIE	32
Internet Relay Chat	34
GOPHER	35
VERONICA	36
WAIS	38
World Wide Web	38
CAPITULO 3. World Wide Web	39
Historia	39
Elementos que integran WWW	41
URL's	41
HTTP	42
Hipermedia	48
Hipertexto	50
CGI	62
Otras tecnologías en el WWW	63
CAPITULO 4. Servidor Web en la Pc	65
Por qué tener un servidor WWW	66
Selección de un servidor	66
Tres opciones para instalar un servidor en un ambiente Windows	68
Internet Information Server	68
Netscape Enterprise Server	70
Apache	71
Instalando un servidor WWW	73

CAPITULO 5. Desarrollo de los CGI's	77
Aplicaciones de los CGI's	78
Variables de entorno	79
Secuencia de un CGI	81
Escogiendo un lenguaje de programación	83
Windows CGI	83
Programando un CGI en Visual Basic	84
Ejemplo 1	85
Ejemplo 2	86
Ejemplo 3	89
CONCLUSIONES	93
APENDICE A. Lista de comandos de HTML	95
APENDICE B. Archivo de entorno del WINCGI	103
APENDICE C. Módulo CGI32.BAS	107
GLOSARIO	127
BIBLIOGRAFIA	142

OBJETIVO

Conocer el World Wide Web, su interacción con los usuarios por medio de los CGI 's, y el desarrollo de estos en un ambiente de programación visual.

INTRODUCCION

Debido a que WWW es un mercado de gran potencial dentro de Internet, hay mucho interés en desarrollar sitios Web, principalmente para compartir información, debido a esto se ha tratado de hacer que el Web sea más atractivo y que tenga más usos que solo presentar páginas estáticas con sólo texto y algunas imágenes, por eso se han desarrollado varias tecnologías que hacen que el Web interactúe con el usuario. Una de estas son los CGI 's que cumplen con esta función.

Los CGI se pueden elaborar en distintos lenguajes de programación, entre los principales se encuentran Perl y C, porque la mayoría de los servidores Web se encuentran en ambientes UNIX; pero debido al auge que se ha alcanzado con el Web, se han desarrollado servidores que trabajan en ambientes distintos de UNIX, como son Windows y Macintosh, así también se tuvieron que desarrollarse CGI 's para estos sistemas.

En un principio se hicieron estos programas en C, pero con el tiempo se fueron desarrollando librerías para lenguajes más adecuados para el ambiente en que se empleaban, así se fueron realizando estos programas en lenguajes visuales, como son Visual Basic, Delphi y Visual C. Haciendo que sea más fácil la programación de los CGI 's.

Capítulo 1. Introducción a redes. Se trata de tener una idea general de los que en computación son las redes de computadoras, en este capítulo se establece una definición, y se trata la historia de como se fueron formando, así como sus diferentes clasificaciones y los componentes que integran una red.

Capítulo 2. Internet. Se conoce como la red de redes, también llamada la supercarretera de la información, hasta hace algunos años sólo era utilizada por militares y científicos en Estados Unidos, pero ahora casi todos podemos utilizarla. En este capítulo se trata la historia de esta red de redes, así como una breve descripción del protocolo que utiliza para comunicarse y algunos de los servicios que proporciona.

Capítulo 3. World Wide Web. Hasta hace unos años Internet era simplemente el envío de mensajes a través de computadoras. La presentación de la información se volvió una cuestión interesante, ya que había que crear un sistema que permitiera un fácil manejo de la misma. Así nació el WWW, o también llamado más comúnmente Web. En este capítulo se ve una breve historia, los elementos que lo integran (como son: el HTTP, que es el protocolo con el cual trabaja; el hipertexto y la hipermedia, que son los que lo hacen atractivo a este servicio); y otras tecnologías que integran al Web.

Capítulo 4. Servidor Web en la Pc. Como el Web se ha convertido en una herramienta necesaria para cualquier empresa, escuela o universidad, existen servidores a la medida de las necesidades de cada una. Existen servidores Web

para distintos sistemas operativos. En este capítulo se analizan los que corren en sistemas Windows, pues es el sistemas más común y fácil de manejar, también se trata de ver los parámetros que se deben de tomar en cuenta a la hora de instalarlo y sus posibles riesgos.

Capítulo 5. Desarrollo de los CGI 's. Utilizando el Web es posible también interactuar con el usuario, una de esa formas es por medio de los CGI (Common Gateway Interface) que es una norma para establecer comunicación entre un servidor WWW y un programa, de tal modo que este último pueda interactuar con Internet. En este capítulo se ve en que se puede emplear un CGI, sus variables de entorno, su funcionamiento, también como es que se pueden programar estos en un ambiente windows por medio de Visual Basic.

INTRODUCCION A REDES

Primeramente tenemos que definir que es una red. Dos de las definiciones más usadas son siguientes:

“Una Red es un conjunto de computadoras independientes capaces de comunicarse electrónicamente”.

“Una red de computadoras, es un conjunto de sistemas conectados entre sí a través de medios de comunicación (líneas telefónicas, cable coaxial, fibra óptica y microondas), y sus objetivos fundamentales son: compartir información, comunicar usuarios, tener flexibilidad en el manejo de la información”.

Para nuestros propósitos nos quedaremos con la ultima definición.

Como se ha dicho las redes constan de dos o más computadoras conectadas entre sí y nos permiten compartir recursos e información. La información para compartir suele consistir en archivos y datos (por ejemplo: información diaria de ventas y gráficas, imágenes prediseñadas y documentos). Los recursos son los dispositivos o las áreas de almacenamiento de datos de una computadora, compartidos por otra computadora mediante la red. Los recursos que una computadora comparta en red incluyen unidades de disco, directorios e impresoras. Al igual que cualquier archivo de computación,

los programas pueden ser compartidos y utilizados mediante algún recurso compartido.

Para comprender lo que son las redes y la forma en que mejoran nuestra capacidad para manejar las computadoras, es importante conocer la manera en que estas han evolucionado hasta llegar a lo que son en la actualidad.

ANTECEDENTES

Las tarjetas perforadas se constituyeron en uno de los primeros medios para alimentar a las computadoras con información para procesamiento. Cada tarjeta representaba una línea de código o de datos del programa, la que después alimentaría a la computadora para que ésta la procesara. En su momento, la información contenida en las tarjetas perforadas la leía un sistema de macrocomputadoras, se procesaba y se imprimían los resultados. El proceso de leer información y procesarla como un todo se conoce actualmente como procesamiento por lotes. La siguiente gran mejora en la alimentación a la macrocomputadora de datos para procesamiento fue el uso de terminales tontas¹. Así, en vez de sentarse ante una perforadora y producir tarjetas perforadas, el usuario se sentaba ante una terminal tonta (una pantalla y un teclado conectados a la macrocomputadora) y tecleaba la información (figura 1.1)

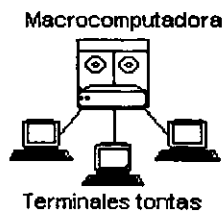


Figura 1.1

En los años sesenta comenzó a florecer un nuevo tipo de servicio de red comercial conocido como tiempo compartido. El tiempo compartido permitió que se instalaran las terminales en lugares geográficamente aislados de la computadora anfitriona, en locales de negocios o en centros de cómputo específicos, desde donde podían servir para acceder a los recursos de cómputo de la computadora anfitriona por medio de líneas telefónicas

¹ Las terminales tontas recibieron este nombre por el hecho de que no se realizaban ningún procesamiento en la terminal misma, sino que se utilizaba para enviar datos a la computadora anfitriona (central o host) por medio del teclado y para recibirlos por medio de la pantalla. Una computadora central es aquella a la que están conectadas las terminales tontas. Las computadoras anfitriona pueden ser macrocomputadoras o computadoras pequeñas.

alquiladas (figura 1.2). La computadora anfitriona asignaba y distribuía su tiempo en diferentes terminales que solicitaban su servicio.

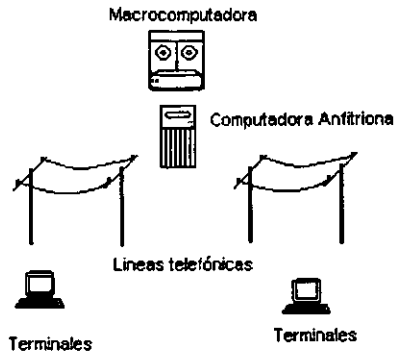


Figura 1.2

Debido a que las macrocomputadoras solían ser grandes y caras, frecuentemente la computadora anfitriona era menos potente. Cuando se enviaba un trabajo de procesamiento por lotes a la computadora central, ésta lo pasaba a la macrocomputadora, para su verdadero procesamiento. Cuando la macrocomputadora terminaba el procesamiento del trabajo, la computadora anfitriona intermedia terminaba guardaba los resultados y los enviaba de regreso a la terminal tonta o a la impresora. Solían usarse varias computadoras anfitrionas pequeñas para proporcionar acceso de las terminales tontas a una sola macrocomputadora.

En esta etapa se utilizaron dos tipos de redes para permitir la comunicación entre las computadoras. La conexión de las terminales tontas con las computadoras anfitrionas se consideró como una red. La conexión entre estas últimas y la macrocomputadora era indirecta, las dos redes en conjunto daban la apariencia de estar en una ruta directa, lo que traía por resultado que al sistema completo se le denominara como si fuese una sola red.

Durante este mismo periodo se dieron muchos avances tecnológicos, los que hicieron todavía más fácil el acceso y el uso de los recursos de cómputo de una macrocomputadora y de los de las otras computadoras anfitrionas. En vez del procesamiento por lotes estándar requerido anteriormente, se tuvo la capacidad para procesar información en tiempo real. Este tipo de procesamiento permitió que los usuarios vieran el resultado de la información procesada en cuanto se tecleaba.

Conforme se dispuso de más servicios de tiempo compartido, los usuarios de esos sistemas se encontraron en un predicamento interesante.

Cada servicio necesitaba de una terminal aparte, debido a que cada computadora anfitriona tenía su propio método para comunicarse con las terminales, pues no había un estándar. Debido a esto se tuvo que adoptar un estándar, en este caso fue el código ASCII.

El ASCII (American Standard Code for Information Interchange o código estándar americano para intercambio de información) fue adoptado en 1964 por la Organización Norteamericana de Patrones. El ASCII permite que 128 caracteres, incluidas las letras del alfabeto, los 10 dígitos numéricos (0-9), así como otros símbolos, puedan representarse en formato binario de 0 y 1. El formato binario para el juego de caracteres ASCII se sirve de una representación de 7 bits, que permite 128 combinaciones posibles.

Una vez establecido el ASCII como el método estándar para transmitir caracteres, se necesitó otro estándar para especificar la manera en que los datos serían transferidos por el cable. Para este propósito se perfeccionó el RS-232C para especificar los voltajes y los parámetros eléctricos de comunicación empleados para conectar dispositivos.

La solución al requerimiento de una línea alquilada para cada terminal fue el módem y la conexión para conmutación telefónica. Un módem conectado a la terminal marcaba el número de la computadora anfitriona. Otro módem conectado a esta contestaba y la terminal quedaba conectada a la computadora anfitriona por medio de líneas telefónicas normales. Al finalizar la llamada, la conexión también terminaba y la línea telefónica servía entonces para otros propósitos. Las conexiones por conmutador telefónica dieron por resultado ahorros importantes, pues ya no era necesaria una línea especial alquilada para abrir el acceso a la computadora anfitriona.

Posteriormente de estos servicios saldrían redes de datos públicos como Tymnet y Telenet. Las redes de las grandes corporaciones (Xerox, General Motors, IBM, Digital Equipment Corporation, AT&T y Burroughs), y las redes de investigación (SERCnet y NPL, inglesas de 1966-1968; HMI-NET de Berlín 1974; CYCLADES, Francia 1972), las redes comerciales, los sistemas de conferencia y las comunidades virtuales (especialmente USENET y FIDOnet).

Conforme avanzaba la tecnología, fue evolucionando la forma de acceso a los servicios de redes de computadoras, lo cual dio una mayor capacidad y un menor costo. Las minicomputadoras, que eran menos poderosas pero también menos caras que las macrocomputadoras, fueron compradas y empleadas por muchas compañías para trabajos de contabilidad y de administración de inventarios. Con la llegada de los estándares ASCII y RS-232C, frecuentemente se utilizaba una sola terminal tonta para acceder a muchos tipos de computadoras anfitrionas y servicios. Se crearon y adoptaron estándares adicionales para especificar protocolos (reglas) que se aplicaron a la comunicación.

Con el tiempo aparecen las primeras computadoras personales (PC) que proporcionan la capacidad de cómputo en una sola unidad para una sola persona. En vez de tener una terminal tonta conectada a una computadora anfitriona, a partir de ese momento se tuvo una computadora independiente puesta en el escritorio. Por supuesto, la PC no eran tan poderosa como una macro o minicomputadora, pero para dar servicio a las necesidades de un solo usuario, era muy buena.

La tecnología siguió avanzando y la combinación de competencia y avance tecnológico hizo que los precios bajaran y se aumentara el poder de cómputo. La flexibilidad, adaptabilidad y bajo precio de las computadoras personales lograron un gran impacto en las ventas. Nació una industria inmensa, lista para responder a las necesidades de sus clientes. Aunque la PC era capaz de ejecutar programas y procesar datos sin la intervención de otra computadora, todavía existía la necesidad de acceder a otros sistemas de cómputo. Se diseñó un software que permitió que la PC reemplazara a la terminal tonta para conectarse con una computadora anfitriona o central mediante servicios de tiempo compartido.

La PC servía para introducir los datos, que normalmente se tecleaban en una terminal tonta mientras estaban en línea con un servicio de tiempo compartido, lo cual ahorra los gastos de la conexión a la línea telefónica y a la computadora central. Después de teclear los datos en la PC y guardarlos en un archivo, se podía hacer conexión a la computadora central y transferirle (o subir) a esta esos datos a mayor velocidad que si fuera tecleados directamente en una terminal tonta. La PC también sirvió entonces para capturar los datos que se enviaban de la computadora anfitriona a un archivo (proceso también llamado bajar datos), y que luego podrían utilizarse en la PC o enviarse a otra computadora.

Con el avance de la tecnología vino una gran evolución de la computadora. A la PC se le añadieron capacidades de almacenamiento adicional y de procesamiento. Además las mini y macrocomputadoras tenían un alto costo de adquisición y mantenimiento, mientras que la PC podía comprarse por poco dinero y los costos de mantenimiento eran bajos.

A medida que las redes de computadoras fueron captando más adeptos, compañías tales como XEROX e IBM comenzaron a desarrollar su propia tecnología en redes de computadoras, comenzando por lo general, con redes de área local. Las redes de amplio alcance entonces, pasaron a ser usadas no solo para la comunicación entre computadoras conectadas directamente sino también para comunicar las redes de área local.

Con el establecimiento de ARPAnet, en USA.-1968, comenzó a entreverse el impacto social de la telemática. La tecnología de ARPAnet fue utilizada para construir en 1976, la red comercial TELENET. En Europa, las

compañías de teléfono, que controlan las redes públicas de transmisión de datos en cada país, adoptaron el protocolo X-25.

En 1987 la red ARPANet (dependiente del departamento de Defensa norteamericano) utilizada al principio exclusivamente para la investigación y desbordada por el interés demostrado por sus usuarios por el correo electrónico, necesito transmitir datos que usaban gran espectro de banda (sonidos, imágenes y videos) y sufrió tal congestión que tuvo que declarar obsoletas sus redes de transmisión de 56.000 baudios por segundo (5.000 palabras por minuto). Posteriormente se convirtió en la espina dorsal de las telecomunicaciones en USA, bajo su forma actual de INTERNET, una vez que quedo demostrada la viabilidad de redes de paquetes conmutados de alta velocidad.

Mientras tanto, se fue desarrollando otra tecnología, basada en conexiones por líneas telefónicas en lugar de conexiones dedicadas. Dos de los primeros productos de esta tecnología fueron ACSNET y UUCP, que sobreviven en una forma modificada. Las redes a través de líneas telefónicas produjeron el más distribuido de los sistemas de conferencia: USENET. También BITNET puso a disposición de la comunidad académica la tecnología en redes de computadoras de IBM y lo difundió aun entre computadoras de otras marcas.

Los servicios prestados por las redes de computadoras se han difundido ampliamente y alcanzan ya a la mayoría en las naciones. A medida que su diversidad continua en aumento, la mayoría de las redes académicas, se conectan entre sí, por lo menos con el propósito de intercambiar correo electrónico.

La comunicación mediante computadoras es una tecnología que facilita el acceso a la información científica y técnica a partir de recursos informativos y de telecomunicaciones. Por eso, se dice que una red es, fundamentalmente, una forma de trabajo en común, en la que son esenciales tanto la colaboración de cada miembro en tareas concretas, como un buen nivel de comunicación que permita que la información circule con fluidez y que pueda llevarse a cabo el intercambio de experiencias.

CLASIFICACION DE LAS REDES

Con el surgimiento de las redes, se tuvo la necesidad de clasificarlas de alguna manera, para poder manejar estándares y poder así estudiarlas mejor. A las redes se les puede clasificar de acuerdo a diferentes parámetros, a continuación se muestra algunos de ellos.

POR TIPO DE PROPIETARIO

Redes Privadas. Estas son las más comunes y normalmente perteneces a universidades, bancos, empresas publicas y privadas. Se caracterizan porque solo un grupo reducido de personas tiene acceso a este tipo de red (los propietarios, los socios, empleados y estudiantes)

Redes Comerciales. Estas rentan sus servicios a personas interesadas en tener acceso a la información en la red. Este tipo de redes pueden pertenecer a revistas científicas, agencias de noticias y grupos que ofrezcan productos de interés común.

Redes Públicas. Estas son administradas generalmente por el gobierno en países subdesarrollados y por grandes consorcios en países capitalistas. Utilizan la infraestructura de la red telefónica y ofrecen sus servicios de transmisión de datos que ofrece, son económicas debido a que se comparten canales de comunicación entre gran cantidad de usuarios.

POR EXTENSION

Red de Area Local

La red de área local (LAN, Local Area Network) nació con los beneficios de conectar las PC o las microcomputadoras a fin de compartir información. Mucho antes de que fuera considerada factible la idea de que las PC reemplazaran a las macro y minicomputadoras, comenzaron a aparecer las primeras LAN de PC.

Una LAN es un sistema de comunicaciones de alta velocidad que conecta microcomputadoras que se encuentren cercanas, por lo general dentro del mismo edificio. Una LAN consta de hardware y software de red y sirve para conectar las PC compartan entre ellas programas, información y recursos, como unidades de disco, directorios e impresoras.

Aunque apareció desde 1983, la LAN ha continuado evolucionando hasta llegar a ser una parte integral de la conectividad de las PC. Las LAN disponibles actualmente son confiables e incluyen muchísimas características poderosas. Esto hace que la LAN sea para las PC una herramienta extremadamente poderosa y flexible para compartir información.

Las LAN son capaces de transmitir datos a velocidades muy rápidas, algunas inclusive más rápido que por línea telefónica; pero las distancias son limitadas. Red de Area Local (Local Area Network) Red formada por computadoras que se encuentran en un mismo edificio, fabrica o campus universitario, es decir en un radio de unos pocos kilómetros cuadrados.

Las LAN están confinadas a un espacio físico restringido y comparten periféricos de costo elevado (graficadores, impresoras láser y unidades de memoria) entre las computadoras que integran la red. No existe un parámetro que indique la longitud máxima de una LAN pero si se puede afirmar que cuando se utilicen sistemas de comunicación remota para comunicar 2 nodos de una misma red se dejara de hablar de una LAN

Red de Area Metropolitana

Con la necesidad de extender más el área de trabajo de una red de área local surge la Red de Area Metropolitana (MAN, Metropolitan Area Network), este tipo de redes pueden extenderse hasta un radio de 50 kilómetros más o menos (esto es, puede abarcar una ciudad o un estado), dicha red satisface una necesidad de una sistema de comunicación de información de tamaño intermedio.

Estas redes también de caracterizan por ser redes híbridas, es decir redes que conectan Pc's, mini y macro computadoras. Se diferencian de las WAN, en que los equipos de comunicación no son tan sofisticados pues no se transmite a distancias muy grandes.

Red de Area Amplia

Una red de área amplia (WAN, Wide Area Network) es una conexión de computadoras que no están geográficamente cercanas, estas necesitan de un hardware especial, así como líneas telefónicas y por lo tanto también software especial.

Estas redes surgen por la necesidad de tener que conectar equipos de comunicaron remota a las computadoras que integran la red, que pueden ser Mainframes, minicomputadoras o Pc's. La extensión geográfica que abarca una red WAN, puede ir desde un estado hasta cubrir en su totalidad el territorio de un país o incluso continentes.

POR TOPOLOGIA

Se le llama topología a la forma en que se conectan los nodos o computadoras dentro de una red. Una red tiene dos diferentes topologías: una física y una lógica. La topología física es la disposición física actual de la red, la manera en que los nodos están conectados unos con otros. La topología lógica es el método que se usa para comunicarse con los demás nodos, la ruta que toman los datos de la red entre los diferentes nodos de la red. Las topologías física y lógica pueden ser iguales o diferentes.

Las tres topologías de red estándar son de bus, de estrella y de anillo. También existen combinaciones de más de una topología. Por ejemplo, una topología de árbol es la combinación de una topología de bus y una de estrella.

Topología de Bus

En una topología de bus, cada computadora esta conectada a un segmento común de cable de red (figura 1.3). El segmento de red se coloca como un bus lineal, es decir, un cable largo que va de un extremo a otro de la red, y al cual se conecta cada nodo de la red. El cable puede ir por el piso, por las paredes, por el techo, o puede ser una combinación de estos, siempre y cuando el cable sea un segmento continuo.

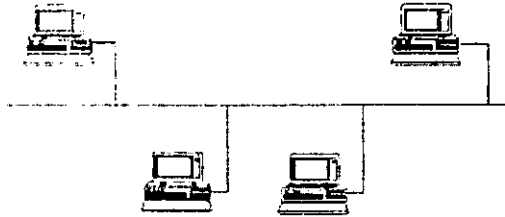


Figura 1.3

El término bus se refiere a la línea de transmisión común a todos los nodos. La información transmitida por cualquier nodo viaja a través del bus con el identificador del nodo al cual va dirigida, y el proceso de recoger la información es, si el nodo reconoce el identificador y coincide con el suyo, recoge el mensaje; en caso contrario, el mensaje sigue recorriendo el bus.

En este tipo de red todos los nodos tienen las mismas posibilidades de transmisión.

Topología de Estrella

En una topología de estrella, cada computadora esta conectada a un concentrador (o hub) ubicado centralmente. (figura 1.4)

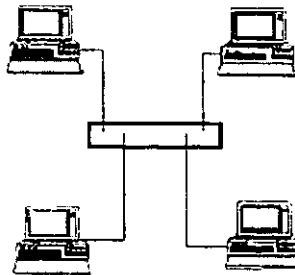


Figura 1.4

Las características principales de este tipo de topología son:

- a) Los nodos se conectan a un concentrador central.
- b) La falta de un nodo no afecta a la red.
- c) La ruptura de un cable afecta solo a 1 nodo conectado a él.
- d) El tráfico de la información aumenta conforme a los puertos que se conectan.
- e) El repetidor envía la información a todas estaciones de trabajo.
- f) El control de la red se asigna al nodo central de tal modo que todos los mensajes que se envían son controlados por él.

Topología de Anillo

En una topología de anillo, cada computadora se conecta en forma de anillo a la red (figura 1.5). En esta topología las estaciones de trabajo y el servidor están conectadas a través de un solo cable de comunicación de trayectoria cerrada, en donde la información fluye en un solo sentido. El método de acceso al cable se llama Token Ring, en el cual si una estación de trabajo quiere transmitir datos, envía un arreglo de bits de información (Token) que son recibidos por una terminal, y así sucesivamente hasta que el mensaje llega a su destinatario.

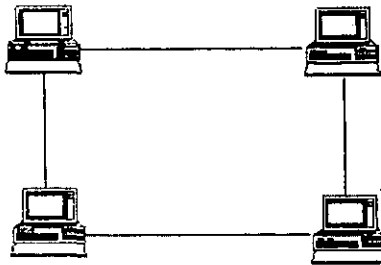


Figura 1.5

Este tipo de topología tiene las siguientes características:

- a) Los tiempos máximos de espera están definidos
- b) Como el servidor sondea primero cual estación de trabajo quiere transmitir no existen interferencias entre las estaciones de trabajo.
- c) Es un método de acceso útil en redes con gran tráfico de información
- d) Cada uno de los nodos retransmite a su vecino más cercano
- e) La ruptura de un cable afecta a toda la red
- f) Se necesita que una máquina sea "monitor" y esto se decide según criterios de jerarquía.

COMPONENTES DE UNA RED

Para poder hablar de una red, también se tiene que saber de los elementos que la componen. Los componentes básicos de una red son los siguientes:

Servidor (server)

El servidor es la máquina principal de la red, la que se encarga de administrar los recursos de la red y el flujo de la información. Muchos de los servidores son "dedicados", es decir, están realizando tareas específicas, por ejemplo, un servidor de impresión solo para imprimir; un servidor de comunicaciones, sólo para controlar el flujo de los datos, etc. Para que una máquina sea un servidor, es necesario que sea una computadora de alto rendimiento en cuanto a velocidad y procesamiento, y gran capacidad en disco duro u otros medios de almacenamiento.

Estación de trabajo (Workstation)

Es una computadora que se encuentra conectada físicamente al servidor por medio de algún tipo de cable. Muchas de las veces esta computadora ejecuta su propio sistema operativo y ya dentro, se añade al ambiente de la red.

Sistema Operativo de Red

Es el sistema (Software) que se encarga de administrar y controlar en forma general la red. Para esto tiene que ser un Sistema Operativo Multiusuario, como por ejemplo: Unix, Netware de Novell, Windows NT, etc.

Recursos a compartir

Al hablar de los recursos a compartir, estamos hablando de todos aquellos dispositivos de Hardware que tienen un alto costo y que son de alta tecnología. En estos casos los más comunes son las impresoras, en sus diferentes tipos: Láser, de color, plotters, etc.

Hardware de Red

Son aquellos dispositivos que se utilizan para interconectar a los componentes de la red, serían básicamente las tarjetas de red (NIC->Network Interface Cards) y el cableado entre servidores y estaciones de trabajo, así como los cables para conectar los periféricos.

Transmisión de datos

La transmisión de datos en las redes, puede ser por dos medios:

Terrestres. Son limitados y transmiten la señal por un conductor físico. Estos pueden ser los siguientes:

- a) Cable par trenzado: Es el que comúnmente se utiliza para los cables de teléfonos, consta de dos filamentos de cobre, cubiertos cada uno por plástico aislante y entrelazados el uno con el otro, existen dos tipos de cable par trenzado: el "blindado", que se utiliza en conexiones de redes y

estaciones de trabajo y el "no blindado", que se utiliza en las líneas telefónicas y protege muy poco o casi nada de las interferencias.

- b) Cable coaxial: Este tipo de cable es muy popular en las redes, debido a su poca susceptibilidad de interferencia y por tener un gran ancho de banda, los datos son transmitidos por dentro del cable en un ambiente completamente cerrado, una pantalla sólida, bajo una cubierta exterior. Existen varios tipos de cables coaxiales, cada uno para un propósito diferente.
- c) Fibra óptica: Es un filamento de vidrio sumamente delgado diseñado para la transmisión de la luz. Las fibras ópticas poseen enormes capacidades de transmisión, del orden de miles de millones de bits por segundo. Además de que los impulsos luminosos no son afectados por interferencias causadas por la radiación aleatoria del ambiente. Actualmente la fibra óptica esta remplazando en grandes cantidades a los cables comunes de cobre.

Aéreos: Son "ilimitados" en cierta forma y transmiten y reciben las señales electromagnéticas por microondas o rayo láser.

INTERNET

Internet se conoce como la red de redes, pero su nombre se deriva de INTERnational NETworking

Aunque Internet sólo hasta hace poco que se ha hablado de ella en los diferentes medios de comunicación, ha existido, de una u otra forma, por más de dos décadas. A continuación se expondrá una breve historia con algunos de los antecedentes que nos ayudarán a comprender parte de su fuerza, de sus debilidades y de su idea principal que rodea a este medio.

HISTORIA

Internet empezó como ARPANET a principios de 1969. Fundada por la Advanced Research Project Agency (ARPA de ahí el nombre), ARPANET fue diseñada para permitir que los investigadores se comunicaran y compartieran información unos con otros. Otra prioridad que tenía era que la red debería ser capaz de sobrevivir, aun cuando parte de ella fuera físicamente destruida (esto principalmente por la amenaza de un ataque nuclear que existía en ese entonces).

De manera fundamental, la idea era: construir una red para investigadores a todo lo largo de los Estados Unidos, para que pudieran utilizarla en sus actividades diarias y, además, tuvieran la seguridad que la destrucción de una máquina en una de las localidades no detendría el funcionamiento de la red.

Como localizaciones de prueba se establecieron cuatro servidores ARPANET: La Universidad de Utah, el Instituto de Investigaciones de Stanford y dos servidores de la Universidad de California, Santa Bárbara y Los Angeles (pudiendo transmitir datos en líneas de transmisión de alta velocidad y programar remotamente computadoras en otros). En septiembre de 1969, se conectó ARPANET. En este tiempo no se le dio mucha importancia por parte de la prensa. Ya en 1971 había quince nodos, y para 1972, treinta y siete.

Poco a poco comenzó a expandirse el uso de ARPANET: no solamente se dedicaba a trabajos de cómputo a larga distancia, sino que se extendió a la comunicación de proyectos y trabajos entre investigadores, y al uso personalizado del correo electrónico. Así también surgen las listas de interés, que son mensajes de correo electrónico retransmitidos automáticamente a los suscriptores en la red.

Sin embargo, años más tarde, mucha gente se interesó. En el otoño de 1972, más de un millón de personas presenció la primera demostración pública de ARPANET y entonces fue cuando verdaderamente la idea de una red nacional (e incluso internacional) empezó a cobrar forma. Todos empezaron a encontrar razones por las que tenían que formar parte de la misma y lo que dicha red podría hacer por ellos. Otra ventaja de ARPANET es que no importaba los tipos o tamaños de las máquinas en las que se estuviera trabajando, mientras cumplieran con los protocolos establecidos, funcionarían dentro de la red.

En 1974, el protocolo original que se conocía como NCP (Network Control Protocol), es cambiado por un nuevo estándar más sofisticado, llamado TCP/IP, publicado en este año por Vint Cerf y Bob Khan. En este nuevo protocolo se tienen dos clases, TCP (Transmission Control Protocol) que convierte mensajes en cadenas de paquetes en el nodo de origen, y los ensambla de nuevo en el punto de destino, e IP (Internet Protocol) que maneja el direccionamiento permitiendo que los paquetes fueran ruteados a través de diferentes nodos y hasta de diferentes redes con varios estándares, como Ethernet, FDDI y X.25.

En 1977 se comenzó a extenderse el uso de TCP/IP en otras redes para vincularse a ARPANET.

A fines de los años 70 y en los años 80, personas de diferentes grupos sociales tuvieron acceso a computadoras de gran capacidad, siendo bastante fácil el conectarse a la creciente red de redes. Como el software de TCP/IP es

de dominio público, y por su misma naturaleza, descentralizante y hasta anárquico, comenzó el auge de la conexión a Internet. Fue en esta época donde surgió USENET, el boletín electrónico más grande del mundo, basándose en UUCP, tecnología desarrollada en los laboratorios Bell de AT&T, junto con el sistema operativo UNIX, que al paso de los años, se ha convertido en el sistema operativo estándar de todas las computadoras de mediano y gran tamaño conectadas a Internet.

A principios de los ochentas ARPANET es dividida en dos redes: en MILNET que se ocupó de asuntos militares, y en ARPANET que transportaba datos de investigación correspondientes a redes y otros campos de acción. Las universidades corrieron a unirse y las redes empezaron a atraer a personas que no pertenecían a la ciencia o a la tecnología. Para mediados de los ochentas, la National Science Foundation enlazó seis centros de supercomputadoras de los Estados Unidos en una red llamada NSFNET, con las velocidades de datos incrementadas de 56000 bps a 1.5 mbps. Este tipo de conexión, conocida como T-1, aún sigue en funcionamiento pero, en 1990, se introdujo la especificación T-3, lo que permitió velocidades de conexión de hasta 45 mbps.

En este tiempo surgió otro punto de desarrollo de estas redes, BITNET (Because It's Time for NETWORK), creada como red cooperativa, proveyendo a sus usuarios de correo electrónico, listas de interés y transferencia de información y archivos.

Con esto la conexión a Internet tiene un costo mínimo, ya que cada nodo es independiente, y maneja por sí mismo sus propias necesidades técnicas y financieras. De esta manera, la red comenzó a extenderse, abarcando mayor número de gentes conectadas y de recursos. Así, la comunicación a través de la computadora comenzó a ser indispensable.

A finales de los ochentas, NSF pasó el financiamiento y la administración de la NSFNET a un grupo no lucrativo de universidades, llamado MERIT (Michigan Educational Research Information Triad). MERIT trabajó junto con MCI e IBM en la expansión y la mejora del acceso nacional a alta velocidad. Temporalmente, las tres organizaciones formaron la ANS (Advanced Network Services), creada con la finalidad de operar a la NSFNET. Esto resultó importante: NSF había establecido una política de uso aceptable (llamada AUP, de acceptable-use policy) que no permitía acceso comercial a la NSFNET. Sin embargo, las empresas ya empezaban a comprometerse. La apariencia de la red empezaba a modificarse.

En 1989 México ingresa a Internet a través de NSFNET, contando además con la red BITNET, que permite que usuarios del ITESM (Instituto Tecnológico de Estudios Superiores de Monterrey) y la UNAM (Universidad Nacional Autónoma de México) tener acceso a los recursos existentes en Estados Unidos y el resto del mundo.

Para 1990, la NSFNET había tomado el lugar de ARPANET y este último había quedado discontinuado.

Mientras ocurría todo esto, otras organizaciones comenzaban a interesarse en las redes globales. La CSNET (Computer Science Network) fue establecida por NSF para ayudar a las universidades que no podían tener acceso a la NSFNET e introducirlas en lo que poco a poco se convertiría en Internet.

Se implementan herramientas que catalogan y facilitan el acceso a Internet: Archie, para la búsqueda de archivos accesibles mediante FTP (File Transfer Protocol); Hytelnet, un catálogo de recursos y bibliotecas en línea accesibles mediante telnet (terminal remota); WAIS (Wide Area Information Servers), para entregar directamente documentos al usuario, solicitándolos a través de palabras clave; Gopher, para ver la información a través de menús; PGP (Pretty Good Privacy), para dar seguridad y privacidad a los mensajes de la comunidad en la red; Veronica, un sistema de búsqueda complementario a Gopher.

En 1991 Comercial Internet eXchange (CIX) Association, Inc, surge a partir de que NSF levanta las restricciones que existían para el uso comercial de la Red.

En 1992, es un año de profundos cambios dentro de Internet. Se funda Internet Society (ISOC), para coordinar el uso de las tecnologías existentes en beneficio de todos los usuarios. Se desarrolla en el CERN la tecnología de WWW (World Wide Web), que permite un acercamiento más fácil a través de hipertexto a los recursos de Internet; también se da la primera muestra de audio y vídeo en tiempo real a través de la Red.

Para 1993 InterNIC es creado por NSF para proveer servicios de información, así como registros, directorios y bases de datos referentes a Internet.

El crecimiento de la red se vuelve exponencial. Mosaic, explorador de Internet desarrollado en la Universidad de Illinois Urbane-Champagne, es el primero en aprovechar la gran capacidad del WWW, teniendo un crecimiento anual de 341,634% en número de usuarios de esta herramienta. El crecimiento de Gopher es del 997%. Las empresas comerciales y los medios de comunicación prestan real atención a Internet.

En 1994 Internet cumple 25 años de servicio. Ahora hay comunidades completas conectadas a Internet (Lexington y Cambridge, Mass., USA), el Senado de los Estados Unidos provee información y los centros comerciales llegan a la red, como Internet Shop Network y JCPenny. El auge es tal que surgen servicios bancarios en la red, como First Virtual y los negocios

comienzan a prosperar. Surgen los centros comerciales de Internet. La mercadotecnia encuentra atractiva a Internet a través del uso masivo de correo electrónico.

En 1995, los sistemas de servicios vía módem (Compuserve, Prodigy, Genie) comienzan a ofrecer servicios de Internet. Gran cantidad de compañías relacionadas con la red se vuelven públicas, encabezadas por Netscape, que tiene el tercer índice de ganancias jamás conseguido en Wall Street.

Los datos aquí proporcionados muestran la evolución de Internet, en la cual se señala la gran diferencia que existe entre su estado actual y sus orígenes. La red, que comenzó como un proyecto de sobrevivencia de la información ante la posibilidad de un ataque nuclear, ha derivado en una red de redes, que comunica de manera amplia y eficiente a un creciente número de personas. Alrededor de esta red se ha generado una nueva cultura, la cybercultura, con su modo de pensar, de hablar, de sentir; un mundo nuevo que aún falta mucho por explorar y que tiene un gran potencial.

Como ya vimos la Internet ha evolucionado de manera muy rápida. ¿Pero como es que se pueden comunicar tantas redes en el mundo?

Para lograr conectar a todas estas redes entre sí, se necesita tener una serie de reglas con las que se puedan comunicar entre ellas, sin crear conflictos; esto es lo que realiza la familia de protocolos TCP/IP, que son los encargados de que todo el mundo de Internet se pueda comunicar de una forma fácil y rápida.

PROTOCOLO TCP/IP

Aunque un poco simplificado, TCP/IP es el nombre de un protocolo de conexión de redes. Un protocolo es un conjunto de reglas a las que se tienen que atener todas las compañías y productos de software con el fin de que todos sus productos sean compatibles entre ellos.

TCP/IP son las siglas de Transmission Control Protocol / Internet Protocol, Protocolo de Control de Transmisión / Protocolo de Internet, que son dos protocolos diferentes. TCP/IP no es un producto único, como piensa mucha gente, sino es toda una familia completa de protocolos relacionados entre sí, diseñados todo ellos para transferir información a través de una red y para proporcionar información de estado sobre la misma red. TCP/IP está diseñado para ser una componente de una red, principalmente la parte del software. TCP/IP no esta limitado a Internet. Se trata del protocolo de red más utilizado en la conexión de redes.

Los principales protocolos tienen las siguientes funciones:

TCP. Es el protocolo que se encarga de hacer la comunicación bidireccional completa mediante circuitos virtuales, proporciona un mecanismo

fiable para la transferencia de datos, al ser un protocolo de alto nivel su función es que grandes volúmenes de información lleguen a su destino correctamente, pudiendo recobrar la pérdida esporádica de paquetes

IP. La principal tarea de este protocolo es dirigir datagramas² entre ordenadores, ya que gestiona el destino, el mejor método de ruta para llegar al destino y también asegura la recuperación en caso de que se presente un problema. Otra tarea importante que tiene es dividir datagramas grandes en partes más pequeñas y después volver a juntarlos en el lugar de destino.

Arquitectura de niveles de TCP/IP

Cuando se diseñó TCP/IP los comités establecidos para crear la familia de protocolos consideraron todos los servicios que se tenían que proporcionar. El mejor enfoque para implementar todos los servicios necesarios era dividir los servicios en categorías, como por ejemplo usuario final (transferencia de ficheros y acceso remoto), transporte (el modo en el que se envían y reciben los datos, invisible para el usuario) y red (como se dividen los datos en paquetes para poder transferirlos). Con este esquema se desarrolló una arquitectura por niveles que aislaba cada uno de los grupos de servicio.

Un enfoque por niveles para diseñar el software requería más trabajo inicialmente, pero a su vez reportaba varios beneficios. En primer lugar, puesto que cada nivel es independiente del resto, los cambios realizados en un servicio no afectaban a los otros. Según se van desarrollando nuevos servicios, esto se pueden añadir sin cambiar ninguna de las partes del sistema del software; y lo que es más importante, esta distribución por niveles permite que se desarrollen un conjunto de pequeños programas, pero eficaces para tareas específicas, cada una de las cuales puede ser independiente de las otras.

La distribución por niveles se utiliza en muchos sistemas de software; una referencia común es la arquitectura ideal de protocolos de conexión de redes desarrollada por la International Organization for Standardization, denominada ISO. ISO desarrolló el modelo de referencia Open System Interconnection (OSI), o Interconexión de Sistemas Abiertos que consta de siete niveles. El modelo OSI por niveles es el que se muestra en la figura 2.1

² Un datagrama es un mensaje que se divide en paquetes y se envía por los niveles a la red.

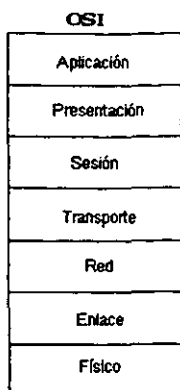


Figura 2.1

Cada nivel en el modelo de referencia OSI tiene una tarea específica que desempeñar. Las tareas son un poco abstractas, porque el modelo OSI es simplemente eso, un modelo. No está diseñado para ser un modelo real, sino un modelo para que lo sigan sistemas como TCP/IP. Los niveles son:

- a) *Aplicación*. Donde se proporciona la interfase de usuario. El nivel de la aplicación es el lugar al que pertenece un programa como la transferencia de ficheros.
- b) *Presentación*. Maneja los formatos de los datos, como ASCII y EBCDIC. El nivel de presentación transforma cualquier dato del nivel a aplicación en un formato estándar para los niveles inferiores.
- c) *Sesión*. La interfase para la red. Este es el lugar donde se gestionan las conexiones entre las aplicaciones y la red.
- d) *Transporte*. Toma datos del nivel de sesión y coloca más información en torno suyo para identificar el destino, o cuando recibe, decodifica la información para determinar a qué aplicación se envían los datos.
- e) *Red*. Envía datos de su sistema a otro. El nivel de red controla la forma en que se envían los datos por la red.
- f) *Enlace de datos*. Proporciona servicios para una transmisión fiable de datos. El nivel de enlace de datos asegura que los datos que se envían coincidan con los que se reciben.
- g) *Físico*. La conexión entre el dispositivo físico y el medio de red. El nivel físico gestiona el envío de bits por el cable de red (o cualquier otro medio que se utilice).

El enfoque OSI por niveles es el que utiliza TCP/IP, aunque con una ligera modificación. La arquitectura de niveles de TCP/IP es la que se muestra en la figura 2.2 junto con los niveles de OSI para facilitar la comparación.

TCP/IP	OSI
Aplicación	Aplicación
	Presentación
	Sesión
Transporte	Transporte
Red	Red
Acceso-a-red	Enlace
	Físico

Figura 2.2

Como puede observarse, los niveles son similares, aunque TCP/IP agrupa varios de los niveles OSI en un único nivel TCP/IP. Esto se realiza principalmente porque era el mejor método de implementar los servicios TCP/IP.

Capas de TCP/IP

Capa de Aplicación. Invoca programas que acceden servicios en la red. Interactúan con uno o más protocolos de transporte para enviar o recibir datos, en forma de mensajes o bien en forma de flujos de bytes.

Capa de Transporte. Provee comunicación extremo a extremo desde un programa de aplicación a otro. Regula el flujo de información. Puede proveer un transporte confiable asegurándose que los datos lleguen sin errores y en la secuencia correcta. Coordina a múltiples aplicaciones que se encuentren interactuando con la red simultáneamente de tal manera que los datos que envíe una aplicación sean recibidos correctamente por la aplicación remota, esto lo hace añadiendo identificadores de cada una de las aplicaciones. Realiza además una verificación por suma, para asegurar que la información no sufrió alteraciones durante su transmisión.

Capa Internet. Controla la comunicación entre un equipo y otro, decide que rutas deben seguir los paquetes de información para alcanzar su destino. Conformar los paquetes IP que serán enviados por la capa inferior. Desencapsula los paquetes recibidos pasando a la capa superior la información dirigida a una aplicación.

Capa de Acceso a Red. Emite el medio físico los flujos de bit y recibe los que de él provienen. Consiste en los manejadores de los dispositivos que se conectan al medio de transmisión.

Una condición que se necesita para permitir que la arquitectura por niveles funcione adecuadamente es que cada nivel debe saber lo que recibe de un nivel por encima o por debajo. Puede que el nivel no se preocupe de los contenidos del mensaje, pero debe saber qué es lo que tiene que hacer con el mensaje.

Para simplificar esta tarea, cada nivel añade un bloque de datos al principio y al final del mensaje que indica qué nivel está implicado, además del resto de información que los otros niveles y la máquina que lo va a recibir necesitan para manejar el mensaje de forma adecuada. Los datos dentro del mensaje se ignoran. Esto se denomina encapsulación, ya que cada nivel añade una cápsula de información en torno a los datos originales.

Cada nivel lleva a cabo su propia encapsulación añadiendo cabecera y bloques finales al mensaje que reciben del nivel superior, lo que tiene como resultado seis conjuntos de cabeceras y bloques finales en el momento en que un mensaje llega a la red. Todas estas cabeceras y bloques finales se pasan a la red, que incluso puede añadir más información al principio o al final. (figura 2.3)

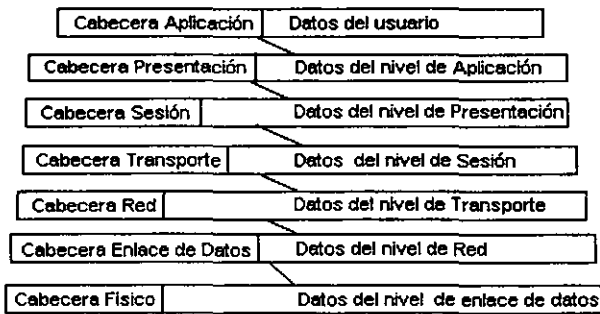


Figura 2.3

Por lo tanto, después de todo esto, ¿cómo es que TCP/IP puede interactuar con el ordenador, independientemente del sistema operativo que utilice? TCP/IP participa en otra arquitectura por niveles aunque es un sistema más obvio que el de OSI o de los modelos TCP/IP. La figura 2.4 nos muestra cómo funciona TCP/IP con el ordenador.

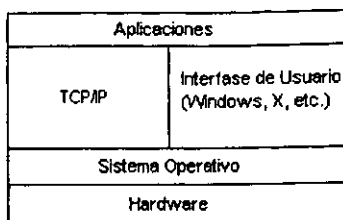


Figura 2.4

TCP/IP en una máquina con Windows se sitúa por debajo de la interfase Windows, pero por encima del sistema operativo. La mayoría de las versiones TCP/IP para Windows disponen de unos controladores especiales añadidos al software de Windows que saben cómo dividir en paquetes en mensaje y enviarlo a la red. Cuando se utiliza una aplicación TCP/IP, el programa toma el mensaje y lo pasa a TCP/IP, que lo procesa y lo envía a través de la red por medio de la utilización del sistema operativo.

Direcciones IP

Todas las máquinas que están conectadas a Internet o cualquier otra red sobre TCP/IP se tiene que identificar de forma única de algún modo. Dentro de una red que se encuentra conectada a Internet, cada máquina tiene que estar identificada de forma única. Sin un identificador único, la red no sabe cómo llevar los mensajes a una máquina. Si existe más de una máquina con el mismo identificador, la red no sabe a dónde tiene que enviar el mensaje.

Internet identifica las redes al asignar una dirección de Internet o, más adecuadamente, una dirección IP, a cada empresa u organismo en la red. Las direcciones IP son todas de 32 bits de largo, y se dividen en cuatro partes de 8 bits, lo que permite que cada parte tenga números que vayan desde 0 a 255, y cada valor de 8 bits está separada por un punto. Por ejemplo 132.248.52.192 y 255.255.27.233 son direcciones IP.

Si la red utilizara los 32 bits de la dirección IP para numerar redes, existirían cerca de cuatro millones de direcciones posibles, más que suficiente para la expansión futura. Sin embargo, algunas combinaciones de bits se reservan para fines especiales, reduciendo el número de direcciones posibles. Los cuatro números de 8 bits se agrupan de formas especiales en función de tipo de red, y así se reduce aun más el número real de direcciones posibles.

Asignar direcciones IP no se reduce a empezar con el número uno y seguir contando ascendentemente, debido a la necesidad de numerar cada máquina (denominada host, un término que se utiliza a menudo cuando se habla de dispositivos de red). En realidad existen dos partes en una dirección IP: el número de red y el número de host dentro de esa red. Al utilizar las dos partes de una dirección IP, las máquinas en redes diferentes pueden tener el

mismo número de host. Sin embargo, puesto que el número de red de las dos redes es diferente, las máquinas se identifican de forma única (los números de las máquinas son los mismos pero los de la red son diferentes). Sin este tipo de esquema, la asignación de números se convierte en poco manejable con gran rapidez.

Las direcciones IP se asignan en función del tamaño de la empresa u organismo. Si la empresa es pequeña, no es necesario que existan muchos identificadores de máquina dentro de la red; por lo tanto, una empresa grande puede tener miles de hosts. Para proporcionar la mayor flexibilidad, las direcciones IP se asignan de acuerdo al tamaño del usuario, denominado Clase A, B o C. También existe una Clase D y E, pero tienen una finalidad especial y no se utilizan como parte de la numeración de direcciones IP.

Estas tres clases permiten que se asignen números de dirección IP en base al tamaño de la empresa. Puesto que 32 bits es el tamaño total permitido, estas clases dividen las cuatro partes de 8 bits en los identificadores de host y la red dependiendo de la clase. Uno o más bits se reservan para el inicio de la dirección IP de 32 bits para identificar el tipo de clase de la que se trate. La organización de las diferentes clases de direcciones IP es la que se muestra en la siguiente figura 2.5

Clases	0	1	2	3	4	B	16	24	31	
A	0	id. de red				id. de nodo				
B	1	0	id. de red				id. de nodo			
C	1	1	0	id. de red				id. de nodo		
D	1	1	1	0	dirección multiemisión					
E	1	1	1	1	0	reservado para usos futuros				

Figura 2.5

Clase	Características	Rango de Direcciones
A	Unicamente tienen una dirección de 7 bits para la dirección de red, pero 24 bits para la dirección de host, lo que permite 16 millones de direcciones de host diferentes, suficientes para las empresas grandes	1.0.0.0 a la 127.0.0.0
B	Tienen 14 bits para la red y 16 bits para el host, lo que permite más redes de clase B pero con menos host. Aún así, 16 bits permiten más de 65,000 hosts.	128.0.0.0 a la 191.255.0.0
C	Pueden tener un máximo de 254 hosts (puesto que los números 0 y 255 están reservados para	192.0.0.0 a la 233.255.255.0

	cada una de las partes de la dirección IP), pero muchas identificaciones de red.	
D y E	Estas redes fueron reservadas para propósitos especiales (Ej. Multicast)	224.0.0.0 a la 255.255.255.255

La mayor parte de las redes son de clase B y C, aunque la decisión final sobre la clase de una red corresponde al Centro de Información de Red (Network Information Center o NIC)

Siempre que se envía un mensaje a un host situado en cualquier parte de Internet, se utiliza la dirección IP para indicar su destino y el host que lo envió. Por supuesto, no se tiene que encargar de guardar registro de todas estas direcciones IP porque ya lo hace otro servicio de TCP/IP denominado Sistema de Nombre de Dominio (DNS).

Sistema de Nombres de Dominio

Cuando una empresa piensa en conectarse a Internet, querrá tener un identificador propio único para ellos. Para obtener uno de estos identificadores únicos de nominados nombre de dominio, la empresa u organismo envía una petición al organismo que controle el acceso a Internet, el NIC. Si el NIC aprueba el nombre de la empresa, se añade a la base de datos de Internet. Los nombres de dominio tienen que ser únicos para impedir confusiones. Parte del nombre de dominio es el identificador de dominio, la parte del nombre que aparece en último lugar (Ej. algo.com). El NIC tiene establecidos seis nombres de dominio que son:

Identificador de dominio	Significado
.arpa	Una identificación ARPANET de Internet
.com	Empresa comercial
.edu	Institución educacional
.gov	Un organismo gubernamental
.mil	Militar
.org	Cualquier cosa que no esté incluida en alguna de las otras categorías

El NIC también permite que letras especiales identifiquen el país de la empresa u organismo. Existen designadores para todos los países del mundo, como por ejemplo .mx para México, .ca para Canadá y .uk para el Reino Unido.

El sistema de Nombres de Dominio (DNS) es un servicio que proporciona TCP/IP que ayuda en el direccionamiento de mensajes. Cuando dirige correo a *hola.saludo.com*, el sistema DNS traduce este nombre simbólico en una

dirección IP al buscar el nombre de dominio en una base de datos. El DNS permite olvidarse de todas esas direcciones IP, permitiendo la utilización de nombres más sencillos: el nombre de dominio. La sintaxis para enviar un mensaje a un usuario en Internet es NombreUsuario@NombreDominio, este ejemplo NombreUsuario es *hola* y NombreDominio es *.saludo.com*

En la práctica, cuando se envía un nombre simbólico a DNS, éste comprueba el host real del usuario, ya que habría millones de direcciones IP en la base de datos; en su lugar, DNS solamente se interesa por la parte de red de la dirección, que traduce a la dirección IP de la red y la envía por la red. Cuando la máquina de Internet de la red que lo recibe toma el mensaje, utiliza una base de datos interna propia para buscar el host del usuario y se ocupa de esa parte del viaje.

SERVICIOS PROPORCIONADOS POR INTERNET

El software que sustenta Internet proporciona un gran número de servicios técnicos sobre los que todo se construye. La mayoría de estos servicios funcionan ocultos, y no hay que preocuparse de ellos. Hay muchos recursos en Internet, y van apareciendo algunos nuevos cada vez que usuarios encuentran una nueva forma de utilizar la red.

En esta sección, se examinarán los recursos más importantes de Internet, con el propósito de dar una idea más precisa de lo que Internet puede ofrecer.

CORREO ELECTRONICO



El correo electrónico (e-mail) es uno de los servicios más utilizados en Internet. El e-mail es fácil de enviar, leer, responder, rápido, práctico y con una gran cobertura. Por estas razones, se ha convertido en el servicio más utilizado, que constituye la columna vertebral de Internet.

Como funciona

El correo electrónico difiere de las otras aplicaciones porque no es un servicio de usuario a usuario: no es necesario que las máquinas emisoras y receptoras del correo electrónico se comuniquen directamente entre sí. Al correo electrónico se le conoce como un servicio de almacenaje y reenvío. El correo pasa de una máquina a otra hasta que llega a su destino final.

La entrega de los mensajes está estandarizada por un sistema llamado SMTP (Simple Mail Transfer Protocol, Protocolo de transferencia Simple de Correo), que es parte de la familia de protocolos TCP/IP. Este describe el formato de los mensajes de correo electrónico y como deben manipularse para

realizar su entrega. Cada computadora de Internet ejecuta un programa de correo que trabaja en "segundo plano" (background) para garantizar que los mensajes son dirigidos y transportados de forma correcta. Este programa, llamado Agente de Transporte, esta siempre listo para responder a cualquier petición que reciba. Un programa llamado Agente de Usuario actúa como interfaz con el sistema de correo de Internet, permitiendo leer los mensajes, redactarlos, eliminarlos, etc.

Cuando se establece una conexión entre dos ordenadores que utilizan SMTP, los dos sistemas intercambian códigos de autenticación, y después de verificar el estado de cada maquina, un sistema envía un comando al otro para identificar al emisor del primer mensaje de correo y proporcionar información básica sobre el mismo. El sistema SMTP receptor devuelve un acuse de recibido después de que se transmite el mensaje.

SMTP es lo suficientemente inteligente para gestionar múltiples destinos para el mismo mensaje de forma eficaz. Si se identifican más de un receptor para el mismo mensaje, dicho mensaje solamente se transmite una vez, pero el sistema receptor se encarga de realizar copias necesarias para cada uno de los destinatarios, lo que disminuye la cantidad de tráfico entre las maquinas.

La mayoría de los sistemas de correo electrónico no imponen limitaciones en los tipos de ficheros que se pueden enviar, siempre y cuando el protocolo de red pueda manejar los caracteres. Sin embargo, obtener en el formato adecuado puede requerir que se ejecute el fichero a través de una utilidad.

Cuando se necesite enviar un archivo binario que no se puede representar como texto habitual, como programas o imágenes, etc., existen facilidades para codificar los datos en texto. De igual forma, una vez que se reciben mensajes codificados, es posible decodificarlos, para guardarlos con su formato original.

Internet se basa en caracteres ASCII de 7 bits, lo que es ideal para la mayoría de los mensajes de texto. Aunque pueden suceder problemas con los archivos binarios y algunos idiomas como el chino o japonés, existen soluciones disponibles para estos ficheros. El sistema de Internet es incapaz de manejar binarios en su formato original. Cualquier cosa que se envíe de esta manera se truncara a ASCII de 7 bits, haciendo virtualmente imposible reconstruir el mensaje original. Para poder enviar archivos binarios en Internet se utiliza dos programas: uno que convierte caracteres de 8 bits en una representación de 7 bits y otro que haga lo inverso.

Una vez que ya se conoce esta advertencia, Internet no se preocupa del contenido de los ficheros que envía. Si suponemos que se han convertido de la forma adecuada para permitir las transmisiones, entonces se puede enviar ficheros de sonido, gráficos, películas multimedia y cualquier otro tipo de

aplicaciones que desee. Internet no hace caso del contenido de los mensajes, sino que solamente se concreta en la información de las cabeceras de los mensajes que TCP/IP añade.

El correo electrónico tiene una ventaja sobre Internet cuando se refiere a transferir ficheros de una aplicación de usuario, ya que permite que se ignoren por completo los formatos de los discos. Un fichero escrito en Macintosh Word se puede enviar a un usuario en un PC. La versión de PC de Microsoft Word reconocerá el formato de fichero como Macintosh Word y le ofrece llevar a cabo una conversión por usted. Este tipo de conversión se aplica a la mayoría de los formatos de aplicaciones, haciendo del correo electrónico un medio eficaz de distribuir ficheros. Por eso, se puede enviar por correo electrónico cualquier tipo de archivo a cualquier persona.

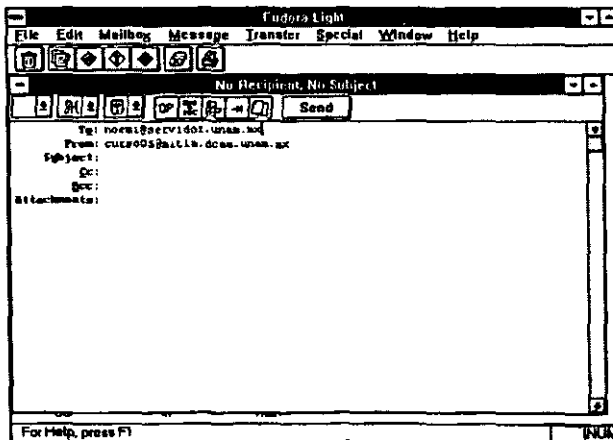
Un ejemplo

Una dirección electrónica esta constituida por dos partes el Nombre de Usuario y el nombre de la máquina donde se recibe el correo. Así que tenemos lo siguiente:

$$\underbrace{\text{noemip}}_{\text{Nombre de Usuario}}@ \underbrace{\text{servidor.unam.mx}}_{\text{Nombre del Servidor}}$$

Para poder enviar un correo, se tienen varias opciones entre las más conocidas y utilizadas esta el programa Eudora, que es una programa que correo en ambiente Windows y es fácil de usar y configurar.

En este tipo de programas se tiene un conjunto de opciones, como son: configuración, enviar mensajes, borrarlos, mandar archivos (opción attachment) con un formato distinto, etc.



CONEXION REMOTA



Hace tiempo, la mayoría de la informática automatizada en el mundo se realizaba en grandes máquinas con una enorme unidad central de procesos cada una de ellas. Los usuarios accedían a estas máquinas a través de estaciones de trabajo individuales denominadas terminales. A diferencia de una PC, un terminal no posee un procesador propio, sino que tiene que estar conectado a un ordenador principal para que pueda funcionar.

Los fabricantes hicieron terminales bastante estándares de modo que se podían utilizar junto una gran variedad de tipos de ordenadores. Digital Equipment Corporation (conocida como digital o DEC) fabricó una línea de terminales que eran bastante comunes para que se utilizaran con ordenadores grandes denominados la serie VT. El VT100 y VT102 (una ligera variación) eran probablemente los tipos de terminales más comunes.

Cuando la gente corriente empezó a utilizar Internet por primera vez, la mayor parte de los ordenadores en la red estaban configurados para enviar y recibir información a través de terminales como la entonces común VT100. Si se desea poner en contacto con uno de estos ordenadores a través de su PC, se necesita un programa que "anuncia" a su PC, haciendo que actúe como una terminal y permitiendo introducir y recuperar datos de las máquinas grandes. Los programas como el programa Terminal (Telnet) incluido con Windows tienen la intención de hacer que una PC se parezca, o emule, a estos terminales.

Telnet es un programa que nos permite establecer una sesión de trabajo con una computadora remota a través de Internet. Una vez que se ha establecido la conexión, se puede utilizar esa computadora en la forma habitual (claro, si se posee una cuenta válida).

Como un servicio público, muchos servidores de Internet permiten a cualquier usuario conectarse utilizando la cuenta especial llamada guest.

La sesión puede ser en una máquina en la misma oficina, en la misma universidad o en cualquier parte del mundo. Con este servicio se puede tener acceso a todos los servicios que esa máquina provee a sus terminales locales.

Como funciona

Esta aplicación consta de dos partes de software que colaboran entre sí: el cliente, que corre en la computadora que solicita el servicio, y el servidor, que corre en la computadora que provee el servicio.

El programa cliente de Telnet, hace lo siguiente:

- 1) Crea una conexión de red TCP con un servidor
- 2) Acepta lo que se escriba en el teclado proveniente del usuario
- 3) Reformatea la entrada de datos a un formato estándar y la envía al servidor
- 4) Acepta la respuesta del servidor en un formato estándar

5) Reformatea esa salida para desplegarla en la pantalla del usuario

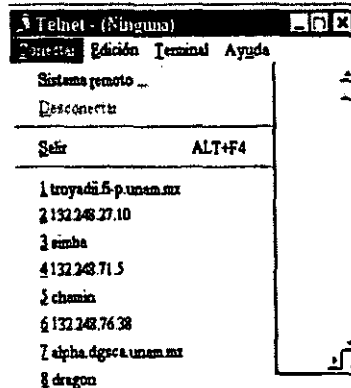
Cuando el servidor se encuentra listo para aceptar las solicitudes de servicio realiza lo siguiente:

- 1) Informa al software de red que esta listo para aceptar conexiones
- 2) Espera una solicitud en un formato estándar
- 3) Atiende la solicitud
- 4) Envía los resultados de regreso al cliente en un formato estándar
- 5) Vuelve a entrar en el proceso de espera

Conectarse a un ordenador remoto utilizando Telnet.

Para iniciar una sesión se necesita ejecutar el programa Telnet.

Se selecciona del menú la opción Conectar para establecer la conexión al servidor que se desea acceder, si el nombre del servidor esta en la lista, solo con seleccionarlo se puede tener una conexión, de lo contrario se tendrá que escribir el nombre del servidor al que se desea acceder.



Cuando ya se tiene la conexión al servidor, se tiene que escribir el nombre de la cuenta del usuario (login), después se escribe el password, si el password es correcto, se inicializa la sesión en el servidor, de lo contrario, este mismo le pedirá que pruebe otra vez, hasta que sea correcto, o hasta un determinado numero de veces. Si se posee un identificador de usuario y palabra clave validos, se puede conectar con cualquier computadora de Internet.

USENET

Usenet (contracción de "User's Network", red de usuarios), es uno de los principales servicios de Internet. Usenet no es una red. Es un sistema de grupos de discusión en el que artículos individuales se distribuyen por todo el

mundo. En el cual se llevan acabo intercambio de noticias, opiniones, boletines, mensajes y artículos entre los usuarios de las grandes redes, una de las más grandes de todas, Internet.

Usenet tiene miles de grupos de discusión. En cada nodo de Internet, el administrador de la red decide que grupos de Internet quiere hacer públicos y que grupos quiere recibir. Por esta razón, Usenet no esta disponible en todas partes. Además, aquellos nodos en los que exista este recurso, no tienen necesidad expresa de importar todos los grupos de discusión.

Como funciona

Funciona como una extensión del Correo Electrónico, bajo un nombre de 'alias' (una dirección ficticia que contiene una lista con varias direcciones electrónicas). Utilizando este recurso cualquier mensaje enviado al 'alias' es automáticamente reenviado a todas las direcciones contenidas en ella.

Las listas de discusión también pueden ser difundidas a través de programas conocidos como servidores de listas, utilizados originalmente en la red BITNET. Aparte del intercambio de mensajes entre los participantes de la lista, los servidores de listas ofrecen recursos adicionales, tales como consultas a registros de mensajes enviados/recibidos, almacenamiento y recuperación de documentos de interés, e información sobre los participantes de la lista.

Las listas de discusión son conocidas también como conferencias electrónicas, son comúnmente usadas como medio de comunicación entre miembros de un proyecto o entre personas interesadas en discutir temas específicos, pudiendo ser abiertas o restringidas para la participación de nuevos miembros. En el caso de las listas abiertas, el pedido de suscripción de un nuevo miembro se hace enviando un mensaje por parte del interesado a una dirección electrónica diferente de la de la lista, creada especialmente para ese fin. Existe una infinidad de listas de discusión sobre los mas variados asuntos, accesibles a una red.

Para hacer uso de listas de discusión es preciso que el usuario tenga acceso al correo electrónico.

FTP



Tanto Internet como sus predecesores se pensaron como medios para intercambio de información. Los servicios más antiguos en Internet se desarrollaron para cumplir este fin. Uno de esos servicios es FTP, que nos permite examinar los ficheros en servidores (hosts) remotos de Internet y transferir ficheros entre su host y estos hosts. FTP son las siglas de File Transfer Protocol, (Protocolo de transferencia de archivos), que es uno de los protocolos más usados, de la familia de TCP/IP.

El servicio FTP es un ejemplo de un sistema cliente / servidor. En este tipo de sistema, se utiliza un programa en el ordenador local (denominado cliente) para solicitar un servicio de un programa en un ordenador remoto (denominado servidor).

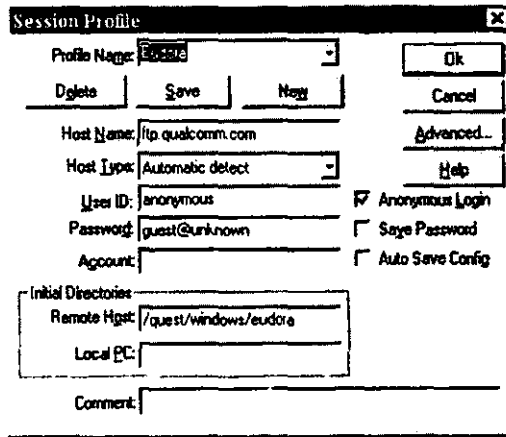
Para que un sistema de ordenador nos permita conectarse a él por medio de un programa FTP, el sistema debe tener funcionando un servidor FTP. Los administradores de la máquina deben configurarla y decidir los ficheros y la información que se va a encontrar disponible en el servidor FTP.

Un tipo común de servidor FTP es un servidor FTP anónimo. Con este tipo de servidor, se puede conectar y cargar o enviar ficheros sin la necesidad de tener cuenta en la máquina. Si el servidor FTP no es anónimo, al conectemos el servidor nos pediría un nombre de usuario y password, del mismo modo como si estuviera accediendo a la máquina. Sin embargo, en un servidor FTP anónimo, se utiliza el nombre de usuario especial anonymous, este nombre de usuario anónimo nos permite entrar al proporcionar cualquier password que desee (a menudo, el servidor le pide que teclee su dirección completa de Internet como el password) a una cuenta especial donde se encuentran los archivos que deseamos.

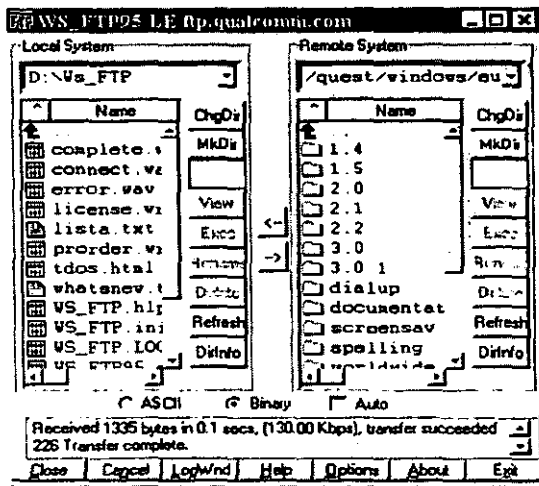
Los servidores FTP anónimos representan uno de los medios más importantes de distribución de software e información por la toda Internet (podría decirse que es un servicio público). Se encuentra disponible una gran cantidad de software, a menudo facilitada gratuitamente. El software se encuentra disponible para muchos tipos diferentes de sistemas de ordenador, como los sistemas UNIX, IBM PC y Macintosh.

Existen programas que nos permiten realizar la transferencia de archivos de forma rápida y sencilla, uno de ellos es el denominado Ws_ftp. Ws_ftp es un cliente FTP bajo Windows.

Cuando ejecutemos este programa, nos pedirá una serie de información, como es el nombre o dirección del hosts al que se desea acceder, tipo de hosts (si es que se conoce, o en caso contrario se deja la opción Automatic detect), nombre del usuario, password (en caso de tenerlo o dirección de correo), etc.



Cuando ya se ha hecho la conexión parecerán los archivos contenidos en el servidor. Para poder copiar cualquier archivo solo con seleccionarlo y accionar la flecha ←, se transferirá el archivo al directorio actual.



ARCHIE



Archie es una herramienta que sirve para la búsqueda de archivos contenidos en servidores FTP anónimo. Este servicio consiste en que los servidores, llamados servidores Archie, acceden regularmente a todos los servidores de FTP anónimo y guardan en una base de datos el listado de todos los archivos que estos servidores de FTP tienen para acceso público. Esta base de datos recibe el nombre de base de datos de archivos Internet (Internet Archives Database). Cuando queremos

saber donde se encuentra un determinado archivo, sólo tenemos que buscarlo en esa base de datos.

Los distintos servidores Archie se encuentran repartidos por todo el mundo, y normalmente cada servidor se encarga de consultar regularmente los servidores de FTP anónimo de su área. La información que recoge cada servidor Archie es compartida con el resto de los servidores, de forma que todos los servidores Archie disponen de la misma información.

El programa Archie fue desarrollado originalmente como un proyecto de los estudiantes y del personal voluntario de la McGill University School of Computer Science de Montreal, Canadá. El software fue escrito por Alan Emtage y Bill Heeln, con la ayuda de Peter Deutsch.

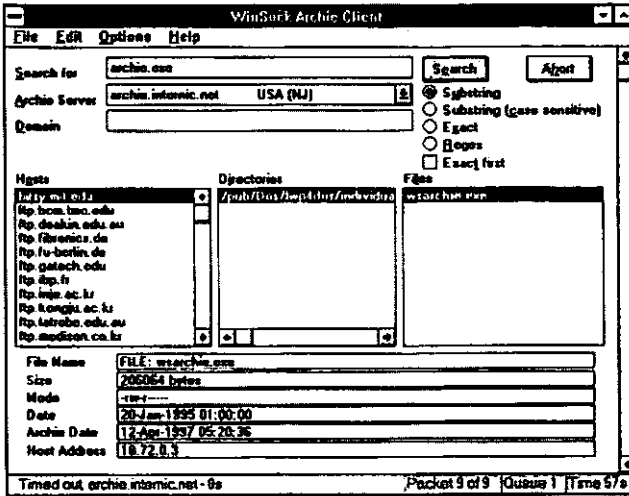
Hay miles de servidores FTP anónimo alrededor del mundo ofreciendo una cantidad inmensa de archivos. El papel de los servidores Archie es ayudar a localizar donde se encuentra la información que se necesita.

Si se considera los servidores de FTP Anónimo de todo el mundo como una enorme biblioteca mundial, que esta cambiando continuamente, se puede considerar a los servidores Archie como un catalogo. Realmente, sin los servidores Archie, la mayoría de los recursos existentes en los servidores FTP Anónimo serian inalcanzables.

Existen tres formas en que se puede utilizar los servidores Archie. La primera es utilizar Telnet para acceder al ordenador donde se ejecutará automáticamente un programa que nos permitirá realizar la búsqueda. La segunda forma consiste en enviar un mensaje vía correo electrónico al servidor de Archie solicitando la información necesaria. A vuelta de correo, Archie responde a la petición. La tercera y la más fácil, es tener en nuestra máquina un programa cliente de Archie, las consultas son realizadas en este caso al programa cliente, y este se pone en contacto con el servidor Archie para buscar las respuestas.

Existe como en todos los anteriores servicios una aplicación que nos permite acceder a este tipo de servicios, en este caso es el programa Ws_archie.

En la siguiente figura se muestra la pantalla con una búsqueda realizada en el programa cliente Ws_archie de Windows.



INTERNET RELAY CHAT



La utilidad Internet Relay Chat (IRC) es análoga a la utilidad Talk pero pueden utilizarla más de dos personas a la vez. Como se es de imaginar, IRC es usado frecuentemente y ofrece mucho más que una simple conversación. Se puede tomar parte en conversaciones públicas con un gran número de personas. Estas conversaciones se organizan sobre distintos temas o ideas.

IRC es un servicio mediante el cual se pueden celebrar conversaciones On Line sobre cualquier tema concreto, con otras personas o grupos con intereses comunes, ubicados en cualquier parte del mundo. Opinar, participar, comunicarse o simplemente divertirse, son los objetivos de este servicio.

Es también la forma más simple y económica que existe de establecer una comunicación en tiempo real con personas que pueden estar al otro lado del planeta.

Para poner tener este tipo de conversación se necesita un programa llamado mIRC, con el se pueden establecer comunicación con diferentes foros en todo el mundo, basta con poner nuestro correo electrónico, un alias, y escoger el tipo de foro al que se desea acceder y listo para hablar.

GOPHER



Gopher es una de las herramientas de búsqueda disponibles en Internet, la cual nos permite buscar recursos utilizando menús. Cuando nos encontramos algo que nos gusta podemos leerlo o tener acceso a él a través de Gopher sin necesidad de preocuparse por los nombres de dominio, los domicilios IP o por cambiar de programas.

La ventaja real consiste en que permite curiosear a través de los recursos de Internet, sin importar su tipo, tal como si se hojeara el catalogo de una biblioteca.

Origen de Gopher. El servicio nació en la Universidad de Minnesota. Este servicio fue diseñado para que cada parte de la burocracia pudiera tener control sobre su propio servidor y sus propios datos. Es decir, que la administración de la escuela pudiera contar con una computadora en su edificio administrativo para distribuir información al personal administrativo; Podrían existir tantos servidores como grupos que quisieran proporcionarlos. Entonces, los inventores de Gopher crearon una aplicación especial que pudiera guiar a los estudiantes a la información que requerían, sin necesidad de capacitación. Para lograr esto, organizaron el sistema por temas, de manera que es posible verlo como una enorme base de datos, en lugar de cientos de bases de dato pequeñas interconectadas. Se puede tener acceso a archivos FTP, números telefónicos de servidores white pages, catálogos de biblioteca y otros servidores de bases de datos con propósito especiales (basados en Telnet). Pero solo Gopher sabe donde se localizan realmente los datos, la forma de tener acceso a ellos y que existen muchos servidores que los proporcionan. No tomo mucho tiempo darse cuenta de que si el sistema funcionaba bien para varios servidores colocados en diferentes departamentos, también podían funcionar para servidores colocados en diferentes partes del mundo. Todo lo que se requería era que Internet enlazara a todos. En cuatro años, mas o menos, la utilización del sistema Gopher creció de un servidor a más de 1300 servidores.

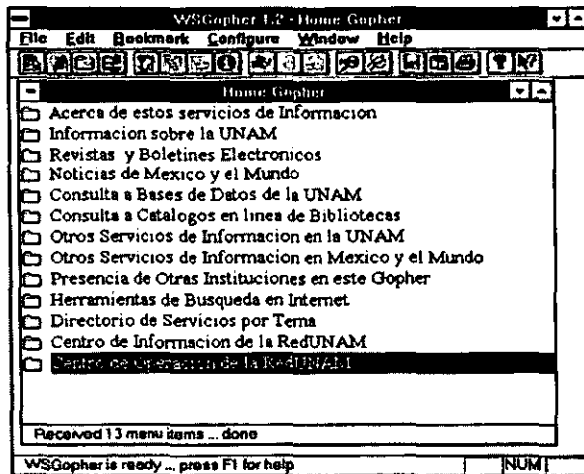
Gopher proporciona una serie de menús desde los cuales se puede acceder virtualmente a cualquier tipo de información textual, incluyendo la que proporcionan otros recursos de Internet. Hay muchos sistemas Gopher en torno a Internet, cada uno administrado localmente. Cada Gopher contiene cualquier información que las personas que administran el Gopher local han decidido compartir.

Mientras algunos servidores Gopher son sistemas aislados, la mayoría de los servidores Gopher están instalados de forma que pueden conectar con otros servidores.

Como funciona

Cuando se pone en funcionamiento por primera vez un cliente Gopher, este se enlaza con su servidor base y le solicita el menú principal. El servidor envía dicho menú y cierta información oculta a su cliente. la información oculta indica a este ultimo lo que representa cada elemento del menú (por ejemplo, un archivo de texto, un directorio, un equipo anfitrión, un numero de puerto para utilizar ,etc.), también le indica el Protocolo Internet (IP) del servidor para ese elemento, el número del puerto que debe utilizar y una ruta de acceso para un archivo. El dominio IP puede ser el servidor mismo, si es ahí donde reside el recurso; pero también puede ser cualquier otro servidor ubicado en un lugar diferente. En realidad no tiene importancia, cuando se selecciona un elemento del menú, el cliente realiza el mismo procedimiento. Su cliente conserva la posición original (en caso de que quiera regresar) y se enlaza con el nuevo servidor. Entonces, el proceso se repite.

Como en todos los servicios, existe también un cliente gopher para Windows, llamado Ws-gopher. Del menú que exhibe el Gopher, sencillamente escoge la información que se desea ver.



VERONICA

Esta es una herramienta utilizada en gopher para la búsqueda de un tema o archivo determinado. Desarrollado en la Universidad de Nevada Veronica (Very Easy Rodent-Oriented Net-wide Index of Computerized Archives, o sea Manejo sencillo de índices de archivos computarizados para redes globales) extrae datos exactamente igual que Archie: visita servidores Gopher alrededor del mundo y consulta sus menús, recuerda lo que hay en cada uno de ellos y construye un índice combinado para presentarlo en un

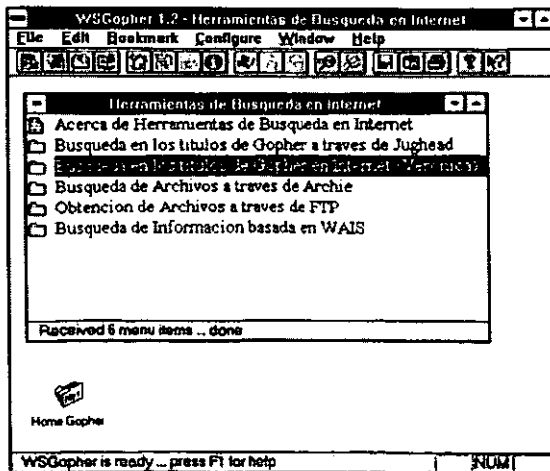
menú Gopher. Parece funcionar como otra búsqueda de índices. Su uso lo selecciona como cualquier otro elemento de un menú de búsqueda. A su vez, Veronica le solicita palabras clave para configurar la búsqueda e integrar los resultados en un menú especial. Los elementos de menú son elementos provenientes de todos los Gopher alrededor del mundo los cuales contienen en su nombre las palabras que se ha especificado. Para tener acceso a Veronica, se revisa cualquier servidor Gopher en busca de elemento de menú como el siguiente:

```
## Search titles in Gopherspace using Veronica <?>
```

Se selecciona ese elemento; y mostrara un siguiente menú de selección, lo primero que debe notar es que existen varias opciones de menú, asociadas a diferentes servidores Veronica. La cantidad de servidores varia según el lugar y de tiempo en tiempo. Algunos servidores Gopher son lo suficientemente inteligentes como para saber cuando el servidor Veronica no se puede utilizar y lo borran de la lista automáticamente. En cambio, otros dejan la lista completa para que se haga el intento y en ocasiones falle.

Veronica es una herramienta que permite mantener la pista de muchos menús de Gopher alrededor del mundo. Se puede utilizar Veronica para realizar una búsqueda y localizar todas las opciones de menú que contiene ciertas palabras clave.

El resultado de una búsqueda con Veronica es un menú, que contiene todos los elementos que se han encontrado. Seleccionando cualquier elemento de este menú, automáticamente conectamos con el Gopher apropiado, donde quiera que este. De hecho, a menos que se especifique lo contrario, no se sabrá que computadora se esta utilizando ni de que país.



WAIS

El nombre de WAIS viene de "Wide Area Information Service" (Servicio de Información de Area Extensa). Esta herramienta es excelente para realizar búsquedas a través de material indexado y localización de artículos basándose en su contenido. Esto significa que WAIS permite realizar exploraciones a través de archivos de Internet en busca de documentos que contengan ciertos grupos de palabras. En realidad WAIS es una herramienta que trabaja con conjuntos de datos o con bases de datos.

Como funciona

WAIS es un sistema distribuido de exploración de texto. Esta basado en una estándar (llamado Z39.50) que describe la manera en que una computadora solicita a otra que realice una búsqueda en lugar de ella. WAIS es uno de los primeros sistemas basados en este estándar de borrador, en la actualidad es también el muy utilizado. Para hacer que un documento este disponible a través de un servidor WAIS alguien debe crear un índice para ese servidor con el fin de que sea utilizado en la búsqueda. Para información textual, cada palabra en el documento por lo regular esta indexada. Cuando se hace una solicitud de una búsqueda desde un cliente WAIS, este se enlaza con el servidor que controla las bibliotecas que se han sugerido. El cliente WAIS solicita a cada servidor, en su momento, que explore su índice en busca de un conjunto de palabras. Entonces, el servidor envía una lista de documentos que pueden ser los apropiados y una "calificación" que le indica que tan apropiado supone que es cada uno. Las calificaciones son comunes, de tal manera que el documento que mejor coincida con su criterio de búsqueda recibe 1000 puntos y los demás van descendiendo proporcionalmente. Después de que todas las bibliotecas han sido exploradas, WAIS proporciona los títulos de los documentos cuya calificación es más alta. Existe un limite para la cantidad de documentos que era reportada, el cual por lo regular varia entre 15 y 50 dependiendo del cliente que se esta utilizando. Después se puede seleccionar los documentos que nos interesan ver y WAIS los desplegará.

WAIS puede acceder a un gran número de bases de datos. Para empezar, se le dice WAIS en que base de datos se quiere hacer la búsqueda. Después WAIS buscara cada palabra en cada articulo en todas las bases de datos que se le indiquen.

WORLD WIDE WEB



El servicio World Wide Web a menudo llamado WEB es una herramienta basada en hipertexto que permite recuperar y mostrar información basada en búsqueda de palabras clave. Lo que hace al servicio World Wide Web tan potente es la idea de hipertexto: datos que contienen a otros datos y también poder ver imágenes.

Sobre este tema se hablará con más detalle en el siguiente capítulo.

WORLD WIDE WEB

La complejidad para su acceso, fue siempre el motivo principal que mantuvo a Internet fuera del interés público en general. Este cúmulo de información carecía de forma y contenido coherente ante los ojos de expertos de otras disciplinas totalmente ajenas a las telecomunicaciones. Esta característica era un beneficio y un obstáculo al mismo tiempo, ya que por un lado se mantenía a una comunidad inmersa en este nuevo proyecto integrada y en contacto, pero en otros aspectos resultaba contraproducente al ser este círculo tan reducido, se incrementaban los costos y se caía en el elitismo.

Hasta hace unos años Internet era simplemente el envío de mensajes a través de computadoras. La recuperación de la información se mostraba compleja e inaccesible, exclusiva para aquellos inmersos en conceptos teleinformáticos y profundos. La presentación de la información se volvió una cuestión interesante, ya que había que crear un sistema que permitiera un fácil manejo de la misma.

HISTORIA

En 1980 Tim Berners-Lee, graduado de la Universidad de Oxford y consultor del CERN (Laboratorio Europeo de Partículas Físicas), escribe un programa que permite crear ligas en modos al azar. Cada nodo tiene un título, un tipo y una lista de ligas con un tipo específico bidireccional; la idea era crear

una nueva clase de sistema de información en el cual los investigadores podrían colaborar e intercambiar información durante el curso de un proyecto en tiempo real, además de que pudieran colaborar a su vez en varios proyectos al mismo tiempo. A principios de los 90 Robert Cailliau, también miembro del CERN, se interesa por el proyecto y colabora en la propuesta. Aparecen nuevos conceptos, entre ellos, el de hipertexto³.

Berners utilizó la tecnología del hipertexto para enlazar de manera conjunta documentos como si fueran una telaraña (de ahí lo de WWW) que podían ser atravesados de cualquier manera para buscar información. El WWW no implica un orden como en los libros o listas; en esencia permite muchas relaciones posibles entre cualquier documento individual y otros documentos. Berners implementó un sistema de navegación de hipertexto permitiendo a los usuarios moverse libremente entre documentos sobre la red, independientemente de su localización.

A fines de 1990, se introdujo el primer software WWW en una computadora NeXT de Steven Job's. El software NeXT permitía ver y transmitir documentos de hipertexto en Internet y facilitaba la edición a los usuarios (el primer visualizador del WWW consto de programas basados en texto que con simples comandos del teclado permitían al usuario seguir los enlaces del WWW). Se hicieron demostraciones de este sistema en los comités del CERN y ante los asistentes a la "Conferencia de Hipertexto" de 1991. En los años siguientes, el sistema WWW se expandió con rapidez.

El Centro Nacional para Aplicaciones de Supercomputo (NCSA) ayuda a la comunidad de investigación científica a desarrollar y distribuir software para la investigación de nueva tecnología que pueda transferirse al sector privado para aplicaciones comerciales. Al iniciar 1991, Marc Andreessen y un grupo de programadores de NCSA conocido como "Grupo diseñador de software", utilizaron fondos gráficos para escribir un programa que se convierte de hecho en la interfaz gráfica del WWW: Mosaic.

Mosaic fue uno de los primeros visualizadores (programas de recorrido del WWW); este continua siendo un programa gratuito y del dominio público, y no hay nada más agradable en el mundo de las computadoras que un software gratis. Mosaic está disponible en la mayoría de los servidores FTP en Internet

El WWW comenzó a hacerse más popular en 1992 y se le puede definir como un sistema de distribución hipermedia de cobertura amplia. Las páginas WWW o Home Pages consisten en enlaces de hipertexto y de hipermedia⁴ que pueden incorporar video en movimiento y sonidos, estas páginas son instaladas en todo el mundo para proporcionar un caudal de información útil.

³ El hipertexto, se refiere de manera específica a documentos computacionales donde los lectores se pueden mover de un lugar a otro, de un documento a otro entre documentos, de una manera no secuencial ni lineal.

⁴ Son conexiones visuales gráficas, mensajes de audio o video, así como de texto

ELEMENTOS QUE INTEGRAN AL WWW

El WWW es un sistema distribuido, pues sus partes y piezas se localizan en diferentes servidores de todo el mundo, cada uno de los cuales utiliza enlaces para conectar información y recursos en otro servidor WWW.

Este sistema enlaza la mayoría de los recursos que existen en Internet. Para los usuarios, el WWW es más fácil de usar que otros sistemas de navegación de Internet como por ejemplo: Archie, Gopher y WAIS. La razón es que el WWW integra diferentes formatos de información: imágenes fijas, texto, audio y video. Los enlaces de WWW pueden apuntar a información no basada en medios. Un enlace puede abrir una localidad FTP o empezar una búsqueda de información con el sistema Gopher.

La primera información que se recibe desde un servidor WWW se conoce como pagina base (home page). La pagina base es una interfaz inicial a una serie de documentos, archivos y recursos que residen en la computadora o en otros servidores. Los enlaces en documentos WWW usan direcciones para conectarse a los recursos a los cuales apuntan. Estas direcciones (o URL's), pueden representar un enlace a casi cada documento o archivo de Internet.

Estructura de las direcciones (URL's)

El formato de las direcciones de sitios WWW para su localización es denominado URL (Uniform Resource Locator). El URL está formado por caracteres que identifican un documento en particular que puede ser casi de cualquier tipo. Cada una de estas direcciones identifica a un solo recurso de información (ya sea, página, imagen, archivo, etc.). El URL consta de cuatro partes:

- Tipo de Protocolo (puede ser telnet, ftp, gopher, http o file.)
- Nombre de la computadora
- Directorio de localización del documento
- Nombre del documento

Para acceder a cualquier dirección en el WWW, se sigue la sintaxis:
Protocolo://Computadora/Directorio/Documento

Para acceder a direcciones de este tipo se cuenta con la posibilidad de ingresar por programas con interfase de texto solamente y con programas de entorno gráfico que son los más utilizados. La ventaja de la interfase de texto es que no requiere de demasiado tiempo para ver el contenido de un documento y transferirlo en caso de requerirlo. En cambio en el ambiente gráfico requiere un poco más de tiempo, por tener que presentar a demás del texto, las imágenes, el audio y el video en caso de tenerlos.

HTTP

HTTP(Hyper Text Transfer Protocol), el protocolo de transferencia de hipertexto, es un protocolo de red creado para el WWW. HTTP es nada menos que un sistema de información de hipermedia de colaboración distribuido. Según Tim Berners-Lee, es un "protocolo orientado a objetos sin fundamento genérico". Este tipo de protocolo de red es muy sencillo de usar y fácil de implementar, características que representan dos de los principios básicos que hay tras HTTP.

Ciertas actuaciones, denominadas métodos, están asociadas con el protocolo orientado a objetos de HTTP. Estos métodos definen extensiones para los comandos de HTTP y pueden ser asociados con tipos particulares de objetos de red tales como documentos, ficheros o servicios de red.

La finalidad de HTTP era hacer posible muchas fuentes de información en red, relacionadas vía Internet. Es posible navegar por estos centros de información para así poder seguir rápidamente referencias desde cualquier recurso de información a otros, los cuales pueden estar localizados en lugares remotos.

Como en cualquier otros sistema de hipermedia, los tiempos de respuesta deberían ser bastante rápidos. Berners-Lee indica que no deberían ser necesarios más de cien milisegundos para un enlace o salto hipermedia. Esto requiere un protocolo de red de recuperación de información muy rápido y mientras no se consiga esto completamente (especialmente en navegadores de WWW conectados vía telefónica), el WWW es muy fácil de utilizar requiriendo poco esfuerzo para movernos a través de los enlaces desde un documento a otro.

El trabajo de HTTP incluye la búsqueda y recuperación, actualizaciones y comentarios en documentos. HTTP permite el diseño y expansión de un conjunto de métodos ampliables. Se construye sobre la disciplina de referencia dada por la URI (Universal Resource Identifier, identificador de recurso universal) un proyecto muy bien estructurado que se origino con el IETF(Internet Engineering Task Force). Hay dos tipos de URI, el URN (Universal Resorce Name, nombre de recurso universal) y el URL (Universal Resource Locator, Localizador de recurso universal).

HTTP también ofrece una representación de datos dinámica a través de negociaciones cliente-servidor. Esto permite que el sistema de información del WWW se construya independientemente del desarrollo y evolución de los nuevos sistemas de representación

Como funciona el HTTP

Dentro de Internet las comunicaciones en red tienen lugar mediante conexiones TCP/IP. Ocasionalmente, se puede dar ciertas situaciones en las

cuales este conjunto de protocolos esta implementado sobre otro protocolo de red. En este caso, los objetos HTTP pueden ser mapeados sobre otros mecanismos de transporte distintos. Se trata de un proceso sencillo y directo que por difícil que parezca, pasa desapercibido a los usuarios del WWW.

El protocolo HTTP es independiente del estado, es decir, que ni el cliente ni el servidor almacenan información sobre el estado del otro lado de una comunicación en curso. El servidor no necesita almacenar información relacionada con el último cliente al que ha servido y, por su parte, el cliente tampoco necesita saber los pasos seguidos por el servidor. Cada componente tiene su propia ocupación y maneja la información por si mismo.

La mayor parte de los protocolos de red dependen del concepto de transacción, que consiste en los siguientes puntos:

- El establecimiento de una conexión entre un cliente y un servidor para que, de esa manera, se produzca la comunicación.
- Una petición por parte del cliente al servidor de servicios específicos, recursos, u otros tópicos de comunicación conocidos.
- Una respuesta por parte del servidor a la petición del cliente, aportando el recurso requerido o una explicación de porque no puede proporcionarlo.
- Fin de la conexión y cierre de la comunicación de petición-respuesta.

Este método (conexión, pregunta, respuesta, cierre) describe la forma de intercambio básico entre clientes y servidores en todo el mundo, ya sea dentro o fuera del ámbito WWW.

Conexión

Se trata del establecimiento de una conexión TCP/IP de red desde el cliente al servidor WWW en Internet. Históricamente, el sistema operativo dominante dentro de Internet ha sido UNIX.

Cuando se utiliza TCP/IP sobre un servidor UNIX, el puerto 80 es el puerto más conocido para establecer las comunicaciones del WWW. Otros puertos no reservados pueden estar específicamente solicitados para que puedan ser usados. Esta especificación se realiza sobre el mismo URL añadiendo el carácter ':' a continuación del nombre del servidor al que queramos conectarnos.

Petición

La petición se inicia desde un cliente del WWW en forma de mensaje de petición dirigido a un servidor en particular. Esta petición incluye el método que se debe aplicar al objeto de red solicitado, el identificador del objeto y la versión del protocolo HTTP utilizado, todo ello seguido de más información codificada en un tipo particular de cabecera.

Respuesta

La respuesta surge por parte del servidor y se dirige al cliente que originó la conexión. Consiste en tres partes: una línea de estado de respuesta, una cabecera de respuesta y los datos de respuesta. La línea de estado contiene los siguientes campos:

- Un identificador que permite reconocer la versión de HTTP utilizada por el servidor.
- Un código de estado que describe los resultados de los intentos del servidor para descifrar y comprender las peticiones del cliente.
- Una explicación muy codificada de la respuesta por parte del servidor.

La cabecera de respuesta puede contener información específica sobre el objeto requerido y, además, se puede pedir cualquier declaración MIME (Multipurpose Internet Mail Extension) para dar la respuesta.

La cabecera de respuesta no contiene nada diferente de la información contenida en la cabecera de petición suministrada por el cliente (y es vista como una forma de confirmación de la solicitud realizada).

Los datos de respuesta aportan todo tipo de información adicional que sigue a la cabecera. Estos datos normalmente están codificados en la forma de un cuerpo de mensaje MIME. La relevancia de estos datos depende por completo del código de estado que devuelve el servidor al cliente.

De igual manera, la declaración MIME (Content-Type) que se usa para los datos puede ser cualquier tipo que el cliente haya especificado que pueda aceptar. Los dos tipos de contenido principales son Text/plain (caracteres ASCII normal) y Text/HTML (documentos de html normales).

De acuerdo con las especificaciones de HTTP, se puede asumir con seguridad que cualquier cliente del WWW es capaz de manejar estos dos tipos de contenido.

Cierre

Supone el fin de una conexión TCP/IP, la cual ha podido ser iniciada previamente tanto por el cliente como por el servidor.

Los formatos específicos de los elementos de petición y respuesta definidos en la especificación de HTTP se pueden encontrar en:

<http://www.w3.org/hypertext/WWW/Protocols/HTTP/HTTP2.html>

Toda la información contenida en una cabecera HTTP se representa normalmente utilizando el conjunto de caracteres ISO-Latin-1, donde cada línea finaliza con un par retorno/avance de línea (normalmente escrito CrLf). En la especificación HTTP también es posible la transmisión de objetos en diferentes métodos binarios.

Un ejemplo

Para comprender mejor el funcionamiento de este sistema aquí se explica un ejemplo común.

El cliente se conecta con el servidor `www.netscape.com` y pide el encabezado de la página inicial de Netscape.

- Cliente: `HEAD / HTTP/1.0`

El método `HEAD` sólo pide el encabezado. La `/` se refiere a la entidad⁵ por defecto, la página inicial `HTTP/1.0` indica la versión de HTML de un mensaje de petición completa

A continuación el cliente envía un encabezado general con la dirección de email asociada al navegador

- Cliente: `From: bob@or.com`

El cliente también indica la página que lo refería a `www.netscape.com`

- Cliente: `Refer: http://search.yahoo.com/bin/search?p=netscape`

El cliente sigue a esto con una línea en blanco, tal como lo requiere HTTP.

- Cliente:

El servidor responde con el código de éxito `200`, que indica que se interpretó satisfactoriamente el mensaje y que la entidad pedida existe

- Servidor: `HTTP/ 1.0 200 OK`

A continuación, el servidor responde primero con los encabezados generales. Esto incluye al tipo de servidor.

- Servidor: `Server Netscape-Communications/4.01`

fecha y la hora actuales

- Servidor: `Date: Thursday, 28-Mar-97 05:30:32 GMT`

y la fecha y la hora en la que se modificó por última vez la entidad pedida

- Servidor: `Last-modified: Wednesday, 27-Mar-97 15:23:10 GMT`

A continuación, el servidor envía los encabezados de entidad, que se aplican específicamente a la entidad pedida. Esto incluye a su tamaño

- Servidor: `Content-length: 11044`

y tipo . El tipo de contenido se indica en el formato de las Extensiones de Multimedia de Internet con Múltiples Objetos (MIME)

- Servidor: `Content-type: text/html`

A continuación `www.netscape.com` cierra la conexión

Métodos Básicos

Ambos mensajes, los de petición completa y los de respuesta completa, se basan en métodos que son legibles para nosotros, comandos de una sola palabra que controlan HTTP, como el `GET` y `POST`.

El mensaje de petición completa de HTTP más simple se compone de una única línea de petición. La línea de petición está formada por: un método (o comando), seguido del URL en el que se debe realizar la tarea específica y la

⁵ Una entidad es un objeto al que se refiere un URL. Por ejemplo una página HTML es un tipo de entidad, un imagen .gif es otra.

versión de HTTP. Por ejemplo, el siguiente es un mensaje de petición completa de HTTP válido:

```
GET index.html HTTP /1.1
```

En este mensaje, GET es el método, index.html es el URL pedido y HTTP/1.1, la versión de HTTP.

A diferencia de la petición simple, una petición completa puede utilizar uno cualquiera entre varios métodos. El ejemplo precedente muestra el método más común, GET. HTTP 1.0 soporta sólo tres métodos, GET, HEAD y POST:

- GET recupera cualquier información que el URL pedido identifique
- HEAD recupera sólo información de cabecera, no el cuerpo de entidad⁶ pedida.
- POST envía información del cliente al servidor. Usualmente, los formularios basados en el WWW se basan en el método POST para transferir datos de formularios al servidor.

La especificación de HTTP 1.1, añade 10 métodos más:

- OPTIONS pide información sobre las opciones de comunicación disponibles que afectan la recuperación del URL pedido. Este método permite al cliente determinar las opciones o requisitos asociados con un recurso sin recuperarlo realmente.
- PUT pide que almacene la entidad adjunta al URL suministrado, lo que le permite actualizar un destino WWW remoto
- PATCH es similar a PUT excepto que la entidad contiene una lista de diferencias entre la versión original del recurso identificado por el URL pedido y el contenido que usted quiere para el recurso después de que haya involucrado la acción PATCH. Esta debe proporcionar un tipo MIME (como el application/diff) para que el servidor sepa qué hacer con la entidad después de abrirla. Por ejemplo, un navegador de WWW puede indicar que se modifique el URL, de destino utilizando el algoritmo diff estándar generando un método PATCH e indicando el tipo MIME de aplicación/diff y, a continuación, la modificación real que se debe aplicar.
- COPY pide que el recurso identificado del URL pedido se copia en el lugar indicado en el campo de cabecera del URL pedido.
- MOVE, DELETE LINK y UNLINK funciona de forma similar a COPY, realizando operaciones básicas sobre los URL
- TRACE es un método de diagnóstico que simplemente refleja los datos de la entidad hacia el cliente.
- WRAPPED permite que un cliente encapsule uno o más mensajes. Después de encapsular un mensaje, por ejemplo, puede cifrarlo para aumentar la seguridad.

⁶ Un cuerpo de entidad es el contenido real de la entidad, en comparación con un encabezado de entidad, que lleva información sobre la entidad (como su tamaño y tipo)

Para pasar tanto el cuerpo de entidad como la información de control requerida e interpretada de forma correcta este cuerpo de entidad, los campos de encabezados de HTTP actúan como modificadores de métodos.

Hay tres tipos de campos de cabecera:

- Los campos de encabezado generales son comunes para los mensajes de petición y para los de respuesta. Los campos de cabecera generales controlan cosas como el estado del cache de una entidad, la fecha y hora en la que se originó este mensaje y el estado de la conexión entre la fuente del mensaje y el destino.
- Los campos de encabezado de petición son específicos de los mensajes de petición completa. Estos controlan cosas tales como los tipos de soporte MIME que acepta el cliente, la dirección de e-mail del usuario del cliente y el URL de la entidad al que se refirió el cliente para realizar la petición (por ejemplo, si un enlace en www.yahoo.com se refiere a una entidad en www.microsoft.com, entonces microsoft puede registrarla).
- Los campos de encabezado de entidad se utilizan por los mensajes de petición y por los de respuesta. Estos encabezados se asocian la mayor parte de las veces con mensajes de respuesta.

El cuerpo de la entidad, tanto si es para una petición como para una respuesta, es simplemente un flujo de octetos. HTTP no restringe la información que circula a través de este cuerpo, sino que simplemente proporciona el mecanismo para etiquetar y transferir esta información. Los métodos de petición utilizados más frecuentemente, GET y HEAD, no envían una entidad del cliente al servidor, así que el cuerpo de la entidad no se aplica usualmente al mensaje de petición.

Existen códigos que indican el estado de la respuesta. Aquí se encuentran los más comunes.

Número de código	Mensaje
200	Correcto
201	Creado
202	Aceptado
204	No hay contenido
301	Traslado de forma permanente
302	Traslado de forma temporal
304	No modificado
400	Solicitud incorrecta
401	Sin autorización
403	Prohibido
404	No encontrado
500	Error interno del servidor
501	No implementado

502	Gateway incorrecto
503	Servicio no disponible

Al código de estado le siguen inmediatamente un conjunto de encabezados y, a continuación, el cuerpo de la entidad.

Los encabezados de respuesta proporcionan la situación exacta del recurso pedido, la versión del software del servidor, las respuestas de autenticación, y métodos opcionales que soporta el servidor. Como ejemplo de un caso en el que el encabezado de respuesta tiene éxito, un navegador puede utilizar un valor devuelto en un campo Server (por ejemplo, Servidor Netscape-Communicator/4.01) para determinar qué tipo de servidor utiliza el destino.

Un mensaje de respuesta tiene usualmente uno o más encabezados de entidad que describen la entidad real que se va a pasar. Los siguientes son algunos de los encabezados de entidad más importantes:

- Content-Type probablemente es el encabezado más importante en todo HTTP. El encabezado indica, en forma MIME, el tipo de datos que contiene la entidad. Por ejemplo Content-type: text/html indica que esa entidad es una página HTML y Content-type: imagen/gif indica una imagen .gif
- Content-Encoding posibilita que HTTP transporte datos con otras representaciones que no sean ASCII, coordinando las acciones entre el servidor y el cliente. Por ejemplo Content-Encoding: gzip permite que el servidor transmita un fichero comprimido gzip al cliente y que el cliente descomprima el fichero
- Content-Length indica el tamaño de la entidad
- Content-MD5, Content-Version y Last-Modified indican la suma de comprobación de la entidad, la información de la versión y la fecha de la última modificación, respectivamente. Un mensaje de petición con un campo de cabecera de petición Unless puede utilizar más tarde estos encabezados para comprobar si ha cambiado la entidad.
- Expires indica cuándo expira la información de la entidad. Después de la fecha de caducidad, el cliente no puede utilizar una copia cache y debe volver a pedir la entidad desde el servidor.
- Link y Title funcionan de forma muy parecida a los elementos <link> y <title> de HTML. Debido a que estos encabezados están en HTTP, puede utilizarlos en todos los tipos de entidad, no tan solo en HTML. Por ejemplo, una entidad de imagen .GIF con encabezado que contenga Content-type: imagen/gif y title: Sample GIF Imagen podría representar la imagen de un título apropiado.

HIPERMEDIA

Cuando hablamos de hipermmedia, nos referimos a cuatro componentes básicos de los documentos que son texto, imágenes, audio y vídeo. El WWW mantiene todos estos ámbitos de una u otra manera. La disponibilidad de

software permite fácilmente que las imágenes se vean junto al texto como parte de los documentos en línea del servidor. Otras imágenes, junto con el audio y vídeo, pueden ser visualizadas utilizando aplicaciones externas, (estas son especificadas del cliente y de la plataforma) para incrementar las capacidades básicas del software.

Hipermedia le da vida a un documento y la computadora personal se convierte en un dispositivo multimedios que puede ser más atractivo e impresionante.

Imágenes en el WWW

Dos formatos de archivos de imágenes, el GIF (formato de intercambio gráfico) y el JPEG (Joint Photographic Expert Group), despliegan imágenes de los documentos y las paginas WWW. El formato GIF lo introdujo CompuServe en 1987, como un medio para intercambiar fotografías dentro de sus servicios comerciales. Los archivos GIF pueden representar imágenes hasta con 256 colores distintos, de una paleta de cerca de 16 millones de colores. Los archivos GIF están comprimidos para ahorrar espacio y tiempo de transmisión.

Cuando nos conectamos a la pagina, al cargar la dirección URL en el visualizador, el documento WWW comienza a transmitirse a la computadora. Se debe notar que la computadora y el servidor remoto no tienen un enlace exclusivo e ininterrumpido. Por lo regular, la computadora personal (cliente) y el servidor remoto se comunican en diversas ocasiones antes de una página aparezca. Si hay más de una imagen GIF o JPEG, estas están cargadas de manera individual en la PC y entonces también están descodificadas (o descomprimidas).

El segundo tipo de imágenes que se pueden encontrar, es el que solicita para verse o copiarse a la computadora personal. Almacenadas en una base de datos, estas imágenes están en formato de archivos gráficos que proveen una resolución mayor que GIF o XBM, como JPEG. El formato JPEG, (grupo de expertos fotográficos unidos), es un estándar de la industria para compresión de archivos de imágenes a color de 24 y 8 bits y de imágenes en escala de grises. Las imágenes GIF ocupan menos espacio que las JPEG, pero solo representan 8 bits, o sea, 256 colores, mientras que el formato JPEG soporta imágenes de 24 bits hasta de 16 millones de colores.

Aunque las imágenes GIF y JPEG son los formatos más comunes, existen muchos otros con los que nos podemos encontrar al navegar en el WWW. Entre los más comunes se encuentran:

- **BMP.** El formato de gráficos de mapas de bits de Windows, permite almacenar datos de imagen hasta de 24 bits. Este formato de imagen, diseñado originalmente para imágenes de Windows, es muy usado en plataformas PC para imágenes de trama. Este formato particular puede incorporar un sistema de compresión interno llamado Run-Length Encoding (RLE).

- **TIFF.** El formato de archivo de imagen etiquetada (Tagged imagen File Fotmat) es especial para gráficos hechos en computadora y se diseña para soportar la salida de imágenes de alta resolución. Este tipo de formato permite imágenes hasta de 32 bits. No es raro que las imágenes TIFF sean bastante grandes. Debido a que está diseñado para contener imágenes para impresión, el formato TIFF usa un sistema de compresión interno llamado Lepel-ZivWelch (LZW).
- **PICT.** La mayoría de los formatos mencionados son de imágenes de trama, lo cual significa que contienen sólo datos del píxel. El formato PICT, que se usa predominantemente en la plataforma Macintosh, es un tipo especial de formato de meta-archivo. Los formatos de meta-archivo pueden contener datos de trama y de vectores. De hecho, pueden almacenar ambos tipos en forma simultánea. Este formato es muy popular en sistemas Macintosh por su tamaño reducido.

Hay cuando menos 15 tipos diferentes de formatos de archivos gráficos, cada uno de los cuales requiere del software adecuado para su correcto despliegue en pantalla. El software de despliegue de imágenes, conocido como visor, procesa y presenta imágenes en el monitor. Se debe instalar y cargar los programas de visualización gráfica, ya sea como productos comerciales o del dominio publico, para ver, manipular y almacenar estas imágenes.

Sonidos

La mayoría de los archivos de sonido que se pueden encontrar en el WWW están en el formato de Interfaz Digital para Instrumentos Musicales (MIDI, Musical Instruments Device Interface) y Real Audio. También se encontrarán formatos comunes como .WAV y .AIFF. En general, los archivos en sí pueden dividirse en dos grupos: los que definen el corte de música mediante matemáticas y los que definen el sonido en términos de ondas. El primer tipo de archivo de audio define en forma matemática las notas de la música y los instrumentos que las tocan. Los archivos MIDI son ejemplos comunes de este tipo de archivos de audio. Estas representaciones que definen las notas y los instrumentos, hacen que los archivos MIDI se consideren archivos de sonido vectorial. Por otra parte, los archivos de sonido como RealAudio, .WAV y .AIFF son descripciones digitales de la variación de frecuencias, también llamadas formas de onda. La diferencia entre esto dos formatos de audio es muy similar a la diferencia que existe entre los archivos de vector y los de trama. Pero la diferencia más notable es el tamaño del archivo

HIPERTEXTO

El hipertexto, un subconjunto de hipermedia, se refiere a la manera de moverse de un documento a otro, de una manera no secuencial ni lineal. En un documento no lineal, nos desplazamos en forma aleatoria. Las palabras o frases e iconos del documento se convierten en enlaces que permiten viajar a una nueva posición en el mismo documento, o incluso, a uno nuevo.

Un tipo de hipertexto es el llamado HTML, utilizado para realizar páginas WWW.

El concepto del HTML es parecido a la estructura interna de un fichero de un procesador de textos, en cuanto a que indica el formato de página. Pero a diferencia de un fichero de procesador de texto, HTML, se basa en la estructura lógica de un documento, no en la estructura física. Mientras que un procesador de textos puede indicar que el título de una sección nueva esté formado por negritas de 18 puntos, el HTML sólo indicará que es un título y dejará que el visualizador o navegador lo interpretara de la forma apropiada.

Debido a que HTML indica una estructura de documentos en lugar de indicar su disposición física, podrá representar documentos de HTML en muchos navegadores, que se ejecuten en muchos sistemas operativos. Por ejemplo un navegador LYNX se ejecuta en computadores UNIX y VAX/VMS con terminales de sólo texto. Mientras que un navegador gráfico puede representar un título en un tipo de letra ancho, LYNX sólo puede hacerlo en negrita, debido a que la terminal no soporta múltiples tipos de letra.

HTML

Lenguaje de Marcas de Hipertexto (HyperText Markup Language) es el lenguaje que define el estilo y la información en un documento de hipermedia del WWW y esta basado en el estándar SGML (Estándares Generalizados del Lenguaje de Marcas, Standard Generalized Markup Language).

El siguiente ejemplo muestra uno de los documentos más básicos de HTML

```
<HTML>
<HEAD>
<TITLE> Un simple ejemplo de HTML</TITLE>
</HEAD>
<BODY>
<H1>Encabezado Nivel uno</H1>
Este es un simple ejemplo de HTML.
Este es un primer párrafo<P>
Este es un segundo párrafo<P>
</BODY>
</HTML>
```

Tal como se ilustra en este ejemplo, los documentos de HTML comienzan con una etiqueta <HTML>, que indica que el documento está en formato HTML, y acaban con </HTML>. La etiqueta <HEAD> introduce el texto que describe un documento HTML. La mayoría de los documentos sólo tienen una etiqueta <TITLE> en la sección de encabezado. La etiqueta <BODY> señala el cuerpo del documento. Aparece después de la sección de cabecera.

Observe que la etiqueta <TITLE> tiene una etiqueta </TITLE> que le corresponde, para señalar el fin del título y el par <H1> funciona en gran parte

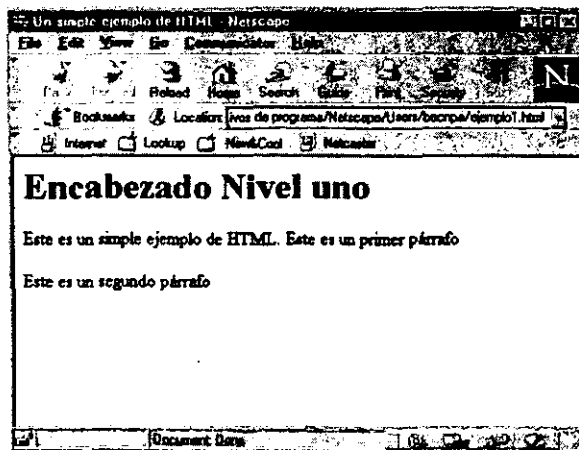
de la misma forma para señalar un título de primer nivel. Casi todas las etiquetas HTML funcionan de la misma manera, con el siguiente formato: `<tag>elemento de la etiqueta</tag>`

Las letras de la etiqueta pueden estar en mayúsculas o minúsculas, indiferentemente. Lo que haya entre ambas etiquetas será influenciada por ellas. Una excepción a esta regla es la etiqueta `<P>`, que señala el fin de un párrafo. Aparece sola, sin la etiqueta `</P>` correspondiente.

Se debe utilizar la etiqueta `<P>` para separar párrafos. Aunque el primer párrafo del ejemplo anterior está dividido en dos líneas separadas, se reconoce como un único párrafo y por este motivo, se trata como tal. El navegador hace caso omiso de la indentación y las líneas en blanco en el texto fuente. Siempre que quiera separar un texto en párrafos múltiples, deberá utilizar `<P>`.

La etiqueta `<TITLE>` señala el título del documento, que la mayoría de navegadores representa en la barra de títulos de la ventana. La etiqueta `<H1>` señala un encabezado de primer nivel, `<H2>` un encabezado de segundo nivel y así sucesivamente, para soportar documentos estructurados como resúmenes.

En la siguiente imagen se muestra el ejemplo anterior desplegado en un navegador.



Enlaces

La característica que más ha influido en el éxito del WWW ha sido, aparte de su carácter multimedia, la posibilidad de unir los distintos documentos repartidos por todo el mundo por medio de enlaces hipertexto. En general, los enlaces tienen la siguiente estructura:

```
< A HREF="xxx" > yyy </A>
```

donde xxx es el destino del enlace
 yyy es el texto indicativo en la pantalla del enlace

Existen cuatro tipos de enlaces:

Enlaces dentro de la misma página. En este caso, se sustituye xxx por #marca (la palabra marca puede ser cualquiera). Esta marca es el sitio donde queremos "saltar". En este sitio se debe poner la siguiente etiqueta . La estructura del enlace es la siguiente:
 Siguiete sección

Con otra página nuestra. Lo más frecuente es que tengamos varias páginas, una inicial y otras conectadas a ella; la forma en que podemos enlazarlas, es simplemente sustituir xxx por el nombre del archivo, de la siguiente forma:

```
<A HREF="pag2.html"> Siguiete página </A>
```

Si queremos hacer un enlace a un sitio específico a otra página en especial, solo hay que utilizar la siguiente estructura:

```
<A HREF="pag1.html#marca"> Sección anterior </A>
```

Con una página fuera de nuestro sistema. Si queremos enlazar con una página que esté fuera de nuestro sistema, es necesario conocer su dirección completa, o URL. El URL podría ser, además de la dirección de una página WWW, una dirección de ftp, gopher, etc. Una vez conocida la dirección, lo colocamos en vez de xxx, de la siguiente forma:

```
<A HREF="http://home.netscape.com/"> Netscape </A>
```

Con una dirección de e-mail. En este caso, sustituimos xxx por mailto: seguido de la dirección de e-mail. La estructura de la etiqueta es:

```
<A HREF="mailto: dirección de e-mail"> Texto de enlace </A>
```

Existen algunos navegadores que no reconocen este tipo de enlace, así que es más seguro poner, en lugar de "Texto de enlace" la dirección de e-mail.

Listas

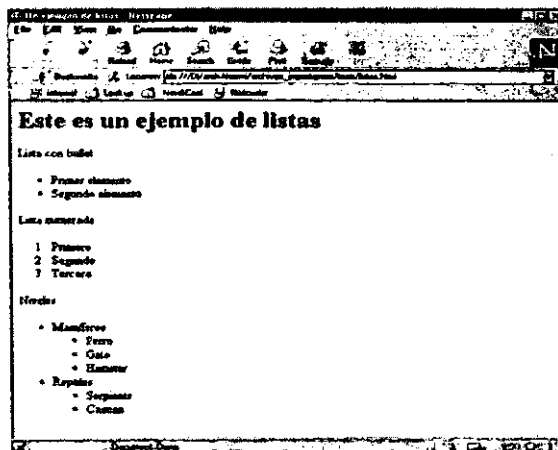
HTML soporta varios tipos de listas, incluyendo a las listas con puntos y con números. Estos tipos de listas dependen de la etiqueta para señalar cada elemento. Una lista con puntos comienza con la etiqueta y una lista numerada, con la etiqueta <LO>. Un ejemplo es:

```

<HTML>
<HEAD>
<TITLE> Un ejemplo de listas</TITLE>
</HEAD>
<BODY>
<H1>Este es un ejemplo de listas </H1>
Lista con bullet
<UL>
    <LI>Primer elemento
    <LI>Segundo elemento
</UL>
Lista numerada
<OL>
    <LI>Primero
    <LI>Segundo
    <LI>Tercero
</OL>
Niveles
<UL>
<LI> Mamíferos
    <UL>
        <LI>Perro
        <LI>Gato
        <LI>Hamster
    </UL>
<LI> Reptiles
    <UL>
        <LI>Serpiente
        <LI>Caiman
    </UL>
</UL>
</BODY>
</HTML>

```

En la siguiente imagen se muestra como se vería.



Formateo de caracteres

A menudo, HTML ofrece varias maneras de representar un texto de exactamente la misma forma, basándose en estilos lógicos y físicos. Por ejemplo, la etiqueta física `` y la lógica `` se representan de forma idéntica como texto en negritas en la mayoría de los navegadores.

Casi siempre se debería utilizar para el formateo las etiquetas lógicas en lugar de las físicas. Debido a que las etiquetas lógicas indican la estructura del documento y no sólo el formateo de los caracteres, ofrecen al usuario un mayor control sobre cómo se representa un documento.

Los siguientes son algunos de los estilos lógicos de HTML utilizados comúnmente:

`<DFN>` indica una palabra que se define, usualmente en cursiva.

`` resalta un término, normalmente en cursiva.

`<CODE>` se utiliza para segmentos cortos de código, que se representan usualmente en un tipo de letra de anchura fija.

`` indica un hincapié fuerte; habitualmente se representa con negritas

Teniendo en cuenta los avisos para que se utilicen etiquetas lógicas y físicas, los estilos físicos son:

`` para negritas

`<U>` para subrayado

`<I>` para cursiva

`<TT>` para texto de anchura fija

Todos los estilos característicos van en pares, como por ejemplo: `` negrita ``

Caracteres especiales

Debido a que el mismo HTML utiliza el signo `&`, los paréntesis en ángulo y las comillas, se tiene que utilizar los siguientes códigos especiales para representar estos caracteres:

`<` representa el paréntesis en ángulo izquierdo (`<`)

`>` representa el paréntesis en ángulo derecho (`>`)

`&` representa el signo `&`

`"` representa las comillas dobles (`"`)

De manera similar, existen códigos para escribir letras específicas de distintos idiomas. Hay muchos de ellos, pero lógicamente, los que más nos interesan son los propios del español (las vocales acentuadas, la ñ y los signos `¿` y `¡`).

Los códigos de las vocales acentuadas se forman comenzando con `&`, seguido de la vocal en cuestión, seguido de la palabra *acute* (aguda) y terminando con el signo;

`á` para la `á`

´ para la é
í para la í
ó para la ó
ú para la ú
Á para la Á
É para la É
Í para la Í
Ó para la Ó
Ú para la Ú

El resto de los códigos son:

ñ para la ñ
Ñ para la Ñ
ü para la ü
Ü para la Ü
¿ para ¿
¡ para ¡

Todo esto, que como se ve es muy laborioso, puede parecer inútil ya que si escribimos nuestro texto sin hacer ningún caso de estas convenciones, escribiendo las letras acentuadas y demás signos directamente, es muy posible que el resultado lo veamos correctamente en nuestro navegador, pero nunca podremos estar seguros que les ocurra lo mismo a todos los que accedan a nuestras páginas con otros navegadores distintos.

Representación de imágenes

La etiqueta que nos sirve para incluir imágenes en nuestras páginas es muy similar a la de enlaces a otras páginas, la única diferencia es que, en lugar de indicar al programa navegador el nombre y la localización de un documento de texto HTML para que lo cargue, se le indica el nombre y la localización de un fichero que contiene la imagen.

La estructura es:

```
<IMG SRC="imagen.gif">
```

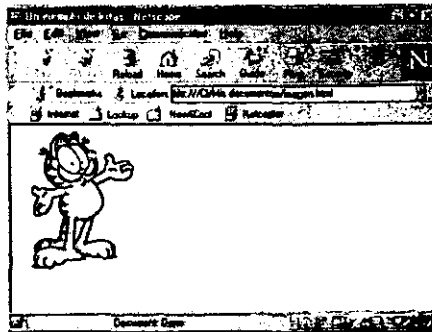
Con el comando IMG SRC (image source, fuente de la imagen) se indica que se quiere cargar una imagen llamada imagen.gif (o el nombre que tenga).

Una opción importante para esta etiqueta es ALT, que indica el texto alternativo que ve el usuario cuando el navegador no soporta imágenes o cuando está deshabilitada la carga de imágenes

```
<IMG SRC="imagen.gif" ALT="una imagen">
```

Además tiene varias opciones para alinear y posicionar una imagen dentro de la ventana del navegador con la opción ALIGN.

En la siguiente imagen se muestra un ejemplo:



Aunque en esta descripción se ha tratado de proporcionar la información básica que se necesita para crear páginas HTML sencillas, las versiones más nuevas de HTML añaden funciones como formateo de texto (fuentes de texto, alineación, tamaño, etc.), tablas, soporte para color, imágenes en segundo plano y marcos

Ahora que ya conoce los fundamentos de HTML, nos preguntamos cómo crear programas que interactúen con él. El método más común de interactuar con HTML en un servidor WWW es utilizando la Interfaz de Entrada Común (CGI)

FORMULARIOS

Los formularios permiten que los usuarios nos envíen la información directamente a nosotros o bien a nuestro servidor, en donde hemos instalado un programa que procese esta información.

Para conseguir esto, los formularios necesitan ejecutar programas o scripts por medio del CGI (Common Gateway Interface). El CGI permite a los formularios ser procesados por programas escritos en cualquier lenguaje, aunque los más usados en Internet son el Perl y el C.

Estructura de un formulario

- 1) Etiqueta de inicio:
<FORM ACTION="..." METHOD="POST" ENCTYPE="TEXT/PLAIN">
- 2) Cuerpo del formulario, con los distintos elementos para poder introducir los datos.
- 3) Botones para enviar el formulario y para limpiarlo.
- 4) Etiqueta de cierre </FORM>

Todos los elementos del formulario deben estar dentro de un FORM, o de lo contrario son ignorados.

Etiqueta de inicio

El atributo ACTION indica la acción que se debe efectuar, en el caso de un CGI, sería precisamente aquí donde indicaríamos su localización en el servidor, que habitualmente es el directorio cgi-bin.

El atributo METHOD=POST indica que los datos sean inmediatamente enviados, nada más con pulsar el usuario el botón de envío.

Con el atributo ENCTYPE="TEXT/PLAIN" se consigue que las respuestas las recibamos como un fichero de texto, perfectamente legible y sin codificar.

Elementos para introducir los datos

Estos elementos se pueden dividir en tres clases:

- Introducción por medio de texto
- Introducción por medio de menús
- Introducción por medio de botones

La introducción de los datos se consigue por medio de la etiqueta:

```
<INPUT TYPE="xxx" NAME="yyy" VALUE="zzz">
```

En donde:

xxx es la palabra que indica el tipo de introducción.

yyy es el nombre que le asignamos nosotros a la variable de introducción del dato.

zzz es la palabra asociada a un elemento.

Todos estos campos de ingreso tienen asociado un NOMBRE, que es obligatorio asignar, pues si no se hace así no habrá ninguna manera de referirse a la información que el usuario ha ingresado.

Ingreso de texto

Para una línea, se define mediante:

```
<INPUT TYPE="text" NAME="nombre" >
```

En este caso es xxx=text, es decir, INPUT TYPE="text". El atributo VALUE no procede en este caso

Ejemplo:

Ingrese su dirección e-mail :

```
<INPUT NAME="email" TYPE="text">
```

La longitud se puede variar incluyendo en la etiqueta el atributo SIZE="número". Por otra parte, sea cual sea la longitud del formulario, si no

se indica nada, el usuario puede introducir el número de caracteres que quiera. Se puede limitar esto, incluyendo en la etiqueta el atributo `MAXLENGTH="número"`.

También se puede hacer que el texto introducido no sea reconocible, es decir que todos los caracteres se representen por asteriscos. Basta con cambiar en la etiqueta `INPUT TYPE="text"` por `INPUT TYPE="password"`.

Para varias líneas

Cuando el texto a introducir puede alcanzar una gran longitud, es conveniente utilizar un formulario de texto de múltiples líneas.

```
<TEXTAREA NAME="nombre" ROWS="lineas" COLS="columnas"> ... </TEXTAREA>
```

(en donde no se utiliza `INPUT TYPE`)

Ejemplo:

```
Escribe tus comentarios <TEXTAREA NAME="Area" ROWS=6 COLS=40></TEXTAREA>
```

Opciones

Si queremos que el usuario, en vez de introducir un texto, como hemos visto en los casos anteriores, escoja entre varias opciones que le presentamos nosotros, haremos uso de un formulario en forma de menú.

Se consigue con la etiqueta de inicio `<SELECT NAME="yyy">` y la de cierre `</SELECT>`.

Las distintas opciones a escoger se consiguen con la etiqueta `<OPTION>`.

Ejemplo:

¿Cuál es tu color preferido?

```
<BR>
<SELECT NAME="ColorPreferido">
<OPTION>Rojo
<OPTION>Verde
<OPTION>Azul
<OPTION>Amarillo
</SELECT >
```

En el ejemplo anterior, sólo es visible en el formulario una opción. Si queremos que sean visibles múltiples opciones a la vez, añadimos en la etiqueta los atributos `MULTIPLE SIZE="número"`, donde especificamos el número de opciones visibles.

Checkboxes

Para permitir al usuario seleccionar varias opciones a la vez, se hace uso de checkboxes, que son pequeños cuadraditos que aparecen al costado del texto, y se pueden seleccionar o deseleccionar de manera independiente, haciendo clic sobre ellos.

Un checkbox se define usando: `<INPUT TYPE="checkbox" NAME="nombre">`.

Ejemplo:

```
<P>Tus deportes de intereses:</P>
<INPUT TYPE="checkbox" NAME="interes_golf"> Golf
<INPUT TYPE="checkbox" NAME="interes_tenis"> Tenis
<INPUT TYPE="checkbox" NAME="interes_squash"> Squash
<INPUT TYPE="checkbox" NAME="interes_natac"> Natacion
<INPUT TYPE="checkbox" NAME="interes_natac"> Futbol
```

Si queremos que el formulario aparezca inicialmente como marcado (el usuario no necesitará hacerlo), basta con añadir el atributo CHECKED dentro de la etiqueta. En el ejemplo anterior si sustituimos la etiqueta equivalente por:

```
<INPUT TYPE="checkbox" NAME="Lista" CHECKED>
```

Botones de radio

Cuando queremos que el usuario elija una única opción entre varias, podemos hacer uso de los botones de radio, que se consiguen con la etiqueta:

```
<INPUT TYPE="radio" NAME="yyy" VALUE="zzz">
```

Donde yyy es el nombre que le ponemos a la variable que se trata de elegir, y zzz es el nombre de cada una de las opciones en concreto. Estos comparten el mismo nombre pero tienen distintos valores. Al momento del envío del formulario, lo que se envía es el nombre, y el valor de la opción que fue elegida.

Ejemplo:

¿Cuál es tu sistema operativo preferido?

```
<BR>
```

```
<INPUT TYPE="radio" NAME="SistemaOperativo" VALUE="PC" CHECKED> PC
<INPUT TYPE="radio" NAME="SistemaOperativo" VALUE="Mac"> Mac
<INPUT TYPE="radio" NAME="SistemaOperativo" VALUE="Unix"> Unix
```

Obsérvese el atributo opcional CHECKED que se ha añadido en la primera etiqueta. Esa será la opción que aparece marcada por defecto y también que no es posible escoger más de una opción.

Botones de envío y de borrado

Faltaba un elemento esencial en cualquier formulario, y es el botón de envío de los datos, que se consigue con la etiqueta:

```
<INPUT TYPE="submit" VALUE="texto">
```

Ejemplo:

```
<INPUT TYPE="submit" VALUE="Enviar datos">
```

Otro botón interesante es el de borrado de los datos introducidos, muy conveniente en un formulario con muchos elementos.

Es muy similar al de envío, pues se consigue con la etiqueta:

```
<INPUT TYPE="reset" VALUE="zzz">
```

En donde zzz es el texto que queremos que aparezca en el botón.

Ejemplo:

```
<INPUT TYPE="reset" VALUE="Borrar datos">
```

En la siguiente imagen se muestra un formulario realizado con los ejemplos anteriores

The screenshot shows a Netscape browser window with the following content:

- Menu: Archivo, Edición, Ver, Comunicador, Ayuda
- Toolbar: Anterior, Siguiendo, Recarga, Inicio, Buscar, Guía, Imprimir, Seguridad, Plus
- Address bar: file:///D:/arch-Noemi/archivos_importantes/tesis/imagenes_servidores/formas.html
- Title: Libro de visitas
- Form fields:
 - Ingrese su dirección e-mail: [text input]
 - Escribe tus comentarios: [text area]
 - ¿Cuál es tu color prefendo?: [Azul]
 - Tus deportes de interes: Golf Tenis Squash Natacion Futbol
 - ¿Cuál es tu sistema operativo prefendo?: PC Mac Unix
- Buttons: [Enviar datos] [Borrar datos]
- Status bar: Documento: Ejecutado

CGI

CGI (Common Gateway Interface) es una interfaz entre programas de aplicación y servicios de información. Hasta el momento, se ha utilizado únicamente como pasarela hacia servidores WWW que utilizan el protocolo HTTP. Sin embargo, ha sido planteada como pasarela independiente del servicio y de la aplicación.

Un documento HTML, es algo estático, permanente, lo que no se adecua a los requisitos de interactividad necesarios en una aplicación de este estilo, como el acceso a información en constante actualización, consulta a bases de datos o seguimiento, control o recuperación de resultados de un determinado proceso. En este tipo de aplicaciones, es necesario acceder a una información que se está generando "en tiempo real". Así pues, es necesario contar con algún tipo de pasarela como la que define CGI.

La interfaz define una forma cómoda y simple ejecutar programas que se encuentran en la máquina en la que se aloja el servidor. Para el cliente presenta una ventaja en el aspecto de la seguridad, ya que no tendrá que ejecutar ningún programa de efectos desconocidos en su sistema local. Desde nuestro punto de vista, además de eliminar la necesidad de aprendizaje, se resuelven los problemas de mantenimiento, operación y distribución de clientes ya que el acceso se realizará a través de cualquier cliente estándar WWW y la comunicación se realizará según el protocolo HTTP.

Sin embargo, aunque en el servidor ya no será necesario permitir el acceso a extraños, disponer de un programa CGI en un servidor implica que virtualmente todo usuario de Internet será capaz de ejecutar un programa en ese servidor, por lo que es necesario atender ciertas precauciones concernientes a la seguridad. La primera de ella suele ser centralizar la localización de todos los CGI's de un servidor en un directorio reservado al efecto al cual se permitirá el acceso únicamente al webmaster. Todo servidor WWW es configurable de tal forma que al recibir una petición de acceso a un documento alojado en tal directorio considerará que se trata de un programa ejecutable.

Como su propio nombre indica, CGI es una interfaz entre servidores de información y programas de aplicación. Por tanto, define una serie de reglas que deben cumplir tanto las aplicaciones como los servidores para hacer posible la presentación de resultados de programas ejecutables en tiempo real a través de servicios de información estandarizados. Al tratarse de una interfaz, no existe ningún tipo de dependencia con el lenguaje de programación empleado. Así pues, es posible encontrar aplicaciones CGI programadas en lenguajes como en C, C++, fortran, perl, Delphi, Visual Basic, AppleScript, Java y cualquier shell de UNIX.

De este tema se hablará con mayor detalle en el capítulo 5.

OTRAS TECNOLOGÍAS EN EL WWW

JAVA

Java es un lenguaje con subproceso múltiple, orientado a objetos, trasladable a una amplia variedad de sistemas operativos y arquitecturas de procesador. Los programas de Java están compilados en pseudocódigo que el interpretador de Java ejecuta en el sistema cliente.

Aunque se basa en C++, Java tiene varias simplificaciones importantes. A diferencia de C++. Java puede recoger basura automáticamente, liberando al programador de la preocupación de tratar la gestión del almacenamiento. No tiene los molestos punteros al estilo de C, ni malloc, ni free. Aunque Java no tiene un operador new, rastrea automáticamente el estado de un objeto y libera memoria de un objeto cuando éste ya no se necesita.

Java está diseñado para soportar un entorno distribuido. Tiene una biblioteca de rutinas llamada java.net, para acceder a protocolos del tipo TCP/IP como HTTP y el protocolo de transferencia de archivos (FTP). Debido a que Java se integra con los servicios del WWW, los programadores pueden acceder a objetos en todo Internet por medio de los URLs de una forma tan fácil como el acceso al sistema de archivos local.

Java está construido desde los fundamentos para que sea un entorno seguro. Java no puede manipular directamente ficheros en disco, e incluso se protege por medio de la seguridad de tipos (un programa no puede convertir un entero en una referencia de objeto u obtener situaciones de memoria privada)

Java es una de las nuevas tecnologías que tiene su auge en estos momentos. Para obtener más información en <http://java.sun.com>

JAVASCRIPT

Se puede imaginar a JavaScript como el hermano menor de Java, debido a que es un subconjunto de Java que efectúa algunas simplificaciones al lenguaje Java. Aunque la sintaxis de JavaScript es similar a la de Java, JavaScript es un lenguaje interpretado completamente, intercalado directamente dentro de las páginas HTML, no compilado en pseudocódigo, como Java. La diferencia entre la complejidad de los dos entornos es muy similar a la diferencia entre otros lenguajes de secuenciación, como DBase, y lenguajes orientados a objetos compilados, como C++ o Delphi.

El ámbito de JavaScript es más pequeño que el de Java y, por este motivo, JavaScript ofrece un entorno de programación más simple, pero más restringido. Un programa Java se basa en el modelo orientado a objetos, pero las secuencias de JavaScript se basan en objetos extensibles incorporados en el navegador, sin clases ni herencia. JavaScript no requiere la escritura de Java ni la comprobación de tipos (las variables se pueden definir en cualquier parte

del programa). JavaScript soporta únicamente unos pocos tipos de datos, como los valores numéricos, booleanos y de cadena.

Las sentencias de JavaScript pueden reconocer y responder a eventos de usuarios como las pulsaciones de ratón, la entrada de formularios y la navegación por páginas.

Una arquitectura probable para aplicaciones futuras basadas en el WWW se basa en una página HTML con comandos JavaScript intercalados, soportados, a su vez, por miniaplicaciones Java

ACTIVEX

Una tecnología de Microsoft que facilita el uso de información compartida entre aplicaciones. Se utiliza principalmente para desarrollar aplicaciones interactivas y contenido de WWW. ActiveX se ha construido sobre la tecnología OLE que se utilizó durante algún tiempo, pero expande el alcance de los objetos compartidos desde el escritorio a todo Internet. Debido a que la tecnología ActiveX es modular en cuanto al diseño, los programas pueden escribirse como aplicaciones independientes, como objetos "smart" incrustados dentro de programas Visual Basic o páginas WWW, o como objetos OLE tradicionales dentro de los documentos.

VBSRIPT

Como competidor directo de JavaScript, VBScript también es un poderoso lenguaje de scripts para desarrollo que puede usarse en el WWW. Fue creado en 1995 por Microsoft y es consecuencia natural de la iniciativa de esta empresa de Internet.

Al igual que JavaScript, VBScript es un lenguaje de scripts que se usa directamente dentro de la página WWW, mediante la etiqueta <SCRIPT>; en este caso no se requiere la etiqueta <APPLET> ni plug-ins. Sin embargo, la principal diferencia es que Java y JavaScript están basados parcialmente en los lenguajes C y C++, mientras que VBScript está basado en Visual Basic, ambiente autónomo de programación de aplicaciones, VbScript comparte elementos y estructuras en común, lo que hace de un buen programador en Visual Basic un buen programador de scripts para WWW.

Uno de los principales atributos de cualquier lenguaje de scripts es la capacidad de ejecutar comandos sin solicitar información del servidor. En un archivo sencillo HTML, se pueden insertar botones y controles para que el usuario navegue por el sitio. Sin embargo, cuando el usuario hace clic en vínculos u otros elementos, la máquina cliente debe solicitarle información al servidor. Cada acción debe pasar por el servidor. Con un lenguaje de scripts, algunas acciones pueden ejecutarse en la computadora cliente, en lugar de solicitar información del servidor. Esto reduce considerablemente el tráfico de red, además de que agiliza las cosas de para el usuario.

SERVIDOR WEB EN LA PC

Un servidor WWW es entonces un programa encargado de responder a requisiciones que se efectúan en el puerto 80, según lo define el protocolo HTTP. Dado la URL de una requisición, el servidor debe encontrar el recurso, archivo o programa que corresponde a dicha URL y, si es necesario, ejecutar los programas CGI (Common Gateway Interfase) y SSI (Server Side Include) asociados a dicha página.

Un cliente de HTTP es normalmente un navegador de Web tal como Internet Explorer o Netscape. Otro tipo común de clientes son los robos de indexado y el mailbot. El primero es un robot encargado de visitar páginas de búsqueda o catalogación de información, mientras que el segundo es un robot que permite el acceso a páginas de Web mediante correo electrónico. Un cliente como IE o Netscape se encarga de desplegar la información en los formatos de uso más común y de llamar aplicaciones auxiliares para el despliegue de información para los formatos que no son tan convencionales, como lo es Postscript.

PORQUE TENER UN SERVIDOR WWW

Debido a que WWW es un mercado de gran potencial dentro de Internet, hay mucho interés en desarrollar sitios Web, para diferentes causas, las principales son para el comercio, la educación, el entretenimiento y sobre todo se utiliza como una herramienta para la difusión de información.

Por eso cada vez son más las empresas, escuelas, universidades y personas en general que desean compartir información, anunciar sus productos o servicios, etc.





Además que este servicio es cada vez más difundido, y se realizan más tecnologías orientadas hacia él.

La principal razón del porqué tener un servidor WWW, es compartir información de toda clase, como pueden ser archivos, documentos, artículos, etc.

SELECCIONANDO UN SERVIDOR

WWW es la parte de Internet que más usuarios atrae y como resultado el número de compañías interesadas en proveer servidores para HTTP ha crecido. En la actualidad hay más de 30 productos diferentes para toda una gama de sistemas operativos. La tabla 1 muestra los 10 más populares según los resultados de un muestreo⁷.

Todo servidor debe ser compatible con HTTP. Sin embargo, cada proveedor incluye características únicas a su servidor para tratar de hacerlo más atractivo en un mercado que es ya bastante competido. Algunos servidores serán más simples que otros, otros más rápidos, otros gratuitos, otros más fáciles de instalar, otros generarán mejores estadísticas de uso, etc.

Servidor	Porcentaje (%)	Plataformas
 Apache	51.85	UNIX, Windows NT, Windows 95/98
 Internet Information Server	22.05	Servidores Windows NT Intel/Alpha
 Netscape Enterprise Server	4.73	UNIX, Windows NT
 NCSA HTTPd	2.08	UNIX, Windows NT, Windows 3.x

⁷ Realizado en 1998. Se puede consultar en la página <http://serverwatch.internet.com>







	WebSite Pro	1.93	Windows NT, Windows 95/98
	WebStar	1.37	Macintosh
	Netscape Fast Track Server	0.86	UNIX, Windows NT, Windows 95/98
	Lotus Domino	0.38	Windows 95/98 Windows NT, UNIX OS/2, Novell NetWare
	AOLServer	0.32	UNIX Windows NT
	Xitami	0.02	Windows 95/98, Windows NT (Intel Alpha); Windows 3.x, OS/2, UNIX, OpenVMS

Tabla 4.1 Principales servidores Web

¿Cuál servidor debe elegirse? La respuesta depende de muchos factores: Dinero disponible para comprarlo, sistema operativo, si se requieren transacciones seguras, cantidad de información que se debe transmitir, etc.

Una buena recopilación de información sobre los servidores que existen actualmente en el mercado, puede encontrarse en <http://serverwatch.internet.com>. Esta página cuenta con una lista de los servidores más utilizados, en la semana o el mes que se consulte. Cuenta además con una sección donde se pueden obtener datos importantes sobre los servidores como son: Dirección donde se puede obtener el servidor, descripción, plataformas donde se puede instalar, condición (Gratis, prueba de evaluación o producto comercial), compañía que lo distribuye, además algunos puntos de evaluación como son: desempeño, facilidad de uso, soporte técnico y seriedad.

Además tiene una sección donde analiza cada uno de los servidores, sus pros y contras.

Notoriamente, la mayor parte de los sitios Web utilizan servidores del dominio público. Según estimaciones, estos servidores funcionan en un 70% del mercado en sitios no comerciales.

Para los sitios comerciales se utilizan, en su mayor parte, servidores comerciales que tienden a hacer más fácil el proceso de instalación y mantenimiento del servidor mediante interfases de uso ameno que hacen posible que la instalación no necesite hacerla un experto -comparado con la tarea de modificar archivos de configuración a través de un editor en los servidores gratuitos. Además, por lo regular, ofrecen soporte técnico especializado.

Ahora bien, en este momento la mayoría de los servidores Web se encuentran en sistemas UNIX, por ser un sistema operativo adecuado para el ambiente de cliente / servidor que necesita un servidor Web. Aunque otra alternativa son los sistemas de red basados en Windows, que están teniendo mucho auge, además que son los sistemas cliente que más se utilizan.

Si la computadora donde se desea instalar un servidor utiliza UNIX, se podrá elegir entre una amplia gama de servidores. En cambio, los sistemas operativos "raros" tendrán pocas opciones —o ninguna. Vale la pena consultar la dirección anterior para conocer cuáles son las opciones disponibles, la cual contiene ligas a las páginas de cada uno de los servidores.

La decisión final de qué servidor es el mejor para una organización dada depende de los factores anteriormente dichos, pero uno de los factores decisivos es el sistema operativo en que se desea instalar el servidor.

TRES OPCIONES PARA INSTALAR UN SERVIDOR EN UN AMBIENTE WINDOWS

Dado que el equipo más popular de trabajo es la computadora personal (PC), nuestro enfoque será para este tipo de equipo, así como para el sistema operativo que comúnmente utiliza (Windows 95/98 o NT)

Analizando los factores arriba descritos el sistema propuesto es el más común (olvidando un poco los problemas de monopolio), con un precio bajo, un sistema operativo amigable, además de que existen varias opciones de servidores Web para este sistema.

INTERNET INFORMATION SERVER

Las empresas buscan tecnologías de Web basadas en estándares para, de una manera más eficiente, intercambiar información con clientes, socios y empleados de todo el mundo. Además, intentan implementar soluciones que aprovechen al máximo las inversiones existentes en conocimientos, formación y tecnología. Microsoft Internet Information Server (IIS) está diseñado para cubrir estas necesidades a un amplio rango de usuarios, desde grupos de trabajo y departamentos de una intranet corporativa hasta proveedores de servicios Internet que alojan sitios Web que reciben millones de visitas diarias.

Requisitos del sistema

Equipo y procesador	Para sistemas Intel y compatibles: Procesador 486 a 66 MHz o superior Para sistemas basados en RISC: Procesador Alpha a 150 MHz o superior
Memoria	Como mínimo, 32 MB
Disco duro	50 MB de espacio disponible en disco duro para Windows NT Server 40 MB de espacio disponible en disco duro para Windows NT Workstation
Sistema operativo	IIS 4.0 requiere la instalación del sistema operativo Windows NT Server 4.0 y Personal Web Server 4.0 requiere la instalación de Windows NT Workstation 4.0 o Windows 95.

Características

Es un servidor Web fácil de instalar y configurar. Si se utiliza el asistente para descarga, se pueden seleccionar los componentes específicos que desea instalar.

Tiene Herramientas flexibles de administración: incluyen una consola integrada de administración basada en Windows, administración mejorada basada en el explorador, secuencias de comandos ejecutables desde la línea de comandos, así como interfases programables para crear herramientas de administración personalizadas.

Seguridad. IIS incluye un servidor de certificados que está estrechamente integrado con el modelo de seguridad de Windows NT Server, lo que permite a las organizaciones emitir y administrar certificados digitales X.509 estándar de Internet.

Motor de búsqueda. IIS incluye capacidades de búsqueda integradas, que permiten a los usuarios crear formularios de búsqueda personalizados con Active Server Pages, objetos de datos ActiveX y consultas SQL para buscar información en el servidor Web.

Página de información:

http://www.microsoft.com/latam/productos/prodref/534_newf.htm

IIS incorpora los estándares existentes y los más recientes de la industria, incluidos: HTTP 1.1 para un mayor rendimiento de Internet, administrando y emitiendo certificados digitales X.509 y enviando mensajes a servidores estándar de noticias NNTP y de correo SMTP.

IIS incluye un conjunto de clases de Java para crear componentes del lado del servidor y Java 1.1 Virtual Machine de Microsoft. Si utiliza Active

Server Pages y Microsoft Transaction Server, puede crear aplicaciones Java y ejecutarlas fácilmente en entornos de servidor.

Ventajas

El precio (hay una versión de distribución libre), control de la administración superior, soporta el estándar HTTP 1.1, soporta servidores virtuales, herramientas de indexación también manejables con Microsoft Office, excelente soporte para desarrollo de aplicaciones distribuidas, excelente colección de herramientas de servidor.

Desventajas

No hay una versión UNIX, no soporta SMTP, solo corre en servidores Windows NT, Documentación mediocre, configuración complicada

NETSCAPE ENTERPRICE SERVER

Si busca un alto desempeño, el servidor de web Netscape Enterprise Server es una buena opción. Es un servidor de fácil uso. Constituye una solución integral para crear y administrar sitios Web en Internet o en una intranet. Es una plataforma abierta para crear aplicaciones centradas en la red utilizando herramientas interplataformas.

Requerimientos

Equipo y procesador	Sistemas Intel con procesador 486 y Pentium
Memoria	Como mínimo, 32 MB
Disco duro	50 MB de espacio disponible en disco duro
Sistema operativo	Windows NT 3.51 o 4.0

Características

El asistente de instalación detecta automáticamente la información de configuración del sistema para ayudar al usuario a optimizar el rendimiento del servidor. El servidor utiliza el navegador como terminal de administración, facilitando su control desde cualquier máquina. Proporciona ayuda contextual mediante documentos HTML accesibles desde cualquier máquina de la red.

Publica estadísticas de todo el sitio como: número total de visitas, número de URL únicos que visitan el sitio, número total de hosts únicos y tráfico total transmitido, indicando el minuto, hora o día. Además identifica qué clientes acuden al mayor número de páginas y descargan la mayoría de información.

Las últimas versiones soportan el protocolo HTTP 1.1, construye buscadores con atributos de documento y vistas, publicación avanzada y manejo de usuarios finales a través de un acceso llamado "Netshare",

agrupamiento de servidores, e integración de Java con soporte para JavaBeans, y JDBC

Incorpora también avanzadas funciones de seguridad para restringir el acceso a los recursos del servidor (como aplicaciones, documentos y herramientas administrativas), y la posibilidad de codificar la información que circula entre el servidor y el cliente. Los usuarios pueden autenticarse mediante certificados X.509v3, contraseñas y nombre de usuario, nombre de dominio, nombre de hosts, dirección IP o pertenencia a grupos.

Soporta aplicaciones en una serie de lenguajes conocidos, como JavaScript, C, Perl y Visual Basic, que se comunican con el servidor a través de la CGI.

Ventajas

Soporte para varias plataformas, capacidad para publicación de usuarios finales, manejo centralizado del servidor, soporta SMTP, motor de búsqueda integrado.

Desventajas.

Precio (en relación con el IIS), complejidad, soporte técnico.

APACHE

Es un servidor originalmente basado en la versión 1.3 de HTTPd. Apache ha sido el servidor que más rápidamente ha crecido en popularidad, ha pasado a ser el servidor más popular. Es también gratuito y está siendo desarrollado por el Proyecto Apache, cuyo objetivo es la creación y mantenimiento de un servidor gratuito de Web compatible con el estándar HTTP.

Requerimientos

Apache 1.3 tiene un diseño para correr sobre Windows NT 4.0. El sistema de instalación solo trabaja sobre procesadores INTEL. Apache puede correr también sobre Windows 95 y Windows NT 3.5.1, pero estos no se han probado. En todos los casos debe instalarse bajo una red TCP/IP.

Si se corre sobre Windows 95, se recomienda usar el "Winsock2" para mejorar el rendimiento pero no es necesario. Si se corre sobre NT 4.0, se recomienda instalar Service Pack 2.

Apache para Windows NT aún no se ha perfeccionado su desempeño. Apache tiene un mejor funcionamiento sobre plataformas UNIX.

Características

Basado originalmente sobre HTTPD de NCSA es un servidor de distribución libre.

Entre los aspectos más notables están: el soporte de diferentes plataformas, soporte del protocolo HTTP/1.1, módulos (API), seguridad, registro, desempeño y robustez. Apache corre en Windows 95/98, OS/2 y en todas las variantes más importantes de UNIX.

El servidor es totalmente compatible con HTTP/1.1 y soporta API e ISAPI (en NT). Apache distribuye un conjunto de módulos que manejan todo desde cookies⁸ y autenticación de usuario hasta corrección de URL.

La seguridad de Apache, su desempeño y la robustez son indiscutibles, muchos de los sitios con más accesos en el mundo corren Apache o alguno de sus derivados. La distribución pública del código original tiene como resultados parches para el software que se distribuyen rápidamente, permitiendo el escrutinio público, esto ayuda a asegurar que los huecos de seguridad en el software sean oportunamente informados. Como resultado de esto, ha permitido a los desarrolladores de Apache crear un paquete que es sumamente estable y seguro y que es capaz de competir con los paquetes comerciales desde el punto de vista de dos aspectos velocidad y integración.

A pesar de todas sus fortalezas, Apache seguramente no es para todos. La configuración y el mantenimiento del servidor se realizan por medio de comandos. A diferencia de la mayoría de los servidores comerciales populares, Apache no ofrece ni un browser o un GUI para la configuración o para las herramientas de administración. Esta es una ventaja para algunos desarrolladores, pero para otros puede traducirse en un mantenimiento y despliegue muy caro, especialmente si los administradores del sitio no están familiarizados con los fundamentos del servidor. La carencia de visual, wizard, y / o browser como herramientas de administración, pueden hacer que los usuarios se alejen. Además, el soporte técnico es por medio de newsgroups.

Ventajas

Precio (es software libre), desempeño y robustez, seriedad, seguridad, soporte para el protocolo HTTP 1.1, extensible, soporte técnico rápido vía grupos Usenet, internase modernizada.

Desventajas

No hay versión disponible Mac, La versión NT esta en la infancia (todavía falta mejorar el desempeño con respecto a las versiones UNIX), faltan interfases wizards y herramientas de administración gráficas para facilitar la

⁸ Las cookies son archivos de texto que contienen información sobre el cliente y que se guardan en la maquina cliente. Permite que los propietarios del ordenador donde esta la pagina que te manda la cookie sepan cuales de sus paginas visitas, cuantas veces al mes las visitas, etc... Todo esto les permite hacerse una idea de tus hábitos de navegación, para realizar estadísticas, para mejorar su publicidad, etc... Generalmente estos archivos se graban en un directorio especial designado por el navegador.

configuración y tareas de administración, para un soporte técnico más extenso se requiere la compra de un contrato de apoyo con terceros.

INSTALANDO UN SERVIDOR WWW

Como se puede ver hay muchos tipos de servidores, de los cuales podemos escoger el que mejor nos convenga.

Existen servidores que se instalan casi solos, pues sólo piden datos mínimos como son: el nombre, la dirección IP, el nombre del administrador y clave de acceso. Hay otros que sólo se instala el programa y hay que configurarlo "a mano" modificando los archivos de configuración. Pero en general casi todos los servidores que están hechos para el ambiente de Windows tienen interfases gráficas o wizards que hacen que las modificaciones o configuraciones sean más fáciles y rápidas.

Los principales parámetros que se deben modificar después de haber instalado el servidor, son principalmente:

- La página principal del sitio
- Los usuarios que pueden tener páginas en el servidor
- Los privilegios de los usuarios
- Los directorios donde se pondrán las páginas, los archivos ejecutables (CGI's), etc.

Después de ver porque instalar un servidor WWW y tener algunas opciones, tenemos que tener en cuenta los riesgos que esto atrae.

Riesgos de instalar un servidor WWW

Un servidor WWW es, actualmente, el servicio de Internet más utilizado, y el que le ha dado a esta red un crecimiento explosivo que no podría haber sido previsto hace algunos años.

La gran mayoría de los documentos en WWW están escritos en HTML, y son enviados a través de la red utilizando un protocolo conocido como HTTP. Por lo tanto, todo servidor de WWW debe estar corriendo algún tipo de servidor de HTTP. Es en las distintas implementaciones de estos servidores donde puede haber errores y problemas que permitan acceso no autorizado a los recursos.

A continuación podemos ver algunos de los principales problemas y sus soluciones

Los principales problemas que puede ocasionar un servidor de WWW son:

- Divulgación de información o documentos a individuos no autorizados.

- Información confidencial enviada por el cliente al servidor (por ejemplo, un número de tarjeta de crédito) y que sea interceptada por alguien no autorizado.
- Divulgación de datos acerca de la máquina que funciona como servidor, dando a los posibles intrusos información que puede servir para montar ataques por otras vías.
- Ejecución de comandos arbitrarios (especificados por el cliente) en el servidor.

Las principales medidas que se deben tomar para incrementar la seguridad de un servidor WWW se pueden resumir en las siguientes:

- Evitar usar un servidor con problemas conocidos. Muchos de los servidores de HTTP ampliamente utilizados han contenido errores que dan lugar a problemas de seguridad. En la gran mayoría de los casos, estos errores han sido corregidos en cuanto fueron detectados, y las correcciones incorporadas en versiones posteriores del servidor. Sin embargo, muchos sitios siguen utilizando versiones viejas, con el consiguiente riesgo de explotación de los huecos de seguridad. Uno de los huecos más notorios ha sido el que fue encontrado en el servidor HTTPD de NCSA, versión 1.3, en febrero de 1995, y que permitía ejecutar comandos arbitrarios. Este problema fue inmediatamente reportado y corregido.
- En términos generales, es recomendable utilizar siempre la última versión del servidor de HTTP que se haya elegido.
- Poner los permisos correctos en los archivos del servidor. Tanto los archivos del servidor (el ejecutable, archivos de configuración, etc.), como los documentos almacenados en él deben tener permisos de acceso cuidadosamente establecidos, para impedir que alguna persona no autorizada, ya sea accidental o intencionalmente, haga modificaciones que pudieran causar problemas. En términos generales:
 1. Crear un usuario y / o grupo dedicado al servidor de http (posiblemente llamado WWW, http o algo semejante).
 2. El programa del servidor de HTTP debe ser ejecutable solamente por el usuario autorizado
 3. Los archivos de configuración del servidor deben ser modificables solamente por el usuario autorizado.
 4. Los documentos deben ser modificables solamente por el usuario y / o grupo autorizado.
 5. Revisar que no sea posible utilizar ligas simbólicas para dar acceso a archivos que están fuera del árbol de documentos del servidor.
 6. Reducir al mínimo las cuentas de usuarios existentes en la máquina que funciona como servidor.
 7. Controlar lo que los usuarios ponen en sus páginas personales y educarlos para que sepan lo que hacen.

8. Monitorear periódicamente las bitácoras del servidor para detectar comportamientos extraños.
9. No confiar en restricciones de acceso por dirección IP, o por contraseñas, para proteger documentos confidenciales. Estas medidas detienen a la gran mayoría de la gente, pero son fácilmente sorteables para un atacante decidido.
10. Desactivar la ejecución de programas CGI salvo en casos necesarios y cuidadosamente controlados.
11. Revisar cuidadosamente los programas CGI que se utilicen (para implementar formas de HTML, por ejemplo), para que no se ejecuten con ninguna clase de privilegios, no modifiquen nada en el servidor y no utilicen ninguna clase de información proporcionada por el usuario sin antes revisarla muy detalladamente.

También es importante tener cuidado del lado del cliente. Si el visualizador que se está utilizando está configurado para ejecutar "ciegamente" cualquier documento de tipo aplicación que se encuentre, es posible ejecutar comandos arbitrarios en la máquina en la que se está ejecutando dicho visualizador. En sus últimas versiones, el mismo visualizador implementa soporte para el lenguaje Java, así como para una serie de extensiones a HTML llamadas JavaScript. Todo esto es muy "bonito", hasta que nos enfrentamos al hecho de que han sido descubiertos serios problemas de seguridad en esta implementación.

Otras recomendaciones

Con el incremento en la popularidad del WWW, Internet se ha visto sobrecargada. Por ejemplo, una página con un número moderado de imágenes puede ocupar un enlace de 64kbps por unos 20 segundos sin permitir ningún otro tráfico durante este intervalo. Por ello se han dado varias propuestas para reducir el tráfico en la red. Una de las más comunes es el uso de cache en un proxy⁹ para el almacenamiento local de páginas de Web. Este proxy reside en una computadora que comúnmente está localizada en la frontera de la red local y la red global. Con esto, el objetivo es, por un lado, aislar computadoras locales de la red global (lo que puede deberse a razones de seguridad, por ejemplo) y, por otro, almacenar información para reducir el tráfico externo de páginas Web. Es posible instalar una PC como proxy, con 2 gigabytes de disco, la cual puede almacenar aproximadamente de 10,000 a 20,000 páginas. Cuando se acaba el espacio, las páginas de menor uso son las primeras en caducar y ser removidas, en tanto que las de uso común se mantienen por

⁹ Un Proxy, es un servidor que actúa como intermediario entre un equipo cliente y uno servidor. Utilizando un cache el proxy se nutre segundo a segundo con los datos de las páginas de Web, Gopher, archivos de FTP, etc. que cada cliente vaya visitando y/o accediendo.

El envío de los datos al cliente, tomados del proxy se efectúa a la velocidad pico de transferencia de información entre el cliente y su proveedor de Internet, con esto aumenta la velocidad y evita posibles colapsos entre servidores remotos, esta es justamente la gran ventaja del uso de esta tecnología.

tiempos más largos, reduciendo así el tráfico externo. Las computadoras proxy son particularmente útiles en organizaciones grandes, tales como universidades o proveedores de acceso a Internet (ISP), que tienen un sinnúmero de computadoras conectadas al gateway principal. La nueva versión de HTTP soporta el modificador de comando "if-modified-since". Este comando es utilizado para verificar la validez de la página en cache local. Si la página ha sido modificada desde el día que fue "cacheada", se baja una nueva copia de la página, si ésta no se ha cambiado el proxy sirve al cliente la página localmente.

Asimismo, el proxy actúa como "representante" del cliente. Cuando el cliente requiere un documento, ya sea por ftp, http o gopher, le pasa la solicitud al proxy el cual a su vez verifica si tiene en su cache la información requerida, esta será devuelta al cliente sin conectar en lo posible con el servidor dueño de la información, si el proxy no tiene dicha información, pedirá al servidor dueño de la información que la remita, guardará en su cache los datos, y luego la entregará al cliente.

El instalar un servidor no es un proceso difícil, prueba de ello es la proliferación de organizaciones que tienen instalados ya servidores. El problema es, una vez instalado el servidor, mantenerlo funcionando, cuidar aspectos de seguridad y crear las páginas que le darán vida al servidor. Habrá arquitecturas para las que sea más difícil que el proceso aquí mostrado; sin embargo el código fuente del servidor de httpd viene ya preconfigurado para una amplia variedad de máquinas y sistemas operativos, lo que le facilitarán el trabajo.

DESARROLLO DE LOS CGI ´ S

El Web no es sólo un medio para colocar información de variados tipos y formatos a disposición de los usuarios de Internet. Utilizando el Web es posible también interactuar con el usuario de forma que pueda, por ejemplo, encargar un producto, suscribirse a algún servicio, reservar un pasaje o hacer una consulta a una base de datos.

Los servidores del Web están configurados de tal manera que cada vez que se requiere un archivo de un directorio determinado (usualmente el "cgi-bin"), dicho archivo no es enviado; sino que es ejecutado como un programa y la salida de este programa es enviada a nuestro navegador para que ésta lo muestre. Esta función es conocida como "Common Gateway Interface" (CGI) y los programas a los que nos referimos son llamados programas CGI.

CGI (Common Gateway Interface) es una norma para establecer comunicación entre un servidor WWW y un programa, de tal modo que este último pueda interactuar con Internet.

Un CGI es un programa que se ejecuta en tiempo real en un Web Server en respuesta a una solicitud de un browser o navegador. Cuando esto sucede el Web Server ejecuta un proceso hijo que recibirá los datos que envía el usuario (en caso de que los haya), pone a disposición del mismo algunos datos

en forma de variables de ambiente y captura la salida del programa para enviarlo como respuesta al browser.

El propósito de los CGI's es proveer "inteligencia" e interactividad a un sitio web, por ejemplo, uno puede querer que su aplicación, por lo general de base de datos, pueda utilizarse a través de páginas del Web. Así, se puede hacer que un programa CGI pinte una interfaz para el usuario (forma del Web) quien llena esta solicitud completando ciertos campos de texto o seleccionando opciones de menús. Esta solicitud es recogida por el CGI quien la transforma en comandos y datos comprensibles para la aplicación, a su vez ésta le regresa al CGI los resultados de la consulta y el CGI se los despliega al usuario en una nueva página Web.

También un CGI puede ejecutar otros CGI's y / o una o varias aplicaciones, o puede ser por sí mismo una aplicación simple, con tal de que cumpla con el estándar.

APLICACIONES DE LOS CGI

El CGI se aplica frecuentemente en "search engines", es decir, en combinación con programas de acceso a bases de datos (del tipo Yahoo).

Como intérprete de formas de recolección de información, donde el CGI recibe la información proporcionada por el usuario, la selecciona y realiza alguna acción con ella, desde almacenarla hasta deducir algún hecho de ella.

El CGI también se emplean en robots, los cuales son programas que "simulan alguna acción humana", en este caso, de interacción con el Web, como puede ser el localizar alguna información de manera autónoma, conectándose a diversas páginas o copiando páginas enteras de servidores a nuestra máquina o quizás, traduciendo y dándole formato a un "e-mail" común y corriente o texto ASCII o documento en algún otro formato, pasándolo al formato HTML e instalándolo automáticamente en el servidor.

Hay CGI's que sirven de interfaz con procesadores de imágenes y que están disponibles en el Web de forma gratuita; otros más sirven de interfaz con programas científicos.

Que se puede y no se puede hacer con los CGI's

El estándar CGI surgió al mismo tiempo que el Web, para ampliar sus posibilidades al incluirle programación. Un CGI siempre corre del lado del servidor, lo que significa que cuando se accede a una página Web que contiene un CGI, éste se ejecuta automáticamente en el servidor y lo que se ve posteriormente en una página, es el resultado de esta ejecución. Esto redundante en lo siguiente:

- Aumento de la carga en el servidor, si se ejecutan simultáneamente varios CGI's complejos o que usen aplicaciones complejas.
- Por lo anterior, el cliente no sufre sobrecarga y, de hecho, el usuario puede no darse cuenta de que la página que está viendo ha sido generada por un CGI (y no proviene de un archivo HTML).
- Si no se toman las debidas (y razonables) precauciones al crearlo, un CGI puede, en un momento dado, comprometer la seguridad del servidor, puesto que dentro de un CGI se puede ejecutar cualquier programa o desplegar cualquier archivo que se encuentre en dicho servidor. A pesar de lo anterior, siempre se puede deshabilitar la ejecución de CGI's (comportamiento preestablecido --default) y habilitar sólo ciertos directorios como mecanismo de protección. El cliente no resulta afectado en este sentido.
- Un CGI tiene a su disposición cierta información proveniente del cliente y a través de variables de ambiente se puede conocer, por ejemplo, la dirección IP y el tipo de cliente y, cuando tanto en el cliente como en el servidor se disponen de los recursos adecuados, se puede conocer el usuario de la página actual y más información, aunque esto no está disponible en el comportamiento normal, ni en todos los servidores y mucho menos en los clientes (aunque esta situación puede cambiar en el futuro).
- Puede hacer todo lo que un programa en el servidor sea capaz de hacer, con tal de que su salida sea HTML. Por ejemplo, puede generar archivos de texto, gráficos o sonido, acceder a bases de datos, llamar a otros CGI's o páginas HTML y mezclar todo éstos.
- Al igual que casi todo en el Web, los CGI's no pueden funcionar en sincronía con el usuario, pues tienen que esperar a que el cliente los mande a ejecutar. Por ejemplo, los programas del tipo query son ideales para ser ejecutados por medio de CGI's, sin embargo, aplicaciones más interactivas como editores de texto o graficadores, no correrían o de hacerlo sería en forma mediocre. Algo similar ocurre con aplicaciones que necesiten estar variando porciones de una misma pantalla Web, ya sea continuamente o luego de la ocurrencia de un evento.
- Por supuesto, un CGI se puede combinar con otros recursos del Web, como recarga dinámica, formas, mapas sensitivos, "server includes", applets de Java, etc.

Variables de entorno

La especificación CGI define una serie de variables que el servidor debe generar en el entorno de ejecución de los scripts que invoca.

Las variables que contiene el CGI son:

AUTH_TYPE

Esta variable es específica a las solicitudes hechas con el HTTP, se utiliza para verificar si un acceso externo es auténtico.

CONTENT_LENGTH

Contiene el tamaño de la información añadida a la solicitud. Si no se añade información, entonces es NULL.

CONTENT_TYPE

Contiene el tipo de documento que será enviado sin modificación al cliente. Sólo en caso de que esta variable no tenga valor, el script debe tratar de deducir el tipo de información examinándola. Si aún así no se puede determinar que tipo es, se asumirá el tipo application/octet-stream

GATEWAY_INTERFACE

La versión de la especificación CGI que emplea el servidor. Por ejemplo: 'CGI/1.1'

HTTP_*

Estas variables son específicas a las solicitudes hechas con el HTTP. La interpretación de estas variables depende del valor de SERVER_PROTOCOL. Todas las variables de ambiente que empiezan con "HTTP_" contienen Información de encabezado leído desde el cliente, si el protocolo usado fue HTTP.

Algunos ejemplos recibidos comúnmente son (no hay garantía de que se reciban siempre, ya que dependen del cliente):

HTTP_ACCEPT. Tipo de datos MIME que el cliente puede aceptar (image/gif, image/jpeg, /*/*).

HTTP_USER_AGENT. Indica el browser empleado por el cliente de la forma software/versión biblioteca/versión (Mozilla/4.03 [es] Win95,I).

HTTP_COOKIE. Las cookies que han sido grabadas por el servidor previamente.

HTTP_REFERER. EL documento desde el cual se llamó al CGI.

HTTP_UA_OS. Sistema operativo usado por el cliente.

HTTP_UA_CPU. El tipo de procesador usado por el cliente.

HTTP_COLOR. Indica como el usuario visualiza los colores.

HTTP_UA_PIXELS. Indica la resolución de pantalla del usuario.

PATH_INFO

La trayectoria que será interpretada por el CGI script. Identifica el recurso o subrecurso que será enviado por el programa CGI.

PATH_TRANSLATED

Contiene la trayectoria del Sistema Operativo para el archivo que el servidor intentará acceder para cumplir la solicitud del cliente que contiene el URL absoluto en PATH_INFO.

QUERY_STRING

Es una cadena codificada URL, que contiene la consulta.

REMOTE_ADDR

Es la dirección IP del agente que invocó al CGI

REMOTE_HOST

Es el nombre del agente que hace la solicitud al servidor, si se conoce, de otra manera es NULL.

REMOTE_IDENT

Si el servidor remoto admite ident facilita el nombre del usuario al que pertenece el cliente. Este valor debe únicamente ser considerado como información complementaria o para la elaboración de estadísticas, pero nunca como información de autenticación.

REMOTE_USER

Si el servidor soporta autenticación de usuario y el CGI está protegido, contiene nombre del usuario que está accediendo al CGI.

REQUEST_METHOD

Esta variable contiene el método con que fue hecha la llamada. Para el protocolo HTTP pueden ser "GET", "HEAD", o "POST", etc.

SCRIPT_NAME

Proporciona el URL que identifica al programa CGI.

SERVER_NAME

Proporciona el nombre del servidor, utilizado en la parte "HOST". Puede ser un nombre de dominio o una dirección IP.

SERVER_PORT

Indica el puerto en el que se recibió la solicitud.

SERVER_PROTOCOL

El nombre y revisión del protocolo de información con que se ha generado la llamada al CGI.

SERVER_SOFTWARE

El nombre y versión del software del servidor de información que responde las solicitud.

SECUENCIA DE UN CGI

La secuencia típica de un programa CGI son:

- 1) Leer los datos de entrada .Comúnmente estos datos son introducidos por una Forma.
- 2) Procesar la información.
- 3) Escribir la página HTML resultante al STDOUT

Quando el usuario envía la forma, el programa recibe los datos en un conjunto de pares nombre - valor . Los nombres son los que se definieron en las etiquetas INPUT (o etiquetas SELECT o TEXTAREA), y los valores son cualquier cosa que el usuario escribió o selecciono.

Este conjunto de pares nombre - valor se envía al programa como una sola cadena de caracteres, la cual se necesita separar. Este paso no es muy complicado, y ya existen una gran cantidad de rutinas que lo implementan.

"nombre1=valor1&nombre2=valor2&nombre3=valor3"

Así que solo se necesita separar los ampersands y los signos de igualdad. Así que, solo se necesita hacer dos cosas:

- 1) Convertir todas los signos de "+" a espacios, y
- 2) Convertir todas las secuencias "%xx" al caracter cuyo valor ASCII es "xx", en hexadecimal. Por ejemplo, convertir "%3d" , ya que 3d hexadecimal es 61 en decimal equivale a "=".

Es necesario este último paso porque la cadena original esta codificada en URL, para permitir signos de igualdad, ampersands y otros símbolos, puedan incluirse en los datos del usuario.

Pero, ¿De donde se obtiene esta cadena?. Eso depende del método HTTP en que fue enviada la información de la forma:
Si se utilizo el método GET , la cadena se encuentra en la variable de ambiente QUERY_STRING.

Si se utilizo el metodo POST , se obtiene del STDIN. El número exacto de bytes que se deben leer se encuentra en la variable de ambiente CONTENT_LENGTH.

Diferencia entre los métodos GET y POST

Los metodos GET y POST son dos métodos diferentes definidos en HTTP que realizan cosas muy diferentes, pero ambos son capaces de mandar información de formas al servidor.

Normalmente, GET es utilizado para obtener un archivo o algun otro recurso, posiblemente con parámetros especificando más exactamente que se necesita. En caso de utilizar una forma para introducir datos, GET incluye estos en el URL, por ejemplo:

```
http://servidor.web.mx/cgi-bin/miprograma.cgi?nombre1=valor1&nombre2=valor2
```

El método GET es el que utilizan comúnmente los navegadores para obtener archivos de Internet, tales como archivos HTML e imágenes. A su vez, puede ser utilizado para la mayoría de los envíos de formas, si no hay mucha información (el límite varia de navegador a navegador).

En particular, navegadores y proxies pueden guardar en memoria las respuestas GET, así que si se envían dos formas idénticas puede que solo una sea procesada por el programa CGI. Si se quiere que cada envío sea procesado, evite utilizar GET.

Normalmente, el método POST es utilizado para mandar un pedazo de información al servidor para que lo procese. (El nombre de POST tal vez venga de la idea de enviar notas a un grupo de discusión o a un Newsgroup). Cuando una forma HTML se envía utilizando POST, los datos de la forma son incluidos al final del envío, en el mismo objeto. Esto no es tan rápido y fácil como el metodo GET, pero es más versatil. Por ejemplo, se pueden mandar archivos enteros utilizando POST. También, el tamaño de los datos no es limitado como en el metodo GET.

ESCOGIENDO UN LENGUAJE DE PROGRAMACION

Antes de comenzar un programa, se necesita escoger cual lenguaje de programación utilizar. Para la mayoría de los proyectos, escoger un lenguaje es solo materia de preferencia. Mientras los amantes de UNIX prefieren los shell scripts, perl o C, un usuario Windows preferirá utilizar Visual Basic , C o tal vez un archivo por tareas de DOS. Se debe escoger un lenguaje de programación que sea familiar, pero a su vez que sea una buena elección para el sistema en que se correrá, y que pueda ejecutar las operaciones para las cuales fue escrito el programa.

La mayoría de los programas CGI están escritos en AppleScript, C, C++, Perl, TCL, cualquier shell de Unix o Visual Basic. Estos no son los únicos lenguajes a utilizar, pero son por mucho los más comunes. Mientras se puede escoger cualquier lenguaje que se desee, existen dos buenas razones para considerar alguno de los más comunes.

Primero, será más rápido elaborar programas CGI ya que existen rutinas disponibles. Muchos programadores del World Wide Web han escrito programas CGI para tareas comunes. Algunos han hecho sus programas disponibles gratuitamente a través de Internet. Se pueden encontrar estos programas buscando en varios recetarios en el Web.

Segundo, si se utilizan uno de los lenguajes comunes de programación CGI, es más fácil obtener ayuda para su depuración. Si se tienen problemas con un programa, tal vez se necesite ayuda. Se pueden enviar las preguntas y leer las respuestas en grupos como USENET o listas de correo electrónico. Si se utiliza uno de los lenguajes populares, más personas serán capaces de ayudarte con tus problemas.

WINDOWS CGI

La especificación CGI estándar fue utilizada por los pioneros de CGI para desarrollar aplicaciones usando scripts para algun shell Unix o en el lenguaje PERL. Los datos pasaban del servidor a la aplicación CGI, y viceversa, usando variables de ambiente. La desventaja de utilizar CGI estándar para servidores Web corriendo bajo Windows es que Windows 3.x y Windows 95 no pueden pasar eficientemente variables de ambiente a aplicaciones CGI.

El Windows Common Gateway Interface (Windows CGI) fue desarrollado por Bob Denny , autor de los servidores WWW para Windows WinHTTPD y WebSite, para sobreponerse de las limitaciones de DOS CGI.

Windows no tiene un interpretador de comandos nativo (Windows NT es una excepción). De tal manera que las aplicaciones Windows CGI necesitan ser archivos ejecutables en Windows. Para minimizar los esfuerzos de

programación y mantener el interfase simple, los datos desde el servidor al cliente, y viceversa, es pasado vía archivos de entrada y salida en vez de variables de ambiente.

El archivo de entrada esta en el mismo formato que los archivos de iniciación (.INI). Los archivos .INI comparten el mismo formato por que Windows contiene servicios que permiten a los programadores leer y escribir estos archivos rápida y fácilmente. Estos servicios son parte del Application Programming Interace (API) y son usualmente referenciados como llamadas API. Usar el formato de los archivos .INI fue una brillante idea por que casi todas las aplicaciones Win CGI son escritas en el lenguaje de programación Visual Basic .

PROGRAMANDO UN CGI EN VISUAL BASIC

Para que un programa común y corriente se convierta en CGI tiene que generar como resultado (por la salida estándar) un documento HTML y debe cumplir con los lineamientos del estándar MIME para especificar qué tipo de documento se generará, comúnmente text/html.

Requisitos para crear un CGI en visual Basic:

- Visual Basic 4 versión a 32 bits, o la última versión disponible
- Obtener la última versión del programa CGI32.BAS (ésta se puede obtener en <http://website.ora.com>)
- Tener un Servidor Web que pueda ejecutar los CGI en ambiente Windows

Pasos para crear un CGI en VB:

- Correr VB y crear un nuevo proyecto
- Remover todas las formas y todos los módulos
- Tener solo los controles necesarios
- Adicionar el archivo CGI32.BAS al proyecto
- Crear un nuevo modulo
- En este módulo crear un procedimiento llamado CGI_Main y otro llamado Inter_Main

El procedimiento CGI_Main es el procedimiento principal donde se escribe el código a ejecutar por el CGI.

El procedimiento Inter_Main es la rutina que se usará solamente cuando no se ejecute por un Servidor Web

Después de crear los procedimientos anteriores se puede ya empezar a programar. El siguiente es un ejemplo sencillo para empezar.

Ejemplo 1:

```
Sub CGI_Main()  
  Send ("Content-type: text/html")  
  Send ("")  
  Send ("<HTML><HEAD><TITLE>Hola!</TITLE></HEAD>")  
  Send ("<BODY><H1>Hola!</H1>")  
  Send ("Hola desde Visual Basic!")  
  Send ("<P>Hoy es: " & Now)  
  Send ("</BODY></HTML>")  
End Sub  
  
Sub Inter_Main()  
  MsgBox "Este es un programa CGI."  
  
End Sub
```

El módulo CGI32.BAS contiene una rutina llamada Sub Main(), la cual inicializa el ambiente CGI, en el que se llama al procedimiento CGI_Main() y en este se ejecuta las líneas de código, en este caso, el programa tiene como salida un documento HTML (especificado en la primera línea).

Después de escribir este código, se guarda el módulo (en este caso hola.bas) y se hace ejecutable. Para poder probarlo se tiene que copiar el programa ejecutable en el directorio especial que el servidor Web tenga para ejecutarlo (en este caso se tiene un directorio cgi-win para este propósito). Para mandarlo llamar se tiene que escribir todo el URL de donde se encuentra. (Ej. servidor.web.mx/cgi-win/hola.exe). La siguiente figura muestra el resultado del programa.

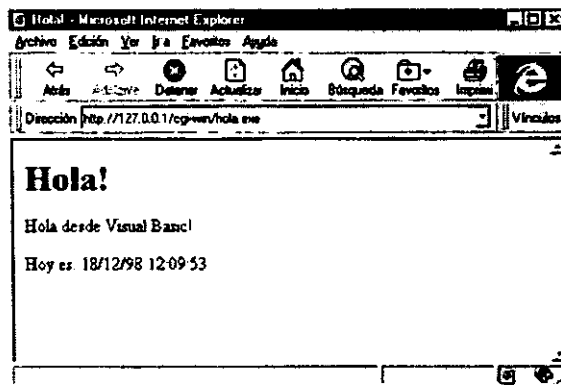


Figura 5.1. Documento generado por el CGI Hola.exe

Ejemplo 2:

El siguiente ejemplo es un programa que tiene como objetivo presentar las variables de entorno que tiene un CGI.

```

Sub CGI_Main()
Dim sel As String
Dim buf As String
Dim i As Integer

Send ("Content-type: text/html")
Send ("X-CGI-Program: VB4 CGI Tester 1.2")
Send ("")
Send ("<HTML><HEAD><TITLE>Resultados del Test</TITLE></HEAD>")
Send ("<BODY><H1>Resultados del test</H1>")
Send ("Program version: 1.3 (32-bit) (16-dlc-98)<BR>")
Send ("Software del servidor: " & CGI_ServerSoftware & "<BR>")
Send ("<HR>")
Send ("HTTP/1.0 200 OK")
Send ("Fecha: " & WebDate(Now))
Send ("<BR>")
Send ("<H2>Variables CGI:</H2>")
Send ("<UL>")
Send ("<LI><I>GATEWAY_INTERFACE: </I>" & CGI_Version)
Send ("<LI><I>SERVER_PROTOCOL: </I>" & CGI_RequestProtocol)
Send ("<LI><I>REQUEST_METHOD: </I>" & CGI_RequestMethod)
Send ("<LI><I>SCRIPT_NAME: </I>" & CGI_ExecutablePath)
Send ("<LI><I>PATH_INFO: </I>" & CGI_LogicalPath)
Send ("<LI><I>PATH_TRANSLATED: </I>" & CGI_PhysicalPath)
Send ("<LI><I>QUERY_STRING: </I>" & CGI_QueryString)
Send ("<LI><I>CONTENT_TYPE: </I>" & CGI_ContentType)
Send ("<LI><I>CONTENT_LENGTH: </I>" & CGI_ContentLength)
Send ("<LI><I>SERVER_SOFTWARE: </I>" & CGI_ServerSoftware)
Send ("<LI><I>SERVER_NAME: </I>" & CGI_ServerName)
Send ("<LI><I>SERVER_PORT: </I>" & CGI_ServerPort)
Send ("<LI><I>Server Admin: </I>" & CGI_ServerAdmin)
Send ("<LI><I>REMOTE_HOST: </I>" & CGI_RemoteHost)
Send ("<LI><I>REMOTE_ADDR: </I>" & CGI_RemoteAddr)
Send ("<LI><I>From: </I>" & CGI_From)
Send ("<LI><I>HTTP_REFERER: </I>" & CGI_REFERER)
Send ("<LI><I>HTTP_USER_AGENT: </I>" & CGI_UserAgent)
Send ("<LI><I>AUTH_TYPE: </I>" & CGI_AuthType)
Send ("<LI><I>REMOTE_USER: </I>" & CGI_AuthUser)
Send ("<LI><I>REMOTE_IDENT: </I>" & CGI_AuthRealm)
Send ("</UL>")
Send ("<H2>Variables del sistema:</H2>")
Send ("<UL>")
Send ("<LI><I>GMT Offset (sec.): </I>" & CLng(CDb1(CGI_GMTOffset) * 86400#))
Send ("<LI><I>Output File: </I>" & UCase$(CGI_OutputFile))
Send ("<LI><I>Content File: </I>" & UCase$(CGI_ContentFile))
Send ("</UL>")

```

```

If CGI_NumAcceptTypes > 0 Then
    Send ("<UL>")
    For i = 0 To CGI_NumAcceptTypes - 1
        If CGI_AcceptTypes(i).value = "Yes" Then
            Send ("<LI>" & CGI_AcceptTypes(i).key)
        Else
            Send ("<LI>" & CGI_AcceptTypes(i).key & " (" & CGI_AcceptTypes(i).value & ")")
        End If
    Next i
    Send ("</UL>")
Else
    Send ("(none)")
End If

Send ("<H2>Encabezados extras:</H2>")
If CGI_NumExtraHeaders > 0 Then
    Send ("<UL>")
    For i = 0 To CGI_NumExtraHeaders - 1
        Send ("<LI><I>" & CGI_ExtraHeaders(i).key & ": </I>" &
CGI_ExtraHeaders(i).value)
    Next i
    Send ("</UL>")
Else
    Send ("(none)")
End If

Send ("<HR>")
Send ("<A HREF=""mailto:" & CGI_ServerAdmin & "">")
Send ("<address>&lt;" & CGI_ServerAdmin & "&gt;</address>")
Send ("</A></BODY></HTML>")

End Sub

Sub Inter_Main()
    MsgBox "Este es un programa CGI para un servidor Web."
End Sub

```

Las siguientes imágenes muestran el resultado del programa anterior.

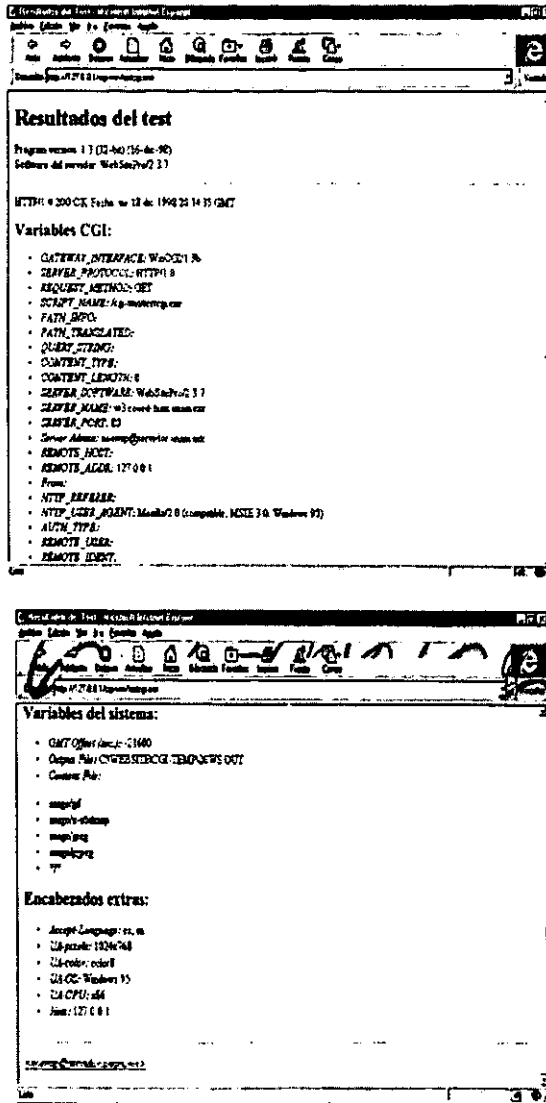


Figura 5.2 Resultado de la ejecución del programa testcgi.exe

Estos son dos ejemplos que nos sirven para tener una idea de lo que se puede hacer con los CGI, pero su principal aplicación son para las bases de datos.

Ejemplo 3:

En este ejemplo se muestra como se puede utilizar los CGI para llenar una base de datos. Primero se tiene que diseñar una forma de llenado con los datos que deseamos que se nos proporcionen.

```
<!-- Este es el código de la forma de llenado -->
<HTML>
<TITLE>Forma de llenado</TITLE>
<BODY>
<FORM ACTION="http://127.0.0.1/cgi-win/llenado.exe" METHOD="GET">
<H2>Forma de llenado</H2>
<HR>
<PRE>
<B>Nombre completo: </B> <INPUT NAME="nombre" TYPE="text" Maxlength="45"
SIZE="45">

<B>Sexo: </B>
    <INPUT TYPE="radio" NAME="sexo" VALUE="Femenino" CHECKED> Femenino
    <INPUT TYPE="radio" NAME="sexo" VALUE="Masculino"> Masculino

<B>Domicilio: </B><INPUT NAME="domicilio" TYPE="text" Maxlength="60"
SIZE="50">

<B>Código postal: </B><INPUT NAME="codigo" TYPE="text" Maxlength="5"
SIZE="5">

<B>Teléfono: </B><INPUT NAME="telefono" TYPE="text" Maxlength="10"
SIZE="10">

    <B>Países: </B> <SELECT NAME="pais">
        <OPTION>México
        <OPTION>Canadá
        <OPTION>Estados Unidos
        <OPTION>Otro
    </SELECT >

</PRE>

<CENTER><INPUT TYPE="submit" VALUE="Enviar datos">
<INPUT TYPE="reset" VALUE="Borrar datos"> </CENTER>
</FORM>
</BODY>
</HTML>
```

Este forma se ve de la siguiente manera:

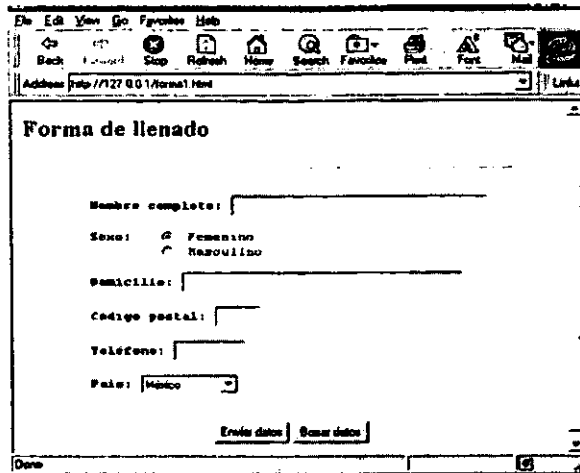


Figura 5.3 Forma1.html

La información que queremos obtener del formulario esta contenida en las variables que se declararon en éste. Estas variables serán enviadas por el método GET, ya que son pocas las variables y de un tamaño definido. En el CGI se obtendrá el valor de las variables y se guardará en una base de datos. El siguiente es el programa que nos permitirá hacer estas operaciones:

```

Sub CGI_Main ()
'Variables para los datos
Dim cNombre As String
Dim cSexo As String
Dim cDomicilio As String
Dim cCp As String
Dim cTelefono As String
Dim cPais As String
'Variables de la base de datos
Dim db As Database
Dim ds As Recordset

'Se asignan los datos obtenidos a las variables
cNombre = GetSmallField("nombre")
cSexo = GetSmallField("sexo")
cDomicilio = GetSmallField("domicilio")
cCp = GetSmallField("codigo")
cTelefono = GetSmallField("telefono")
cPais = GetSmallField("pais")

'Se crea el ambiente para la base de datos
Set db = DBEngine.Workspaces(0).OpenDatabase(App.Path & "\dbsample.mdb")
Set ds = db.OpenRecordset(dbOpenTable)
ds.AddNew ' Agrega un registro a la base
'Cada uno de los datos se escribe en los campos respectivos

```

```

ds!Nombre = cNombre
ds!Sexo = cSexo
ds!Domicilio = cDomicilio
ds!Codigo = cCp
ds!Telefono = cTelefono
ds!Pais = cPais
ds.Update ' Se actualiza el registro
ds.Close ' Se cierra la base

```

' Se manda un documento HTML con los datos

```

Send ("Content-type: text/html")
Send ("")
Send ("<HTML><HEAD><TITLE>Resultado</TITLE></HEAD>")
Send ("<BODY>")
Send ("<H2>Datos recibidos</H2>")
Send ("<HR>")
Send ("<PRE>")
Send ("      Nombre: " & cNombre)
Send ("      Sexo: " & cSexo)
Send ("      Domicilio: " & cDomicilio)
Send ("      Codigo postal: " & cCp)
Send ("      Telefono: " & cTelefono)
Send ("      Pais: " & cPais)
Send ("</PRE>")
Send ("<HR>")
Send ("</BODY></HTML>")

```

End Sub

Sub Inter_Main()

MsgBox "Este es un programa CGI para windows"

End Sub

El resultado de correr el programa anterior es guardar en una base de datos llamada dbsample.mdb los datos contenidos en la forma, como se puede observar el manejo de la base datos es sencillo, como si fuera un programa normal de VB lo único que cambia es que los datos son obtenidos mediante una formulario de HTML y además que se envía un documento HTML como respuesta de que los datos ya están guardados, como se muestra en la siguiente figura.

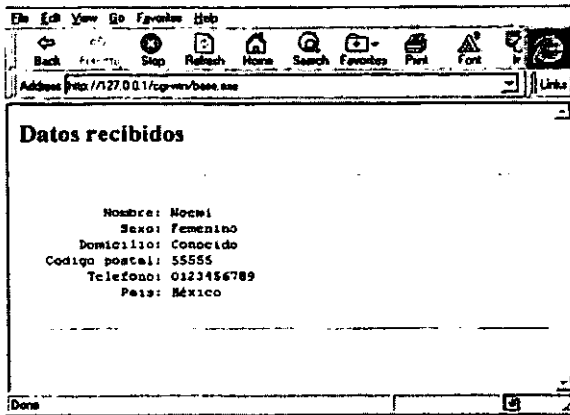


Figura 5.4 Resultado del CGI base.exe

Como se ha visto el manejo de CGI's en un ambiente Windows es fácil, ya que en esta caso (con Visual Basic) solo se necesita de un módulo que nos permita interactuar con el ambiente del Web. Además de que el modo para hacer un programa para Internet es igual a cualquier otro programa, solo cambian algunos comandos.

CONCLUSION

Los CGI's han existido desde el principio del HTML y a mi parecer permanecerán entre nosotros todavía un tiempo más por dos razones:

Facilidad de uso y porque se ejecutan en el servidor.

El lenguaje Java, con mucho auge actualmente, desde cierto punto de vista se puede ver como un fuerte competidor de los CGI's y muchos ya lo toman como sustituto, aunque se trata de dos herramientas diferentes o dos puntos de vista para resolver el mismo problema: programar en el Web.

Primero que nada, Java es un lenguaje de programación con todas las ventajas que esto implica, incluyendo manejo sencillo de gráficos, y cada vez que se accede a un applet, éste se carga en el cliente y es allí donde se corre; esto es, aprovecha todos los recursos y tiene todas las limitaciones del cliente (actualmente esto último se vuelve cada vez más irrelevante), además, la parte "complicada" que significa el programar, se salva cada vez mejor con porciones de código (bibliotecas de applets) ya disponibles. Por otra parte, CGI es un estándar, es decir, una forma de trabajar y no un lenguaje específico; hay que usar un lenguaje externo para crear el programa que cumpla con dicho estándar. Este programa siempre residirá y correrá del lado del servidor, usando sus recursos y observando también sus limitaciones. La diferencia entre ambas herramientas es similar a la observada entre modelos centralizados y modelos distribuidos de comunicación, es por ello que siempre que se desee mantener una aplicación o datos en una sola máquina se preferirá el CGI, mientras que cuando esto no sea importante, se puede preferir la distribución del software con Java.

Los CGI's surgen como necesidad de programar para el Web y específicamente para crear puentes o interfases a programas disponibles en una sola máquina, centralizando la información y el software que la maneje. Los programas Java que quieran servir para este propósito, necesitan dividirse en dos, una parte en el servidor y la otra en el cliente. La parte en el servidor sirve de verdadero puente de la aplicación con el Web; y la que está en el cliente, sirve de puente entre el Web y el usuario. Esto se puede hacer, gracias a que es incidental (o intencional) que un programa Java corra en conjunción con el Web (pues originalmente se diseñó para funcionar como cualquier otro lenguaje independiente).

Por supuesto, un programa en Java puede servir para muchas más cosas que como puente, puede ser una aplicación por sí mismo al igual que con los CGI's, pero sin las restricciones de dicho estándar y por supuesto, sus características no dependen (ni necesitan) del estándar HTML. Por como surge y por todo el apoyo que ha tenido de parte de las empresas, puede hacer todo lo que el cliente le permita desde la primera versión liberada, y esto es

suficiente para la mayoría de los propósitos generales y muchos de los específicos. Por su parte, los CGI's están restringidos a seguir los caminos ya existentes del estándar HTML y sólo pueden crecer en la medida que crezca dicho estándar.

Como conclusión, se puede observar que los CGI's sirven muy bien para el propósito específico de funcionar como puentes para aplicaciones ya existentes y, además, son muy simples. Los programas Java pueden, en un momento dado, servir para dicho propósito pero además tienen muchas otras ventajas y un crecimiento mucho más claro y veloz, tomando como punto de comparación el que con ambos se puede programar en el Web.

APENDICE A

Lista de comandos de HTML

COMANDO	DESCRIPCION
<!-->	Comentario
<A>	Es la tarea para realizar un vínculo. Un vínculo está compuesto por dos partes: el origen o fuente (source) y el destino u objetivo (target). Por ejemplo: Referencias donde "archivo.htm" es el archivo-destino y "Referencias" es el origen
<APPLET>	Inserta una aplicación en el texto
	Fuente negrita
<BLINK>	Hace parpadear el texto.
<BLOCKQUOTE>	Es la orden para indentar párrafos. No sirve para el sangrado de la primera frase
<BODY>	Es el documento mismo. Este parámetro incluye todo el texto y fuera de él se colocan direcciones y frases finales. Atributos: BGCOLOR: permite seleccionar un color de fondo para el documento. Se realiza de dos maneras, con la tabla RGB(red-green-blue) o por el nombre en inglés del color. <BODY BGCOLOR="White"> equivale a <BODY BGCOLOR="#FFFFFF">. LINK: permite seleccionar el color en que se verán los vínculos. Por defecto, es el azul. ALINK: permite seleccionar el color en que se verán los vínculos activos. Un vínculo activo es la que está seleccionándose y se ve debajo del cursor. VLINK: permite seleccionar el color en que se verán los vínculos ya visitados. TEXT: permite seleccionar el color del texto. Por defecto, es el negro.
 	Corte de línea. Para iniciar otra línea.
<CENTER>	Alineación en el centro.
	Permite determinar el tamaño de uno o varios caracteres. El tamaño por defecto de todo texto en HTML es 3. Con FONT puedo designar, por ejemplo, el tamaño 7, o, en relación +1 / -1 con los caracteres anteriores. No afecta las imágenes, y es ignorada por los browser que no la soportan. Atributos: SIZE: es obligatorio. Se utiliza para cualquiera de los dos modos de determinación del tamaño de letra. COLOR: permite elegir el color para la fuente.

<p><FORM></p>	<p>Es la tarea que permite incluir uno o varios formularios dentro de cualquier sección del cuerpo del documento. Dentro de un formulario pueden incorporarse imágenes.</p> <p>Atributos:</p> <p>ACTION: especifica la URL -dirección- a la que se envían los datos contenidos en el formulario. La dirección es del programa que procesa los datos, generalmente localizado en un directorio llamado cgi-bin. Puede usarse los protocolos HTTP o MAILTO. Este atributo es obligatorio.</p> <p><FORM ACTION="http://www.xxx/cgi-bin/respuesta"></p> <p>METHOD: indica el método en que se enviarán los datos hacia el servidor. Solamente hay dos valores (métodos): GET (por defecto) y POST. Este atributo es obligatorio. El método que será internacionalizado es POST.</p> <p>ENCTYPE: especifica el tipo de envío MIME de datos, con el método POST. Por defecto, asume el valor application/x-www-form-urlencoded.</p> <p>ACCEPT: especifica una lista MIME para el correcto procesamiento del formulario.</p>
<p><FRAME></p>	<p>Determina cada sección dentro de un grupo de secciones (FRAMESET). Solamente puede ir dentro de FRAMESET.</p> <p>Atributos:</p> <p>SRC: da la URL del archivo que contiene la FRAME. Si no tiene este atributo, el browser presentará una sección vacía.</p> <p>NAME: le da un nombre a la FRAME, para ser el objetivo de un vínculo. No puede vincularse una FRAME, sino su contenido. El valor de NAME debe ir entre comillas.</p> <p>SCROLLING: cuando una sección excede la pantalla, incorpora barras horizontales o verticales o ambas para ver el resto del documento. Puede asumir los valores: yes (se asegura que las barras aparezcan, aún cuando la sección no excede la pantalla), no (que impide que aparezcan las barras, aún cuando la sección excede la pantalla), y auto (por defecto).</p> <p>TARGET: especifica el nombre de una sección-objetivo en la que un documento debe abrirse.</p> <p>MARGINHEIGHT: Permite cambiar los márgenes verticales entre el borde y el contenido de la FRAME.</p> <p>MARGINWIDTH: Permite cambiar los márgenes horizontales entre el borde y el contenido de la FRAME.</p> <p>FRAMEBORDER: borde de la sección. Valores: 1 (agrega un separador o borde en la intersección de las secciones) y 0 (remueve el borde). Por defecto, es 1.</p> <p>NORESIZE: es para impedir el cambio de tamaño que puede realizar el usuario. También es útil para textos que deban permanecer fijos.</p>

<FRAMESET>	<p>Define un conjunto de secciones de pantalla, o un conjunto de conjuntos de secciones (es anidable). En el documento se coloca en reemplazo de BODY, siendo en general incompatible una tarea con otra.</p> <p>Atributos:</p> <p>COLS: especifica el tamaño horizontal de una sección, en números absolutos de columnas o porcentuales en relación con la pantalla.</p> <p>ROWS: especifica el tamaño vertical de una sección, en números absolutos de líneas o porcentuales en relación con la pantalla.</p> <p>MARGINWIDTH: establece el ancho de los márgenes. Por defecto es 10 píxeles.</p> <p>MARGINHEIGHT: establece el ancho de los márgenes. Por defecto es 10 píxeles.</p>
<H1... H6>	Títulos del BODY, del tamaño 1 al 6.
<HEAD>	Encabezado; parte inicial del documento.
<HR>	Línea horizontal.
<HTML>	Al principio y al fin de todo documento.
<I>	Fuente Itálica.
	<p>Es la tarea que nos permite insertar una imagen en el documento. Es vacía. El atributo SRC es necesario para indicar el archivo que guarda la imagen que queremos insertar. No todos los browsers pueden leer imágenes, por eso es bueno colocar un texto alternativo que se introduce con ALT. Las extensiones de los archivos de imagen son: .gif (el único universalmente leído), .xbm (el primer formato de gráficos de la red), .xpm, .jpeg o .jpg (la extensión más indicada para alta definición de imágenes), .png (formato independiente de alguna empresa). Para los videos, .avi.</p> <p>Atributos:</p> <p>SRC: es obligatorio. Señala el nombre del archivo de imagen a insertar.</p> <p>ALT: opcional. Para aquellos usuarios que tengan un browser que no ve imágenes, señala un texto alternativo (lo más relacionado posible con la imagen) de hasta 1024 caracteres.</p> <p>ALIGN: ubica el texto que rodea a la imagen;. Tiene los valores top, middle, bottom. Netscape ha agregado: texttop, absmiddle, baseline, y absbottom. Los valores no normalizados de left y right solamente son vistos con Netscape, Internet Explorer y Mosaic.</p>
<INPUT>	Permite incluir diferentes campos para ingresos del usuario. Un campo es una caja vacía para incorporar texto. Solamente puede usarse dentro de FORM, y no

	<p>hay límite para la cantidad de cajas (de INPUTs). Los atributos TYPE y NAME son necesarios. El primero lleva un valor normalizado en inglés, mientras el segundo puede llevar un valor en español. El atributo VALUE también puede tener un valor en español. NAME, porque su contenido está relacionado con la base de programación en el servidor; VALUE porque su contenido es lo que se ve con el browser.</p> <p>VALUE: especifica el valor inicial del elemento del INPUT.</p> <p>NAME: designa un nombre-variable al valor dentro de INPUT, es decir, le da un nombre al campo. Es utilizado en pareja con value en el procesamiento del formulario. Es obligatorio.</p> <p>TYPE: especifica el tipo de dato ingresado. Este atributo es obligatorio. Puede asumir los siguientes valores:</p> <p><u>checkbox</u>: genera pequeñas cajas donde se marcarán las respuestas u opciones. Con cada INPUT, se coloca un TYPE=checkbox. Puede elegirse más de una opción. Requiere la presencia de los atributos NAME y VALUE.</p> <p><u>radio</u>: genera pequeñas círculos donde se marcarán las respuestas u opciones. Con cada INPUT, se coloca un TYPE=radio. Sólo puede elegirse una opción.</p> <p><u>hidden</u>: es para incluir información a salvo de posibles cambios producidos por el usuario o el programa browser. Requiere la presencia de los atributos NAME y VALUE.</p> <p><u>image</u>: permite cambiar las cajas vacías de las opciones por imágenes.</p> <p><u>password</u>: enmascara el texto ingresado por el usuario, de modo que éste no aparece en la pantalla. El autor debe proveer una palabra clave.</p> <p><u>reset</u>: es un atributo activo, porque significa que el botón generado puede reinicializar el formulario después de haberlo llenado. Vuelve a presentarlo como al inicio, borrando lo ya escrito.</p> <p><u>submit</u>: es un atributo activo, porque significa que el botón generado indica un asentimiento e inmediato envío del formulario al servidor.</p> <p><u>text</u>: indica que los datos serán texto.</p> <p><u>file</u>: es de Netscape. El usuario puede seleccionar un archivo propio y enviarlo al servidor. Exige que en el ENCTYPE del formulario sea multipart/form-data. Requiere la presencia de los atributos NAME y VALUE.</p>
<p><LH></p>	<p>Es para un encabezamiento dentro de una OL, UL o DL. pertenece a la extensión. Puede incluir vínculos, imágenes.</p>

	Determina cada ítem en una lista ordenada, no ordenada, menú o directorio. Es continente, aunque generalmente se omite el cierre de tarea.
<NOFRAME>	Permite incorporar datos que no pertenecen y no están bajo la ingerencia de la tarea FRAMESET. Las tareas de secciones son incompatibles con BODY y otros elementos del lenguaje, con NOFRAME éste puede incluirse porque genera una isla dentro de FRAMESET.
	La lista ordenada, es una de las dos más usadas (junto a la lista no ordenada). Ordena los ítems según números o letras. Cada elemento va precedido de LI, que es el único elemento que puede contener una lista ordenada. Las listas pueden ser anidadas, esto es, pueden acumularse.
<OPTION>	Define las diversas opciones a seleccionar dentro de un campo de un formulario. Esta tarea se ingresa dentro de SELECT, y sólo puede contener texto (no puede incluir lenguaje de marcado). Atributos: VALUE: especifica un valor para cada OPTION.
<P>	Es la tarea para iniciar un párrafo, componente primordial de un documento. Implica un salto de línea, por ello, entre un párrafo y otro queda espacio en blanco (a diferencia de BR, que solamente implica un corte de línea).Puede usarse aislado para dejar un espacio en blanco. Es la tarea más básica, aunque no obligatoria.
<PRE>	Es una tarea muy usada que marca al texto que contiene como texto premarcado. Es muy flexible, por ejemplo, permite hacer tablas cuando el browser es sólo texto, y, en general, para secciones de documentos cuyas características requieren un comando HTML que no existe. Los caracteres son vistos en letra monoespaciada. También puede usarse para dejar espacio en blanco entre un párrafo y otro, con dos espacios hechos con la barra entre principio y fin de tarea. PRE solamente puede incluir vínculos y caracteres de estilo. Imágenes, tablas, etc. no son posibles. Tampoco incluye los signos precedidos por &, éstos deben ser usados normalmente.
<SELECT>	Permite al usuario seleccionar uno o varios valores de un campo, dentro de un formulario. Esos valores son ingresados por OPTION.
<SUB>	Es la tarea que permite que uno o más caracteres estén subcriptos. Fue creada para ser utilizada en fórmulas matemáticas, pero puede ser usada en otros casos.
<SUP>	Permite colocar uno o varios caracteres sobrescritos en relación con el carácter anterior. Como SUB, fue creada para fórmulas matemáticas.
<TABLE>	Indica la presencia de una tabla en el documento.

	<p>Pertenece a la extensión del lenguaje, pero es soportada por casi todos los browsers. En el caso de browsers de sólo texto, pueden realizarse tablas con <PRE>. Las tablas se ordenan en columnas, líneas y celdas. Los datos dentro de las celdas, el contenido en sí de la tabla, pueden ser palabras, formularios, vínculos, listas o imágenes. TABLE implica un corte de línea.</p> <p>Atributos:</p> <p>ALIGN: permite alinear la tabla. Por defecto asume la izquierda. Puede asumir los valores: right y left, como valores básicos; también son posibles center, justify.</p> <p>BORDER: Por defecto algunos browser (la mayoría) presentan las tablas sin borde; Netscape da por defecto el valor de 1. Puede aumentarse el grosor del borde, o, para Netscape, dar el valor 0 para que no haya.</p>
<TD>	<p>Dato de tabla, es la tarea que define una celda con datos. Se coloca dentro de TR. Cada inicio y fin de TD genera una columna dentro de una línea. A lo largo de las líneas, colocar siempre la misma cantidad de TD, de lo contrario pueden producirse columnas erróneas. El fin de tarea es opcional, porque una segunda TD indica el fin de la primera y /TR el fin de la línea. Puede contener vínculos, imágenes, formularios, listas, etc., aunque no está normalizado.</p> <p>Atributos:</p> <p>ALIGN: define la alineación horizontal del contenido de la celda. Puede asumir los valores: right y left, como valores básicos; también son posibles center, justify. Si hay otra alineación dada por TR, la de TD tiene preeminencia dentro de su contenido.</p> <p>COLSPAN: especifica cuántas columnas ocupará la TD. El valor por defecto es 1. ROWSPAN: especifica cuántas líneas ocupará la TD. El valor por defecto es 1.</p>
<TEXTAREA>	<p>Permite al usuario ingresar un texto en bloque dentro de un formulario. Es un campo más extenso que los INPUTs.</p> <p>Atributos:</p> <p>COLS: define en valores absolutos el ancho del área de ingreso, en columnas. Es obligatorio.</p> <p>ROWS: define en valores absolutos la altura del área de ingreso, en líneas. Es obligatorio.</p> <p>NAME: da el nombre-variable del contenido del área. Es obligatorio.</p> <p>DISABLED: desactiva el control para ingresos de usuario.</p>
<TH>	<p>Table header, determina un encabezamiento de tabla del tamaño de una celda. Puede ser un encabezamiento para una o varias columnas y una o varias líneas. Si entre</p>

	<p>inicio y fin de tarea no se ingresa ningún dato, aparecerá un encabezamiento vacío (en blanco). Puede contener vínculos, imágenes, formularios, listas, etc., aunque no está normalizado.</p> <p>Atributos:</p> <p>ALIGN: define la alineación horizontal del contenido de la celda. Puede asumir los valores: right y left, como valores básicos; también son posibles center, justify. Si hay otra alineación dada por TR, la de TH tiene preeminencia dentro de su contenido.</p> <p>COLSPAN: especifica cuántas columnas ocupará la TH. El valor por defecto es 1.</p> <p>ROWSPAN: especifica cuántas líneas ocupará la TH. El valor por defecto es 1. ")" indica que la celda se expande la cantidad de líneas indicadas</p>
<TITLE>	<p>Es la tarea más relevante del encabezado. Es un título del documento tomado como totalidad. Cuando se lea el documento, aparecerá en la barra del tope superior de la pantalla, como indicador. Para que la ventaja sea realmente efectiva, no se recomienda colocar el encabezado de un capítulo, sino del documento en su totalidad. Se recomienda una longitud de hasta 60 caracteres.</p>
<TR>	<p>Table row, es la tarea para generar una línea de tabla. Es imprescindible para colocar cada grupo de datos contiguos.</p> <p>Atributos:</p> <p>ALIGN: define la alineación horizontal del contenido de las celdas en la línea. Puede asumir los valores: right y left, como valores básicos; también son posibles center, justify.</p> <p>VALIGN: define la alineación vertical del contenido de las celdas en la línea. Puede asumir los valores top, middle, bottom, baseline. Baseline especifica que la base del contenido de la celda debe alinearse verticalmente con la base del texto de las otras celdas de la línea. Este atributo pertenece a la extensión del HTML.</p>
<TT>	<p>Es la orden para que uno o varios caracteres sean visualizados en letra monoespaciada. También con la tarea PRE se visualiza en esta letra, sin embargo con TT no pueden marcarse nada más que letras.</p>
<U>	<p>Texto subrayado.</p>
	<p>Lista no ordenada, es, junto con la anterior, la lista más usada. Cada renglón es precedido por una LI, que es el único elemento que puede ir dentro de una UL. Organiza verticalmente los ítems, cada uno de los cuales se</p>

	visualiza con un símbolo previo, como un asterisco o un círculo, etc. Como el resto de las listas, la orden UL incluye un corte de línea. También cada LI incluye un corte de línea.
--	--

APENDICE B

Archivo de entorno del WINCGI

Como la CGI sólo define una interfaz, no entra en la especificación del lenguaje de programación con el que se deben crear los programas CGI. Sólo indica la forma en que debe realizarse la comunicación con la interfaz, pero no define cómo debe implementarse, ni sobre qué sistema operativo. Esto abre inmensas posibilidades ya que, en la práctica, se puede utilizar desde cualquier lenguaje interpretado del estilo de Perl en máquinas UNIX a cualquier lenguaje de programación compilado del estilo de Visual Basic en máquinas Intel bajo Windows.

La especificación CGI fue realizada pensando en máquinas UNIX. En éstas, la información CGI del servidor web se suministraba a través de la entrada/salida estándar y de variables de entorno. Sin embargo hay una serie de características que la hacen inutilizable en máquinas Windows (por ejemplo el manejo de stdin/stdout). Es por ello que los creadores del servidor Website (para Windows), transformaron la especificación para adaptarla a los ordenadores con entorno Windows. A esta nueva especificación se la denominó WinCGI. Con ella, en lugar de la entrada y salida estándar (stdin/stdout) se utilizan ficheros de intercambio, lo que permite programar aplicaciones con cualquier compilador: Visual Basic, C++, Delphi, etc. Actualmente su versión oficial es la 1.3.a.

La línea de comandos que el servidor web empleará para ejecutar una aplicación CGI (WinCGI, en realidad), constará del nombre del programa CGI ejecutable y de un nombre de fichero con extensión ini, que servirá de archivo de intercambio entre el servidor web y el programa CGI.

Esto quiere decir que será una línea del estilo: C:\cgi-win\application_cgi.exe
C:\temp\fichero_de_intercambio.ini

La aplicación aplicacion_cgi.exe será el programa CGI que invoca el servidor web ante una petición de un cliente, y el archivo fichero_de_intercambio.ini será el archivo que el servidor y el programa CGI utilizarán para informar y ser informados, respectivamente, de la petición que haga el cliente.

El hecho de que al realizar la especificación de WinCGI los autores se inclinaron por utilizar un archivo con extensión ini como archivo de intercambio, se debe a la facilidad de manejo de este tipo de archivos en entornos Windows, ya que la propia API de Windows dispone de funciones de acceso directo a cualquier línea de cualquier sección de un fichero de inicialización. En nuestro caso esta labor será incluso más fácil, ya que existe un módulo en Visual Basic (CGI32.BAS), especializada en la lectura y escritura de este tipo de ficheros, y que convierte el manejo de los mismos en algo trivial.

El fichero ini es pues la parte estándar a la que deberán acomodarse tanto el servidor como el programa CGI si quieren entenderse. Es decir, no podremos inventarnos las secciones de este archivo, sino que éstas deberán seguir unas normas concretas.

En concreto este fichero consta de ocho secciones, con los siguientes títulos: Accept, System, ExtraHeaders, FormLiteral, FormExternal, FormHuge y FormFile. Cualquier dato de intercambio debe ir insertado en una de esas ocho secciones. Vemos estas secciones una a una, con sus posibles entradas y valores. Cada elemento de cada sección, se muestra en la forma entrada=valor. El valor será siempre una cadena (string), y si esta cadena está vacía, la entrada se omite.

SECCION [CGI]

Esta sección contiene la identificación del programa CGI e información de acceso.

- CGI Version. Versión de la especificación WinCGI utilizada por el servidor.
- Query String. Si la URL contiene el signo ? entonces, en esta entrada se guarda la cadena que sigue a dicho signo (que contendrá los parámetros de la consulta). El servidor debe suministrar este valor tal cual lo envió el cliente, quedando bajo la responsabilidad del programa CGI, la descodificación del mismo en las partes clave=valor.
- Server Name. Nombre de dominio del servidor (para poder reconstruir la URL completa).
- Server Port. Puerto a través del cual está escuchando el servidor.
- Server Software. Nombre y versión del servidor web.
- Referer. La URL de la página web que contiene el enlace al programa CGI (esta entrada puede no estar presente si el enlace fue escrito con el teclado o activado a través de un bookmark).
- Request Method. Método de petición (GET ó POST) utilizado. El método POST suele utilizarse cuando los datos que deben transmitirse son tan largos que el método GET no puede manejarlos.
- Request Protocol. Protocolo y versión, a través del cual se ha recibido la petición.
- Remote Address. Dirección IP del cliente.
- Remote Host. Nombre de dominio del cliente.
- Content File. Ruta completa del fichero con los datos que envió el cliente (y que ha rellenado el servidor con esos datos).
- Content Length. Longitud de los datos enviados.
- Content Type. Tipo MIME del contenido de los datos
- Executable Path. Ruta del fichero WinCGI que está siendo ejecutado.

SECCION [ACCEPT]

Esta sección contiene la lista de los tipos de datos que acepta el cliente. La sintaxis será: tipo/subtipo=valor. Si el valor no se incluye, se entiende por defecto que es yes (sí). Por ejemplo, esta sección podría ser como la siguiente:

```
[Accept]
image/gif=yes
image/x-xbitmap=yes
image/jpeg=yes
image/pjpeg=yes
/**=yes
```

SECCION [SYSTEM]

La sección System contiene información específica de la implementación del CGI.

- GMT Offset. Tiempo (en segundos) que es necesario añadir a la hora GMT (Greenwich Mean Time, Hora Respecto al Meridiano Greenwich) para convertirla a la hora local.
- Debug Mode. Informa si el servidor está en modo "depuración".
- Output File. Ruta completa del fichero en el que el servidor web espera recibir los resultados del programa CGI.
- Content File. Ruta completa del fichero que almacena el contenido de la petición.

Por ejemplo, esta sección podría ser como la siguiente:

```
[System]
GMT Offset=-3600
Debug Mode=no
Output File=c:\home\cgi-temp\salida_cgi.out
Content File=c:\home\cgi-temp\salida_cgi.con
```

SECCION [EXTRA HEADERS]

Cabeceras "extra" incluidas en la petición. Por ejemplo:

```
[Extra Headers]
User Agent=Mozilla/4.03 [es] (Win95; I)
Connection=Keep-Alive
Pragma=no-cache
Host=localhost
```

SECCION [FORM LITERAL]

Si se usa el método POST en la petición y ésta proviene a su vez de un formulario, en esta sección se escribirán las entradas correspondientes a las

entradas del formulario HTML. Los datos vendrán en la URL en la forma "entrada1=valor1&entrada2=valor2..." El servidor deberá dividir esta sentencia en pares "entradaX=valorX" y reemplazar los signos + por espacios en blanco (ya que los espacios en blanco se transmiten de esta forma en la URL).

Esas entradas son las que el servidor web introducirá en el fichero de extensión ini. Por ejemplo, esta sección podría ser como la siguiente:

```
[Form Literal]
Articulo1=ServidorWeb2.htm
Articulo2=ClienteEcho.htm
Nombre=Noemi
Apellidos=Parra
Email=noemip@servidor.unam.mx
```

SECCION [FORM EXTERNAL]

Si los valores de una entrada del formulario tienen más de 255 caracteres o contienen caracteres de control, el servidor web escribe el valor descodificado en un fichero temporal y añade una línea en esta sección.

Esta sección se podría parecer a lo siguiente:

```
[Form External]
field300chars=C:\home\cgi-temp\HS19AF6C.000 300
fieldwithlinebreaks=C:\home\cgi-temp\HS19AF6C.001 43
```

SECCION [FORM HUGE]

Esta sección amplía a la anterior, y se utiliza cuando el valor de una entrada es mayor de 64K. En ese caso, el servidor no descodifica el valor, pero añade una línea en esta sección, en la que se indica el fichero de contenido donde aparecerá dicho valor y la posición del primer byte del valor, en dicho archivo. Por ejemplo:

```
[Form Huge]
field230K=c:\home\cgi-temp\HS19AF6C.002 276920If
```

SECCION [FORM FILE]

Esta sección se utiliza cuando la petición contiene varios uploads y no la utilizaremos en ningún caso.

APENDICE C

Módulo CGI32.BAS

Attribute VB_Name = "CGI_Framework"

```

'-----
' *****
' * CGI32.BAS *
' *****
'

```

' VERSION: 1.9 (July 17, 1997)

' AUTHORS: Robert B. Denny <rdenny@dc3.com>
' Christopher J. Duke <duke@ora.com>

' Common routines needed to establish a VB environment for
' Windows CGI programs that run behind the WebSite Server.

' INTRODUCTION

' The Common Gateway Interface (CGI) version 1.1 specifies a minimal
' set of data that is made available to the back-end application by
' an HTTP (Web) server. It also specifies the details for passing this
' information to the back-end. The latter part of the CGI spec is
' specific to Unix-like environments. The NCSA httpd for Windows does
' supply the data items (and more) specified by CGI/1.1, however it
' uses a different method for passing the data to the back-end.

' DEVELOPMENT

' WebSite requires any Windows back-end program to be an
' executable image. This means that you must convert your VB
' application into an executable (.EXE) before it can be tested
' with the server.

' ENVIRONMENT

' The WebSite server executes script requests by doing a
' CreateProcess with a command line in the following form:

```
' prog-name cgi-profile
```

' THE CGI PROFILE FILE

' The Unix CGI passes data to the back end by defining environment
' variables which can be used by shell scripts. The WebSite
' server passes data to its back end via the profile file. The
' format of the profile is that of a Windows ".INI" file. The keyword
' names have been changed cosmetically.

' There are 7 sections in a CGI profile file, [CGI], [Accept],

' [System], [Extra Headers], and [Form Literal], [Form External],
' and [Form huge]. They are described below:

' [CGI] <== The standard CGI variables
' CGI Version= The version of CGI spoken by the server
' Request Protocol= The server's info protocol (e.g. HTTP/1.0)
' Request Method= The method specified in the request (e.g., "GET")
' Request Keep-Alive= If the client requested connection re-use (Yes/No)
' Executable Path= Physical pathname of the back-end (this program)
' Logical Path= Extra path info in logical space
' Physical Path= Extra path info in local physical space
' Query String= String following the "?" in the request URL
' Content Type= MIME content type of info supplied with request
' Content Length= Length, bytes, of info supplied with request
' Request Range= Byte-range specification received with request
' Server Software= Version/revision of the info (HTTP) server
' Server Name= Server's network hostname (or alias from config)
' Server Port= Server's network port number
' Server Admin= E-Mail address of server's admin. (config)
' Referer= URL of referring document
' From= E-Mail of client user (rarely seen)
' User Agent= String describing client/browser software/version
' Remote Host= Remote client's network hostname
' Remote Address= Remote client's network address
' Authenticated Username=Username if present in request
' Authenticated Password=Password if present in request
' Authentication Method=Method used for authentication (e.g., "Basic")
' Authentication Realm=Name of realm for users/groups

' [Accept] <== What the client says it can take
' The MIME types found in the request header as
' Accept: xxx/yyy; zzzz...
' are entered in this section as
' xxx/yyy=zzzz...
' If only the MIME type appears, the form is
' xxx/yyy=Yes

' [System] <== Windows interface specifics
' GMT Offset= Offset of local timezone from GMT, seconds (LONG!)
' Output File= Pathname of file to receive results
' Content File= Pathname of file containing raw request content
' Debug Mode= If server's CGI debug flag is set (Yes/No)

' [Extra Headers]
' Any "extra" headers found in the request that activated this
' program. They are listed in "key=value" form. Usually, you'll see
' at least the name of the browser here as "User-agent".

' [Form Literal]
' If the request was a POST from a Mosaic form (with content type of
' "application/x-www-form-urlencoded"), the server will decode the

' form data. Raw form input is of the form "key=value&key=value&...",
 ' with the value parts "URL-encoded". The server splits the key=value
 ' pairs at the '&', then splits the key and value at the '=',
 ' URL-decodes the value string and puts the result into key=value
 ' (decoded) form in the [Form Literal] section of the INI.

' [Form External]

' If the decoded value string is more than 254 characters long,
 ' or if the decoded value string contains any control characters
 ' or quote marks the server puts the decoded value into an external
 ' tempfile and lists the field in this section as:

' key=<pathname> <length>

' where <pathname> is the path and name of the tempfile containing
 ' the decoded value string, and <length> is the length in bytes
 ' of the decoded value string.

' NOTE: BE SURE TO OPEN THIS FILE IN BINARY MODE UNLESS YOU ARE
 ' CERTAIN THAT THE FORM DATA IS TEXT!

' [Form File]

' If the form data contained any uploaded files, they are described in
 ' this section as:

' key=[<pathname>] <length> <type> <encoding> [<name>]

' where <pathname> is the path and name of the tempfile containing the
 ' uploaded file, <length> is the length in bytes of the uploaded file,
 ' <type> is the content type of the uploaded file as sent by the browser,
 ' <encoding> is the content-transfer encoding of the uploaded file, and
 ' <name> is the original file name of the uploaded file.

' [Form Huge]

' If the raw value string is more than 65,536 bytes long, the server
 ' does no decoding. In this case, the server lists the field in this
 ' section as:

' key=<offset> <length>

' where <offset> is the offset from the beginning of the Content File
 ' at which the raw value string for this key is located, and <length>
 ' is the length in bytes of the raw value string. You can use the
 ' <offset> to perform a "Seek" to the start of the raw value string,
 ' and use the length to know when you have read the entire raw string
 ' into your decoder. Note that VB has a limit of 64K for strings, so

' Examples:

' [Form Literal]

' smallfield=123 Main St. #122

' [Form External]

' field300chars=c:\website\cgi-tmp\1a7fws.000 300

' fieldwithlinebreaks=c:\website\cgi-tmp\1a7fws.001 43

' [Form Huge]

```

' field230K=c:\website\cgi-tmp\1a7fws.002 276920
'
' =====
' USAGE
' =====
' Include CGI32.BAS in your VB4 or VB5 project. Set the project options for
' "Sub Main" startup. The Main() procedure is in this module, and it
' handles all of the setup of the VB CGI environment, as described
' above. Once all of this is done, the Main() calls YOUR main procedure
' which must be called CGI_Main(). The output file is open, use Send()
' to write to it. The input file is NOT open, and "huge" form fields
' have not been decoded.
'
' NOTE: If your program is started without command-line args,
' the code assumes you want to run it interactively. This is useful
' for providing a setup screen, etc. Instead of calling CGI_Main(),
' it calls Inter_Main(). Your module must also implement this
' function. If you don't need an interactive mode, just create
' Inter_Main() and put a 1-line call to MsgBox alerting the
' user that the program is not meant to be run interactively.
' The samples furnished with the server do this.
'
' If a Visual Basic runtime error occurs, it will be trapped and result
' in an HTTP error response being sent to the client. Check out the
' Error_Handler() sub. When your program finishes, be sure to RETURN
' TO MAIN(). Don't just do an "End".
'
' Have a look at the stuff below to see what's what.
'
'-----
' Author: Robert B. Denny <rdenny@netcom.com>
'        April 15, 1995
'
' Revision History:
' 15-Apr-95 rbd Initial release (ref VB3 CGI.BAS 1.7)
' 02-Aug-95 rbd Changed to take input and output files from profile
'              Server no longer produces long command line.
' 24-Aug-95 rbd Make call to GetPrivateProfileString conditional
'              so 16-bit and 32-bit versions supported. Fix
'              computation of CGI_GMTOffset for offset=0 (GMT)
'              case. Add FieldPresent() routine for checkbox
'              handling. Clean up comments.
' 29-Oct-95 rbd Added PlusToSpace() and Unescape() functions for
'              decoding query strings, etc.
' 16-Nov-95 rbd Add keep-alive variable, file uploading description
'              in comments, and upload display.
' 20-Nov-95 rbd Fencepost error in ParseFileValue()
' 23-Nov-95 rbd Remove On Error Resume Next from error handler
' 03-Dec-95 rbd User-Agent is now a variable, real HTTP header
'              Add Request-Range as http header as well.
' 30-Dec-96 rbd Fix "endless" loop if call FieldPresent() with

```

```

'      zero form fields.
' 23-Feb-97 rbd  Per MS Tech Support, do not do Exit Sub in sub Main
'              as this can corrupt VB40032.DLL. Use End instead.
' 17-Jul-97 cjd  1.9: Removed MAX_FORM_TUPLES (was previously a
'              constant set to 100) and replaced it with a
'              varying-sized array in GetFormTuples().
-----

```

Option Explicit

```

' =====
' Manifest Constants
' =====

```

```

Const MAX_CMDARGS = 8      ' Max # of command line args
Const ENUM_BUF_SIZE = 4096 ' Key enumeration buffer, see GetProfile()
' These are the limits in the server
Const MAX_XHDR = 100      ' Max # of "extra" request headers
Const MAX_ACCTYPE = 100   ' Max # of Accept: types in request
Const MAX_HUGE_TUPLES = 16 ' Max # "huge" form fields
Const MAX_FILE_TUPLES = 16 ' Max # of uploaded file tuples

```

```

' =====
' Types
' =====

```

```

Type Tuple          ' Used for Accept: and "extra" headers
  key As String     ' and for holding POST form key=value pairs
  value As String
End Type

```

```

Type FileTuple      ' Used for form-based file uploads
  key As String     ' Form field name
  file As String    ' Local tempfile containing uploaded file
  length As Long    ' Length in bytes of uploaded file
  type As String    ' Content type of uploaded file
  encoding As String ' Content-transfer encoding of uploaded file
  name As String    ' Original name of uploaded file
End Type

```

```

Type HugeTuple      ' Used for "huge" form fields
  key As String     ' Keyword (decoded)
  offset As Long    ' Byte offset into Content File of value
  length As Long    ' Length of value, bytes
End Type

```

```

' =====
' Global Constants
' =====

```

```

' -----
' Error Codes
' -----
'
Global Const ERR_ARGCOUNT = 32767
Global Const ERR_BAD_REQUEST = 32766      ' HTTP 400
Global Const ERR_UNAUTHORIZED = 32765     ' HTTP 401
Global Const ERR_PAYMENT_REQUIRED = 32764 ' HTTP 402
Global Const ERR_FORBIDDEN = 32763        ' HTTP 403
Global Const ERR_NOT_FOUND = 32762        ' HTTP 404
Global Const ERR_INTERNAL_ERROR = 32761   ' HTTP 500
Global Const ERR_NOT_IMPLEMENTED = 32760  ' HTTP 501
Global Const ERR_TOO_BUSY = 32758         ' HTTP 503 (experimental)
Global Const ERR_NO_FIELD = 32757         ' GetxxxField "no field"
Global Const CGI_ERR_START = 32757        ' Start of our errors

' =====
' CGI Global Variables
' =====
'
' -----
' Standard CGI variables
' -----
'
Global CGI_ServerSoftware As String
Global CGI_ServerName As String
Global CGI_ServerPort As Integer
Global CGI_RequestProtocol As String
Global CGI_ServerAdmin As String
Global CGI_Version As String
Global CGI_RequestMethod As String
Global CGI_RequestKeepAlive As Integer
Global CGI_LogicalPath As String
Global CGI_PhysicalPath As String
Global CGI_ExecutablePath As String
Global CGI_QueryString As String
Global CGI_RequestRange As String
Global CGI_REFERER As String
Global CGI_From As String
Global CGI_UserAgent As String
Global CGI_RemoteHost As String
Global CGI_RemoteAddr As String
Global CGI_AuthUser As String
Global CGI_AuthPass As String
Global CGI_AuthType As String
Global CGI_AuthRealm As String
Global CGI_ContentType As String
Global CGI_ContentLength As Long
'
' -----
' HTTP Header Arrays

```



```

'-----
Global CGI_AcceptTypes(MAX_ACCTYPE) As Tuple ' Accept: types
Global CGI_NumAcceptTypes As Integer ' # of live entries in array
Global CGI_ExtraHeaders(MAX_XHDR) As Tuple ' "Extra" headers
Global CGI_NumExtraHeaders As Integer ' # of live entries in array
'-----
' POST Form Data
'-----
Global CGI_FormTuples() As Tuple ' Declare dynamic array for POST form
key=value pairs
Global CGI_NumFormTuples As Integer ' # of live entries in array
Global CGI_HugeTuples(MAX_HUGE_TUPLES) As HugeTuple ' Form "huge tuples"
Global CGI_NumHugeTuples As Integer ' # of live entries in array
Global CGI_FileTuples(MAX_FILE_TUPLES) As FileTuple ' File upload tuples
Global CGI_NumFileTuples As Integer ' # of live entries in array
'-----
' System Variables
'-----
Global CGI_GMTOffset As Variant ' GMT offset (time serial)
Global CGI_ContentFile As String ' Content/Input file pathname
Global CGI_OutputFile As String ' Output file pathname
Global CGI_DebugMode As Integer ' Script Tracing flag from server
'-----
' =====
' Windows API Declarations
' =====
' NOTE: Declaration of GetPrivateProfileString is specially done to
' permit enumeration of keys by passing NULL key value. See GetProfile().
' Both the 16-bit and 32-bit flavors are given below. We DO NOT
' recommend using 16-bit VB4 with WebSite!

#If Win32 Then
Declare Function GetPrivateProfileString Lib "kernel32" _
Alias "GetPrivateProfileStringA" _
(ByVal lpApplicationName As String, _
ByVal lpKeyName As Any, _
ByVal lpDefault As String, _
ByVal lpReturnedString As String, _
ByVal nSize As Long, _
ByVal lpFileName As String) As Long
#Else
Declare Function GetPrivateProfileString Lib "Kernel" _
(ByVal lpSection As String, _
ByVal lpKeyName As Any, _

```

```

    ByVal IpDefault As String, _
    ByVal IpReturnedString As String, _
    ByVal nSize As Integer, _
    ByVal IpFileName As String) As Integer
#End If
'
'
' =====
' Local Variables
' =====
'
Dim CGI_ProfileFile As String      ' Profile file pathname
Dim CGI_OutputFN As Integer       ' Output file number
Dim ErrorString As String

'-----
'
' Return True/False depending on whether a form field is present.
' Typically used to detect if a checkbox in a form is checked or
' not. Unchecked checkboxes are omitted from the form content.
'
'-----
Function FieldPresent(key As String) As Integer
    Dim i As Integer

    FieldPresent = False          ' Assume failure

    If (CGI_NumFormTuples = 0) Then Exit Function ' Stop endless loop

    For i = 0 To (CGI_NumFormTuples - 1)
        If CGI_FormTuples(i).key = key Then
            FieldPresent = True    ' Found it
            Exit Function          ' ** DONE **
        End If
    Next i

    ' Exit with FieldPresent still False
End Function

'-----
'
' ErrorHandler() - Global error handler
'
' If a VB runtime error occurs during execution of the program, this
' procedure generates an HTTP/1.0 HTML-formatted error message into
' the output file, then exits the program.
'
' This should be armed immediately on entry to the program's main()
' procedure. Any errors that occur in the program are caught, and
' an HTTP/1.0 error message is generated into the output file. The

```

' presence of the HTTP/1.0 on the first line of the output file causes
 ' NCSA httpd for WIndows to send the output file to the client with no
 ' interpretation or other header parsing.

 Sub ErrorHandler(code As Integer)

```

    Seek #CGI_OutputFN, 1 ' Rewind output file just in case
    Send ("HTTP/1.0 500 Internal Error")
    Send ("Server: " + CGI_ServerSoftware)
    Send ("Date: " + WebDate(Now))
    Send ("Content-type: text/html")
    Send ("")
    Send ("<HTML><HEAD>")
    Send ("<TITLE>Error in " + CGI_ExecutablePath + "</TITLE>")
    Send ("</HEAD><BODY>")
    Send ("<H1>Error in " + CGI_ExecutablePath + "</H1>")
    Send ("An internal Visual Basic error has occurred in " + CGI_ExecutablePath + ".")
    Send ("<PRE>" + ErrorString + "</PRE>")
    Send ("<I>Please</I> note what you were doing when this problem occurred,")
    Send ("so we can identify and correct it. Write down the Web page you were
using,")
    Send ("any data you may have entered into a form or search box, and")
    Send ("anything else that may help us duplicate the problem. Then contact the")
    Send ("administrator of this service: ")
    Send ("<A HREF=""mailto:" & CGI_ServerAdmin & "">")
    Send ("<ADDRESS>&lt;" + CGI_ServerAdmin + "&gt;</ADDRESS>")
    Send ("</A></BODY></HTML>")
  
```

Close #CGI_OutputFN

```

'=====
End ' Terminate the program
'=====
  
```

End Sub

 ' GetAcceptTypes() - Create the array of accept type structs
 ' Enumerate the keys in the [Accept] section of the profile file,
 ' then get the value for each of the keys.

 Private Sub GetAcceptTypes()

```

    Dim sList As String
    Dim i As Integer, j As Integer, l As Integer, n As Integer

    sList = GetProfile("Accept", "") ' Get key list
    l = Len(sList) ' Length incl. trailing null
    i = 1 ' Start at 1st character
    n = 0 ' Index in array
    Do While ((i < l) And (n < MAX_ACCTYPE)) ' Safety stop here
  
```

```

j = InStr(i, sList, Chr$(0))      ' J -> next null
CGI_AcceptTypes(n).key = Mid$(sList, i, j - i) ' Get Key, then value
CGI_AcceptTypes(n).value = GetProfile("Accept", CGI_AcceptTypes(n).key)
i = j + 1                        ' Bump pointer
n = n + 1                        ' Bump array index
Loop
CGI_NumAcceptTypes = n           ' Fill in global count
End Sub

```

```

-----
'
' GetArgs() - Parse the command line
'
' Chop up the command line, fill in the argument vector, return the
' argument count (similar to the Unix/C argc/argv handling)
-----

```

```

Private Function GetArgs(argv() As String) As Integer
Dim buf As String
Dim i As Integer, j As Integer, l As Integer, n As Integer

buf = Trim$(Command$)           ' Get command line

l = Len(buf)                     ' Length of command line
If l = 0 Then                    ' If empty
    GetArgs = 0                  ' Return argc = 0
    Exit Function
End If

i = 1                            ' Start at 1st character
n = 0                            ' Index in argvec
Do While ((i < l) And (n < MAX_CMDARGS)) ' Safety stop here
    j = InStr(i, buf, " ")        ' J -> next space
    If j = 0 Then Exit Do        ' Exit loop on last arg
    argv(n) = Trim$(Mid$(buf, i, j - i)) ' Get this token, trim it
    i = j + 1                    ' Skip that blank
    Do While Mid$(buf, i, 1) = " " ' Skip any additional whitespace
        i = i + 1
    Loop
    n = n + 1                    ' Bump array index
Loop

argv(n) = Trim$(Mid$(buf, i, (l - i + 1))) ' Get last arg
GetArgs = n + 1                  ' Return arg count

End Function

```

```

-----
'
' GetExtraHeaders() - Create the array of extra header structs
'

```

' Enumerate the keys in the [Extra Headers] section of the profile file,
' then get the value for each of the keys.

```
-----
Private Sub GetExtraHeaders()
    Dim sList As String
    Dim i As Integer, j As Integer, l As Integer, n As Integer

    sList = GetProfile("Extra Headers", "") ' Get key list
    l = Len(sList) ' Length incl. trailing null
    i = 1 ' Start at 1st character
    n = 0 ' Index in array
    Do While ((i < l) And (n < MAX_XHDR)) ' Safety stop here
        j = InStr(i, sList, Chr$(0)) ' J -> next null
        CGI_ExtraHeaders(n).key = Mid$(sList, i, j - i) ' Get Key, then value
        CGI_ExtraHeaders(n).value = GetProfile("Extra Headers",
        CGI_ExtraHeaders(n).key)
        i = j + 1 ' Bump pointer
        n = n + 1 ' Bump array index
    Loop
    CGI_NumExtraHeaders = n ' Fill in global count

```

End Sub

```
-----
'
' GetFormTuples() - Create the array of POST form input key=value pairs
'
-----

```

```
Private Sub GetFormTuples()
    Dim sList As String
    Dim i As Integer, j As Integer, k As Integer
    Dim l As Integer, m As Integer, n As Integer
    Dim s As Long
    Dim buf As String
    Dim extName As String
    Dim extFile As Integer
    Dim extlen As Long

    n = 0 ' Index in array
    ReDim Preserve CGI_FormTuples(n) As Tuple ' Increase array size
    '
    ' Do the easy one first: [Form Literal]
    '
    sList = GetProfile("Form Literal", "") ' Get key list
    l = Len(sList) ' Length incl. trailing null
    i = 1 ' Start at 1st character
    Do While i < l ' Safety stop here
        j = InStr(i, sList, Chr$(0)) ' J -> next null
        CGI_FormTuples(n).key = Mid$(sList, i, j - i) ' Get Key, then value
        CGI_FormTuples(n).value = GetProfile("Form Literal", CGI_FormTuples(n).key)
    
```

```

i = j + 1           ' Bump pointer
n = n + 1         ' Bump array index
ReDim Preserve CGI_FormTuples(n) As Tuple ' Increase array size
Loop
'
' Now do the external ones: [Form External]
'
sList = GetProfile("Form External", "") ' Get key list
l = Len(sList)           ' Length incl. trailing null
i = 1                   ' Start at 1st character
extFile = FreeFile
Do While i < l         ' Safety stop here
    j = InStr(i, sList, Chr$(0)) ' J -> next null
    CGI_FormTuples(n).key = Mid$(sList, i, j - i) ' Get Key, then pathname
    buf = GetProfile("Form External", CGI_FormTuples(n).key)
    k = InStr(buf, " ") ' Split file & length
    extName = Mid$(buf, 1, k - 1) ' Pathname
    k = k + 1
    extlen = CLng(Mid$(buf, k, Len(buf) - k + 1)) ' Length
    '
    ' Use feature of GET to read content in one call
    '
    Open extName For Binary Access Read As #extFile
    CGI_FormTuples(n).value = String$(extlen, " ") ' Breathe in...
    Get #extFile, , CGI_FormTuples(n).value 'GULP!
    Close #extFile
    i = j + 1           ' Bump pointer
    n = n + 1         ' Bump array index
    ReDim Preserve CGI_FormTuples(n) As Tuple ' Increase array size
Loop

CGI_NumFormTuples = n           ' Number of fields decoded
n = 0                          ' Reset counter
'
' Next, the [Form Huge] section. Will this ever get executed?
'
sList = GetProfile("Form Huge", "") ' Get key list
l = Len(sList)           ' Length incl. trailing null
i = 1                   ' Start at 1st character
Do While i < l         ' Safety stop here
    j = InStr(i, sList, Chr$(0)) ' J -> next null
    CGI_HugeTuples(n).key = Mid$(sList, i, j - i) ' Get Key
    buf = GetProfile("Form Huge", CGI_HugeTuples(n).key) ' "offset length"
    k = InStr(buf, " ") ' Delimiter
    CGI_HugeTuples(n).offset = CLng(Mid$(buf, 1, (k - 1)))
    CGI_HugeTuples(n).length = CLng(Mid$(buf, k, (Len(buf) - k + 1)))
    i = j + 1           ' Bump pointer
    n = n + 1         ' Bump array index
    ReDim Preserve CGI_FormTuples(n) As Tuple ' Increase array size
Loop

```

```

CGI_NumHugeTuples = n          ' Fill in global count

n = 0                          ' Reset counter
'
' Finally, the [Form File] section.

sList = GetProfile("Form File", "") ' Get key list
l = Len(sList)                  ' Length incl. trailing null
i = 1                          ' Start at 1st character
Do While ((i < l) And (n < MAX_FILE_TUPLES)) ' Safety stop here
    j = InStr(i, sList, Chr$(0)) ' J -> next null
    CGI_FileTuples(n).key = Mid$(sList, i, j - i) ' Get Key
    buf = GetProfile("Form File", CGI_FileTuples(n).key)
    ParseFileValue buf, CGI_FileTuples(n) ' Complicated, use Sub
    i = j + 1                    ' Bump pointer
    n = n + 1                    ' Bump array index
Loop

CGI_NumFileTuples = n          ' Fill in global count

End Sub

'-----
'
' GetProfile() - Get a value or enumerate keys in CGI_Profile file
'
' Get a value given the section and key, or enumerate keys given the
' section name and "" for the key. If enumerating, the list of keys for
' the given section is returned as a null-separated string, with a
' double null at the end.
'
' VB handles this with flair! I couldn't believe my eyes when I tried this.
'-----
Private Function GetProfile(sSection As String, sKey As String) As String
    Dim retLen As Long
    Dim buf As String * ENUM_BUF_SIZE

    If sKey <> "" Then
        retLen = GetPrivateProfileString(sSection, sKey, "", buf, ENUM_BUF_SIZE,
CGI_ProfileFile)
    Else
        retLen = GetPrivateProfileString(sSection, 0&, "", buf, ENUM_BUF_SIZE,
CGI_ProfileFile)
    End If
    If retLen = 0 Then
        GetProfile = ""
    Else
        GetProfile = Left$(buf, retLen)
    End If
End Function

End Function

```

```

-----
'
' Get the value of a "small" form field given the key
'
' Signals an error if field does not exist
'
-----
Function GetSmallField(key As String) As String
    Dim i As Integer

    For i = 0 To (CGI_NumFormTuples - 1)
        If CGI_FormTuples(i).key = key Then
            GetSmallField = Trim$(CGI_FormTuples(i).value)
            Exit Function      '** DONE **
        End If
    Next i

    ' Field does not exist
    '
    Error ERR_NO_FIELD
End Function

-----
'
' InitializeCGI() - Fill in all of the CGI variables, etc.
'
' Read the profile file name from the command line, then fill in
' the CGI globals, the Accept type list and the Extra headers list.
' Then open the input and output files.
'
' Returns True if OK, False if some sort of error. See ReturnError()
' for info on how errors are handled.
'
' NOTE: Assumes that the CGI error handler has been armed with On Error
'
-----
Sub InitializeCGI()
    Dim sect As String
    Dim argc As Integer
    Static argv(MAX_CMDARGS) As String
    Dim buf As String

    CGI_DebugMode = True      ' Initialization errors are very bad
    '
    ' Parse the command line. We need the profile file name (duh!)
    ' and the output file name NOW, so we can return any errors we
    ' trap. The error handler writes to the output file.
    '
    argc = GetArgs(argv())
    CGI_ProfileFile = argv(0)

```



```

sect = "CGI"
CGI_ServerSoftware = GetProfile(sect, "Server Software")
CGI_ServerName = GetProfile(sect, "Server Name")
CGI_RequestProtocol = GetProfile(sect, "Request Protocol")
CGI_ServerAdmin = GetProfile(sect, "Server Admin")
CGI_Version = GetProfile(sect, "CGI Version")
CGI_RequestMethod = GetProfile(sect, "Request Method")
buf = GetProfile(sect, "Request Keep-Alive") ' Y or N
If (Left$(buf, 1) = "Y") Then ' Must start with Y
    CGI_RequestKeepAlive = True
Else
    CGI_RequestKeepAlive = False
End If
CGI_LogicalPath = GetProfile(sect, "Logical Path")
CGI_PhysicalPath = GetProfile(sect, "Physical Path")
CGI_ExecutablePath = GetProfile(sect, "Executable Path")
CGI_QueryString = GetProfile(sect, "Query String")
CGI_RemoteHost = GetProfile(sect, "Remote Host")
CGI_RemoteAddr = GetProfile(sect, "Remote Address")
CGI_RequestRange = GetProfile(sect, "Request Range")
CGI_Referer = GetProfile(sect, "Referer")
CGI_From = GetProfile(sect, "From")
CGI_UserAgent = GetProfile(sect, "User Agent")
CGI_AuthUser = GetProfile(sect, "Authenticated Username")
CGI_AuthPass = GetProfile(sect, "Authenticated Password")
CGI_AuthRealm = GetProfile(sect, "Authentication Realm")
CGI_AuthType = GetProfile(sect, "Authentication Method")
CGI_ContentType = GetProfile(sect, "Content Type")
buf = GetProfile(sect, "Content Length")
If buf = "" Then
    CGI_ContentLength = 0
Else
    CGI_ContentLength = CLng(buf)
End If
buf = GetProfile(sect, "Server Port")
If buf = "" Then
    CGI_ServerPort = -1
Else
    CGI_ServerPort = CInt(buf)
End If

sect = "System"
CGI_ContentFile = GetProfile(sect, "Content File")
CGI_OutputFile = GetProfile(sect, "Output File")
CGI_OutputFN = FreeFile
Open CGI_OutputFile For Output Access Write As #CGI_OutputFN
buf = GetProfile(sect, "GMT Offset")
If buf <> "" Then ' Protect against errors
    CGI_GMTOffset = CVDate(Val(buf) / 86400#) ' Timeserial GMT offset
Else

```

```

    CGI_GMTOffset = 0
End If
buf = GetProfile(sect, "Debug Mode") ' Y or N
If (Left$(buf, 1) = "Y") Then      ' Must start with Y
    CGI_DebugMode = True
Else
    CGI_DebugMode = False
End If

GetAcceptTypes      ' Enumerate Accept: types into tuples
GetExtraHeaders    ' Enumerate extra headers into tuples
GetFormTuples      ' Decode any POST form input into tuples

End Sub

'-----
'
' main() - CGI script back-end main procedure
'
' This is the main() for the VB back end. Note carefully how the error
' handling is set up, and how program cleanup is done. If no command
' line args are present, call Inter_Main() and exit.
'-----
Sub Main()
    On Error GoTo ErrorHandler

    If Trim$(Command$) = "" Then ' Interactive start
        Inter_Main              ' Call interactive main
        End                      ' Exit the program
    End If

    InitializeCGI               ' Create the CGI environment

    '=====
    CGI_Main                    ' Execute the actual "script"
    '=====

Cleanup:
    Close #CGI_OutputFN

    End                          ' End the program
'-----
ErrorHandler:
    Select Case Err              ' Decode our "user defined" errors
        Case ERR_NO_FIELD:
            ErrorString = "Unknown form field"
        Case Else:
            ErrorString = Error$ ' Must be VB error
    End Select

    ErrorString = ErrorString & " (error #" & Err & ")"

```

```

On Error GoTo 0          ' Prevent recursion
ErrorHandler (Err)     ' Generate HTTP error result
Resume Cleanup
-----
End Sub

-----
'
' Send() - Shortcut for writing to output file
-----
Sub Send(s As String)
    Print #CGI_OutputFN, s
End Sub

-----
'
' SendNoOp() - Tell browser to do nothing.
'
' Most browsers will do nothing. Netscape 1.0N leaves hourglass
' cursor until the mouse is waved around. Enhanced Mosaic 2.0
' oputs up an alert saying "URL leads nowhere". Your results may
' vary...
-----
Sub SendNoOp()

    Send ("HTTP/1.0 204 No Response")
    Send ("Server: " + CGI_ServerSoftware)
    Send ("")

End Sub

-----
'
' WebDate - Return an HTTP/1.0 compliant date/time string
'
' Inputs:  t = Local time as VB Variant (e.g., returned by Now())
' Returns: Properly formatted HTTP/1.0 date/time in GMT
-----
Function WebDate(dt As Variant) As String
    Dim t As Variant

    t = CDate(dt - CGI_GMTOffset)    ' Convert time to GMT
    WebDate = Format$(t, "ddd dd mmm yyyy hh:mm:ss") & " GMT"

End Function

```

```
' PlusToSpace() - Remove plus-delimiters from HTTP-encoded string
```

```
Public Sub PlusToSpace(s As String)
    Dim i As Integer
```

```
    i = 1
    Do While True
        i = InStr(i, s, "+")
        If i = 0 Then Exit Do
        Mid$(s, i) = " "
    Loop
```

```
End Sub
```

```
' Unescape() - Convert HTTP-escaped string to normal form
```

```
Public Function Unescape(s As String)
    Dim i As Integer, l As Integer
    Dim c As String
```

```
    If InStr(s, "%") = 0 Then          ' Catch simple case
        Unescape = s
        Exit Function
    End If
```

```
    l = Len(s)
    Unescape = ""
    For i = 1 To l
        c = Mid$(s, i, 1)              ' Next character
        If c = "%" Then
            If Mid$(s, i + 1, 1) = "%" Then
                c = "%"
                i = i + 1                ' Loop increments too
            Else
                c = x2c(Mid$(s, i + 1, 2))
                i = i + 2                ' Loop increments too
            End If
        End If
        Unescape = Unescape & c
    Next i
```

```
End Function
```

```
' x2c() - Convert hex-escaped character to ASCII
```

```
-----
Private Function x2c(s As String) As String
    Dim t As String
```

```
    t = "&H" & s
    x2c = Chr$(CInt(t))
```

```
End Function
```

```
Private Sub ParseFileValue(buf As String, ByRef t As FileTuple)
    Dim i, j, k, l As Integer
```

```
    l = Len(buf)
```

```
    i = InStr(buf, " ")          ' First delimiter
    t.file = Mid$(buf, 1, (i - 1)) ' [file]
    t.file = Mid$(t.file, 2, Len(t.file) - 2) ' file
```

```
    j = InStr((i + 1), buf, " ") ' Next delimiter
    t.length = CLng(Mid$(buf, (i + 1), (j - i - 1)))
    i = j
```

```
    j = InStr((i + 1), buf, " ") ' Next delimiter
    t.type = Mid$(buf, (i + 1), (j - i - 1))
    i = j
```

```
    j = InStr((i + 1), buf, " ") ' Next delimiter
    t.encoding = Mid$(buf, (i + 1), (j - i - 1))
    i = j
```

```
    t.name = Mid$(buf, (i + 1), (l - i - 1)) ' [name]
    t.name = Mid$(t.name, 2, Len(t.name) - 1) ' name
```

```
End Sub
```

```
' FindExtraHeader() - Get the text from an "extra" header
```

```
' Given the extra header's name, return the stuff after the ":"  
' or an empty string if not there.
```

```
-----
Public Function FindExtraHeader(key As String) As String
    Dim i As Integer
```

```
    For i = 0 To (CGI_NumExtraHeaders - 1)
        If CGI_ExtraHeaders(i).key = key Then
            FindExtraHeader = Trim$(CGI_ExtraHeaders(i).value)
```

```

        Exit Function      ' ** DONE **
    End If
Next i
'
' Not present, return empty string
'
    FindExtraHeader = ""
End Function
*****

```

GLOSARIO

Active X

Lenguaje de programación apoyado en controles OLE, Visual Basic y Librerías del entorno Windows (OCX) de Microsoft (<http://www.microsoft.com>), Active X permite que interactuen aplicaciones Windows con el World Wide Web. Actualmente solo es soportado por el Internet Explorer, aunque existen planes para integrarlo a plataformas Macintosh y UNIX.

API (Application Program Interface)

Conjunto de reglas de programación que determinan como una aplicación debe acceder a un servicio.

Archie

Un sistema para la localización de archivos que están disponibles públicamente por FTP anónimo. Es necesario conocer el nombre del archivo o una subcadena del mismo para utilizar archie.

ARPANET (Advanced Research Projects Agency Network)

Red experimental con fines militares establecida en los setenta, en la cual se probaron las teorías y el software en los que esta basado Internet. ARPANET era una red experimental que apoyaba la investigación militar, en particular la investigación sobre cómo construir redes que pudieran soportar fallas parciales (como las producidas por los bombardeos) y aún así funcionar. La red fue diseñada para requerir un mínimo de información de las computadoras que forman parte de ella. La filosofía era que cada computadora en la red se pudiese comunicar, como un elemento particular con cualquier computadora.

ASCII (American Standard Code for Information Interchange)

Es el estándar del World Wide Web para el código utilizado por computadoras para representar todas las letras (mayúsculas, minúsculas, letras latinas, números, signos de puntuación, etc.). El código estándar ASCII es de 128 letras representadas por un dígito binario de 7 posiciones (7 bits), de 0000000 a 1111111.

BBS (Bulletin Board System)

Servicio que proporcionan desde grandes compañías (Compuserve, America On Line, etc.) hasta pequeños proveedores, que consiste en intercambiar información con otros usuarios, descargar archivos etc., sin estar conectados a Internet por lo que actualmente están cayendo en desuso. Las BBS se cuentan por miles en el mundo, corriendo desde una simple PC con una o dos líneas telefónicas. La

tendencia actual es que las BBS se vayan convirtiendo en proveedores de servicio de Internet.

BITNET (Because It's Time Network ó Because It's There Network)

Red de sitios educativos (investigación y universitarios) separada de Internet, pero el correo electrónico es libremente intercambiado entre BITNET e Internet. Los Listserves son la forma más popular de los grupos de noticias originados en BITNET. Las computadoras de BITNET son usualmente mainframes corriendo el sistema operativo VMS (variante de UNIX).

Browser

Vea Visualizador.

CERN

Laboratorio Europeo de Física de Partículas. Fue el desarrollador inicial del World Wide Web. Actualmente los estándares del Web son desarrollados por la World Wide Web Organization (3W0). El web site del CERN se encuentra en <http://www.cern.ch>

Chat

Termino utilizado para describir la comunicación de usuarios en tiempo real. Vea IRC

CGI (Common Gateway Interface)

Una interfaz escrita en un lenguaje de programación (perl, c, C++, Visual Basic, etc.) y posteriormente ejecutada o interpretada por una computadora servidor para contestar a pedidos del usuario desde una computadora con una aplicación cliente; casi siempre desde el World Wide Web. Esta interfaz permite obtener los resultados pedidos, como los que resultan al consultar una base de datos.

Cliente

- a) Una aplicación que permite a un usuario obtener un servicio de un servidor localizado en la red.
- b) Un sistema o proceso que solicita a otro sistema o proceso que le preste un servicio.

Cookie

Procedimiento ejecutado por el servidor que consiste en guardar información acerca del cliente para su posterior recuperación (proceso realizado por el Internet Explorer cuando utiliza Microsoft Network (<http://www.msn.com>)). En la práctica la información es proporcionada desde el visualizador al servidor del World Wide Web vía una forma o un método interactivo que puede ser recuperado nuevamente cuando se accede al servidor en el futuro. Es utilizado por ejemplo para el registro a un servicio.

Correo Electrónico (e-mail)

Permite el intercambio de mensajes entre personas conectadas a una red de manera similar al correo tradicional. Entre las aplicaciones cliente de correo electrónico tenemos a Eudora, Mail, Pine, Pegasus, etc. La definición acerca del correo electrónico fue especificada en el RFC#822. Para más información consulte <http://www.internic.net/rfc/rfc822.txt>

Dirección electrónica (address)

Dirección de un usuario en Internet. Por medio de ella es posible enviar correo electrónico a un usuario. Esta es única para cada usuario y se compone por el login de un usuario, arroba y el nombre del servidor de correo electrónico. Ej. usuario@computadora.com.

Dirección IP

La dirección del protocolo de Internet (IP) es la dirección numérica de una computadora en Internet. Cada dirección electrónica se asigna a una computadora conectada a Internet y por lo tanto es única. La dirección IP esta compuesta de cuatro octetos como 132.248.53.10

DNS

Sistema de nomenclatura de dominios (Domain Name System) Es un sistema que se establece en un servidor (que se encarga de un dominio) que traduce nombres de computadoras (como servidor.unam.mx) a domicilios numéricos de Internet (direcciones IP) (como 132.248.10.1).

Dominio

Conjunto de computadoras que comparten una característica común, como el estar en el mismo país, en la misma organización o en el mismo departamento. Cada dominio es administrado por un servidor de dominios.

Los dominios se establecen de acuerdo al uso que se le da a la computadora y al lugar donde se encuentre. Ejemplo:

- .com Comercial
- .edu educación (USA),
- .mx México
- .es España, etc.

E-mail

Vea Correo Electrónico.

Eudora

Aplicación cliente de correo electrónico. Existen dos versiones la comercial y la versión Freeware.

FAQ

Preguntas más frecuentes (Frequently Asked Questions) Se refiere a una pregunta o preguntas más frecuentes y a sus respuestas.

Firewall

Una combinación de hardware y software que separa una red de área local (LAN) de Internet. La regla básica de un Firewall es asegurar que todas las comunicaciones entre la red e Internet se realicen conforme a las políticas de seguridad de la organización o corporación.

Freeware

Aplicaciones que pueden obtenerse directamente de Internet y que no es necesario pagar por su utilización.

FTP

- a) Protocolo de transferencia de archivos (File transfer Protocol).
- b) Aplicación que desplaza archivos utilizando el Protocolo de transferencia de archivos

FTP anónimo

Procedimiento que se utiliza para descargar archivos públicos de una computadora remota a un local. Es a veces necesario introducir un password que puede ser la palabra guest (huésped), o nuestra dirección electrónica.

Gateway (Puente)

Sistema de información que transfiere información entre sistemas o redes incompatibles.

GIF (Graphics Interchange Format)

Formato binario de archivos que contienen imágenes. Este formato es utilizado por su alta capacidad de compresión de la información de una imagen. Fue desarrollado en 1987 (GIF87) por Compuserve (<http://www.compuserve.com>) como solución para compartir imágenes a través de plataformas. Posteriormente fue revisado en 1989 generándose la nueva versión GIF89a. Otro formato binario es el JPG.

GNU

Fundación para el Software Libre (FSF - Free Software Foundation) Busca eliminar las restricciones de uso, copia, modificación y distribución del software. Actualmente se encuentra apoyando el desarrollo de sistemas operativos (Linux), compiladores (compilador GNU C Compiler (gcc), Perl), etc. Trata de promover, desarrollar y usar del software libre en todas las áreas de la computación. Específicamente, la Fundación pone a disposición de todo el mundo un completo e integrado sistema de software llamado GNU. La mayor parte de este sistema está ya siendo utilizado y distribuido. El costo del software únicamente esta determinado por el costo del material utilizado para distribuirlo.

Gopher

Un servicio basado en menús que sirve para explorar recursos de Internet.

Gopher es un sistema de entrega de información distribuida. Utilizando gopher podemos acceder a información local o bien a acceder a servidores de información gopher de todo el mundo. Gopher combina las características de BBS (Bulletin Board Service) y bases de datos, permitiendo establecer una jerarquía de documentos, y permitiendo búsquedas en ellos por palabras o frases clave. Concebido y desarrollado en la Universidad de Minnesota en el año 91 es de libre distribución para fines no comerciales.

Gopher soporta directorios, ficheros de texto, ítem de búsqueda, sesiones telnet y tn3270, multimedia y texto formateado (postscript y otros).

Grupos de noticias

Area de mensajes automatizada, operada normalmente a través de USENET, en la que los suscriptores dejan mensajes a todo el grupo sobre temas específicos.

Herramientas de búsqueda

Programas que permiten a los usuarios definir criterios o palabras relacionadas con una información requerida, siendo otras computadoras de la red las que efectúan la búsqueda indicando los sitios donde se encuentran los datos.

Hipermedia

Combinación de texto y multimedia. Actualmente es un recurso ampliamente explotado en el World Wide Web. El término hipermedia incluye el concepto de hipertexto pero superando la restricción que la información sea solamente textual, ofreciendo por lo tanto, la posibilidad de incluir información multimedia.

Hipertexto

Documentos que contienen vínculos con otros documentos, al seleccionar un vinculo automáticamente se despliega el segundo documento.

Es la consulta de un documento de texto pero no obligatoriamente de forma secuencial.

Homepage (Página inicial)

Es la pagina web de entrada a un lugar del World Wide Web. Es considerada la página principal.

Host (Anfitrión)

Computadora a la que tenemos acceso de diversas formas (telnet, FTP, World Wide Web, etc.). Es el servidor que nos provee de la información que requerimos para realizar algún procedimiento desde una aplicación cliente.

HTML Lenguaje de marcado de hipertexto, (HyperText Markup Lenguaje)

Es el lenguaje con que se escriben los documentos en el World Wide Web. A la fecha existen tres versiones de HTML.

HTML 1, donde se sientan las bases para la disposición del texto y las gráficas, HTML 2 donde se agregan formas y HTML 3 (llamado también extensiones Netscape) donde se añaden tablas, mapas, etc.

HTTP Protocolo de Transferencia de Hipertextos (Hyper-Text Transfer Protocol)

Es el protocolo en que se basa la tecnología de World Wide Web. HTTP es el conjunto de reglas que gobiernan el software que transporta los documentos HTML a través de Internet.

Internet

Es una red de cómputo a nivel mundial que agrupa a distintos tipos de redes usando un mismo protocolo de comunicación. Los usuarios en Internet pueden compartir datos, recursos y servicios. Internet se apoya en el conjunto de protocolos TCP/IP. De forma más específica, Internet es la WAN más grande que hay en el planeta, e incluye decenas de MAN's y miles de LAN's. Las computadoras que lo integran van desde modestos equipos personales, minicomputadoras, estaciones de trabajo, mainframes hasta supercomputadoras. Internet no tiene una autoridad central, es descentralizada. Cada red mantiene su independencia y se une cooperativamente al resto respetando una serie de normas de interconexión. El organismo que se encarga de regular, establecer estándares, administrar y hacer operacional a Internet es la ISOC (Internet Society).

Internet address (dirección internet)

Vea dirección IP.

Internet Explorer

Programa visualizador del World Wide Web desarrollado por Microsoft. Disponible gratuitamente desde <http://www.microsoft.com/ie>. La versión 3 de este programa soporta Java y controles Active X.

InterNIC

Es el nombre que se le da al conjunto de proveedores de servicios de registro. El InterNIC define los nombres de dominio a nivel mundial. El sitio de la Internic (<http://www.internic.net>) es mantenido además por la National Science Fundation (NSF <http://www.nsf.gov>) y la compañía de telecomunicaciones ATT (<http://www.att.com>).

Intranet

Una red privada dentro de una compañía u organización que utiliza el mismo software que se encuentra en Internet, pero que es solo para uso interno. Por ejemplo, muchas compañías tienen servidores World Wide Web disponibles solo para sus empleados.

IP (Protocolo Internet)

Permite a un paquete de datos viajar a través de múltiples redes hasta alcanzar su destino. Se encarga de la capa de red del modelo OSI

IRC

- a) Programa basado en el modelo cliente servidor que permite conversar con múltiples usuarios en red sobre un tema común.
- b) Protocolo mundial para conversaciones simultáneas que permite comunicarse por escrito entre sí a través de ordenador a varias personas en tiempo real. El servicio IRC está estructurado mediante una red de servidores, cada uno de los cuales aceptan conexiones de programas cliente, uno por cada usuario.

Java

Un lenguaje de programación que permite ejecutar programas escritos en un lenguaje muy parecido al C++, llamados applets, a través del World Wide Web. La diferencia contra un CGI es que la ejecución se realiza totalmente en la computadora cliente, en lugar del servidor. Java fue originalmente desarrollado por Sun Microsystems (<http://www.sun.com>). El principal objetivo de JAVA fue hacer un lenguaje que fuera capaz de ser ejecutado de una forma segura a través de Internet. Esta característica requiere la

Internet address (dirección internet)

Vea dirección IP.

Internet Explorer

Programa visualizador del World Wide Web desarrollado por Microsoft. Disponible gratuitamente desde <http://www.microsoft.com/ie>. La versión 3 de este programa soporta Java y controles Active X.

InterNIC

Es el nombre que se le da al conjunto de proveedores de servicios de registro. El InterNIC define los nombres de dominio a nivel mundial. El sitio de la Internic (<http://www.internic.net>) es mantenido además por la Nacional Science Fundation (NSF <http://www.nsf.gov>) y la compañía de telecomunicaciones ATT (<http://www.att.com>).

Intranet

Una red privada dentro de una compañía u organización que utiliza el mismo software que se encuentra en Internet, pero que es solo para uso interno. Por ejemplo, muchas compañías tienen servidores World Wide Web disponibles solo para sus empleados.

IP (Protocolo Internet)

Permite a un paquete de datos viajar a través de múltiples redes hasta alcanzar su destino. Se encarga de la capa de red del modelo OSI

IRC

- a) Programa basado en el modelo cliente servidor que permite conversar con múltiples usuarios en red sobre un tema común.
- b) Protocolo mundial para conversaciones simultáneas que permite comunicarse por escrito entre sí a través de ordenador a varias personas en tiempo real. El servicio IRC está estructurado mediante una red de servidores, cada uno de los cuales aceptan conexiones de programas cliente, uno por cada usuario.

Java

Un lenguaje de programación que permite ejecutar programas escritos en un lenguaje muy parecido al C++, llamados applets, a través del World Wide Web. La diferencia contra un CGI es que la ejecución se realiza totalmente en la computadora cliente, en lugar del servidor. Java fue originalmente desarrollado por Sun Microsystems (<http://www.sun.com>). El principal objetivo de JAVA fue hacer un lenguaje que fuera capaz de ser ejecutado de una forma segura a través de Internet. Esta característica requiere la

eliminación de muchas construcciones y usos de C y C++. El más importante, es que no existen punteros. Java no puede acceder arbitrariamente a direcciones de memoria. Java es un lenguaje compilado en un código llamado "código-byte" (byte-code). Este código es interpretado "en vuelo" por el interprete Java.

JPG o JPEG (Joint Photographic Experts Group)

Un formato para guardar imágenes que las hace ocupar poco espacio en la memoria de la computadora y en disco. Por esta razón son más rápidas de transmitir a través del web. A diferencia del formato GIF, este formato no es aceptado por todos los visualizadores del World Wide Web.

LAN Red de área local (local area network)

Red cuyas dimensiones no exceden 10 Km. Puede tratarse de computadoras conectadas en una oficina, en un edificio o en varios.

Link

Vea Vínculo.

Listas de correo o listas de discusión o foros de discusión

Servicio automatizado de mensajes, a menudo moderado por un propietario en el que los suscriptores reciben mensajes dejados por otros suscriptores por un tema dado. Los mensajes se envían por correo electrónico.

MAN Red de área metropolitana (Metropolitan area Network)

Red que no va más allá de los 100 Km. Equipos de computo y sus periféricos conectados en una ciudad o en varias forman una MAN

MIME Extensiones de Correo de Internet de Múltiples propósitos (Multipurpose Internet Mail Extensions)

Técnica para codificar archivos y anexarlos a un mensaje de correo electrónico. Permite principalmente enviar archivos binarios como parte de un mensaje.

Modelo Cliente-Servidor

El modelo cliente-servidor se apoya en terminales (clientes) conectadas a una computadora que los provee de un recurso (servidor). De esta manera los clientes son los elementos que necesitan servicios del recurso y el servidor es la entidad que poseen el recurso. Los clientes sin embargo no dependen totalmente del servidor. Ellos pueden realizar los procesamientos para desplegar la información (por ejemplo en forma gráfica). El servidor los provee únicamente de la información sin hacerse cargo de otros procesos. El tráfico en la red de esta forma se ve aligerado y las comunicaciones entre las computadoras se realizan más rápido.

Módem

Equipo utilizado para adecuar las señales digitales de una computadora a una línea telefónica o a una red digital de servicios integrados (ISDN), mediante un proceso denominado de modulación (para transmitir información) y demodulación (para recibir información), de ahí su nombre. La velocidad máxima que puede alcanzar un módem para línea telefónica es de 33 kBps. Un módem debe cumplir con los estándares de MNP5 y V42.bis para considerar su adquisición.

Motor de búsqueda

Una aplicación de software o un servicio que se utiliza para buscar archivos en una intranet o en Web. Entre los motores de búsqueda más comunes se incluyen algunos como Excite, Yahoo!, WebCrawler, Infoseek y Lycos, pero se crean nuevos motores de búsqueda constantemente.

Mosaic

Visualizador para el World Wide Web. Fue el primer visualizador para los ambientes Macintosh, UNIX y Windows, desarrollado por la NCSA (<http://www.ncsa.uiuc.edu/>).

MPEG (Moving Pictures Expert Group)

MPEG es un estándar de compresión de vídeo. Se adhiere a los visualizadores por medio de un Plug-in. La versión 3 del Internet Explorer lo trae ya integrado.

MUA Agente usuario de correo electrónico (Mail User Agent)

Son todo aquellos programas que permiten la edición, lectura y respuesta de correo electrónico.

Multimedia

Término que se utiliza para cualquier contenido que combine texto, sonido, gráficos y vídeo.

NCSA Centro Nacional de Aplicaciones de Supercómputo (National Center for Supercomputing Applications)

Desarrolladores del visualizador Mosaic para el World Wide Web. Localizado en <http://www.ncsa.uiuc.edu/>

Netscape Navigator

Visualizador para el World Wide Web. Disponible desde <http://www.netscape.com>, en las siguientes plataformas X-Windows (UNIX), Macintosh y Windows.

NIC Centro de Información de red (Network Information Center)

Organización responsable de proporcionar información de una red a los usuarios.

Octeto (octect)

Termino para referirse a los ocho bits que conforman un byte.

OSI Interconexión de Sistemas Abiertos (Open Systems Interconnect)

Es el protocolo en el que se apoya Internet. Establece la manera como se realiza la comunicación entre dos computadoras a través de siete capas: Física, Datos, Red, Transporte, Sesión, Presentación y Aplicación.

Página web

Es el resultado en hipertexto e hipermedia que proporciona un visualizador de World Wide Web después de obtener la información solicitada.

Password

Palabra clave que se le asigna a un usuario -además de su login- como contraseña para la utilización de los recursos de una computadora. El password no es visible en la pantalla al momento de teclearlo.

Perl

Lenguaje de programación utilizado en el World Wide Web a través de un CGI, principalmente para realizar consultas a bases de datos como Oracle, SQL-Server, SyBase, etc., o a herramientas locales como WAIS. Perl es un lenguaje para manipular textos, archivos y procesos, proporciona una forma fácil y legible para realizar trabajos que normalmente se realizarían en C o en un shell. Perl nació y se ha difundido bajo el sistema operativo UNIX, aunque existe para otras plataformas. Perl fue desarrollado por Larry Wall, y está distribuido libremente bajo la filosofía de GNU.

POP Protocolo de Oficina de Correos (Post Office Protocol)

Programa cliente que se comunica con el servidor, identifica la presencia de nuevos mensajes, solicita la entrada de los mismos y utiliza al servidor como oficina despachadora de correo electrónico cuando el usuario envía una carta. Los mensajes enviados a la aplicación cliente son inmediatamente eliminados del servidor, sin embargo las aplicaciones modernas pueden omitir este paso. Entre los programas que utilizan dicho protocolo se encuentra Eudora.

Protocolo

- a) Es la definición de como deben comunicarse dos computadoras, sus reglas de comportamiento, etc.
- b) Definición de reglas.

Proveedor de Servicios de Internet. (Internet Service Provider)

Organización que provee la conexión de computadoras a Internet, ya sea por líneas dedicadas o por líneas conmutadas.

Proxy

El proxy es un servidor que actúa como un cache. Los usuarios que se conecten al proxy, le pedirán las paginas web o los ficheros que deseen y el se encargara de traerlos de Internet. Además, el servidor de proxy recuerda las paginas, así que una pagina que ya haya sido pedida por cualquier persona, el nuevo usuario que la pida se la traerá de nuestro servidor, no de Internet. Con esto se mejora la velocidad y se evitan posibles colapsos en servidores remotos.

Puerto

- a) Es un numero que identifica a una aplicación particular de Internet.
- b) Uno de los canales de entrada/salida de una computadora

Red

Agrupación tanto de equipos como de programas que comparten recursos entre sí, observando "reglas de comportamiento" a partir del uso de un lenguaje y medios de transmisión comunes, sin importar -en lo esencial- la naturaleza de cada elemento dentro de la red.

RFC Solicitud para comentarios (Request for Comments)

Es un conjunto de documentos en los cuales los estándares de Internet, los estándares propuestos y, generalmente las ideas en proceso de aceptación son documentadas y publicadas. Los documentos pueden ser consultados en <http://www.internic.net/rfc>.

Robots

Los robots en el contexto del World Wide Web son programas que viajan en el Web, indexando páginas, localizando errores, etc. Estos programas son enviados y mantenidos por varias herramientas de búsqueda.

Server Side Include (SSI)

Un server side include, o SSI, es un pedazo de código que es insertado en una página HTML e interpretado por el servidor antes

de que la página sea enviada al browser. SSI permite de incluir información en los archivos HTML tal como la fecha de la última modificación del archivo, o inclusive otro archivo HTML.

Servidor

Computadora dedicada a gestionar el uso de la red por otras computadoras llamadas clientes. Contiene archivos y recursos que pueden ser accedidos desde otras computadoras (terminales).

Sesión remota

Uso de los recursos de una computadora desde una terminal que no precisamente se encuentra cercana a ella.

SGML (Standard Generalized Markup Language)

Estándar internacional que provee de un esquema de representación para crear información textual. HTML es un subconjunto de SGML.

Shareware

Programas que pueden ser obtenidos por Internet de computadoras con archivos públicos. La regla de utilización es que se paguen después de un periodo de evaluación (por lo regular 30 días).

Sitio

Una colección de páginas Web relacionadas, que residen en el mismo servidor y están conectadas entre sí mediante vínculos.

TCP

Protocolo de control de transmisión (Transfer Control Protocol). Es el protocolo que se encarga de la transferencia de los paquetes a través de Internet. Se encarga de que los paquetes lleguen al destino sin ningún error o pide su reenvío. Se encarga de la capa de transporte del modelo OSI

Telnet

Protocolo de emulación de terminal que permite establecer una sesión remota a otra computadora en Internet.

Topologías de anillo

Topología en donde las estaciones de trabajo se conectan físicamente en un anillo, terminando el cable en la misma estación de donde se originó.

Topología de bus

Topología en donde todas las estaciones se conectan a un cable central llamado "bus". Este tipo de topología es fácil de instalar y requiere menos cable que la topología de estrella.

Topología de estrella

Topología donde cada estación se conecta con su propio cable a un dispositivo de conexión central, bien sea un servidor de archivo o un concentrador o repetidor.

Topología de red

Se refiere a cómo se establece y se cablea físicamente una red. La elección de la topología afectará la facilidad de la instalación, el costo del cable y la confiabilidad de la red. Tres de las topologías principales de red son la topología de bus, de estrella, y de anillo.

UDP Protocolo de Datagramas de usuario (User Datagram Protocol)

Protocolo que no pide confirmación de la validez de los paquetes enviados por la computadora emisora. Este protocolo es actualmente usado para la transmisión de sonido y vídeo a través de Internet. El UDP esta diseñado para satisfacer necesidades concretas de ancho de banda, como no reenvía los datos perdidos, es ideal para el tráfico de voz digitalizada, pues un paquete perdido no afecta la calidad del sonido. Entre las aplicaciones que utilizan este protocolo encontramos a Real Audio.

URL Localizador Uniforme de recursos (Uniform Resorce Locator)

Sistema de direccionamiento estándar para archivos y funciones de Internet, especialmente en el Word Wide Web. El url esta conformado por el servicio (p. e. <http://>) más el nombre de la computadora (p. e. www.unam.mx) más el directorio y el archivo referido.

USENET (USEr NETWORK)

Sistema de redes que contiene artículos llamados NEWS que pueden ser consultados sin necesidad de estar inscrito.

Usuario

Un usuario es la persona que tiene una cuenta en una determinada computadora por medio de la cual puede acceder a los recursos y servicios que ofrece una red. Un usuario que reside en una determinada computadora tiene una dirección electrónica única.

Veronica Índice de Red Amplia muy fácil Orientado hacia Roedores para Archivos Computarizados (Very Easy Rodent-Oriented Net-Wide Index to Computerized Archives)

Herramienta de localización que permite realizar búsquedas basadas en palabras clave en directorios y dominios de Gopher, Herramientas de búsqueda.

Vínculo (link)

Es un indicador de texto o una imagen que sirve como enlace a otro documento.

Visual Basic

Lenguaje de programación de Microsoft orientado a eventos, utilizado principalmente en el World Wide Web para realizar consultas a bases de datos de Microsoft como Fox Pro, SQL-Server, etc., que funciona en servidores de Windows NT.

Visualizador (Browser)

Programa que despliega la información almacenada en páginas HTML que se encuentran disponibles en servidores del World Wide Web. Como ejemplo de visualizadores tenemos Cello, Internet Explorer, Mosaic, Netscape, Plugins, etc.

VRML (Virtual Reality Modeling Language antes Virtual Reality Markup Language)

Lenguaje de programación utilizado para hacer presentaciones de realidad virtual en el Word Wide Web. Puede ser un visualizador propio o integrado a los visualizadores WWW a través de un Plug-in. En agosto de 1995 se anuncio la especificación 2.0 como un nuevo estándar. VRML 1.0 permite crear mundo estático en 3-D, que contienen objetos que pueden girar libremente alrededor de su eje, pero sin ningún movimiento interactivo real. VRML 2.0 por su parte permite manipular los objetos, cuenta con sensores de proximidad, sonido etc.

WAIS. Servicio de Información de Area Amplia (Wide Area Information Service)

Es una herramienta que permite encontrar información almacenada en archivos o en bases de datos a través de Internet.

WAN. Red de área mundial (World Area Network)

Puede extenderse a todo un país o a muchos a través del mundo.

Web

Vea World Wide Web.

Website

Conjunto de páginas web que comparten un mismo tema e intención y que generalmente se encuentra en un sólo servidor, aunque esto no es forzoso.

Windows 95

Sistema operativo gráfico de 32 bits desarrollado por Microsoft (<http://www.microsoft.com>) y diseñado específicamente para computadoras con procesadores compatibles con Intel.

Windows NT

Sistema operativo gráfico de 32 bits desarrollado por Microsoft muy similar al Windows, pero con más prestaciones.

World Wide Web

Sistema basado en hipertextos cuya función es buscar y tener acceso a documentos a través de la red.

Write Pages

Listas de usuarios de Internet. Existen varios lugares donde los usuarios pueden registrarse y realizar búsquedas de personas.

WWW

Vea World Wide Web

Yahoo!

Herramienta pública de búsqueda del World Wide Web disponible en <http://www.yahoo.com>, la búsqueda se realiza a partir de los títulos de los documentos de HTML. Yahoo! Se distingue por tener un gran índice de documentos ordenados por categorías.

BIBLIOGRAFIA

Conéctate al Mundo de Internet Guía y Catalogo

Ed Krol (O'Reilly & Associates, Inc.)
Mc Graw Hill (A nutshell handbook)
Segunda Edición.

La Biblia de Internet

Que Development Group
Anaya Multimedia

Internet

Manual de Referencia
Harley Hahn
Serie McGraw-Hill de informática

Todo acerca de redes de computación

Kevin Stoltz
Prentice Hall Hispanoamericana

Word Wide Web Paso a Paso

Traducido del inglés de la obra: Using the World Wide Web.
Grupo QUE
Prentice Hall Hispanoamericana, S.A.

Special Edition. Using CGI

Jeffy Dwight, Michael Erwin
QUE

Revista Byte México

Número Diciembre 1995
Redes una nueva visión

Revista Soluciones Avanzadas

Número: 33, 35, 37
Dirección: <http://www.fciencias.unam.mx/revista/soluciones.html>

<http://www.microsoft.com/Spain/ie/guide/default2.htm>
Guía Internet

<http://pubweb.nexor.co.uk/public/rfc/index/rfc.html>

RFC Index Search Form

http://a01-unix.uc3m.es/~lewis/mi_practica.html
Laboratorio de Redes de Ordenadores 1995/96

<http://www2.ceniai.inf.cu/travel/page21.htm>
Internet

<http://www.netvillw.com.br/~dippold/palestra/www.htm#servidor>
INTERNET - WWW (Rede Gráfica)

<http://www.dic.uchile.cl/~manual/form/index.htm>

<http://www.mat.upna.es/tutorial/cgi/http.html>
Protocolo HTTP

<http://www.w3.org/Protocols/HTTP/HTTP2.html>
HTTP: A protocol for networked information

<http://www.w3.org/>
W3C - The World Wide Web Consortium

<http://www.wmaestro.com/webmaestro/>
WebMaestro - Crea tu pagina del Web

<http://www8.zdnet.com/products/content/pcmg/1515/pcmg0056.html>
Netscape Communications Corp.: Netscape Enterprise Server, Netscape FastTrack Server

<http://website.ora.com/>
WebSite Professional 2.0

<http://alek.pucp.edu.pe/~rieee/Coloquio/Grupo15.html>
Instalación del servidor HTTP en la Sección Electricidad y Electrónica

<http://www.hnet.es/programas/win31/servidor/servidor.htm>
Servidores

<http://tech.west.ora.com/win-httpd/>
Windows httpd

<http://www.apache.org>
Apache Organization

[http://www3.service.digital.com/apache/dist/
Index of /apache/dist](http://www3.service.digital.com/apache/dist/Index_of_apache/dist)

<http://webcompare.internet.com/resources.html>
Resources for Web site administrators

<http://serverwatch.internet.com/top10monthly.html>
ServerWatch Top 10 Web Servers

http://www.sicoar.com.ar/proxy_web/proxy.html
<http://vip.interplanet.es/faq2/general.html#¿Qué es un Proxy?>
<http://www.activ.es/csm/help/index.html>
Qué es un Proxy

<http://vip.interplanet.es/faq2/general.html#¿Qué son las cookies?>

<http://statline.vb.cbs.nl/erwin/ekcgi/ekcgi.html>
CGI with Delphi

<http://www.stars.com/Vlib/Providers/CGI.html>
The WDWL: VL-WWW: CGI

<http://highland.dit.upm.es:8000/CGI/cgi.html>

<http://web.jet.es/justino/portada.shtml>
Biblioteca de rutinas CGI y JavaScript

<http://www.ok.cl/cgi/>
CGI Master: Manual de CGI e ISAPI para programadores, especificaciones y tutorial en castellano (español)

<http://www.uaemex.mx/publica/informatica/infuni17/cgi.htm>

http://fciencias.ens.uabc.mx/~smorales/sistemas_operativos/bdwww.html
Acceso a Bases de Datos a través del World Wide Web

<http://www.ur.mx.division/chepe/homepage/tutorvb/vbhtml/home.htm>
Programación en Visual Basic 3.0 de Microsoft

http://guille.costasol.net/indice_cf.htm
el Guille -La Web del Visual Basic y Programación

<http://www.users.dircon.co.uk/~tbrown/vb/cgi.htm>
Visual Basic CGI Programming

<http://www.op.net/~jstrollo/vblinks.html>
Strollo Software

<http://www.umich.edu/~websvcs/umweb/wwwcgi.html>
web services . . . CGI Programs on www.umich.edu

<http://www.geocities.com/SiliconValley/6742/>
CGI scripts and other Web Utilities for Windows NT/95

<http://phonos.upf.es/~jordinas/esp/cgi.htm>
Common Gateway Interface

<http://www.primelink.com/joda/cgiexam.htm>
CGI Examples

<http://www.microsoft.com/spain/scripting/default.htm?/spain/scripting/vbscript/vbstutor/vbswhat.htm>
Tecnologías de Microsoft Scripting

<http://ciencias.ens.uabc.mx/~redes2/Internet/estandares/cgi1.html>