

Servicios de información sobre WWW

Daniel Rodríguez García

Versión 1.0

6 de junio de 1997.

Nota importante:

Algunas partes de este manual, que no son originales han sido obtenidas de revistas, libros, y cómo no, de WWW. Los libros y revistas utilizados están referenciados en la bibliografía. Algunas de otras direcciones de donde han sido obtenidos partes importantes son:

- Manual de Internet Information Server de Microsoft ©.
- Simedia Proxy en <http://www.simedia.es/>
- El tejedor en <http://www.dic.uchile.cl/~manual/>
- Perdón si me dejo alguno...

Conceptos Generales

Conocimientos Previos

Internet

Internet es una red mundial de equipos que se comunican usando un lenguaje común. Es similar al sistema telefónico internacional: nadie posee ni controla todo el sistema, pero está conectado de tal manera que hace que funcione como una red muy grande.

Intranet

Intranet hace referencia a cualquier red que no esté conectada a Internet pero que utilice estándares y herramientas de comunicación de Internet para suministrar información a los usuarios una la red privada. Por ejemplo, una organización puede instalar servidores Web a los que los empleados puedan tener acceso para publicar boletines informativos de la organización, cifras de ventas y otros documentos de la organización. Los empleados tienen acceso a la información mediante exploradores de Web.

Los servidores Web pueden configurarse para suministrar una intranet con las mismas características y servicios que se encuentran en Internet, como páginas de hipertexto (que pueden contener texto, enlaces, imágenes y sonidos), que respondan a las peticiones de información de los clientes del Web y tengan acceso a bases de datos.

Direcciones IP y DNS

Internet es una colección de redes TCP/IP particulares repartidas por todo el mundo. Cada ordenador dispone de una o más direcciones (direcciones IP), que es una secuencia de 4 números entre 1 y 254, que identifican de manera única a cada ordenador. La información se transmite en Internet mediante paquetes de datos donde Cada paquete va dirigido a la dirección IP de un equipo concreto, como 10.212.57.189.

Este formato (dirección IP) es adecuado para las máquinas pero difícil para nosotros que preferimos darles un nombre. Es decir, 158.227.112.88 es "sisf00.si.ehu.es", y para ello está el sistema de nombres de dominio (DNS). El DNS asocia una dirección IP específica, como 158.227.112.180, a un nombre de dominio, como "sisf00.si.ehu.es" es decir, el que convierte el nombre en dirección IP. La estructura de los nombres de dominio es jerárquica.

Cuando un usuario explora Internet usando un nombre de dominio, el explorador debe contactar primero con un servidor DNS para resolver el nombre de dominio a una dirección IP y después contactar con el equipo que tiene esa dirección IP.

Esto tiene dos implicaciones para su servidor:

- Debe tener una dirección IP permanente asignada a cada servidor de Internet.
-

- Debe registrar un nombre de dominio en el DNS por cada dirección IP permanente.

Normalmente, su proveedor de Internet (ISP) le suministrará las direcciones IP y también puede registrar los nombres de sus dominios.

WWW

Un conjunto de servicios hipermedia, ofrecidos en todo el mundo a través de INTERNET, constituyen lo que se llama WWW (World Wide Web – o simplemente Web). World Wide Web ofrece una interfaz gráfica y sencilla para recorrer y consultar los documentos de Internet / intranet. Dichos documentos, así como los vínculos entre ellos, componen una red o “web” de información. Las páginas pueden contener texto, imágenes, películas, sonidos, casi cualquier cosa. Estas páginas se pueden encontrar en equipos situados en cualquier parte del mundo. Cuando se conecta con Web, dispone del mismo acceso a la información en todo el mundo. Los archivos o *páginas* de Web están interconectados. Para conectarse a otras páginas puede hacer clic en el texto o en los gráficos especiales, que se llaman *hipervínculos* o enlaces.

Arquitectura Cliente – Servidor

El WWW está basado en un modelo **cliente – servidor**, utilizando el protocolo HTTP: (HyperText Transfer Protocol, –protocolo de transferencia de hipertexto–). Un ordenador actúa como servidor, ofreciendo la información, y otro como cliente, recibiendo la. El ordenador que actúa como servidor, debe estar corriendo un programa especial, generalmente llamado httpd (http daemon), que es capaz de recibir los requerimientos de información y atenderlos, es decir, el cliente al indicársele que se conecte a algún servidor (servicio en el Web), envía una señal al ordenador indicado / solicitando la información. El servidor la recibe y contesta este requerimiento, enviando el documento solicitado.

Navegadores / clientes / browsers

Clientes WWW, también llamados navegadores, browsers, exploradores, son una herramienta para recorrer y tener acceso a la información en WWW. Un navegador es un programa que reside en el cliente y que permite visualizar páginas HTML, imágenes o directorios de ficheros, pudiendo encontrarse todo esto en un ordenador remoto.

También es posible “navegar” en nuestro propio ordenador, con lo que podremos aprender a manejar el browser o probar nuestras páginas HTML sin tener que configurar un servidor, y dicho sea de paso, sin que se produzca gasto telefónico.

Los clientes o navegadores más comunes son el Navigator y Communicator de Netscape y el Internet Explorer de Microsoft.

Servidor WWW o Web

Un servidor Web (o demonios WWW) es un programa que le permite publicar información en una intranet de la organización o en Internet, es decir, los servidores Web son los programas que atienden las peticiones de los clientes mediante el Protocolo de transferencia de hipertexto (HTTP).

Existen multitud de Servidores Web, alojados con frecuencia en máquinas Unix ó NT. Algunos de los más comunes son: Commerce Server, Fast Track, Enterprise Server, y Communications Server de Netscape; Net Action de Lotus; Internet Information Server y FrontPage de Microsoft; y WebSite de O'Reilly, etc.

PLUG-INS

Los plug-ins son programas que extienden las capacidades de los navegadores de una forma específica, proporcionando, por ejemplo, capacidades de audio, vídeo, etc. Las compañías están desarrollando continuamente multitud de plug-ins que cubren un gran número de funciones. Es habitual navegar por la red e instalar automáticamente un nuevo plug-in o una nueva versión de uno que ya se posee.

Consideraciones en la instalación de servicios de Internet

En este capítulo se comentan una serie de requisitos y factores que influyen en la instalación de servidores WWW en una organización. No se comenta la opción de alquilar estos servicios, que para muchos tipos de empresas, puede ser una opción más barata.

Requisitos

Para instalar un servidor WWW es necesario que la organización tenga acceso permanente a Internet, un router / ordenador con software de proxy, que conectará la red de nuestra organización a Internet y evidentemente, los servidores que se quieran: FTP, Gopher, WWW, news IRC, correo electrónico, Telnet...

Acceso permanente a Internet

Para que el resto del mundo pueda tener acceso a un servidor, es necesario contratar una línea dedicada a Internet con un mínimo de 64KB de ancho de banda. El coste de la línea depende del proveedor de servicio, y de parámetros como distancia, tasa de información asegurada,...

Las conexiones con Internet se alquilan normalmente a los ISP (proveedor de Internet). Además de suministrar la conexión física con Internet y la dirección IP (y la máscara de subred si fuera necesario), el ISP puede suministrar muchos de los servicios de Internet, como el registro de nombres de dominio, encaminadores y servicios DNS.

La conexión con Internet correcta

La conexión con Internet se realiza a través de una tarjeta adaptadora de red o de otros dispositivos de red, como modems o tarjetas RDSI (Red digital de servicios integrados). El ancho de banda de Internet se mide en bits por segundo (bps).

La configuración de su servidor y el ancho de banda de Internet determinan la velocidad de transmisión a su equipo y la cantidad de peticiones simultáneas que se pueden atender. A medida que aumente el número de equipos que obtengan información a través de su conexión con Internet, se producirán retrasos o fallos a menos que disponga del ancho de banda necesario.

Cuando alquila una conexión con Internet, el ISP instala un cable de red en su sitio. Las conexiones alquiladas tienen velocidades que van desde 56.000 bps (con Frame Relay) hasta 45.000.000 bps (con conexiones T3). Las líneas telefónicas RDSI (ISDN) pueden ofrecer velocidades de hasta 128.000 bps.

Tipos de conexión con Internet

Los tipos de conexiones que se describen en la siguiente tabla representan los niveles de servicios típicos para una conexión completa con Internet desde Norteamérica y Japón. Es posible que en otros países los servicios de Internet que se proporcionan a través de los proveedores de servicios de Internet varíen considerablemente. Puede que observe otras diferencias, dependiendo de la naturaleza del hardware, el contenido que haga disponible desde su sitio y otras variables.

Conexión	BPS máximo	Usuarios simultáneos aceptados
Frame Relay	56.000	10–20
ISDN	128.000	10–50
T1	1.500.000	100–500
T1 fraccional	varía según las necesidades	
T3	45.000.000	Más de 5000

Los servidores con poca carga de trabajo pueden usar conexiones Frame Relay o RSDI (ISDN). Los servidores con tráfico medio pueden usar una línea T1 o una fracción de línea T1. Las grandes organizaciones que prevén gran tráfico en Internet pueden necesitar varias líneas T1 completas o fraccionadas, o incluso líneas T3 para poder dar servicio a miles de usuarios.

Existen conexiones con Internet a través de modems, pero se usan normalmente para exploración de clientes individuales y no se recomienda su uso para servidores. Una conexión con Internet a través de módem y una línea telefónica sólo puede dar servicio a dos o tres usuarios a la vez. (Los servidores de Internet en modo texto con un número reducido de usuarios potenciales podrían usar conexiones a través de modems.) Las conexiones por módem se denominan “vínculos lentos” porque los datos se transmiten a la velocidad del módem, normalmente entre 9.600 y 28.800 bps, muy lenta para el correcto funcionamiento de un servidor de Internet.

Encaminadores / Router

Dispositivo hardware. TCP/IP es un protocolo enrutable, lo que significa que cada unidad de información (paquete) tiene una dirección específica a la que se envía. Los encaminadores / routers conectan dos redes, encaminando los paquetes entre ellas. Estos dispositivos comprueban el destino de cada paquete que le llega y si el destino se encuentra en la otra red, envía el paquete a otro router (encaminar). Si el paquete destino se encuentra en la red propia, lo manda a su destinatario.

Puede configurar los encaminadores para que sólo permitan el paso entre redes de determinados paquetes, en un proceso llamado *filtrado de paquetes*. El filtrado de paquetes se puede usar para evitar que los usuarios vean o se conecten a equipos y recursos internos.

Muchas veces a los router se le llama CISCO, pero esté es el nombre de una empresa comercial.

Servidores de Internet (WWW, FTP,...)

Se instalan los servicios que se quieran de Internet (FTP, Gopher, WWW, news IRC, correo electrónico, Telnet) sobre un ordenador que soporte el protocolo TCP/IP. Además es necesario que se disponga de un servidor de nombres (DNS, Domain Name Server).

Parámetros a considerar

El router y la conexión a Internet tienen un coste relativamente fijo. El router, que puede ser comprado o alquilado, una vez instalado y configurado ya no es necesario volver a tocarlo. La conexión a Internet suele ser una cuota fija cada cierto tiempo, cuyo coste depende de la calidad del servicio.

El servidor, sin embargo es más dinámico. Puede variar la información que continúe, los servicios y la expectativa que tengamos de él.

Servicios necesarios

Además de WWW, es posible que se quieran dar otros servicios como FTP, News, Telnet, correo, acceso por teléfono (con SLIP o PPP), en la actualidad o en un futuro cercano. Se deben descartar las configuraciones que no cumplan todos estos servicios.

Precio

El precio de la máquina que opera como servidor, puede variar enormemente. Desde un clónico PC con Linux y software gratuito hasta soluciones "llave en mano", que también pueden variar desde el Workgroup Server de Apple hasta WebFORCE de Silicon Graphics pasando por los servidores Netra de Sun. Cada compañía tiene un gran rango de posibilidades.

Prestaciones

Con muchas consultas y un servidor poco potente, se pierde calidad de servicio. La información sufre retrasos e incluso puede llegar a no poder servir información alguna o cerrarse la comunicación.

Facilidad de instalación y mantenimiento del sistema.

En el servidor hay que hacer un determinado número de tareas para comprobar que todo marcha correctamente. Además se puede querer tener programas asociados de estadísticas de acceso al servidor, acceso a Bases de datos, seguridad,... Dependiendo de la complejidad puede estar una persona con conocimientos medios hasta un técnico dedicado en exclusiva a esas tareas.

Versatilidad de manejo.

Los sistemas sencillos son más fáciles de mantener pero suelen ser más rígidos. Esta rigidez puede dar lugar a que no se puedan instalar opciones que sí son posibles en sistemas más complejos.

Diseño de contenido.

Si la empresa se diseña sus páginas HTML, es necesario tener algún sistema que soporte los programas gráficos, editores, escáner,... para la realización de las páginas. Este sistema puede ser el propio servidor o un pequeño ordenador personal.

Sistemas UNIX genéricos sobre PC.

Bajo un ordenador PC, con características de gama media-alta, se instala uno de los muchos sistemas operativos UNIX que hay actualmente:

- SCO UNIX, Solaris,...
- FreeBSD, Linux,... (gratuitos)

Estas versiones de software ya poseen servidores de FTP, Telnet, News,...,e incluso WWW. Pero podemos añadir otro software de WWW a nuestro gusto:

- NCSA, CERN, Apache,... (gratuitos)
- Netscape Communications Server,
- Netscape Commerce Server,... (comerciales)

Dependiendo de si el PC es clónico o no, y de si el software escogido es gratuito, ésta puede ser la configuración más barata, que no significa la que peor.

Microsoft

El servidor WWW de Microsoft se llama Internet Information Server (IIS), que además proporciona los servicios de FTP y Gopher trabaja sobre el sistema operativo de red Windows NT Server. Destaca su sencillez de instalación y el DNS lo proporciona el propio Windows NT Server, pero para los demás servicios (correo,...) hay que añadir paquetes de software adicionales.

Microsoft también dispone de Internet Merchant Server es un servidor seguro para transacciones electrónicas de compras basado en el protocolo SET definido por VISA y Mastercard.

Microsoft dispone de una *“suite de aplicaciones servidor”* llamado **BackOffice Server**, que se compone de:

- Windows NT Server
- SNA Server
- SQL Server Exchange Server
- Systems Management Server (SMS)
- Proxy Server
- Se le pueden añadir Transaction Server y Merchant Server,

con lo que Microsoft pretende agrupar todos los servicios de administración de sistemas, Internet, bases de datos y mensajería.

MacIntosh

Apple se basa en los servidores Workgroup con PowerPC más software para crear los Apple Internet Server.

El software se compone del servidor WebStar y las herramientas necesarias para la creación de documentos (editor BBedit, Netscape, y el paquete completo de Adobe Acrobat). Además se incluyen algunos motores de Bases de Datos.

SUN Microsystems

Han estado en la red desde su creación y algunos servidores claves como el del CERN o NCSA han sido desarrollados en este tipo de máquina. El producto integrado de Sun se llama Netra Internet Server y se compone de estaciones SPARCStation 2/10 con servidor WWW de NCSA para los modelos inferiores y Netscape Communications Software para los modelos superiores.

Al ser UNIX, es versátil y por el mismo motivo más complejo que los sistemas vistos anteriormente. Cualquier innovación se desarrolla antes para este sistema que para los demás y se ha convertido en un “estándar de facto”.

Además Sun a desarrollado Java, lenguaje que permite ejecutar aplicaciones en la red. Al ser una plataforma UNIX tiene muchas posibilidades de ampliación de servicios (FTP, mail, news,...).

Silicon Graphics

Los servidores de Silicon Graphics se llaman WebForce e incluyen herramientas para la creación de páginas HTML:

Netscape, WebMagic para el diseño de las páginas y WebForce para crear espacios en 3D. Performance Copilot es un software que permite evaluar la carga y el rendimiento del servidor.

Silicon Graphics a desarrollado un lenguaje llamado VRML (Virtual Reality Modeling Language) para espacios en tres dimensiones que ha sido adoptado por la industria.

Pese a ser un entorno UNIX tiene un potente entorno de ventanas que hace que el sistema sea sencillo de administrar, configurar y utilizar.

Parámetros en la configuración de WWW

En este punto se comentan los aspectos a configurar en un servidor WWW. En cada servidor se hace diferente pero parecida por lo que sólo se comentan. Aunque, cada día que pasa, las herramientas de configuración son más visuales y los servidores resultan más fáciles de administrar.

Parámetros de red.

Como puede ser el nombre del servidor, puerto por el que escucha, número máximo de conexiones, direcciones desde las que se permite acceder y direcciones de las cuales no se permite, si el servidor puede actuar como proxy, si se configura como proxy, etcétera.

Directorios

Los directorios de las direcciones en WWW, no suelen coincidir con los reales en el ordenador, así como donde están los directorios de programas CGI, cuál es el directorio base, etc.

Ficheros de registro

Nombre y dónde se guardan los ficheros de los accesos al servidor.

Seguridad

Hay directorios en los que se necesita una clave para entrar, así como el tipo de seguridad para esos directorios, etc.

MIME Types

Extensiones de los ficheros que corresponden a cada tipo (Ver MIME Type).

Navegar por directorios

Si el servidor permite devolver la lista de ficheros y directorios cuando en la petición no se especifica fichero y no existe el fichero por defecto en ese directorio.

Servidores Virtuales

Un servidor puede albergar varios nombres de dominio (como `www.organizacion1.com` y `www.organizacion2.com`) en el mismo equipo, asignando varias direcciones IP a la tarjeta adaptadora de red conectada a Internet.

Otros...

Aquí se pueden englobar aspectos como el nombre de la página por defecto, *keep alive*, que es el número de segundos que el servidor mantiene viva una conexión, ficheros a devolver cuando se producen errores, etc.

Descripción del servicio WWW

URL (Uniform Resource Locator)

URL (Uniform Resource Locator) constituyen una manera estandarizada de dar una dirección en Internet a un recurso de información.

De una forma más simple, se puede pensar en ello, con el mismo concepto de fichero pero a través de una red, en la que el fichero puede o no estar en el propio ordenador. Además puede ser enviado por medio de diferentes formas o protocolos. Pero los puede referenciar a preguntas, bases de datos, etc. La forma del URL dependerá de lo que se está referenciando.

Referenciando un hipertexto

```
http://host:puerto/dir1/dir2/.../dirn/archivo.html#seccion
```

Ejemplos:

```
http://www.ji.si.ehu.es/
```

```
http://sunsite.dcc.uchile.cl/wm/paint/auth/index.html#dali
```

`http://`, que indica que lo que viene es una referencia a un hipertexto. A continuación se indica el nombre del ordenador en que se encuentra el documento, luego el directorio en que se encuentra, y finalmente el nombre del fichero en que está guardado el hipertexto.

Puerto por donde se hacen las peticiones al servidor WWW. Generalmente se utiliza el que esta por defecto, que el 80, y no hace falta ponerlo.

Mediante el signo # se indica que es una sección dentro del una página html (hipertexto) y no al comienzo (ver como se crean enlaces dentro de html).

Notas:

- Si no se especifica el nombre del fichero, se suele enviar un archivo llamado por defecto, que generalmente se llama, `index.html` o `Default.html`, pero esto es una de las opciones a configurar en el servidor WWW. Por eso, en la mayoría de los casos, indicar sólo el nombre del ordenador es suficiente para iniciar una búsqueda.
- El directorio que se indica está tomado en referencia al directorio base del servicio HTML (no coincide con la raíz del árbol de subdirectorios del ordenador)
- Los servidores también se pueden configurar para acceder a directorios usuario. Que generalmente, es un subdirectorio, dentro del directorio HOME del usuario, al que se se suele llamar `public_html`, `pub_www` (también a configurar).

Referenciando un documento gopher

`gopher://host:puerto/dirl/dir2/.../dirn/archivo`

Ejemplo: `gopher://sunsite.dcc.uchile.cl/cec/readme.txt`

`gopher://`, que indica que lo que viene es una referencia a un servicio de tipo gopher. A continuación se indica el nombre del ordenador en que se encuentra el documento, luego el directorio en que se encuentra, y finalmente el nombre del fichero en que está guardado el documento.

Nota: El sistema Gopher es anterior al sistema WWW y actualmente existe mucha información en este tipo de sistemas, que gradualmente ha ido trasladándose a servicios hipermediales.

Referenciando un servicio FTP

`ftp://host:puerto/dirl/dir2/.../dirn/archivo`

Ejemplo: `ftp://sunsite.dcc.uchile.cl/OS/packages/`

`ftp://`, que indica que lo que viene es una referencia a un servicio ftp (transferencia de archivos) . A continuación se indica el nombre del ordenador que ofrece el servicio ftp, luego el directorio en que se encuentra, y finalmente el nombre del archivo.

En caso de que no se especifique archivo, se estará referenciando a un índice con los contenidos del directorio especificado. En caso de que no se especifique el directorio, se referenciará a la raíz del servicio FTP.

Referenciando un archivo cualquiera

`file://host:puerto/dirl/dir2/.../dirn/archivo`

Ejemplo: `file://sunsite.dcc.uchile.cl/OS/packages/`

`file://`, que indica que lo que viene es una referencia a un archivo cualquiera. A continuación se indica el nombre del ordenador que donde se encuentra el archivo, luego el directorio en que se encuentra, y finalmente el nombre del archivo. Debe especificarse el nombre de archivo

Referenciando un grupo de noticias

`news:grupo`

Ejemplo: `news:es.alt.chistes`

`news`, que indica que lo que viene es una referencia a un grupo de noticias. A continuación se indica el nombre del grupo de noticias.

Nota: La forma en que el cliente recibirá la información del grupo de noticias depende exclusivamente de él. Es decir, debe estar adecuadamente configurado para leer noticias desde algún ordenador. No es posible, por lo tanto, especificar en el URL desde que ordenador deberá tratar de obtener la información del grupo de News.

Referenciando una dirección e-mail

<mailto:dir1>, dir2, ... dirN

Ejemplo: `mailto:alguien@algunsitio.es, persona@donde.com`

`mailto:` indica que lo que viene es una referencia a uno o varios usuarios que recibirán el correo electrónico.

Nota: El cliente debe estar adecuadamente configurado para poder enviar el mail.

HTTP (HyperText Transfer Protocol)

El protocolo HTTP (HyperText Transfer Protocol) ha sido usada en la Web desde 1990, define los términos en que se produce la comunicación entre un cliente HTTP (un browser como Netscape, por ejemplo) y un servidor HTTP (un servidor Web).



La conversación que se lleva a cabo entre un cliente y un servidor se realiza mediante texto normal. Sobre Internet el protocolo está sobre TCP y generalmente sobre el puerto 80. El flujo de información que se produce es cómo se indica a continuación.

El cliente envía al servidor la siguiente información (de manera absolutamente transparente):

1. El método de la petición, la tabla 1 muestra algunos de los métodos posibles.
2. El nombre del documento que desea.
3. La versión de protocolo HTTP que se empleará en la comunicación.
4. Una lista de los tipos de datos que está dispuesto a aceptar (por ejemplo, si tenemos deshabilitada la opción de cargar gráficos, no incluirá el tipo gráfico en la lista).
5. Su propio nombre y versión.
6. Más cosas como el cliente que es, etc. y una **línea en blanco** para indicar el final.

GET	Devolver un fichero
HEAD	Igual a GET, pero el servidor no devuelve la el cuerpo del fichero solo información a cerca del fichero (meta-información). Generalmente usado para testear enlaces, accesibilidad, modificaciones,...
POST	Enviar datos al servidor
PUT	Enviar datos al servidor

Tabla 1. Métodos de petición

Los siguientes ejemplos los envía un cliente a un servidor. Junto a la URL que se solicitaría, está la petición que se envía:

http://www.ehu.es/	GET / HTTP/1.0
http://www.ehu.es/index.html	GET /index.html HTTP/1.0
http://www.ehu.es/prog/appl.html	GET /prog/appl.html HTTP/1.0

Ejemplo de lo que manda un navegador es:

```
GET /index.html HTTP/1.0
Referer: http://turtle.ee.ncku.edu.tw/~antony/test/get.html
User-Agent: Mozilla/2.01Gold (Win95; I)
Host: libsun1.lib.ncku.edu.tw:8080
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
```

El servidor responde con la siguiente información, que el cliente interpreta adecuadamente para mostrarnos el documento solicitado en pantalla:

1. La versión de protocolo HTTP, indicando que está de acuerdo en la indicada por el cliente.
2. Un código indicando lo que a ocurrido. Ver tablas de códigos de estados.
3. La fecha.
4. Nombre y versión del servidor.
5. Más información, entre la que se encuentra el tipo de documento enviado, así como su longitud.
6. Una línea en blanco, seguida del documento en sí.

Por ejemplo:

```
HTTP/1.0 200 OK
Date: Monday, 3-Jun-96 20:00:53 GMT
Server: NCSA/1.4
MIME-version: 1.0
Content-type: text/html
Last-modified: Wednesday, 21-Jun-96 14:31:45 GMT
Content-length: 2719

<HTML><HEAD>Servidor WWW de ...</HEAD>
<BODY>
...
</HTML>
```

Tablas de códigos de estados:

1xx: Información	No usado, reservado para el futuro
2xx: Success	La acción fue exitosamente recibida y ejecutada
3xx: Redirection	Debe ser realizada otra acción para completar la petición
4xx: Client Error	La petición contiene un error sintáctico o no puede ser cumplida.
5xx: Server Error	El servidor falló pese a realizarse una petición aparentemente correcta

Tabla 2. Códigos de estados (general).

200	OK
201	Created
202	Accepted
204	No Content
301	Moved Permanently
302	Moved Temporarily
304	Not Modified
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable

Tabla 3. Tabla de código de estados (específica).

En otras ocasiones, lo que el cliente HTTP requiere del servidor no es un documento HTML, sino un documento que contiene un programa. En ese caso, lo que hace el servidor es ejecutar dicho programa (llamado programa CGI). El trabajo del programa CGI suele ser construir un documento HTML sobre la marcha para que el servidor se lo mande al cliente. El documento devuelto no existe en realidad en el servidor, sino que se crea por el programa partiendo de la nada. La comunicación entre cliente y servidor es igual que en un intercambio normal de documentos, con la diferencia de que el documento que devuelve el servidor no existe de forma permanente.

La versión 1.0 de HTTP (HTTP/1.0) –la más utilizada actualmente- abre una nueva conexión por cada petición. Con la versión 1.1 posee la característica de conexión persistente. Con lo mejora mucho el rendimiento del sistema, entre 8 y 10 veces mejor.

Otras características de la versión 1.1 sobre la 1.0 son:

- Características de Cache
- Identificación mejorada (Enhanced Authentication): Basic, Digest (Explicada en RFC2069)
- Rango de petición (Range Request) en bytes.

MIME (Multipurpose Internet Mail Extensions)

MIME "Multipurpose Internet Mail Extensions" (MIME definido en RFC 1521.) es el estándar para enviar por correo electrónico, datos binarios usando el sistema de correo electrónico. Los usos típicos de MIME incluyen el envío de imágenes, audio, documentos de procesadores de texto o incluso ficheros de texto cuando es importante que sistema de correo no modifique ninguna parte del fichero. MIME permite además etiquetar partes de un mensaje para que el receptor (o programa de correo) puede determinar que hacer con él. Anteriormente a la existencia de los MIME-Types, se utilizaban programas como UUENCODE y UUDECODE para convertir los ficheros binarios, a sólo texto y viceversa, antes de enviarlos por el correo electrónico. Pero van cayendo en desuso rápidamente, ya que los programas que gestionan correo con MIME, lo utilizan una codificación más eficiente, llamada "Base64" pese a que no lleva muchos años.

Este mismo concepto se a aplicado en WWW para el envío sobre el protocolo HTTP de las diferentes partes de las que se compone un documento en hipertexto (HTML), aunque mejor sería decir hipermedia.

Entonces, suele haber dos ficheros, uno en lado del servidor y otro en el lado del cliente que sirven para hacer la correspondencia entre los ficheros y MIME Type, y entre los ficheros que le llegan al cliente y las aplicaciones de ayuda que lanza el navegador. Hoy día, la mayoría de las clientes y servidores tienen suelen tener herramientas visuales que abstraen de usar esos ficheros directamente. Esos ficheros se suelen llamar `mailcap` y `mime.types`.

Fichero Mailcap

El fichero “mailcap” está definido en RFC 1524. Básicamente un fichero mailcap es un fichero de configuración que hace un mapping de MIME types con visores externos. Por ejemplo podría ser el mapping del MIME type `image/gif` con algún programa de tratamiento de gráficos como por ejemplo `xv`, `Paint Shop Pro`, etc.

Ejemplo de fichero mailcap:

```
# This is a simple example mailcap file.
# Lines starting with '#' are comments.

# This maps all types of audio data (audio/basic, audio/x-aiff,
# etc.) to the viewer 'showaudio'. Note that '%s' means 'put the
# datafile name here when the viewer is executed'.
audio/*; showaudio %s

# This maps all types of images (image/gif, image/jpeg, etc.)
# to the viewer 'xv'.
image/*; xv %s

# This maps MPEG video data to the viewer 'mpeg_play'.
video/mpeg; mpeg_play %s

# This maps all types of video *other than MPEG* to the viewer
# 'genericmovie'.
video/*; genericmovie %s

application/postscript; ghostview %s
application/x-dvi; xdvi %s
.
.
.
```

NOTA: No oficiales o experimentales tipos MIME, están marcados en el subtipo con 'x-'. Por ejemplo como AIFF audio no es todavía un MIME.-Type registrado, aparecerá generalmente como `audio/x-aiff`. (Cuando esté registrado será probablemente `audio/aiff`.)

Fichero MIME.TYPES

Asocia las extensiones de los ficheros con sus tipos MIME.

Ejemplo de fichero mime.types:

```
# This is a comment.

application/activemessage
application/remote-printing
application/x-sv4cpio          sv4cpio
application/x-ustar            ustar
```

audio/basic	au snd
audio/x-aiff	aif aiff aifc
audio/x-wav	wav
image/gif	gif
image/jpeg	jpeg jpg jpe
image/tiff	tiff tif
image/x-portable-bitmap	pbm
image/x-xwindowdump	xwd
message/external-body	
message/news	
multipart/parallel	
text/html	html htm
text/x-sgml	sgml sgm
text/plain	txt
video/mpeg	mpeg mpg mpe
video/quicktime	qt mov
video/x-msvideo	avi

Ejecución de aplicaciones sobre WWW en el lado del servidor

En los comienzos de la Web, abundaban las páginas estáticas y los beneficios económicos sólo se podían generar gracias a la publicidad o el pagar por ver. La conexión a Internet y el desarrollo de un Web propio podría considerarse relativamente caro y poco práctico. Mucha gente que se conectaba por primera vez a Internet se sentía desilusionada y gran número de empresas de informática consideraban Internet como una moda pasajera, sin aplicación en los negocios. Todos ellos pudieron tener razón en su momento. Sin embargo ha producido un gran incremento del número de personas conectadas en todo el mundo, y ha mejorado en gran medida la calidad y la oferta de los Web. El éxito de Internet se debe a una vieja idea, a la vez sencilla y ambiciosa: permitir acceder a información alojada en cualquier parte del mundo, en cualquier tipo de máquina, desde cualquier tipo de máquina.

A continuación se explican como se crean formularios, que en el fondo también son marcas HTML pero que ayudarán a enviar información que introduzca el cliente hacia el servidor y después a continuación, veremos como se pueden recoger esos datos con CGIs, etc....

CGI (Common Gateway Interface)

Introducción

En un documento HTML plano el texto no cambia, es estático, pero en infinidad de ocasiones es necesario que las respuestas al cliente (browser) se creen dinámicamente, es decir, en tiempo real.

Para poder ejecutar esas aplicaciones externas, ha surgido un estándar, un protocolo, llamado CGI (Common Gateway Interface), es decir, pasarela e interfaz, porque normalmente el CGI ha de ser el intermediario entre el Servidor Web y otro programa,.. CGI define una serie de parámetros conocidos como variables de entorno, que describen las peticiones del cliente. Esto define una interfaz independiente entre los programas o scripts y el servidor HTTP.

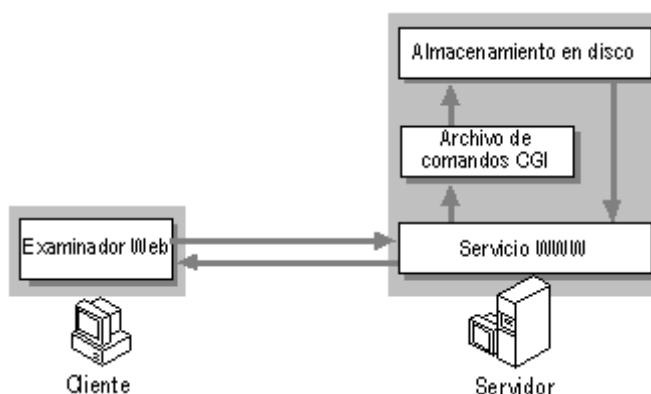
Los usos de los CGI pueden ser de lo más variado como se comentó en los formularios aunque su mayor uso será el de acceder a Sistemas de Gestión de Base de Datos (SGBD), o Servidor de Base de Datos. Es decir, se utiliza un programa (generalmente se le llama a este programa CGI) cuya función es realizar todos los accesos a la base de datos.

Para unir una base de datos a WWW, necesitaríamos construir un programa CGI que el servidor Web ejecuta para que pregunte al motor de la BD, recoja el resultado y se lo devuelva al cliente. Además, la base de datos no tiene por qué estar en el mismo ordenador que el servidor Web.

Otros ejemplos pueden ir a casos más triviales como añadir la fecha a una página junto con su santoral, mostrar el número de usuarios que han visitado dicha página.

El CGI reside en el servidor, el lenguaje a utilizar para crearlo dependerá del tipo de servidor Web de que dispongamos. En un servidor Windows NT se pueden usar C++ o Visual Basic; en un servidor Macintosh, Apple Script o aplicaciones programables MacOS; en un servidor Unix o VMS tenemos C, C++, Fortran, Perl, TCL, y todos los shells Unix. Mucha gente prefiere script o shells por que son más fáciles de mantener y depurar que los lenguajes compilados. Hay escritas muchas bibliotecas que nos facilitan el trabajo, más adelante hay referencias a ellas.

Una limitación importante en estos programas es el tiempo, ya que estas páginas o respuestas son creadas en tiempo real y el cliente está esperando por ellas. Además hay que añadir lo que tarda la página en llegar (que pasa por el servidor), por eso debemos ver cuando el tiempo de respuesta supera lo razonable.



CGI. Entrada

■ Con el método GET:

Si se ha utilizado el `METHOD="GET"` en el formulario, cuando se envían los contenidos son unidos a la pregunta URL de la forma:

```
programa?nombre=valor&nombre=valor
```

Y el programa CGI recibirá la entrada del input codificado en una variable de entorno llamada `QUERY_STRING`.

■ Con el método POST:

Para peticiones que tienen información después de la cabecera (header), esto ocurre cuando es enviada con el método `POST`. Debe haber un método para leer esos datos, a menos que se defina otra forma, ésta es mediante la entrada estándar. El servidor enviará `CONTENT_LENGTH` y el `CONTENT_TYPE` de los datos.

Nota: el servidor no está obligado a poner EOF (End Of File). Así que debe controlarse con `CONTENT_LENGTH`.

Ejemplo:

Si tenemos un formulario con `METHOD="POST"`, entonces por la entrada estándar (stdin) podemos tener algo así:

```
nombre=Juan+Valdes&dpto=lsi
```

En este caso el servidor pondrá `CONTENT_LENGTH=26` y `CONTENT_TYPE = application/x-www-form-urlencoded`. (variables de entorno).

Bibliotecas de funciones

Hay una serie de bibliotecas de funciones que nos ayudan a decodificar la entrada sin tener que hacer nosotros los cambios comentados anteriormente. Además existen en casi todos los lenguajes aunque los más comunes son C/C++ y Perl.

Muchas de ellas son freeware o shareware. y disponibles en internet. Algunas de ellas son:

En C/C++:

- `cgic` de Thomas Boutell en <http://www.boutell.com/cgic/>
- EIT CGI Library en <http://wsk.eit.com/wsk/dist/doc/libcgi/libcgi.html>

Para Macintosh: <http://arpp.carleton.ca/grant/mac/grantscgi/>

En Perl:

- <http://www-genome.wi.mit.edu/WWW/tools/scripting/CGIperl/>

En Borland Delphi, la biblioteca WIN-CGI en:

- <http://www.href.com/>

En Turbo Pascal, WINCGI interface escrita por Markus Schlarmann.

- <http://141.2.61.48/tpwcgi/tpwcgi.htm>

Marc Hedlund mantiene las FAQ sobre CGI en:

- <http://www.best.com/~hedlund/cgi-faq/>

y Thomas Boutell sobre WWW, donde también hay información en las FAQ:

- <http://www.boutell.com/faq/>

CGI. Salida.

Non-parsed Header Output

El script debe devolver una respuesta HTTP completa, es decir nuestro programa o script se comunica directamente con el cliente. Esto permite una respuesta HTTP/0.9 a una pregunta HTTP/1.0, aunque esto no es aconsejable. Y lo mismo con el protocolo HTTP/1.1

(Se puede consultar la especificación de HTTP <URL:<http://www.w3.org/>>)

Ejemplo:

Este script habla directamente con el cliente. Si el script es referenciado con `SERVER_PROTOCOL = HTTP/1.0`, el script debería responder con una respuesta HTTP/1.0:

```
HTTP/1.0 200 OK
Content-type: text/plain
SERVER: NCSA/1.5

Aquí va el texto generado por el script.
```

Parsed Header Output

La salida de nuestro programa es tratada por el servidor WWW.

Debe devolver un mensaje de respuesta CGI. Cuya notación en BNF es:

```
CGI-Resp=*(CGI-Header|HTTP-Header) NL [Cuerpo]
CGI-Header= Content-type
           Location
           Status
           extension-header
```

NOTA: NL es Nueva Línea.

Donde:

Content-type: éste es el MIME-type del documento que se está devolviendo (text/plain, text/html, ...).

Location: se devuelve una referencia a un documento, en vez del documento.

Status: usado para darle a un servidor una línea de estado en HTTP/1.0 que éste mandará al cliente. Cuyo formato: 3 dígitos string, ej: 200 OK.

Ejemplos

```
Content-type: text/html

<HTML>
...
</HTML>
```

```
Location: /path/hasta/donde/esta/doc.txt
```

```
Location: gopher://gopher.ehu.es
```

```
Content-type: text/plain

Esto es un fichero en texto normal.
```

Variables de entorno

Para pasar los datos del servidor WWW al script CGI, el servidor usa argumentos en la línea de comandos y variables de entorno, estas variables de entorno se explican en las tablas que vienen a continuación.

Variables de entorno no específicas de la petición y que están en todas las peticiones:

SERVER_SOFTWARE	Nombre y versión del servidor de información (que está ejecutando el programa pasarela). Formato: nombre/versión. Ejemplo:NCSA/1.4
SERVER_NAME	El nombre de host del servidor (o la dirección IP) como debería aparecer en la dirección URL con referencia propia.
GATEWAY_INTERFACE	La revisión de la especificación CGI (Interfaz de puerta de enlace o gateway común) con la que cumple el servidor Formato:CGI/versión. Ejemplo:CGI/1.1

Variables de entorno puestas por el gateway:

SERVER_PROTOCOL	El nombre y la versión del protocolo de recuperación de la información relativa a esta petición, habitualmente HTTP/1.0.
SERVER_PORT	El puerto TCP/IP en el que se recibió la petición.
REQUEST_METHOD	El método de peticiones de HTTP. Puede ser: GET, HEAD, POST.
PATH_INFO	Información adicional acerca de la ruta de acceso, como la proporciona el cliente. Incluye la parte la parte final de la dirección URL, tras el nombre del archivo de comandos pero antes de la cadena de consulta (si la hay).
PATH_TRANSLATED	El valor de PATH_INFO, pero con cualquier nombre de ruta de acceso virtual expandido a una especificación de directorio.
SCRIPT_NAME	Ruta virtual al script que es ejecutado. El nombre del programa de archivos de comandos que se ejecuta.
QUERY_STRING	La información que aparece después del signo de interrogación (?) en la dirección URL que hizo referencia a este archivo de comandos.
HOST_NAME	Nombre del host que hace la petición.
REMOTE_ADDR	Dirección IP del host que hace la petición..
AUTH_TYPE	El tipo de autorización que se esté utilizando. Si el servidor ha autenticado el nombre del usuario, ésta contendrá Basic. De lo contrario, no estará presente.
REMOTE_USER.	El nombre del usuario proporcionado por el cliente y autenticado por el servidor
REMOTE_IDENT	Si el servidor HTTP soporta RFC 931. Se utiliza para conectarse.
CONTENT_TYPE	El tipo de contenido (MIME-Type) de la información proporcionada en la parte principal de una petición POST.
CONTENT_LENGTH	El número de bytes que el archivo de comandos espera recibir del cliente.
HTTP_*	Todos los encabezados HTTP que aún no se hayan analizado en alguna de las variables de la lista. Estas variables tienen el formato HTTP_<nombre del campo encabezado>, por ejemplo: HTTP_ACCEPT: */*, q=0.300, audio/x-aiff, audio/basic, image/jpeg, image/gif, text/plain, text/html HTTP_USER_AGENT: Microsoft Internet Explorer/0.1 HTTP_REFERER: http://server/samples/dbsamp/dbsamp3.htm HTTP_CONTENT_TYPE: application/x-www-

	form-urlencoded
	HTTP_CONTENT_LENGTH: 10

Seguridad en los CGIs

Un CGI es un programa ejecutable, es decir, se permite a todo el mundo ejecutar programas en el sistema, lo cual no es muy seguro. Por eso deben ser tomadas algunas medidas de seguridad.

Una de las medidas suele ser que los programas CGI residan en un subdirectorio especial (normalmente llamado /cgi-bin) y así el servidor Web sabe que lo tiene que ejecutar en vez de simplemente mostrárselo al cliente (browser). Este directorio suele estar bajo el control del administrador Web, no permitiendo que un usuario medio pueda crear CGIs.

Problemas con los CGI

El principal problema de esta opción es que cada cliente solicita la ejecución, en la máquina remota, de una instancia de un ejecutable, existiendo tantas instancias como clientes conectados, y consumiéndose gran cantidad de recursos (memoria RAM) en el servidor.

Otro problema es que el lenguaje con el que programar estos CGI debe tener un fácil acceso a la entrada y salida, como ocurre con el C; sin embargo, estos conceptos no son manejados por los lenguajes de alto nivel.

Más información ...

En NCSA, The Common Gateway Interface.

<http://hoohoo.ncsa.uiuc.edu/cgi/>

En W3:

<http://www.w3.org/>

Alternativas a uso de CGI. Interfaces de programación de Aplicaciones (API)

Servlets (Sun Microsystems)

Java Server APIs, se escriben servlets que se incorporan al servidor:

Ejemplo de un sencillo servlet:

```
import java.io.*;
import java.servlet.*;

public class HelloServlet extends GenericServlet {

    public void service(ServletRequest req, ServletResponse res)
        throws ServletException, IOException {

        PrintStream out = new PrintStream(res.getOutputStream());
        out.println("Hello world!");
    }

    public String getServletInfo() {
        return "Hello World Servlet";
    }
}
```

NSAPI, SSJ (Netscape)

Las API de Netscape se llaman NSAPI.

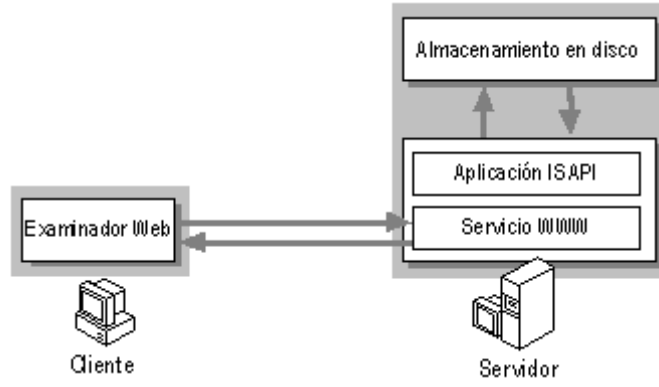
También han creado unas clases Java que muy parecidas a los Servlets, llamadas Server Side Java (SSJ).

ISAPI (Microsoft)

Las API de Microsoft se llaman ISAPI.

Las aplicaciones que usan ISAPI se compilan como bibliotecas de vínculos dinámicos (DLL) que el servicio WWW carga al iniciarse. Como los programas residen en memoria, los programas ISAPI son considerablemente más rápidos que las aplicaciones escritas con la especificación CGI.

ISAPI se puede usar para crear aplicaciones que se ejecuten como DLL en su servidor Web. Si ha utilizado archivos de comandos CGI anteriormente, encontrará que las aplicaciones ISAPI tienen un mejor rendimiento porque se cargan en memoria durante la ejecución del servidor. Requieren menos tiempo de espera porque cada petición no inicia un proceso distinto.



Ejecución de aplicaciones sobre WWW en el lado del cliente

Aquí se ven pequeños ejemplos de cómo ejecutar scripts en el cliente, y cómo por medio del lenguaje Java con los applets se pueden llegar a ejecutar aplicaciones en los clientes, pudiendo llegar a dar una gran funcionalidad a estos.

Este capítulo no se adentra en ello y sólo pretende dar una visión general.

Scripts en el cliente

El código HTML puede llegar puro al cliente con código en otros lenguajes (scripts). Será tarea del navegador o de sus plug-ins, interpretar correctamente estos scripts. En el caso de no ser admitidos por nuestro navegador o browser, simplemente serán ignorados.

Los lenguajes de script más comunes son Visual Basic Script y JavaScript. El VBScript es un subconjunto del lenguaje VBA (Visual Basic para Aplicaciones) y necesita que el navegador sea Internet Explorer, aunque ya existe el plug-in para Netscape. El JavaScript en cambio puede ser interpretado por ambos. Java es un lenguaje de programación parecido al C, del cual JavaScript es un subconjunto, paralelamente a lo que ocurre con VB. Java es más potente y VisualBasic más fácil de aprender.

VisualBasic Srippts

Un ejemplo de página Web a la que se le ha añadido código VBScript es el cuadro siguiente.

```
<HTML>
<HEAD>
<TITLE>Mi Titulo </title>
</HEAD>

<BODY>
Ejemplo VBS.
<input type=text name=txtXY size=50>
</BODY>
</HTML>

<SCRIPT language="VBS">
'comentario
ultimo_x=0
ultimo_y=0

Sub link1_MouseMove (s,b,x,y)
    ultimo_x=x
    Ultimo_y=y
    txtXY.value="Coordenadas: x=" & ultimo_x & "y=" & ultimo_y
End Sub

Sub link1_OnClick
    ired = msgbox ("Ha pulsado Vd. en el punto x=" & ultimo_x & " ,y=" &
ultimo_y , 1,"Mi aplicacion")
```

```
if i ret = 1 then
  msgbox "Gracias por pulsar Aceptar", 0, "Mi_Aplicacion"
end if
End Sub

</script>
```

JavaScript

JavaScript es un nuevo lenguaje. Los 'scripts' de Javascript pueden ser introducidos dentro de sus páginas de HTML. Con Javascript se puede dar respuesta a eventos iniciados por el usuario (el observador de nuestras páginas, por ejemplo), eventos tales como la entrada de una forma o algún enlace.

Esto sucede sin ningún tipo de transmisión. De tal forma que cuando un usuario escribe algo en una forma, no es necesario que sea transmitido hacia el servidor, verificado y devuelto. Las entradas son verificadas por la aplicación cliente y pueden ser transmitidas después de esto. También se puede pensar de programa que se ejecuta en la versión cliente. Por ahora existe un gran número de calculadores en Internet, algunos son proporcionados por Netscape..

Aunque JavaScript se parece a Java, no es lo mismo! Java es un lenguaje de programación mucho más complejo que JavaScript. JavaScript está hecho para ser un lenguaje bastante fácil de entender. A los autores del JavaScript no les debió haber importado mucho el tema de la programación. Por esta razón, algunos elementos de Java no son aceptados en JavaScript.

Pequeños ejemplos de JavaScript

Se muestran algunos pequeños scripts, de modo que se pueda aprender de cómo estos son implementados dentro de los documentos de HTML y mostrar con que posibilidades se cuenta en este nuevo lenguaje script.

El siguiente un script solo mostrará texto dentro de un documento HTML.

```
<html>
<head>
Mi primer JavaScript!
</head>
<body>
<br>
Este es documento normal en HTML.
<br>
  <script language="LiveScript">
    document.write("Esto es JavaScript!")
  </script>
<br>
Otra vez en HTML.
</body>
</html>
```

La salida será:

```
Este es documento normal en HTML.
Esto es JavaScript
Otra vez en HTML.
```

No es muy funcional, pero muestra como se deben usar las marcas.

Las funciones son invocadas por eventos iniciados por el usuario. Los scripts pueden ser ubicados como comentario para asegurarse de que los browsers obsoletos no muestren el script por si solos.

```
<html>
<head>
  <script language="LiveScript">
    function pushbutton() {
      alert("Hola Mundo!");
    }
  </script>
</head>
<body>
<form>
  <input type="button" name="Button1" value="Presióneme"
onclick="pushbutton()">
</form>
</body>
</html>
```

Este script creará un botón que cuando es presionado muestra una ventana diciendo 'Hola!'.

Lo que sucede es lo siguiente. Primero la función se carga y es guardada en memoria. Entonces un botón es hecho con el tag normal `<form>` (HTML). Hay algo completamente nuevo con la etiqueta `<input>`. Allí puede ver 'onclick'. Esto le dice al browser que función tiene que invocar cuando este botón es presionado (lógicamente si el browser es compatible con JavaScript). La función 'pushbutton()' se aclara en el encabezado. Cuando el botón es presionado esta función se ejecuta. Existe otra cosa nueva en este script el método 'alert'. Este método ya es declarado en JavaScript- solo se necesita invocarlo. Existen muchos métodos diferentes los cuales se pueden invocar.

Más información...

<http://home.netscape.com/>

Java

Java es un lenguaje originalmente desarrollado por un grupo de ingenieros de Sun, utilizado por Netscape posteriormente como base para JavaScript. Si bien su uso se destaca en el Web, sirve para crear todo tipo de aplicaciones (locales, intranet o Internet).

Java es un lenguaje:

- Orientado a objetos
- Independiente de la plataforma

Algunas características notables:

- robusto
- gestiona la memoria automáticamente
- no permite el uso de técnicas de programación inadecuadas
- multithreading
- cliente-servidor
- mecanismos de seguridad incorporados

- herramientas de documentación incorporadas

Los applets son una de las particularidades de Java que contribuyen a explicar su éxito. Permiten cargar a través de la red una aplicación portable que se ejecuta en el navegador. Sin embargo, es necesario que el navegador sepa hacerlos funcionar.

Diferencias entre Applet y Aplicación

Los applets presentan las siguientes diferencias respecto a las aplicaciones Java:

Son cargados mediante el navegador (Netscape, Hotjava,...) y no son lanzados directamente por el núcleo Java.

Se cargan a través de la red por medio de páginas HTML y no residen en el disco duro de la máquina que los ejecuta.

Por razones de seguridad, tienen menos derechos que una aplicación clásica; de modo predeterminado, en el puesto que los ejecuta, no pueden ni leer ni escribir ficheros, ni lanzar programas, ni cargar DLL; sólo pueden comunicar con el servidor del que han salido. Estas restricciones de seguridad tienen por objetivo evitar cargar, por error, un applet que podría: destruir datos en su máquina; espiar de una forma u otra.

Ejemplo de applet:

```
import java.applet.*;
import java.awt.*;
import java.util.*;
public class ciclo extends Applet {
    // la tabla en la que se almacenan los nombres de los métodos
    // llamados
    Vector lasLlamadas = null;
    public void paint (Graphics g ) {
        int yActual = 20;
        g.drawString (Llamadas a los métodos:",20,yActual);
        yActual += 15;
        lasLlamadas.addElement ("paint");
        for (Enumeration e= lasLlamadas.elements();
e.hasMoreElements();) {
            g.drawString (e.nextElement ().toString
            ( ),20,yActual);
            yActual +=15;
        }
    }
    public ciclo ( ) { // constructor
        lasLlamadas = new Vector ( );
        lasLlamadas.addElement ("constructor");
    }
    public void init () {
        lasLlamadas.addElement ("init");
    }
    // . . .
    // igual para Start ( ), destroy ( ), stop ( )
    // . . .
}
```

El programa HTML que permite cargar el applet en el navegador es el siguiente:

```
<HTML>
<HEAD><TITLE> El applet</TITLE></HEAD>
<BODY>
<H2> Ejemplo Ciclo de vida de un Applet </H2>
<APPLET CODE="ciclo.class" WIDTH=400 HEIGHT=200>
</APPLET>
</BODY>
</HTML>
```

Más información ...

JavaScript

- <http://www.webconn.com/java/javascript/intro/>

VBScript,

Manual de referencia de Microsoft: <http://www.microsoft.com/workshop>

- <http://www.swlink.net/~lifey/>

Java

- <http://java.sun.com/>
- <http://www.gamelan.com/>

Conexión WWW – Bases de Datos

Hoy día todos los fabricantes de Sistemas de Gestión de Bases de Datos (DBMS –Data Base Management Systems) han desarrollado software para acceder e introducir datos en sus DBMS.

Actualmente existen dos filosofías distintas para construir aplicaciones Cliente/Servidor que accedan a una base de datos remota a través de Internet, que son ejecutando programas en el servidor (como por ejemplo mediante los CGI, servlets,...) y los scripts en el Servidor, que son páginas que antes de ser enviadas por medio del servidor son “interpretadas” y creadas dinámicamente. Algunos CGI son llamados también scripts, lo que puede dar lugar a confusión, pero a lo largo de este capítulo se verá la diferencia entre las dos formas.

A continuación en este capítulo se explican los razones que llevan a llevar unir a las bases de Datos con WWW y después las distintas alternativas mencionadas anteriormente, centrándose en la conexión WWW con Access mediante ODBC (Consultar el manual de IIS).

Motivos que llevan a la unión de WWW con BD

Uso de interfaces gráficas

Las interfaces gráficas hacen más sencillo el acceso a Bases de Datos. Como en la mayoría de los casos los browsers poseen una interface gráfica, hacen que sea más sencillo acceder a las bases de datos.

Con los formularios de HTML, simplemente tienen que rellenar algunos campos y se obtienen respuestas en una forma agradable de leer.

Estandarización

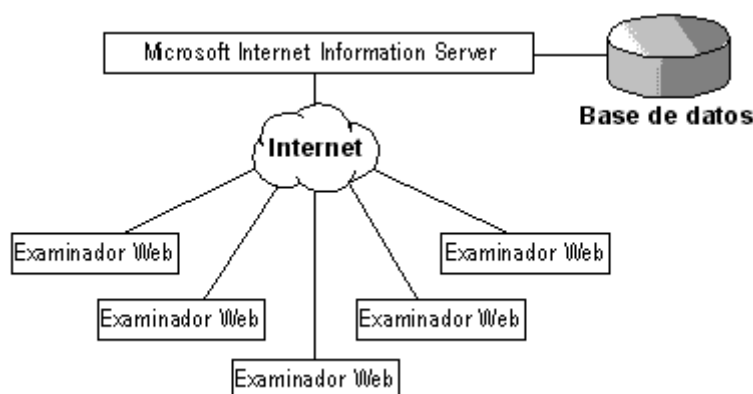
Además, al estar los documentos escritos en HTML, los programadores, para acceder a esas bases de datos sólo tienen que aprender HTML, y los usuarios sólo un tipo de GUI.

Cross-plataform support

Clientes de Web están disponibles en casi todas los tipos de máquinas, así que podemos acceder a bases de datos desde cualquier plataforma.

Acceso a través de redes

Podemos acceder a las bases de datos desde cualquier lugar.



Ejecutando CGIs, API en el servidor...

En este caso se ejecuta un programa que accede a la Base de Datos, puede ser por medio de CGI, ISAPI, NSAPI, Servlets ...

El uso de CGI es la forma tradicional, que se basa en el siguiente proceso. Ciertas opciones realizadas sobre una página HTML en el cliente provocan la solicitud de la ejecución de un programa (CGI) residente en el servidor, al hacer referencia a la URL del fichero ejecutable. Este CGI, al ejecutarse, realiza un acceso a una base de datos a partir de los datos proporcionados por el formulario, creando una página HTML dinámicamente con el resultado de la consulta a la base de datos, que es enviada de vuelta al cliente. Aunque se están haciendo cada día más populares las interfaces de programación de aplicaciones (API) como Netscape Server API (NSAPI) o Internet Server API (ISAPI) de Microsoft debido a que no requieren memoria ni recursos diferentes.

Puede que las consultas SQL y los ficheros HTML se almacenen van por separado. Esta es la forma de funcionar ISS (Internet Information Server) y Access, ambos de Microsoft. O puede que las consultas SQL estén en el mismo fichero que el código HTML y esté fichero se le pase al CGI y cree de ahí la nueva página a ser enviada, esta es la forma de funcionar de miniSQL (Base de Datos) con W3-msql (CGI).

Scripts en el servidor

El Script en el Servidor es tenemos un único fichero HTML que será enviado a todos los clientes; sin embargo, este fichero contiene de pequeños grupos de sentencias escritas en un lenguaje sencillo. El fichero HTML, antes de ser enviado a cada cliente, se tratan, ejecutando dichas sentencias. Este código es el responsable de tomar los datos del formulario del cliente, accederá la base de datos y construir HTML en función de todo ello.

Entre las ventajas de esta opción están el poder utilizar un lenguaje de programación de alto nivel, y el que el programa que se ejecuta en el servidor sea uno sólo, reduciéndose los requerimientos del sistema.

Además, el código de script en el servidor no es de acceso público, al contrario de lo que ocurre con el HTML o el script en el cliente. Ejemplos de scripting en el servidor son

Cold Fusion de Alliare, Live Wire de Netscape e Informix, y Active Server Pages de Microsoft.

COLD FUSION, LIVEWIRE Y ACTIVE SERVER PAGES

Cold Fusion permite desarrollar aplicaciones combinando HTML con CFML (Cold Fusion Markup Language). No se ha de confundir con Cool Fusion, que es un Plug-in para animación y multimedia.

En un servidor Cold Fusion, en vez de documentos HTML existen plantillas de documentos HTML a los que se les ha anadido código CFML. En vez de enviar el documento HTML directamente al cliente, Cold Fusion pre-procesa la plantilla sustituyendo el código CFML por la información solicitada, y el documento dinámico HTML así generado es el que se envía al cliente. A continuación se muestra un ejemplo de una plantilla Cold Fusion (.cfm).

```
<CFQUERY NAME="ListaResultado" DATASOURCE="MiBaseDeDatos">
    SELECT producto Precio FROM Productos
</CFQUERY>

<HTML>
<HEAD>
<TITLE>Listado de Productos</TITLE>
</HEAD>
<BODY>
<H1>Productos</H1>
<P>
<CFOUTPUT QUERY="ListaResultado">
#Producto# #NumberFormat(Precio, "($____.____)"# <BR>
</CFOUTPUT>
</BODY>
</HTML>
```

Netscape e Informix se han unido para lanzar LiveWire, que posee una filosofía similar. El cliente realiza un acceso a una base de datos remota en el servidor donde se ejecutan scripts de Java que producen la generación de código HTML y su envío como respuesta a una consulta. Por ejemplo sencillo lo vemos a continuación:

```
<HTML>
<HEAD>
<TITLE>Ejemplo LW</TITLE>
</HEAD>
<BODY>
Bienvenido. Tu dirección es:
<SERVER>
write(request.ip)
</SERVER>
</BODY>
</HTML>
```

Microsoft por su parte propone Active Server Pages, con la misma filosofía. Los ficheros ASP soportan VBScript y JScript. El siguiente ejemplo de fichero ASP con código VBScript muestra uno u otro formulario en función de los parámetros de la llamada. Supongamos que llamamos a este fichero ej.asp. La primera vez, al no haber parámetros, aparecerá "introduzca su dirección de correo:". Podemos teclear "mherran@tsai.es" y pulsar "Aceptar" en el formulario. Al hacer esto, el formulario se llamará a sí mismo, esta vez con P=2, lo que producirá el mensaje: Su dirección es mherran@tsai.es. Los scripts en el cliente y en el servidor se pueden combinar para ganar velocidad. Por ejemplo, se puede utilizar scripts en el cliente para el control de errores de los datos introducidos por el cliente, cuando éstos

no requieran de un acceso a la base de datos, y scripts en el servidor para el resto de tratamientos.

```
<HTML>
<HEAD>
<TITLE>Ejemplo ASP</TITLE></HEAD>
<BODY>
<%
Esto es VBScript
p=Request.geueryString("P")
if p = ("0" or p = "") then
%>

<FORM METHOD="POST" ACTION=ej.asp?P=2">
Introduzca su dirección de correo:
<INPUT TYPE=text NAME="NOMBRE" SIZE=23 MAXLENGTH=9>
<P>
<INPUT TYPE="submit" VALUE="Aceptar" NAME="ACEPTAR">
<INPUT TYPE="reset" VALUE="Borrar" NAME="BORRAR">

<%else%>

Su dirección es <%= Request.Form("NOMBRE")%>
<%end if%>

<A HREF="ejl.asp?P=0">[Reiniciar]</A>
</FORM>
</BODY>
</HTML>
```

Conexión Web con Bases de Datos Access

Para proporcionar acceso a una base de datos SQL desde su página Web, se necesitacrear un archivo del Conector de bases de datos de Internet (extensión de archivo .idc) y un archivo de extensión HTML (extensión de archivo .htm).

Este punto está íntegramente obtenido del manual de IIS de Microsoft®. Para más información sobre el ODBC (Open DataBase Connectivity), configuración del servidor con las Base de Datos, consultar el manual.

Ejemplo de una consulta

Este ejemplo comienza con una sencilla página Web llamada `Sample.htm`. Dicha página contiene un vínculo que dará como resultado la ejecución de una consulta utilizando el controlador ODBC para Microsoft SQL Server, cuyos resultados se devuelven como otra página Web. La ilustración siguiente muestra `Dbssampl.htm` tal como aparece en Microsoft Internet Explorer (suponiendo que Internet Information Server se ha instalado en un equipo llamado "servidorweb").

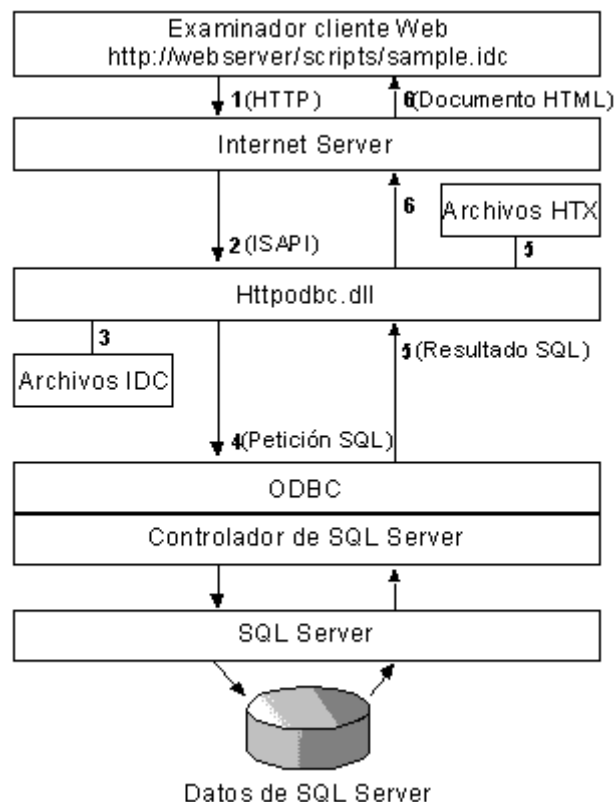


Cuando se hace clic en el vínculo "Haga clic aquí para ejecutar la consulta", se envía al servidor otra dirección URL. La dirección URL precede al texto del vínculo (tiene formato de texto oculto):

```
<A
  HREF="http://servidorweb/samples/dbsamp/dbsamp1.idc">H
  aga clic aquí para ejecutar la consulta</A>
```

En la dirección URL ya se ha hecho referencia al archivo del Conector de bases de datos de Internet que va a utilizar el IDC (Dbsamp1.idc). La asignación de un archivo de extensión hace innecesario que se haga referencia a Httpodbc.dll en la dirección URL.

En IIS (Internet Information Server), todo el proceso de utilizar el Conector de bases de datos de Internet para este ejemplo se realiza en seis pasos, tal como se muestra en el siguiente diagrama.



1. Internet Information Server.

El explorador de Web es quien envía la dirección URL.

2. Internet Information Server carga Httpodbc.dll y le suministra la información restante de la dirección URL.

Los archivos .Idc se asignan a Httpodbc.dll. Httpodbc.dll se carga y obtiene el nombre del archivo del Conector de bases de datos de Internet (y otros elementos) de la dirección URL pasada a Internet Information Server.

3. Httpodbc.dll lee el archivo del Conector de bases de datos de Internet.

El archivo del Conector de bases de datos de Internet contiene varias entradas con el formato

campo: valor

En el archivo Ejemplo.idc, el origen de datos ODBC se especifica mediante:

Origen de datos: Web SQL

Y el archivo de extensión HTML se especifica mediante:

Plantilla: sample.htx

Este es todo el contenido del archivo .idc al que se hace referencia en la dirección URL:

```
Origen de datos: Web SQL
Nombre de usuario: sa
Plantilla: sample.htx
Instrucción SQL:
+SELECT au_lname, ytd_sales
+ from pubs.dbo.titleview
+ where ytd_sales>5000
```

En el archivo .idc de ejemplo el nombre del origen de datos es "Web SQL". Las notas de instalación de ODBC le indican cómo crear un origen de datos llamado Web SQL.

Los demás elementos contenidos en el archivo .idc de ejemplo incluyen:

Nombre de usuario, que debe ser una conexión válida con el origen de datos ODBC; en este ejemplo, la conexión es con la cuenta "sa" en un Microsoft SQL Server.

Plantilla, que especifica el archivo que se va a utilizar para combinar los resultados.

Instrucciones SQL, que contiene la instrucción SQL que se va a ejecutar.

Para obtener las definiciones de todos los campos que pueden especificarse en el Conector de bases de datos de Internet, consulte "Aprender las características del Conector de bases de datos de Internet", más adelante en este mismo capítulo.

La instrucción SQL de Sample.idc devuelve los apellidos de todos los autores y las ventas del año hasta la fecha, en unidades, de la base de datos de ejemplo "pubs" de SQL Server para aquellos autores cuyos libros tengan unas ventas anuales de más de 5000 dólares.

4. El IDC conecta con el origen de datos ODBC y ejecuta la instrucción SQL contenida en el archivo del Conector de bases de datos de Internet.

Se ha realizado la conexión con el origen de datos ODBC mediante el IDC, que en este ejemplo carga el controlador ODBC de SQL Server y se conecta con el servidor especificado en la definición del origen de datos. Una vez realizada la conexión, la instrucción SQL del archivo del Conector de bases de datos de Internet se envía al controlador ODBC de SQL Server, que a su vez la envía a SQL Server.

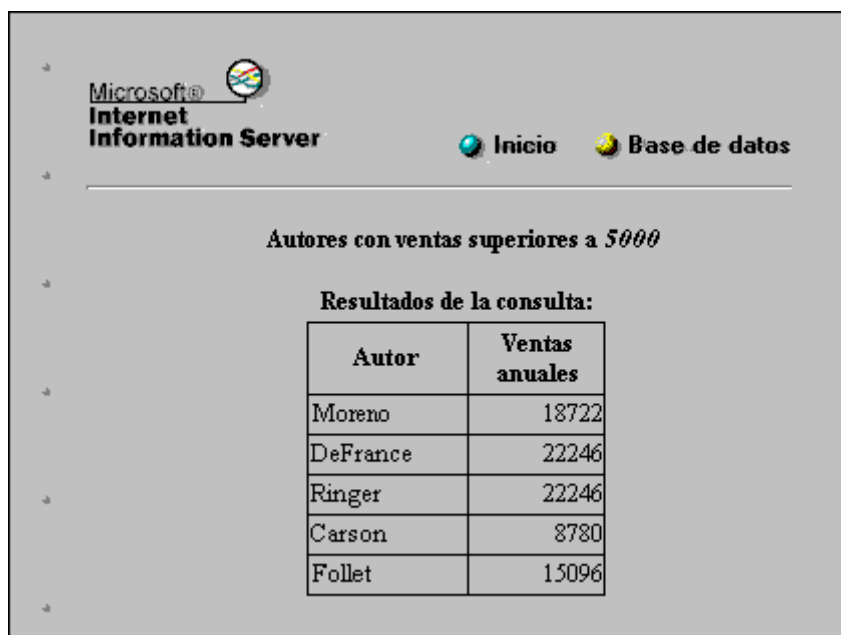
5. El IDC recopila los datos de la base de datos y los combina en el archivo de extensión HTML.

Tras la ejecución de la instrucción SQL, IDC lee el archivo de extensión HTML especificado en Sample.idc (Sample.htx). Los archivos de extensión HTML (.htx) contienen etiquetas especiales de HTML utilizadas por IDC para controlar dónde y cómo se combinan los datos devueltos por la instrucción SQL.

6. El IDC devuelve el documento combinado a Internet Information Server que, a su vez, lo devuelve al cliente.

Después de que hayan combinado todos los datos en Sample.htx, el documento HTML ya completo se devuelve al cliente.

La página Web resultante de este proceso aparece en Microsoft Internet Explorer como se muestra a continuación.



The screenshot shows the Microsoft Internet Information Server interface. At the top, there is a logo and the text "Microsoft® Internet Information Server". To the right, there are two buttons: "Inicio" (Home) and "Base de datos" (Database). Below this, the text "Autores con ventas superiores a 5000" is displayed. Underneath, the heading "Resultados de la consulta:" is shown. A table with two columns, "Autor" and "Ventas anuales", contains the following data:

Autor	Ventas anuales
Moreno	18722
DeFrance	22246
Ringer	22246
Carson	8780
Follet	15096

Descripción del archivo Sample.htx

Para devolver los datos al cliente de WWW, el archivo .idc combina el archivo de extensión HTML, .htx, y los datos de ODBC. Tras ello los datos combinados se adjuntan a encabezados HTTP estándar (200 estado OK, tipo de contenido, etc.), pasan al servicio WWW y se devuelven al cliente.

El archivo .htx es un documento HTML con algunas etiquetas adicionales que van entre <%%> o <!--%%-->, que el archivo .idc utiliza para agregar datos dinámicos al

documento. El formato HTML del archivo .htx suele dar formato a los datos que se devuelven. Hay seis palabras claves (`begindetail`, `enddetail`, `if`, `else`, `endif` y `%z`) que controlan la forma en que se combinan los datos de la base de datos y el formato HTML en el archivo .htx. Los nombres de las columnas de la base de datos especifican cuáles son los datos que se devuelven en el documento HTML. Por ejemplo, la línea siguiente de un archivo .htx combina datos de la columna `Nombrecorreoelectrónico` de todos los registros procesados:

```
<%begindetail%><%Nombrecorreoelectronico%><%enddetail%>
```

El archivo `Sample.htx` es un documento HTML que contiene etiquetas del Conector de bases de datos de Internet para los datos devueltos por la base de datos (por motivos de claridad, las etiquetas aparecen en negrita). Para resaltar las etiquetas del IDC se ha eliminado una parte del formato HTML.

Para que el ejemplo sea más claro, se ha eliminado casi todo el formato HTML.

```
<HTML>
<BODY>
<HEAD><TITLE>Autores y ventas anuales</TITLE></HEAD>
<%if idc.sales eq ""%>
<H2>Autores con ventas superiores a<I>5000</I></H2>
<%else%>
<H2> Autores con ventas superiores a<I><%idc.sales%></I></H2>
<%endif%>
<P>
<%begindetail%>
<%if CurrentRecord EQ 0 %>
Resultados de la consulta:
<B>Ventas anuales del autor<BR></B>
<%endif%>
<%au_lname%><%ytd_sales%>
<%enddetail%>
<P>
<%if CurrentRecord EQ 0 %>
<I><B>Ninguno de los autores tiene unas ventas anuales
superiores a </I><%idc.sales%>.</B>
<P>
<%else%>
<HR>
<I>
La página Web que verá aquí se creó combinando los resultados de
la consulta SQL y el archivo de plantilla Sample.htx.
<P>
La combinación la realizó el Conector de bases de datos de
Internet de Microsoft y Microsoft Internet Information Server
devolvió los resultados a este explorador de Web.
</I>
<%endif%>
</BODY>
</HTML>
```

Las secciones `<%begindetail%>` y `<%enddetail%>` delimitan el lugar del documento donde aparecerán las filas devueltas por la base de datos. Las columnas devueltas por la consulta aparecerán rodeadas por `<%%>`; en este ejemplo son `<%au_lname%>` y `<%ytd_sales%>`.

Características del Conector de bases de datos de Internet

El Conector de bases de datos de Internet tiene varias características que facilitan la creación de páginas Web que contengan datos de una base de datos.

Archivos del Conector de bases de datos de Internet (.idc)

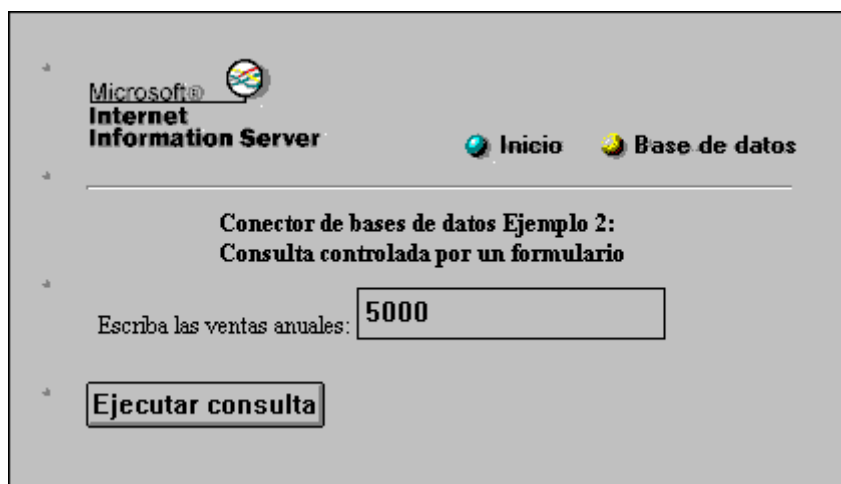
Los archivos del Conector de bases de datos de Internet contienen la información que se utiliza para tener acceso a la base de datos. La siguiente sección describe las características de los archivos del Conector de bases de datos de Internet.

Parámetros

El ejemplo de la sección anterior sólo muestra el tipo de consulta más sencillo, una consulta que se ha definido completamente en un archivo del Conector de bases de datos de Internet. Aunque este tipo de consulta es útil, pueden crearse páginas Web aún más potentes mediante la utilización de parámetros. Los parámetros son los nombres y valores de los controles del formulario de HTML, como por ejemplo `<INPUT...>`, así como los nombres especificados directamente en las direcciones URL. Estos nombres y valores los envían los exploradores de Web y pueden utilizarse en instrucciones SQL del servidor.

Por ejemplo, en la última sección, la consulta de `Sample.idc` sólo ha devuelto aquellos autores cuyas ventas anuales superan 5000. Utilizando un parámetro, podría crear una página Web que preguntara al usuario si desea cambiar el número 5000.

La página Web debe solicitar al usuario la cifra de ventas anuales y, a continuación, dar el nombre "sales" a la variable asociada. `Dbsamp2.htm` muestra un formulario con un campo de entrada que se utiliza para obtener el número:



Microsoft®
Internet
Information Server

Inicio Base de datos

**Conector de bases de datos Ejemplo 2:
Consulta controlada por un formulario**

Escriba las ventas anuales:

La sintaxis del campo de entrada y el botón de `Sample2.htm` en formato HTML es la siguiente:

```
<FORM METHOD="POST" ACTION="/scripts/samples/sample2.idc">
<P>
Escriba las ventas anuales hasta la fecha: <INPUT NAME="sales"
VALUE="5000" >
<P>
<INPUT TYPE="SUBMIT" VALUE="Ejecutar consulta">
</FORM>
```

En el archivo `Sample2.idc` del Conector de bases de datos de Internet se utiliza el parámetro que aparece en **negrita** en lugar del número 5000:


```
SQLStatement:
+SELECT au_lname, ytd_sales
+ from pubs.dbo.titleview
+ where ytd_sales > %sales%
```

Aquí el nombre del parámetro debe ser “sales” para que corresponda a <INPUT NAME= “sales” ...> de la página Web. Los parámetros deben estar entre signos de porcentaje (%) para distinguirlos de un identificador normal de SQL. Cuando el Conector de bases de datos de Internet encuentra el parámetro en el archivo .idc, sustituye el valor enviado por el explorador de Web y, posteriormente, envía la instrucción SQL al controlador ODBC.

El signo de porcentaje (%) es también un carácter comodín de SQL. Los comodines se utilizan en las consultas SQL para buscar un elemento de una tabla que contenga determinados caracteres. Para insertar un único signo “%” en un comodín SQL, utilice “%%”. Esto evita que el IDC intente utilizar % como marcador de parámetro. Por ejemplo:

```
SQLStatement:
+SELECT au_lname, ytd_sales, title
+ from pubs.dbo.titleview
+ where title like '%%título%%'
```

Para que un signo de porcentaje se reconozca como un comodín de SQL debe especificarlo dos veces y, a continuación, agregar los caracteres de porcentaje alrededor del parámetro para distinguir la cadena como parámetro. En el ejemplo, la consulta busca la palabra *título* en todas las entradas de la columna de títulos. Esta consulta devuelve lo siguiente:

```
título
título y hecho
página del título principal
autor y título
```

Para devolver todas las entradas que contienen la palabra *título* en las seis primeras letras, debe dar el siguiente formato a las consultas:

```
SQLStatement:
+SELECT au_lname, ytd_sales, title
+ from pubs.dbo.titleview
+ where title like '%título%%'
```

En este ejemplo se devuelven los siguientes resultados:

```
título
título y hecho
```

Para devolver todas las entradas que contengan la palabra *título* como las seis últimas letras, debe dar el siguiente formato a las consultas:

```
SQLStatement:
+SELECT au_lname, ytd_sales, title
+ from pubs.dbo.titleview
+ where title like '%%título%'
```

En este ejemplo se devuelven los siguientes resultados:

```
título
autor y título
```

Es posible crear potentes conjuntos de páginas Web utilizando el resultado de una consulta para proporcionar vínculos con otras consultas. Por ejemplo, para mostrar los títulos de un autor individual, en lugar de devolver el nombre del autor como texto normal, puede darle formato como un vínculo y, a continuación, utilizar el vínculo para realizar otra consulta.

Campos de los archivos del Conector de bases de datos de Internet (.idc)

En las siguientes tablas aparece una lista con los campos que pueden especificarse en un archivo del Conector de bases de datos de Internet. Observe que los parámetros o las variables del servidor pueden aparecer en cualquier lugar del archivo .idc.

Campos necesarios en un archivo del Conector de bases de datos de Internet (.idc)

Campo	Descripción
Datasource	El nombre que corresponde al Nombre del origen de datos (DSN) del sistema ODBC que ha creado previamente utilizando el Administrador de ODBC o la herramienta proporcionada con los ejemplos.
Template	El nombre del archivo con extensión HTML que da formato a los datos devueltos por esta consulta. Por convención, estos archivos utilizan la extensión .htx.
SQLStatement	La instrucción SQL que se va a ejecutar. La instrucción SQL puede contener valores de parámetros, que deben ir entre signos de porcentaje (%), del cliente. En el archivo del Conector de bases de datos de Internet, la instrucción SQL puede ocupar varias líneas. Después del campo SQLStatement, todas las líneas que comiencen con un signo más (+) se considerarán parte del campo SQLStatement. En el mismo archivo pueden aparecer varias instrucciones SQL.

Campos opcionales en un archivo del Conector de bases de datos de Internet (.idc)

Campo	Descripción
DefaultParameters = <i>parám=valor</i> [, <i>parám=valor</i>] [...]	Los valores de los parámetros, si hay alguno, que se van a utilizar en el archivo del Conector de bases de datos de Internet siempre que el cliente no especifique algún parámetro.
Expires	El número de segundos que hay que esperar antes de actualizar una página con salida en caché. Si una petición posterior es idéntica, la página de caché se devolverá sin tener acceso a la base de datos. Este campo es útil cuando desee forzar una nueva consulta de la base de datos tras un determinado periodo de tiempo. De forma predeterminada, el IDC no pone en caché las páginas de salida. Sólo las incluye en caché cuando se utiliza el campo Expires.
MaxFieldSize	El espacio de búfer máximo por campo que asigna el IDC. Los caracteres posteriores a éste se truncarán. El parámetro se aplica sólo a aquellos campos de la base de datos que superen los 8192 bytes. El valor predeterminado es 8192.
MaxRecords	El número máximo de registros que el IDC devolverá de cualquier consulta. El valor <i>MaxRecords</i> no se define de forma predeterminada, lo que significa que una consulta puede devolver hasta 4.000 millones de registros. Defina este valor para limitar los registros devueltos.
ODBCConnection	Inserte este campo con el valor de <i>agrupación</i> para agregar la conexión a la agrupación de conexiones, que conserva la conexión con la base de datos abierta para futuras peticiones. A continuación, el IDC envía los datos a través de una conexión agrupada para la

	<p>posterior ejecución de un archivo .idc que contiene los mismos valores de Datasource, Username y Password. Defina esta opción para mejorar el rendimiento utilizando el Conector de bases de datos de Internet. Además, hay una opción <i>sin agrupación</i>, que especifica que la conexión del archivo .idc en la que se define esta opción no debe tomarse de la agrupación de conexiones. Para administrar la caché de conexiones con más precisión, defina el valor de este campo como nopool. Además, si hay un límite en cuanto al número de conexiones actuales, probablemente no desee que la agrupación de conexiones monopolice todas las conexiones; de lo contrario, nadie podría conectarse a SQL Server.</p> <p>Nota Para definir el valor predeterminado de la agrupación de conexiones, debe definir como 1 la entrada del Registro PoolIDCConnections</p>
Passwords	La contraseña que corresponde al nombre de usuario. Si no hay ninguna contraseña, este campo puede dejarse en blanco.
RequiredParameters	Los nombres de los parámetros, si hay alguno, que Httpodbc.dll se asegura que pasarán del cliente; de lo contrario, devolverá un error. Los nombres de los parámetros se separan mediante comas.
Translationfile	<p>La ruta de acceso al archivo que asigna caracteres no ingleses (como à, ô o é) para que los exploradores puedan mostrarlos adecuadamente en formato HTML. Si el archivo de traducción no está en el mismo directorio que el archivo .idc, debe escribir la ruta de acceso completa al archivo de traducción. Sintaxis:</p> <p>Translationfile: C:\nombre_directorio\nombre_archivo. Si publica una base de datos en un idioma que no sea inglés, utilice el campo Translationfile. Un archivo de traducción es un archivo de texto en el que los caracteres especiales se asignan en el siguiente formato: <i>valor=cadena</i><CR>, donde <i>valor</i> es un carácter internacional y <i>cadena</i> es el código de traducción de HTML.</p>
Username	<p>Un nombre de usuario válido para el origen de datos proporcionado por el campo Datasource.</p> <p>Nota Si utiliza Microsoft SQL Server con la opción de seguridad integrada, los campos de nombre de usuario y contraseña del archivo .idc se ignoran. La conexión con SQL Server se realiza utilizando las credenciales del usuario Web. Si la petición se realiza como un usuario anónimo, el nombre de usuario y la contraseña están determinados por la configuración del usuario anónimo (el valor predeterminado es IUSR_nombreequipo) del Administrador de servicios de Internet. Si la petición del cliente contenía credenciales para la conexión, el nombre de usuario y la contraseña proporcionados por el usuario final se utilizan para conectarse a SQL Server.</p>
Content-Type	Cualquier tipo MIME válido que describa qué se devolverá al cliente. Si el archivo .htx contiene HTML, casi siempre será "text/html".

Uso de cuadros de lista de selección múltiple en formularios HTML

Cuando se utiliza un formulario HTML que contiene una etiqueta `<SELECT MULTIPLE...>`, el Conector de bases de datos de Internet convierte los elementos seleccionados en una lista separada por comas; la lista puede utilizarse en el archivo .idc de la misma forma que otros parámetros. No obstante, puesto que el parámetro es realmente una lista, sólo suele utilizarse en instrucciones SQLSelect con una cláusula IN, como en los ejemplos siguientes.

Si el nombre del parámetro del archivo .idc está entre comillas simples, cada elemento de la lista también estará entre comillas simples. Siempre que la columna de la cláusula IN sea una columna de caracteres o de cualquier otro tipo en el que los literales vayan

entre comillas (por ejemplo, fechas y horas) debe escribir el nombre del parámetro entre comillas simples. Si no se encuentran comillas simples alrededor del nombre del parámetro, no se colocará ninguna comilla alrededor de cada elemento de la lista. Si la columna de la cláusula IN es de tipo numérico o de cualquier otro tipo en el que los literales no vayan entre comillas simples, no debe escribir el nombre del parámetro entre comillas simples.

Por ejemplo, si un formulario HTML contenía el cuadro de lista de elección múltiple que aparece a continuación:

```
<SELECT MULTIPLE NAME="región">
<OPTION VALUE="Oeste">
<OPTION VALUE="Este">
<OPTION VALUE="Norte">
<OPTION VALUE="Sur">
</SELECT>
```

Es posible construir un archivo .idc con una instrucción SQL:

```
SQLStatement: SELECT nombre, región FROM cliente WHERE región IN
('%región%')
```

Si el usuario ha seleccionado “Norte”, “Oeste” y “Este” en el formulario HTML, la instrucción SQL se convertiría en:

```
SELECT nombre, región FROM cliente WHERE región IN ('Norte',
'Oeste', 'Este')
```

A continuación aparece otro ejemplo de formulario HTML, pero esta vez se utilizan datos numéricos, por lo que esta vez el parámetro del archivo .idc no aparece entre comillas.

```
<SELECT MULTIPLE NAME="año">
<OPTION VALUE="1994">
<OPTION VALUE="1995">
<OPTION VALUE="1996">
</SELECT>
```

Es posible construir un archivo .idc con una instrucción SQL:

```
SQLStatement: SELECT producto, año_ventas FROM ventas WHERE
año_ventas IN (%año%)
```

Si el usuario ha seleccionado “1994” y “1995” en el formulario HTML, la instrucción SQL se convertiría en:

```
SELECT product, sales_year FROM sales WHERE sales_year IN (1994,
1995)
```

Uso de consultas por lotes y consultas múltiples

En un archivo .idc, es posible agrupar consultas SQL de dos formas distintas, como consultas por lotes o como consultas múltiples.

Consultas por lotes

Si consulta bases de datos que pueden procesar simultáneamente varias consultas en una única instrucción SQL (como una base de datos de SQL Server), debe dar a las instrucciones el formato por lotes con el fin de optimizar el rendimiento. Por ejemplo:

```
SQLStatement:
+insert into perf(testtime, tag) values (getdate(),
'%etiqueta%')
+SELECT au_lname, ytd_sales from pubs.dbo.titleview where
ytd_sales>5000
+SELECT count(*) as nrecs from pubs.dbo.titleview where
ytd_sales>5000
```

Consultas múltiples

Si consulta bases de datos que no pueden procesar una serie de consultas SQL simultáneamente, formule las consultas como consultas múltiples. Por ejemplo:

```
SQLStatement:
+insert into perf(testtime, tag) values (getdate(),
'%etiqueta%')
SQLStatement:
+SELECT au_lname, ytd_sales from pubs.dbo.titleview where
ytd_sales>5000
SQLStatement:
+SELECT count(*) as nrecs from pubs.dbo.titleview where
ytd_sales>5000
```

Las consultas por lotes se procesan juntas inmediatamente, mientras que las consultas múltiples se procesan una a una. Por tanto, si su base de datos puede administrar consultas por lotes, al dar a las consultas un formato por lotes obtendrá mejor rendimiento.

Archivos de extensión (htx) HTML

Los archivos de extensión HTML contienen un número de palabras clave que controlan la construcción del documento HTML de salida. Estas palabras clave se explican en las siguientes secciones.

<%begindetail%>, <%enddetail%>

Las palabras clave <%begindetail%>, <%enddetail%> rodean una sección del archivo de extensión HTML donde se combinarán los archivos de salida de la base de datos. Dentro de la sección, los nombres de columna delimitados por <% y %> o <!-- %--> se utilizan para marcar la posición de los datos devueltos por la consulta. Por ejemplo:

```
<%begindetail%>
<%au_lname%>: <%ytd_sales%>
<%enddetail%>
```

presentará las columnas `au_lname` e `ytd_sales`. Es posible hacer referencia a cualquier columna de este modo. También puede hacerse referencia a los nombres de columna desde cualquier parte de un archivo de extensión HTML.

Nota: Si la consulta no ha devuelto ningún registro, la sección <%begindetail%> se saltará. Por cada instrucción SQL que genere un conjunto de resultados (por ejemplo,

SELECT), debería existir una sección `<%begindetail%>` `<%enddetail%>` correspondiente en el archivo `.htx`.

`<%if%>`, `<%else%>`, `<%endif%>`

Los archivos de extensión pueden contener lógica condicional con una instrucción `if-then-else` para controlar la construcción de la página Web. Por ejemplo, una utilización habitual es insertar una condición para mostrar los resultados de la consulta en la primera fila, dentro de una sección `<%begindetail%>`; en el caso de que la consulta no devuelva ningún registro, se mostrará el texto “No hay ningún autor cuyas ventas anuales sean superiores a” `%idc.sales%`. Mediante la utilización de la instrucción `<%if%>` y una variable incorporada denominada “CurrentRecord” es posible personalizar el resultado para que se imprima el mensaje de error cuando no se devuelva ningún registro. A continuación se ofrece un ejemplo que muestra la utilización de la instrucción `<%if%>`.

`<%begindetail%><%if CurrentRecord EQ 0 %>`

Resultados de la consulta:

```
<B>Author YTD Sales<BR></B>
<%endif%>
<%au_lname%><%ytd_sales%>
<%enddetail%>
<P>
<%if CurrentRecord EQ 0 %>
<I><B>No hay ningún autor cuyas ventas anuales sean superiores a
</I><%idc.sales%>.</B>
<P>
<%else%>
<HR>
<I>
La página Web que está viendo se ha creado combinando los
resultados de la consulta SQL con el archivo de plantillas
Sample.htx.
<P>
La combinación la ha realizado el Conector de bases de datos de
Internet de Microsoft y Microsoft Internet Information Server ha
devuelto los datos a este explorador de Web.
</I>
<%endif%>
</BODY>
</HTML>
La sintaxis general es:
<%if condición%>
texto HTML
[<%else%>
texto HTML]
<%endif%>
```

Donde *condición* tiene el formato siguiente:

valor1 *operador* *valor2*

y *operador* puede ser uno de los siguientes:

EQ	si <i>valor1</i> es igual a <i>valor2</i>
LT	si <i>valor1</i> es menor que <i>valor2</i>
GT	si <i>valor1</i> es mayor que <i>valor2</i>

CONTAINS	si cualquier parte de <i>valor1</i> contiene la cadena <i>valor2</i>
----------	--

Los operandos *valor1* y *valor2* pueden ser nombres de columnas, una de las variables incorporadas (vea a continuación CurrentRecord o MaxRecords), un nombre de variable HTTP (vea a continuación) o una constante. Cuando se utilizan en una instrucción `<%if%>`, los valores no están delimitados por `<%` y `%>`. Por ejemplo, para que procese al autor "Pérez" de una forma especial, utilice la condición:

```
<%begindetail%>
<%if au_lname EQ "Pérez"%>
¡Este hombre es perez!
<%endif%>
<%enddetail%>
```

La instrucción `<%if%>` también puede utilizarse para hacer procesos especiales basados en información procedente de variables HTTP. Por ejemplo, para dar un formato diferente a una página basándose en el tipo de explorador de Web del cliente, puede incluir lo siguiente en el archivo de extensión HTML.

```
<%if HTTP_USER_AGENT contains "Maruja"%>
el cliente acepta características avanzadas de HTML
<%else%>
el cliente es <%HTTP_USER_AGENT%>
<%endif%>
```

CurrentRecord, MaxRecords

La variable incorporada CurrentRecord contiene el número de veces que se ha procesado la sección `<%begindetail%>`. La primera vez que se procese la sección `<%begindetail%>`, el valor será cero. Posteriormente, el valor de CurrentRecord cambia cada vez que se toma otro registro de la base de datos.

La variable incorporada MaxRecords contiene el valor del campo MaxRecords del archivo del Conector de bases de datos de Internet. Tanto MaxRecords como CurrentRecord sólo pueden utilizarse en instrucciones `<%if%>`.

Parámetros de los archivos del Conector de bases de datos de Internet

Es posible tener acceso a archivos del Conector de bases de datos de Internet en el archivo de extensión HTML colocando el prefijo "idc" y un punto delante del nombre del parámetro. En Sample3.htx, mostrado anteriormente en este mismo capítulo, puede mostrar el valor del parámetro %sales% incluyendo la línea:

El valor del parámetro sales es: `<%idc.sales%>`

Variables HTTP

Algunas variables de los archivos de extensión HTML pueden proporcionar una gran cantidad de información acerca del entorno y del cliente Web conectado al servidor. Además, estarán disponibles todos los encabezados enviados por el cliente. Para tener acceso a ellos utilizando el Conector de bases de datos de Internet debe convertirlas:

1. Agregue HTTP_ al principio.
2. Convierta todos los guiones en caracteres de subrayado.
3. Convierta todas las letras a mayúsculas.

Puede consultarse la tabla de variables de entorno para las aplicaciones CGI ya que son las variables HTTP para las aplicaciones IDC.

Seguridad en WWW

Conceptos básicos de seguridad en la Web.

Cuando un usuario de Internet introduce datos en un formulario, esos datos muchas veces deben comunicarse al programa que los procese en el servidor de una forma segura ya que algunos de estos datos son confidenciales, como por ejemplo el número de tarjeta de crédito o el mero hecho de que una persona realice una reserva en un hotel.

El protocolo de seguridad más extendido es SSL (Secure Socket Layer), y asegura:

- Autenticidad, nadie puede hacerse pasar por nosotros.
- Privacidad, nadie puede leer nuestros datos.
- Integridad, nadie puede modificar los datos que nosotros enviamos.

A nivel más técnico, SSL se sitúa, en los niveles de red, entre el protocolo de aplicación HTTP y por encima del protocolo de conexión TCP/IP. Esto permite a SSL operar con distintas aplicaciones como FTP, Telnet, ...

Este protocolo utiliza encriptación por clave pública y clave privada y utiliza el algoritmo RC4 desarrollado por la empresa RSA Data Security Inc.

Existe un problema, el gobierno americano no permite exportar algoritmos de encriptación fuera de EE.UU. y Canadá, ya que los considera armas secretas. Este algoritmo utiliza una clave de 128 bits y lo que sí permite es exportar este mismo algoritmo pero utilizando una clave de 40 bits. Por lo tanto todos los usuarios de SSL de EE.UU. y Canadá utilizan 128 bits y los demás 40 bits.

¿Es suficiente esta clave?, según Netscape se necesita un computador que proporcione 64 MIPS -un Pentium a 100 MHz proporciona 50 MIPS- trabajando un año para saltar una clave de 40 bits, lo cual parece suficientemente disuasorio, sin embargo existen computadores más potentes que pueden realizar ese trabajo en mucho menos tiempo.

La prueba más rotunda de que la clave de 40 bits no es suficiente es que la propia empresa Netscape vende su software por Internet, y mientras los usuarios americanos y canadienses pueden enviar su número de tarjeta de crédito a través de Internet, el resto de usuarios lo deben hacer por teléfono.

Servidores Seguros

Un servidor seguro es un servidor que establece una conexión con el cliente, de manera que la información circula a través de Internet encriptada mediante algoritmos que aseguran que sea inteligible solo para el servidor y el visualizador que accede al Web. Un servidor seguro es la plataforma necesaria que permite establecer sobre servicios que conlleven transferencia de información cuya confidencialidad ha de ser protegida.

Es un requisito imprescindible para el establecimiento de servicios de banca electrónica o de comercio electrónico.

Si hemos conectado con un servidor seguro. En primer lugar, la dirección de URL comienza por `https://` en vez de `http://` (a esta dirección se accede, a veces, sin intervención del usuario, debido a que se pulsa una palabra clave que la lleva incorporada, o bien intencionadamente cuando se desea acceder a un servidor en modalidad segura). Además, en la mayoría de los visualizadores tendremos una indicación de que la conexión segura se ha establecido. Así, por ejemplo con el visualizador de Netscape se presentan las siguientes indicaciones:

- La llave de la parte inferior izquierda, que habitualmente aparece partida se ve completa.
- Aparece una línea azul en el límite superior del área de visualización.

La información del documento incluye los datos del certificado. Según la versión de Netscape, esta información está en la sección File o View.

Otros browsers presentan iconos distintos para mostrar conexiones seguras, como el Microsoft Internet Explorer.

Funcionamiento de un servidor seguro

Un cliente accede a la dirección del web seguro a través de la URL correspondiente, como por ejemplo, `https://www.seguro.com`. Una vez establecida la conexión, el visualizador solicita una conexión segura. Si el servidor a que se accede es un servidor seguro, responderá afirmativamente a la solicitud, enviándole un certificado electrónico de tipo RSA. Tras recibir este certificado, el visualizador lo desempaqueta con la clave de la autoridad de certificación, ya integrada en el software, obteniendo de este modo la clave según el algoritmo RSA. Por último, el cliente genera una clave de encriptación simétrica según el algoritmo RC4 y se la envía encriptada al servidor (con su clave pública). A partir de este momento, tanto el cliente como el servidor pueden establecer una comunicación segura basada en esta clave simétrica, que ambos, y sólo ellos conocen.

Las claves simétricas son generadas aleatoriamente en cada sesión, por lo cual no hay posibilidad de que estas sean conocidas por eventuales hackers.

La forma de funcionar es el protocolo SSL que se explica en el siguiente apartado.

Secure Sockets Layer (SSL)

En esta sección se describen los protocolos que usan la criptografía para proteger las transmisiones de datos desde y hacia su servidor.

Secure Sockets Layer (SSL) es un protocolo para proporcionar seguridad de datos por niveles entre sus protocolos de servicio HTTP y TCP/IP. Este protocolo de seguridad, proporciona codificación de datos, autenticación de servidor e integridad de mensajes para una conexión TCP/IP.

SSL es un protocolo propuesto por Netscape Communications Corporation y enviado al grupo de trabajo W3C acerca de la seguridad para su consideración como método de seguridad estándar para exploradores de Web y servidores de Internet. SSL proporciona un "protocolo de intercambio" de seguridad que se utiliza para iniciar la conexión TCP/IP. Este protocolo de intercambio tiene como resultado que el cliente y el servidor se pongan de acuerdo sobre el nivel de seguridad que utilizarán y lleva a cabo

cualquier requisito de autenticación para la conexión. A partir de entonces, el único papel de SSL es codificar y decodificar la corriente de bytes del protocolo de aplicación que se esté utilizando (por ejemplo, HTTP). Esto significa que toda la información tanto en la petición HTTP como en la respuesta HTTP está totalmente codificada, incluyendo la dirección URL que está solicitando el cliente, el contenido de cualquier formulario (como por ejemplo números de tarjetas de crédito), cualquier información de autorización de acceso HTTP (nombres de usuario y contraseñas) y todos los datos devueltos por el servidor al cliente.

Un servidor con SSL habilitado puede enviar y recibir comunicaciones privadas por Internet a clientes con SSL habilitado.

Las transmisiones codificadas SSL son más lentas que las que no lo están. Para evitar reducir el rendimiento de todo su sitio, considere la posibilidad de utilizar SSL sólo para carpetas virtuales que contienen información muy delicada, como por ejemplo el envío de un formulario que contenga información acerca de tarjetas de crédito.

La activación de la seguridad SSL en un servidor de Web requiere los siguientes pasos:

1. Generar un archivo de pares de claves y un archivo de petición.
2. Solicitar un certificado a una autoridad de certificación.
3. Instalar el certificado en el servidor.
4. Activar la seguridad SSL en una carpeta de servicio WWW.

Importante: Al activar la seguridad SSL, tenga en cuenta lo siguiente:

- Puede activar la seguridad SSL en la raíz de su sitio Web o en una o varias carpetas virtuales.
- Una vez activado y correctamente configurado, sólo los clientes con SSL activado podrán comunicarse con las carpetas WWW con SSL activado.
- Las direcciones URL que señalen a documentos en carpetas WWW con SSL activado tienen que usar “https://” en lugar de “http://” en la dirección URL. Todos los vínculos que usen “http://” en la dirección URL no funcionarán sobre una carpeta protegida.

Instalación de SSL

Generación de un par de claves

Como parte del proceso de habilitar la seguridad Secure Sockets Layer (SSL) en su servidor de Web, necesita generar un par de claves y, a continuación, adquirir un certificado SSL. La nueva aplicación Administrador de claves (instalada con el producto y ubicada en el grupo de programas Internet Server) simplifica este procedimiento.

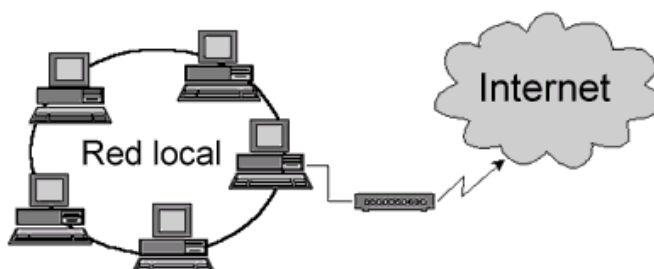
Adquisición de un certificado

La clave generada por el Administrador de claves no es válida para su utilización en Internet hasta que obtenga para ella un certificado válido de una Autoridad de certificación como por ejemplo VeriSign. Envíe el archivo de petición del certificado a la Autoridad de certificación para obtener un certificado válido. Hasta que haga esto, la clave existirá en su equipo host, pero no se podrá utilizar. Para obtener instrucciones acerca de la adquisición de un certificado de VeriSign, consulte el sitio Web de VeriSign en <http://www.verisign.com/>.

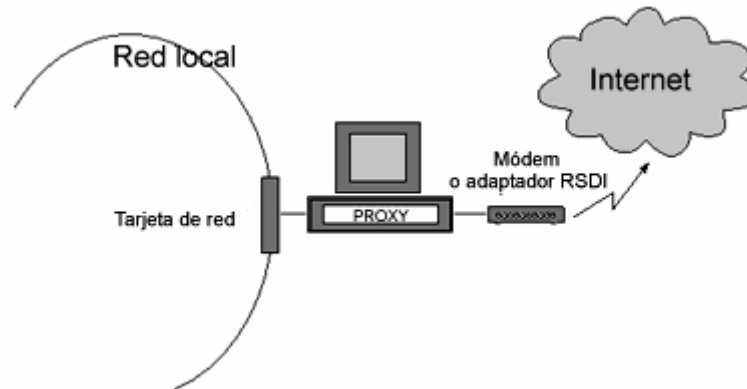
Tras haber realizado la petición de su certificado, recibirá un certificado firmado de la Autoridad de certificación (consulte con su Autoridad de certificación para obtener todos los detalles). Después hay que instalar el par de claves, que en IIS de Microsoft se hace con el programa administrador de claves y creará un archivo similar al del siguiente ejemplo:

Proxy

Imagine la siguiente situación: Se dispone de una red local en la cual varios ordenadores están conectados entre sí. Uno de ellos tiene además una conexión telefónica con Internet, ya sea propia o a través de un proveedor de acceso.



La solución más sencilla consiste en emplear un proxy, que es un programa que actúa como pasarela (gateway) entre redes. En este caso conecta la red local e Internet. Cada vez que alguno de los ordenadores de la red intenta acceder a Internet, realiza esta



petición al proxy, que utiliza la única conexión existente para transmitir la información de Internet al ordenador que la solicita.

Funcionamiento de un Proxy

Desde el punto de vista del usuario de la red local, el sistema funciona como si tuviera realmente un acceso directo a Internet. El usuario accede inmediatamente desde su ordenador a una página Web o recibe su correo electrónico sin siquiera saber que el proxy existe.

En realidad al abrir un programa como Netscape o solicitar el correo pendiente la petición se realiza al proxy, no al servidor de Internet. El programa Proxy es el encargado de redireccionar estas peticiones al ordenador correspondiente (el servidor de la página Web o el servidor de correo) y una vez recibida la información transmitirla al ordenador que la solicitó.

Ventajas de un Proxy

El programa Proxy ofrece las siguientes ventajas en una red local:

- Menor coste: El programa y la instalación tienen un precio mucho menor que cualquier router.
- Una sola línea telefónica: Sólo es preciso disponer de una línea telefónica normal o RDSI.
- Fácil instalación: La instalación emplea los dispositivos de la propia red local, por lo que se reduce a la configuración de los programas. El asistente de configuración de Proxy simplifica la introducción de los datos necesarios.
- Seguridad: El proxy también actúa como una barrera (firewall) adicional para limitar el acceso a la red local desde el exterior.
- Dirección IP única: El número IP es el que identifica a un ordenador en Internet. Si se utiliza un proxy basta un IP para toda la red local en lugar de un IP para cada uno de los ordenadores.
- Conexión automática (autodialing): No es necesario que el ordenador que actúa como proxy esté conectado permanentemente a Internet. Con esta función cada vez que un usuario realiza una petición el proxy establece la conexión. Del mismo modo se desconecta cuando no hay ninguna petición, todo ello automáticamente.

Cortafuegos / Firewall

Un cortafuegos es un dispositivo lógico que protege una red privada del resto de la red (público). Funcionan así:

1. Se coge un ordenador con capacidad de encaminar (por ejemplo un PC con Linux)
2. Se le ponen dos interfaces (por ejemplo interfaces serie, o ethernet, o de paso de testigo en anillo (Token Ring), etc...)
3. Se le deshabilita el reenvío de paquetes IP (IP forwarding)
4. Se conecta una interfaz a la Internet
5. Se conecta la otra interfaz a la red que se quiere proteger

Ahora hay dos redes distintas que comparten un ordenador. El cortafuegos puede comunicarse tanto con la red protegida como con la Internet. La red protegida no puede comunicarse con la Internet, y la Internet no puede comunicarse con la red protegida, dado que hemos deshabilitado el reenvío IP en el único ordenador que las conecta.

Si se quiere llegar a la Internet desde la red protegida, hay que hacer primero un telnet al cortafuegos, y acceder a la Internet desde él. Del mismo modo, para acceder a la red protegida desde la Internet, se debe antes pasar por el cortafuegos.

Inconvenientes de los Cortafuegos

El mayor problema de los cortafuegos es que restringen mucho el acceso a la Internet desde la red protegida. Básicamente, reducen el uso de la Internet al que se podría hacer desde un terminal. Tener que entrar en el cortafuegos y desde allí realizar todo el acceso a Internet es una restricción muy seria. Programas como Netscape, que requieren una conexión directa con la Internet, no funcionan desde detrás de un cortafuegos. La solución a todos estos problemas es un Servidor Proxy.

La solución más directa para aumentar la seguridad es crear una zona de acceso interno accesible solo para los ordenadores de la LAN, es decir, colocar el servidor Web detrás del cortafuegos:

```
Otros ordenadores LAN
      |
Servidor Web <-----> Cortafuegos<-----> Internet
      |
Otros Ordenadores LAN
```

Sin embargo, si quiere que el resto del mundo acceda a su servidor, deberá poner algo detrás del cortafuegos. Desde el punto de vista de la seguridad de su organización lo mejor es que el servidor Web esté detrás del cortafuegos:

```
Ordenadores LAN <--> Cortafuegos <--> Servidor Web <--> Internet
```

No es bueno que el cortafuegos se ejecute en la misma máquina que el servidor WWW, ya que un problema en el servidor puede comprometer la seguridad de toda la organización.

Existen en la actualidad soluciones que permiten tener servidores para los usuarios internos (de la organización) y para los que acceden desde fuera, que se pueden configurar y trabajar con derechos específicos.

Más información...

SSL: "The SSL Protocol": <http://home.netscape.com/newsref/std/SSL.html>

RSA: "RSA Data Security Inc.": <http://www.rsa.com/>

Ficheros de Registros. Estadísticas

Los servidores registran información acerca de quiénes han accedido al servidor y a qué información. Estos datos le facilitan el ajuste del servidor, el número de usuarios que suelen tener acceso al sitio, la evaluación del contenido y las auditorías de seguridad, etc.

Almacenen de registros

La elección del almacén de datos, es decir, donde queremos que el servidor guarde las entradas que se efectúan en el servidor, puede ser en un sistema de ficheros -lo más común hasta ahora) o en una base de datos, como se permite en algunos servidores como IIS mediante Microsoft SQL Server. Esos ficheros de registros (log files) son creados por el servidor para almacenar datos referentes a la transmisión. Cada acceso queda registrado con la información necesaria para conocer cualquier característica que conlleva la transmisión.

Cuando la información se guarda en archivos, los formatos de los archivos pueden variar pero los más comunes son: Formato de archivo de registro "común" del National Center for Supercomputing (NCSA) - el más utilizado-, el formato del European Microsoft Windows NT Academic Centre (EMWAC) o el formato del Internet Information Server de Microsoft (IIS).

Lectura de los archivos de registro

Todos los formatos son parecidos, por ello a continuación se explica el más usado, el formato del archivo de registro común ("common" log file format -NCSA-), en el cual, cada línea del archivo se compone de:

Host	rfc931	authuser	[DD/Mon/YYYY:hh:mm:ss]	"petición"	ddd	bbbb
------	--------	----------	------------------------	------------	-----	------

host : El nombre o el número IP del *host* del cliente.

rfc931 : cualquier información devuelta por *identd* para esa persona, si no -.

Authuser : cuando el usuario envía un *userid* , para autenticarse, si no -.

dd:Mon:YYYY: fecha.

hh:mm:ss : hora en formato de 24 horas.

Petición : La primera línea de la petición HTTP enviada por el cliente.

ddd : código de estado devuelto por el servidor, - si no es posible.

bbbb : número total de bytes enviados, sin incluir la cabecera HTTP/1.0 ó - si no es aplicable.

Ejemplo de un fichero de registro (access log file) en formato común:

```
sungeet.isi.edu - - [01/Apr/1997:02:14:57 -0100] "GET /lsi/BIWIT97/new.gif HTTP/1.0" 200 901
green12.nitco.com - - [01/Apr/1997:04:15:39 -0100] "GET / HTTP/1.0" 200 872
green12.nitco.com - - [01/Apr/1997:04:15:49 -0100] "GET /lsi/imagenes/upv.gif HTTP/1.0" 200 7105
dial15.uio.satnet.net - - [01/Apr/1997:04:21:46 -0100] "GET /lsi/Activities/csc/HTTP/1.0" 200 914
dial15.uio.satnet.net - - [01/Apr/1997:04:22:17 -0100] "GET /lsi/imagenes/upv.gif HTTP/1.0" 200 7105
158.227.112.22 - - [01/Apr/1997:06:36:41 -0100] "GET /robots.txt HTTP/1.0" 404 -
158.227.112.22 - - [01/Apr/1997:06:36:50 -0100] "GET /lsi/Members.shtml HTTP/1.0" 200 1316
```

Programas analizadores de los ficheros de registros

Como puede verse, dada la gran cantidad de información que se guarda y la forma de almacenarla en el fichero hace difícil sacar conclusiones tras un estudio del mismo. Por ello se recurre a programas que dan una visión digerida de esos datos, los cuales realizan unos procesos de filtrado y clasificación de los datos. El resultado de esta operación es la generación de páginas HTML o gráficos, que contienen las estadísticas clasificadas según se le haya indicado inicialmente al programa (día, lugar de acceso, hora, ...).

Estos analizadores en una empresa pueden ayudar en tareas como:

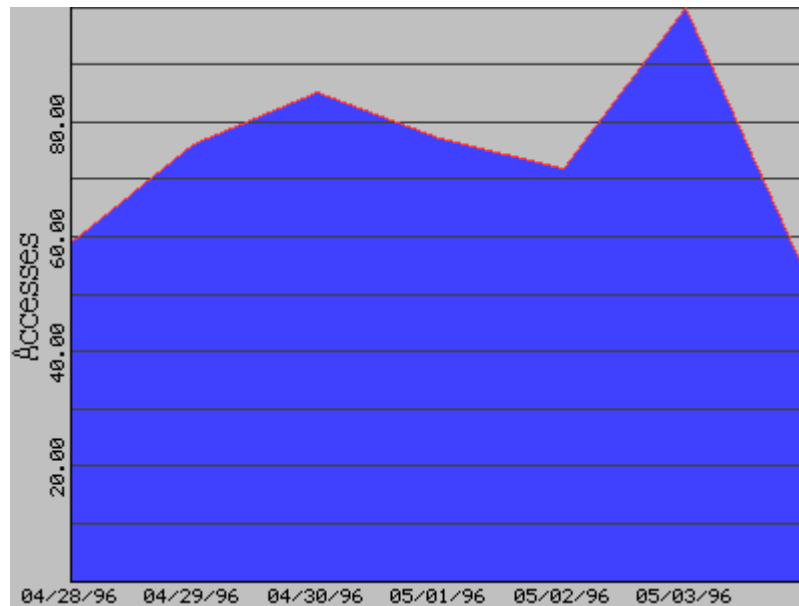
- Estudios de mercado para futuras ofertas. Saber hacia que sector del espacio dirigir los esfuerzos en cuanto a ofertas a realizar, idiomas en los que hay que dirigirse, etc.
- Capacidad de captar visitantes. Análisis de la cantidad de accesos que se llevan a cabo a las primeras páginas de un servicio y cuántos acceden a la totalidad de las páginas.
- Análisis del tipo de visitante / cliente. Conociendo el tipo de visitante de las páginas (universitario, empresario, particular, ...) se puede orientar el producto ofertado para, de esta forma, adecuarnos a los clientes.
- etc.

Algunos de estos programas se nombran a continuación:

- wwwstat
- gwstat
- wusage
- getstats
- MacHTTP Tuil
- WebStat

Ejemplos

Ejemplos de salida con wusage, este programa genera los resultados en gráficos:



Ejemplo de salida con wwwstat, este genera los resultados en una página HTML de la siguiente manera:

LSI Web Access Statistics

Last updated: Wed, 02 Apr 1997 00:01:01 (GMT -0100)

[Daily Transmission Statistics](#)
[Hourly Transmission Statistics](#)
[Total Transfers by Client Domain](#)
[Total Transfers by Reversed Subdomain](#)
[Total Transfers from each Archive Section](#)
[Previous Full Summary Period](#)

Totals for Summary Period: Apr 1 1997 to Apr 1 1997

Files Transmitted During Summary Period	99
Bytes Transmitted During Summary Period	1020023
Average Files Transmitted Daily	99
Average Bytes Transmitted Daily	1020023

Daily Transmission Statistics

%Reqs	%Byte	Bytes Sent	Requests	Date
100.0	100.0	1020023	99	Apr 1 1997

Hourly Transmission Statistics

%Reqs	%Byte	Bytes Sent	Requests	Time
2.02	0.90	9192	2	02
2.02	0.90	9192	2	03
5.05	5.29	54001	5	04
2.02	0.17	1757	2	06
1.01	0.02	180	1	08
1.01	0.17	1751	1	09

5.05	2.80	28581	5		10
5.05	3.71	37893	5		11
5.05	21.75	221809	5		12
5.05	3.62	36935	5		14
2.02	0.90	9192	2		15
2.02	1.80	18337	2		16
9.09	0.58	5931	9		17
2.02	0.20	2038	2		18
14.14	12.46	127061	14		19
33.33	44.15	450368	33		20
1.01	0.03	284	1		21
2.02	0.51	5236	2		22
1.01	0.03	285	1		23

Total Transfers by Client Domain					
%Reqs	%Byte	Bytes Sent	Requests		Domain
8.08	6.15	62777	8		br Brazil
2.02	0.90	9192	2		ca Canada
2.02	0.90	9192	2		fr France
1.01	0.33	3389	1		se Sweden
14.14	3.26	33231	14		com US Commercial
2.02	0.90	9192	2		edu US Educational
3.03	3.50	35664	3		net Network
26.26	33.02	336767	26		ehu.es
41.41	51.04	520619	41		unresolved

Total Transfers by Reversed Subdomain					
%Reqs	%Byte	Bytes Sent	Requests		Reversed Subdomain
41.41	51.04	520619	41		Unresolved
8.08	6.15	62777	8		br.usp.larc.kiwi
2.02	0.90	9192	2		ca.uqam.ppp-ext.as6804
10.10	1.29	13137	10		com.atext.crawl1
2.02	0.17	1757	2		com.lycos.lycosidae
2.02	1.80	18337	2		com.nitco.green12
2.02	0.90	9192	2		edu.isi.sungeet
1.01	0.10	1052	1		es.ehu.sc
25.25	32.91	335715	25		es.ehu.si
2.02	0.90	9192	2		fr.jussieu.cicrp.ibm0
3.03	3.50	35664	3		net.satnet.uio.dial15
1.01	0.33	3389	1		se.swipnet.dialup95-5-8

Total Transfers from each Archive Section					
%Reqs	%Byte	Bytes Sent	Requests		Archive Section
1.01	9.96	101598	1		/estadisticas/
1.01	9.96	101598	1		/estadisticas/wwwstats.htm
1.01	0.33	3389	1		
9.09	0.93	9468	9		/index.shtml
1.01	0.80	8111	1		/lsi/Actividades/BIWIT97/
1.01	0.17	1751	1		/lsi/Actividades/Prode/
1.01	0.17	1751	1		/lsi/Activities/Prode/
2.02	0.21	2188	2		/lsi/Activities/csc/
3.03	2.39	24333	3		/lsi/BIWIT97/
3.03	0.32	3243	3		/lsi/BIWIT97/new.gif
1.01	0.15	1496	1		/lsi/Members/Members.shtml
11.11	18.64	190135	11		/lsi/imagenes/upv.gif
2.02	0.14	1399	2		/pacte/
1.01	0.05	548	1		/~jibrogad/Index.class
1.01	0.38	3858	1		/~jippefet/
1.01	0.73	7420	1		/~jippefet/Publi.shtml

1.01	1.80	18354	1		/~jippefet/Tomas.gif
1.01	0.39	3979	1		/~jippefet/Tomas_esp.shtml
6.06	0.30	3068	6		Code 302 Redirected Req
16.16	0.44	4514	16		Code 404 Not Found Req

This summary was generated by wwwstat-1.0

SSI

Server Side Includes

SSI (Server Side Includes) o LiveHTML (como también se denominan en el servidor de ORACLE -Oracle WebServer-) permite crear documentos que proporcionan información adicional al vuelo. Esa información puede ser, por ejemplo, la fecha actual, la última modificación del fichero o su longitud. Incluso en su modo más avanzado, se pueden utilizar para ejecutar CGI scripts.

Hay que considerar varios factores si se opta por permitir esta opción en los servidores. La primera es que el servidor tiene que ir analizando el documento mientras lo manda, lo cual puede tener un coste en el rendimiento del servidor, según sea la carga de este. Por lo tanto, sólo debería utilizarse cuando realmente merezca la pena. Lo segundo, es que hay un riesgo en la seguridad si se permite que a usuarios medios usar comandos como si fueran el usuario “servidor”.

La extensión suele cambiar para que sepa el servidor que en el documento hay una instrucción SSI, que debe de “interpretar”. En los servidores esta extensión suele ser shtml, shtm (NCSA, Apache, etc.) o .stm (Internet Information Server). Si da un nombre al archivo con una extensión .htm o .html, el servidor WWW pasará las instrucciones SSI por alto (ver el siguiente punto). Sin embargo, esas la extensiones (.shtml, .stm,) pueden configurarse por cualquier otra. Aunque por lógica, no es aconsejable que fuesen .html, .htm, lo cual haría que hiciese el análisis a todas las páginas. IIS no permite poner las extensiones .htm o .html para SSI.

Formato SSI.

Todas las directivas tienen el formato de comentarios SGML / HTML dentro del documento. Cada directiva tiene el siguiente formato:

```
<!--#comando etiqueta1="valor1" etiqueta2="valor2" -->
```

Cada comando toma diferentes argumentos, la mayoría de ellos sólo aceptan una etiqueta cada vez. A continuación viene una lista de los comando y sus etiquetas asociadas.

- **config.** La directiva `config` controla varios aspectos del análisis/tratamiento de un fichero. Tiene dos etiquetas válidas:
 - **errmsg** controla que mensaje es enviado de vuelta al cliente si ocurre un error mientras se interpreta el documento. Cuando ocurre un error es añadido en el fichero de registros del servidor.

- `timefmt` determina el formato a usar cuando se proporcionan fechas / horas. Este es una cadena compatible con la biblioteca `strftime` de la mayoría de las versiones de UNIX.
- `sizefmt` determina el formato a usar cuando se muestran tamaños de fichero. Pueden ser `bytes` para formato de bytes (como 1,234,567), o `abbrev` para la versión abreviada que muestra número de kilobytes o megabytes que ocupa el fichero.
- `include` incluirá el texto de un documento en el documento tratado. Cualquier fichero a incluir esta sujeto a los usuales accesos de control. Este comando acepta dos etiquetas:
 - `virtual` una ruta virtual a un documento en un servidor. Se debe acceder a un fichero normal de esta manera, aunque no a un script CGI. Se puede acceder sin embargo, a otro documento interpretado.
 - `file` un fichero dado un camino relativo al directorio actual. Pero “`..`” no puede ser usado ni tampoco rutas absolutas. Como antes se pueden enviar otros documentos interpretados, pero no scripts CGI.
- `echo` muestra el valor de una de las variables definidas debajo. Cualquier fecha está sujeta a la configuración de `timefmt`. La única etiqueta válida a este comando es `var`, cuyo valor es el nombre de la variable que se desea mostrar.
- `fsize` muestra el tamaño del fichero especificado. Las etiquetas válidas son las mismas que con el comando `include`. El formato resultante de este comando esta sujeto al parámetro `sizefmt` del comando `config`.
- `Lastmod` muestra la fecha de la última modificación del fichero especificado, sujeto al formato dado por el parámetro `timefmt` del comando `config`. Las etiquetas válidas son las mismas que con el comando `include`
- `exec` ejecuta un comando del shell o un script CGI. . Suele tener que estar activado en al configuración del servidor para que permita usarlo. Las etiquetas válidas son:
 - `cmd` ejecutará la cadena de caracteres dada usando `/bin/sh` (UNIX). Todas las variables del punto siguiente están definidas y pueden ser usadas en el comando.
 - `cgi` ejecutará el script CGI de la ruta virtual dada e incluirá su salida.

Variables de entorno de SSI.

Hay un cierto número de variables disponibles para los documentos que son analizados / tratados. Estas variables son las del protocolo CGI, y además, están disponibles las siguientes:

- `DOCUMENT_NAME`: El nombre del fichero actual.
- `DOCUMENT_URI`: La ruta virtual al propio documento. Por ejemplo: `/docs/tutorial/foo.shtml`.
- `QUERY_STRING_UNESCAPED`: La pregunta de búsqueda que el cliente envíe. Los caracteres especiales de la shell son marcados con `\`.

- **DATE_LOCAL:** La fecha actual, hora de la zona local. Sujeto al parámetro timefmt del comando config.
- **DATE_GMT:** Igual que DATE_LOCAL pero en formato Greenwich.
- **LAST_MODIFIED:** La última modificación del archivo actual. También sujeto al parámetro timefmt.

Ejemplos

Añadiendo ficheros.

Puede agregar información repetitiva a un archivo HTML justo antes de enviarlo al usuario. Esta característica es interesante cuando se incluye el mismo texto en muchas páginas HTML, como la información de derechos de autor o un vínculo con la página principal, o quizás también muchas veces dentro de la misma página, en los subapartados. El formato es el siguiente:

```
<!--#include file="valor"-->
```

El valor tiene que contener una ruta relativa o la ruta completa, desde el directorio particular del servicio WWW.

Por ejemplo, para incluir un vínculo con la página principal en todos los documentos HTML:

1. Cree el archivo Linkhome.htm, que contiene los códigos HTML que quiere repetir; por ejemplo, un botón que conduzca a su página principal. El archivo contendría código HTML parecido al siguiente:

```
<A HREF="/homepage.htm"><IMG  
SRC="/imagenes/button_h.gif"></A>
```

2. Use la extensión de archivo `.stm` al crear las páginas Web (en lugar de `.htm` o `.html`).
3. En cada archivo `.stm`, use una instrucción **include file** donde quiera que aparezca la información que se repite. Por ejemplo:

```
Puede volver a: <!--#include file="/linkhome.htm"--> en  
cualquier momento
```

Observe que todas las rutas son relativas al directorio particular de WWW y que pueden incluir directorios virtuales.

Inserción de Fecha y hora.

La mayoría de los usos de SSI, está relacionado con fechas y horas. A menos que se quiera aceptar como los pone el servidor por defecto, los comandos de echo deberían ir tras el comando config con el parámetro timefmt como el siguiente:

```
<!--#config timefmt="%A, %B %d, %Y, at %l:%M %p"-->
```

así luego los echo de variables generarán salidas como las siguientes:

```
GMT date/time is <!--# echo var="DATE_GMT" -->
```

```
GMT date/time is Wednesday, March 26, 1997, at 03:45 PM
```

LOCAL date/time is :<!--# echo var="DATE_LOCAL" -->	LOCAL date/time is Wednesday, March 26, 1997, at 10:45 AM
Updated on <!--# flastmod file="test.shtml" -->	Updated on Thursday, March 28, 1996, at 10:43 PM

Tamaños de fichero.

Las directivas #fsize y #flastmod, permiten dar el tamaño y fecha de cualquier documento.

Si se configura con: <!--#config sizefmt="bytes"-->

Tamaño fichero : <!--# fsize file="test.shtml" -->.	Tamaño fichero: 6405 bytes
--	----------------------------

Pero si se configura con: <!--#config sizefmt="abbrev"-->

Tamaño fichero: <!--# fsize file="test.shtml" -->.	Tamaño fichero: 6 Kbytes
---	--------------------------

Uno de los mejores usos de fsize es cuando se quiere dar a un usuario la posibilidad de que vea el tamaño de los ficheros que puede querer llevar a su ordenador, sin tener que mirar y teclear su longitud en las páginas HTML.

Otros ejemplos de echo útiles ...

Nombres de Documentos y rutas

Este documento es <!--#echo var="PATH_TRANSLATED"-->	Este documento es /usr/local/www/htdocs/ejemplos/test.shtml
Su ruta virtual es <!--#echo var="DOCUMENT_URI"-->	Su ruta virtual es /ejemplos/test.shtml

Web browser Utilizado

Web Browser : <!--#echo var="HTTP_USER_AGENT"-->	Web Browser: Mozilla/4.0b4 (Win95; I)
---	--

Información del Servidor

Host: <!--#echo var="REMOTE_HOST"-->	Host: sip113.si.ehu.es (158.227.112.71)
---	--

(<!--#echo var="REMOTE_ADDR"-->)	
Server: <!--#echo var="SERVER_NAME"--> (<!--#echo var="SERVER_SOFTWARE"-->)	Server: www.ji.ehu.es (Oracle Web listener2.0/1.20in 2)

Glosario

API

Application Program Interface; Interface de programación

Applet

Un término de Java para designar las aplicaciones que se ejecutan en un cliente WWW.

Base de Datos Remota

Un SGBD (Sistema de Gestión de Bases de Datos) que se ejecuta en máquina diferente de donde está el servidor Web.

Base de Datos Local

Igual que el anterior, pero en este caso están en la misma máquina.

CGI

Ver Interface CGI.

Cliente

Un programa, como por ejemplo un navegador Web el cual hace peticiones a un servidor y muestra los resultados que este le envía.

Direcciones IP

Cuatro números menores de 255 separados por “.”, que identifican de manera única a un ordenador en Internet.

Domain Name Services (DNS) Servidor de Nombre de Dominio

Mecanismo que identifica nombres de dominio con direcciones IP.

Encriptación

Codificación de los datos que son enviados a través de una red, con el fin de que la información enviada sea ininteligible para terceros.

Extensiones de fichero

Los caracteres que van tras el último punto “.” en el nombre de un fichero. Generalmente indican el tipo de archivo que se trata. Ej. :index.html (extensión html)

Firewall

Un ordenador que regula el acceso entre una red local e Internet y el acceso desde el exterior a la red local.

FTP

File Transfer Protocol. Protocolo para transmitir ficheros a través de Internet.

HTTP

HiperText Transfer Protocol. Protocolo para el envío de documentos hipermedia a través de una red (Internet, intranets).

HTML

El lenguaje HTML (Hypertext Markup Language) es el usado actualmente para escribir documentos hipermedia sobre WWW ó Web. Esta basado en SGML (Standard Generalized Markup Language).

Image Map

Una imagen en una página Web, que tiene asociados enlaces en regiones de la imagen.

Interface CGI

Un estándar para ejecutar aplicaciones a través de un servidor WWW.

ISAPI

API del servidor de Microsoft, Internet Information Server (IIS). Más eficiente que la interface CGI, pero no es un estándar.

Java

Lenguaje desarrollado por Sun Microsystems.

JavaScript

Lenguaje desarrollado por Netscape para emplearlo en sus navegadores y servidores a través de NSAPI.

JDBC

ODBC (Open DataBase Connectivity) para Java.

LiveHTML

Ver Server Side Includes (SSI).

Router

Máquina que conecta dos redes.

Secure Socket Layer (SSL)

Una capa que permite a protocolos con HTTP, FTP, telnet, etc, que sus datos son enviados criptografiados a través de la red.

Servidor

Proceso que espera peticiones de clientes para atenderles.

Server-Parseable file

Fichero que contiene códigos que el servidor interpreta antes de ser enviados al cliente.

Server Side Includes (SSI)

SSI o LiveHTML permite crear documentos que proporcionan información adicional al vuelo, analizando el documento antes de transmitirlo.

SQL (Structured Query Language)

Lenguaje estándar para acceder a Bases de Datos relacionales.

VRML

Virtual Reality Markup Language. Lenguaje de descripción de imágenes en 3D y de realidad virtual.

HTML

HyperText Markup Language

El lenguaje HTML (Hypertext Markup Language) es el usado actualmente para escribir documentos hipermedia sobre WWW ó Web. Esta basado en SGML (Standard Generalized Markup Language) una codificación que permite compartir información mediante ficheros, bien por otros sistemas de publicación, como clientes WWW / editores, o bien, por aplicaciones para envío electrónico, tratamiento de bases de datos, etc.

HTML (HyperText Markup Language)

Lo primero es saber que un documento HTML es un archivo de texto simple (también conocido como ASCII), luego, se puede editar con cualquier editor de texto, aunque existen editores HTML visuales (WYSIWYG) (por ejemplo, HotMetal, , Netscape Composer, FrontPage, ...) que hacen la vida más sencilla, aunque puede resultar útil saber que código HTML generan de antemano.

HTML esta basado en elementos. Los elementos son el componente principal de la estructura de un documento HTML. Algunos ejemplos de elementos son las cabeceras, tablas, párrafos, listas, etc. Las etiquetas HTML pueden contener texto simple, otros elementos o ambos. Este es le modo correcto en que se debería de ver las páginas HTML.

Para denotar los elementos HTML, se usan etiquetas. Las etiquetas HTML consisten en un menor que (<), usado para indicar el comienzo de una etiqueta HTML y un mayor que (>), usado para indicar el fin de la etiqueta HTML. Las etiquetas van generalmente a pares, por ejemplo <H1> y </H1>. Los fin de etiqueta, son como los de comienzo, excepto que contienen un barra (/) antes del texto.

Algunos elementos pueden incluir atributos, los cuales son información adicional incluida dentro de la etiqueta de comienzo. Por ejemplo se puede alinear un párrafo (centro, dcha, izq) incluyendo el atributo apropiado en la etiqueta de párrafo.

Algunas consideraciones es HTML son:

- HTML no es sensible a mayúsculas - minúsculas. Por ejemplo, <title> es equivalente a <TITLE> o <TiTle> .
- No todas las marcas son soportadas por los browsers de WWW. Generalmente, si no las soportan, simplemente las suelen ignorar.
- No importan las tabulaciones (tab), los saltos de línea ni los espacios en blanco extra.

Los documentos HTML

La estructura de un documento HTML se compone una cabecera y un cuerpo. En la cabecera generalmente va un título y en el cuerpo los elementos de los que se compone el documento como párrafos, listas, tablas, etc.

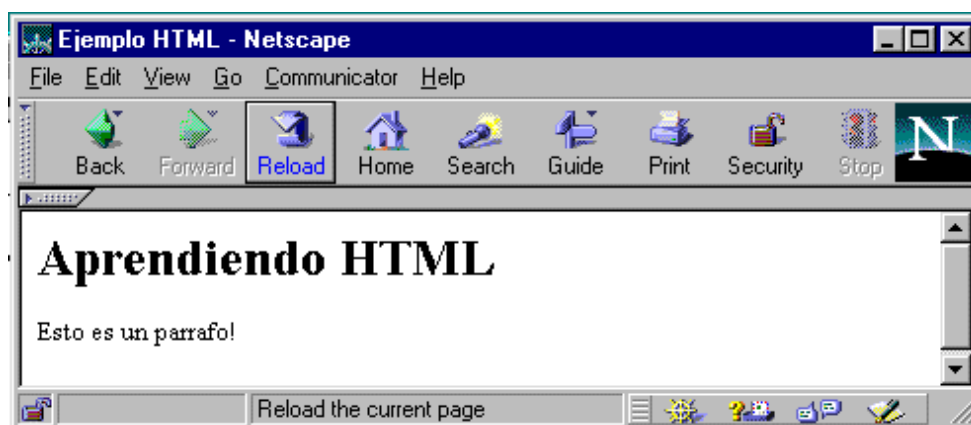
La cabecera se inicia mediante la etiqueta `<HEAD>` y se termina con `</HEAD>`. Por lo general se incluye aquí el título del documento, mediante `<TITLE> ... </TITLE>`. Y el cuerpo se inicia mediante la etiqueta `<BODY>` y se termina con la etiqueta `</BODY>`.

Ejemplo:

```
<html>
<head>
<TITLE>Ejemplo HTML</TITLE>
</head>
<body>
<H1>Aprendiendo HTML</H1>
<P>Esto es un parrafo!</P>
</body>
</html>
```

Las etiquetas necesarias son `<html>`, `<head>`, `<title>`, y `<body>` con sus correspondientes fines de etiqueta. Todos los ficheros deben incluir estas etiquetas, así que puede ser útil crear un fichero de plantilla con ellas.

Así se ve en el cliente WWW:



Etiquetas

HTML

Las etiquetas `<HTML> ... </HTML>` se usan para delimitar el documento en hipertexto completo. Este comando permite además indicar al cliente la versión de HTML que estamos usando, entre otras cosas.

HEAD

Se delimita entre `<head> ... </head>`. Es la primera parte de un documento HTML, debe llevar un título, pero además suele llevar más información sobre el documento.

TITLE

Titulo del documento que se delimita entre `<TITLE> ... </TITLE>`. Debe de ir dentro la cabecera, en decir, entre las etiquetas `<HEAD> ... </HEAD>`. Suele aparecer en la barra superior del cliente WWW.

BODY

El contenido del documento debe ir entre `<BODY> ... </BODY>`.

Cabeceras

Los textos en HTML poseen seis niveles de cabeceras. El nivel 1 es el que pone el texto más grande y con mayor separación hasta el siguiente texto, hasta el nivel 6 con letra más pequeña y menor separación hasta el siguiente texto.

Todas las etiquetas de encabezado se escriben de a pares, e insertan un salto de línea al final del texto que delimitan.

Ejemplo:

Nivel 1 Nivel 6	<code><H1>Nivel 1</H1></code> ... <code><H6>Nivel 6</H6></code>
--------------------------------------	---

Párrafos

El elemento párrafo en HTML se crea con la etiqueta `<P> ... </P>`. Inserta un retorno de carro y un salto de línea al tras el párrafo. Además, permite alinear el párrafo al centro, a la izquierda o a la derecha, con el atributo `align`.

Listas

Existen tres tipos de listas: no numeradas, numeradas y glosarios. Las listas se pueden anidar unas dentro de otras, aunque sean de diferentes tipos.

Listas no numeradas

También se llaman listas no ordenadas. Se declaran entre las etiquetas `...` y cada ítem de la lista se marca con `` y se puede omitir su etiqueta de fin. El resultado final es que el ítem se separa del borde izquierdo y aparece marcado con un símbolo que se puede definir con el atributo `TYPE`. Cuyos valores posibles son: `disc`, `square` y `circle`.

Ejemplo:

<ul style="list-style-type: none">■ Primer ítem■ Segundo ítem<ul style="list-style-type: none">• Primer subítem• Segundo subítem■ Tercer ítem	<pre><UL TYPE="square"> Primer ítem Segundo ítem <UL TYPE="disk"> Primer subítem Segundo subítem Tercer ítem </pre>
--	--

Listas numeradas

También se llaman listas ordenadas, se declaran entre las etiquetas ` ... `. Cada ítem de la lista se marca con un ``. El resultado final es una lista numerada, el la cual se pueden especificar los siguiente atributos: `start`, el cual indica el número por el cual se empieza a numerar, aunque se puede dar un valor concreto con el atributo `value` y `type` el cual especifica el tipo de numeración que se dará (1 para números arábigos, `i` ó `I` para números romanos, `a` ó `A` para letras) .

Ejemplo:

I. Primer item II. Segundo item i. Primer subitem ii. Segundo subitem III. Tercer item	<pre><OL TYPE="I"> Primer item Segundo item <LO TYPE="i"> Primer subitem Segundo subitem </LO> Tercer item </pre>
--	--

Listas descriptivas o glosarios.

Una lista descriptiva es muy útil al momento de hacer glosarios, índices, referencias a otros documentos o descripciones. Una lista descriptiva se declara usando las etiquetas `<DL> ... </DL>`. Donde cada ítem de la lista consta de un título, marcado mediante `<DT> ... </DT>`, y una descripción para dicho título, que se indica mediante la etiqueta `<DD>`.

Ejemplo:

1er punto Descripción 1 2do punto Descripción 2	<pre><DL> <DT>1er punto </DT> <DD>Descripción 1 <DT>2^{do} punto</DT> <DD>Descripción 2 </DL></pre>
--	---

Texto preformateado

Si queremos que el texto final aparezca tal y como lo hemos escrito en el documento, es decir, con los retornos de carro, tabulaciones, etc., que hayamos utilizado, tendremos que introducir ese texto en el documento entre las etiquetas `<pre> ... </pre>`. El tipo de letra usado por el texto preformateado es el mismo que el de `<TT> ... </TT>`, parecido al de una máquina de escribir y más pequeño que el del texto normal. Además, en el texto preformateado no se interpretan los caracteres especiales HTML : `<`, `>` y `&`.

Se pueden insertar espacios en blanco, saltos de línea y tabulaciones.	<pre><PRE> Se pueden insertar espacios en blanco, saltos de línea y tabulaciones. </PRE></pre>
--	--

Citas

Para incluir citas se puede utilizar la etiqueta `<BLOCKQUOTE>`. Les da un formato diferente del normal.

Address

Las direcciones pueden aparecer en un formato especial, y para ello se puede utilizar la etiqueta `<ADDRESS>`. Generalmente las muestra en *itálica*.

Ejemplo:

<i>alguien@algunsitio.com</i>	<code><ADDRESS></code> alguien@algunsitio.com <code></ADDRESS></code>
-------------------------------	---

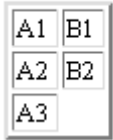
Líneas Horizontales

La etiqueta `
` realiza un retorno de carro. Además permite realizar saltos más largos o más cortos, dejar atrás las imágenes, etcétera.

Tablas

Una tabla en HTML se entiende como un conjunto de filas y cada fila contiene a su vez un conjunto de celdas. Las tablas se declaran usando la etiqueta `<TABLE>` ... `</TABLE>`. Dentro de la tabla, se usa `<TR>` ... `</TR>` para indicar una fila y dentro de una fila, `<TD>` ... `</TD>` para delimitar el contenido de una celda. Los elementos de cierre de fila y celda son opcionales. Si queremos dotarle un título, se utiliza `<CAPTION>`. Estas etiquetas aceptan numerosos atributos con los que se puede por ejemplo, unir celdas, especificar la alineación del contenido de la celda, etc.

Ejemplo de tabla

<p>Título</p> 	<pre><TABLE BORDER="2"> <CAPTION>Título</CAPTION> <TR> <TD> A1</TD><TD> B1</TD> </TR> <TR> <TD> A2</TD><TD>B2</TD> </TR> <TR> <TD> A3</TD> </TR> </TABLE></pre>
---	---

Comentarios

Para poner comentarios al texto fuente lo haremos de la siguiente manera:

```
<!-- Esto es un comentario -->
```

Su contenido no aparecerá en la visualización del documento final.

También pueden verse con SSI (Server Side Includes) y JavaScript, aunque en ese caso no serían comentarios.

Formato de caracteres

HTML tiene dos tipos de estilos para mostrar frases o palabras individuales: lógico y físico. El estilo lógico, es acorde con el significado/semántica y el estilo físico acorde con la apariencia, aunque luego puedan verse iguales en el navegador. Por ejemplo que se muestre igual `` (bold, negrita) que `` (énfasis).

El porqué de los dos estilos es debido al divorcio en SGML entre contenido y presentación. Por ejemplo, las cabeceras H1, no especifican el tipo de letra y tamaño a presentar la cabecera (Times New Roman 24, negrita y centrado) y en muchos navegadores este tipo de cosas se pueden definir.

Estilos lógicos

Algunos de ellos son:

`<DFN>` definición. Generalmente en itálica.

`` énfasis. Generalmente en itálica.

`<CITE>` títulos de libros, películas, etc. Generalmente en itálica.

`<KBD>` entrada de teclado. Generalmente como tamaño fijo (TypeWriter).

`` fuerte énfasis. Generalmente en negrita.

Estilos físicos

Algunos de los estilos físicos son:

`` Negrita (Bold)

`<I>` Cursiva (Italic)

`<TT>` Tamaño fijo (TypeWriter)

`<U>` Subrayado (Underline)

`<BLINK>` Parpadeante (Blink)

`<CENTER>` Centrado

Caracteres especiales

Tienen dos funciones: marcar caracteres especiales y mostrar caracteres que no están disponibles en ASCII, como caracteres con tildes, diéresis y acentos. Para incluir caracteres especiales, se usa la etiqueta `&código;`.

Como hemos visto antes hay tres caracteres especiales en ASCII (`<`, `>`, `&`) que tienen un significado especial para HTML. y para poder mostrarlos deben ser marcados con caracteres de escape de esta forma:

<code>&lt;</code>	<code><</code>
<code>&gt;</code>	<code>></code>
<code>&amp;</code>	<code>&</code>

Otros caracteres especiales:

<code>&aacute;</code>	á
<code>&eacute;</code>	é
<code>&iacute;</code>	í
<code>&oacute;</code>	ó
<code>&uacute;</code>	ú

ü	ü
ñ	ñ
Á	Á
É	É
Í	Í
Ó	Ó
Ú	Ú
Ü	Ü
Ñ	Ñ
¡	¡
¿	¿

NOTA: A diferencia del resto de HTML, las secuencias de escape si son sensibles a mayúsculas y minúsculas como puede verse en la tabla.

Enlaces / Referencias en documentos HTML

Es lo que permite a HTML, navegar entre documentos o saltar a partes de un mismo documento, es por tanto la principal característica del hipertexto.

Para definir un enlace en HTML usamos la etiqueta <a> (anchor) para indicar el objeto desde el cual parte el enlace. Dentro de la etiqueta se incluye el parámetro href="URL" para especificar el destino del enlace, es decir, la dirección URL con la que queremos enlazar.

Para referenciar a otro documento, es necesario conocer la ubicación / ruta / path exacta del documento que se referenciará. Una ubicación, puede ser referenciada en forma relativa o absoluta.

Ruta relativa

Se indica especificando la posición del documento en la estructura de subdirectorios a partir de la ubicación del documento actual. Sólo se puede usar para documentos ubicados en el mismo ordenador. Una referencia relativa a otro documento se hace usando la etiqueta

Ej. usando un path relativo al documento actual:

```
<a href="../../../Donostia/hoteles.html">Hoteles de Donostia</a>
```

Ruta absoluta

Se indica especificando el URL (Uniform Resource Locator) de la página que se está referenciando. Los URL son una manera universal de especificar una dirección. La forma más básica de referenciar un hipertexto es usar la etiqueta El host indica la ordenador donde se encuentra el documento. El directorio y archivo indican su posición dentro de ese computador. Los URL tienen muchas formas distintas para referenciar distintos objetos.

Ej.: Zona

Enlaces a sección del mismo documento.

Si el enlace lo queremos hacer a una sección del propio documento, debemos declarar primero el ancla que nos definirá el lugar al cual iremos, es decir, el destino. Esto lo haremos añadiendo a la etiqueta <A> , mediante el atributo name.

Ej.: `texto destino`.

Posteriormente, en el enlace / referencia indicaremos que se debe llegar a ese destino mediante el carácter #.

Ej.: `texto`

Si nos encontramos dentro del mismo documento bastará con llamarlo sin utilizar el nombre de la página HTML

Ej.: `texto`.

Multimedia en páginas HTML

Para hacer más atractivo y fácil de entender un hipertexto, es posible incluir imágenes, sonido, animaciones, videos y realidad virtual. Las imágenes pueden aparecer directamente en el hipertexto (en los casos en que el cliente provee de una interfaz gráfica). En cambio, algunos de los otros deben de estar asociados a programas o tener los plug-ins necesarios (depende del cliente WWW).

Imágenes

Para incluir una imagen en un hipertexto, se usa la etiqueta:

``.

La imagen puede ser referenciada en forma absoluta o relativa, de la misma forma en que se referenciaba a los documentos hipertexto en la sección anterior.

Aunque hay muchos tipos de formatos, los fundamentales para el uso en WWW son GIF y JPEG.

Ejemplos:

``

``

Para browsers que no utilizan formato gráfico o mientras no llega la imagen al browser, se puede incluir un texto alternativo. Para ello hay un parámetro en el que decimos que inserte un texto donde debería aparecer el gráfico mediante `alt`. Esta utilidad conviene usarla sobre todo cuando el gráfico es un hiperenlace, ya que de otra forma los clientes sin capacidad gráfica no presentarán el enlace.

Ej.: ``.

Para conseguir que un gráfico se presente como un enlace a otra dirección basta con meterlo entre `<a>` y ``.

Ej.: ``

Cuando queremos que un gráfico aparezca sin el borde simplemente tenemos que añadirle el parámetro `BORDER=0`. También podemos indicar la alineación del gráfico con respecto al texto mediante `ALIGN` que puede tomar los valores `TOP`, `MIDDLE` y `BOTTOM`.

Ej.: `Texto asociado`

Hay un formato especial de GIF, GIF89A que permite guardar varias imágenes en el mismo fichero permitiendo crear animaciones con ellas. Además permite fondos transparentes.

Imágenes externas

La forma de insertar una imagen externa es:

` ... `

Imágenes de fondo

La forma de insertar una imagen como fondo en un cliente es:

`<BODY BACKGROUND="url_de_la_imagen" ... >`

Audio

Los archivos de audio son, por lo general, bastante grandes en cuanto al espacio en disco que ocupan; por ello, son lentos de transferir a través de la red. Por esto los archivos de audio no se cargan de manera automática, sino que requieren son transferidos sólo cuando el usuario lo solicita. Para incluir un sonido:

` ... `

Los formatos de audio más comunes son: AU, AIF / AIFC, Wav, MPEG.

Vídeo

Para incluir un vídeo, se procede igual que la manera anterior, apuntando al URL del vídeo.

Los formatos de vídeo más comunes son MPEG, QuickTime y AVI.

Mapas sensibles / Imágenes Activas

Permiten que una imagen tenga varios destinos dependiendo de la coordenada en que se pulse. Dependiendo de si la coordenada pulsada se analiza en el cliente o en el servidor hay dos formas de generarlos. Server-Side Imagemap la forma antigua donde se necesita procesamiento a ambos lados y Client-Side Imagemap, la forma actual y más correcta de hacerlo ya que sólo se necesita procesamiento por parte del cliente.

Server Side Imagemaps

Se necesita procesamiento tanto por parte del cliente como por parte del servidor. En el servidor se guarda información mediante un fichero que contiene las formas y coordenadas de una de las regiones que se quiere que sean activas en la imagen. Es decir, que cuando se pulse en determinada región, se salte a una URL.

El formato del fichero cambia el orden de los datos si se utiliza el formato del servidor NCSA o CERN (en los demás servidores suelen utilizar uno de estos dos formatos). Las herramientas para hacer los mapas suelen permitir guardar en cualquiera de estos dos formatos.

Formato en NCSA (pionero y más utilizado):

Método URL coord1 coord2 ... coordn. Donde método puede ser:

Default: cuando se pulsa en la imagen fuera de áreas sensibles
Circle: Círculo. Coordenadas: centro radio.
recty: rectángulo. Coordenadas: esquina-sup-izq esquina-inf-dcha.
poly: polígono. Coordenadas: cada coordenada es un vértice.
point: punto. Coordenadas del punto.

¿Cómo funciona?

Hay un programa CGI en el servidor, que se ejecuta en el servidor cuando se pulsa una imagen activa (generalmente llamado `Imagemap`). Este programa interpreta la coordenada pulsada por el cliente WWW en la imagen para devolverle la localización del nuevo destino consultando el fichero de coordenadas visto anteriormente.

Client-Side Imagemaps

Este método tiene ciertas ventajas sobre el anterior como pueden ser:

- Reduce la carga del servidor.
- Se pueden mostrar los destinos en el cliente mientras los usuarios se mueven sobre la imagen.
- Los mapas pueden ser creados y probados sin necesidad de un servidor.
- Los mapas también pueden ser accedidos usando otros protocolos que no sean HTTP. Server-side imagemaps sólo pueden ser accedidos con el protocolo HTTP.
- Los mapas no dependen del servidor, haciéndolos más portables.
- El mapa está incluido en una página HTML, después sólo hay que añadirle a la marca IMG, el atributo USEMAP tal y como se ve en el ejemplo.

Ejemplo:

```
<MAP NAME="botones">
<AREA SHAPE="RECT" COORDS="10,10,49,49" HREF="nosotros.html">
<AREA SHAPE="RECT" COORDS="60,10,99,49" HREF="productos.html">
<AREA SHAPE="RECT" COORDS="110,10,149,49" HREF="indice.html">
<AREA SHAPE="RECT" COORDS="0,0,159,59" NOHREF>
</MAP>

<IMG SRC="../../../images/tech/bar.gif" USEMAP="#botones">
```

NOTA: Si USEMAP empieza por #, el cliente supone que el mapa está en la misma página, sino hay que añadir el fichero donde esta delante: ``

Si el navegador no soporta client-side imagemap, se puede añadir ISMAP para que lo haga con el servidor :

```
<A HREF="/cgi-bin/imagemap/directorio/imagen.map"> <IMG
SRC="/directorio/imagen.gif" USEMAP="#nombre" ISMA></A>
```

Más información ...

NCSA – A beginner's Guide to HTML Home Page:

<http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html>

Official HTML specification:

<http://www.w3.org/pub/WWW/MarkUp/>

Tejedores del Web: Manual de html

<http://www.dic.uchile.cl/~manual/>

WWW FAQ (Frequently Asked Questions)

<http://www.boutell.com/faq/>

Otros manuales en la Web:

<http://www.utirc.utoronto.ca/HTMLdocs/NewHTML/htmlindex.html>

http://www.webcom.com/~webcom/html/tutor/html_overview.html

http://www.indiana.edu/ip/ip_support/learn_html.html

<http://WWW.Stars.com/Vlib/Misc/Tutorials.html>

<http://home.mcom.com/home/how-to-create-web-services.html>

<http://cclx1.unican.es/~uc1892/html2.html>

<http://www.geocities.com/SiliconValley/2915/manual.htm>

<http://www.um.es/~psibm/tutorial/>

Formularios

en HTML

HTML permite usar formularios en el documento, como una manera básica de proveer al usuario de un método para enviar información hacia el servidor desde el cliente hacia el servidor, mediante campos que se pueden rellenar, lo mismo que un formulario de papel. Estos elementos son cajas de texto, combos, listas, botones de opción, etc. De esta forma se completa la interacción Cliente/Servidor, al permitir el flujo de información no sólo desde el servidor hacia el cliente (flujo HTML normal) sino también en el sentido opuesto. Un formulario supone, casi con toda probabilidad, la existencia de código HTML generado dinámicamente.

La mayoría de las páginas HTML son idénticas para todo el mundo, y no admiten acciones sobre ellas, salvo la navegación. Sin embargo, es posible que el contenido de las páginas que sea variable. Esto puede hacerse, o bien generando físicamente un fichero HTML en el servidor, que enviado al cliente, o enviando e fichero HTML sin que exista una copia físicamente en el servidor que es el caso más habitual.

Las aplicaciones que tienen lugar a través de los formularios son muy variadas: ordenar un producto, enviar comentarios, hacer una operación bancaria, inscribirse en una base de datos, etc. Por ejemplo, cada vez que utilizamos un buscador en Internet, éste nos devuelve un documento que ha sido confeccionado dinámicamente, específicamente para nuestra petición, en función de las palabras clave que hemos proporcionado, sin existir una copia del documento en el servidor.

Introducción de texto

Campos de texto

Introduce datos...

Primer campo...

Segundo campo...

Código HTML del ejemplo:

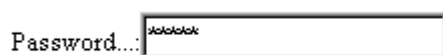
```
Introduce datos:<INPUT TYPE="text" NAME="datos">  
Primer campo...:<INPUT NAME="campo1" VALUE="Algo por defecto">  
Segundo campo...:<INPUT NAME="campo2" SIZE=5 MAXLENGTH=5>
```

Los atributos que tiene son:

- `TYPE = text`. Tipo texto. Es el valor que pone por defecto. No hace falta ponerlo.

- **NAME** nombre simbólico. Es colocado junto con su valor cuando se envía el formulario.
- **SIZE** tamaño del campo INPUT en caracteres.
- **MAXLENGTH** máximo número de caracteres que son aceptados en la entrada.
- **VALUE** valor que pone en el campo por defecto.

Campos de Password



Código HTML del ejemplo:

```

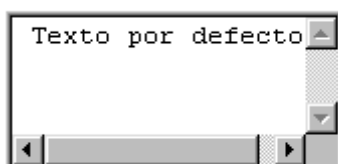
Password...:<INPUT TYPE="password" NAME="password"
VALUE="quest">

```

Los atributos que tiene son:

- **TYPE** = password. Tipo password.
- **NAME** nombre simbólico. Es colocado junto con su valor cuando se envía el formulario.
- **SIZE** tamaño del campo INPUT en caracteres.
- **MAXLENGTH** máximo número de caracteres que son aceptados en la entrada.
- **VALUE** valor que pone en el campo por defecto.

Areas de texto.



Código HTML del ejemplo:

```

<TEXTAREA COLS=18 ROWS=3>
  Texto por defecto.
</TEXTAREA>

```

Atributos de TEXTAREA:

- **NAME** nombre simbólico que será unido al texto cuando se envíe el formulario.
- **ROWS** número de filas en caracteres.
- **COLS** número de columnas en caracteres.

Introducción de opciones

CheckBoxes

- ☐ Giro postal.
- ☐ Cheque.
- ☐ Tarjeta de crédito.

Código HTML del ejemplo:

```
<INPUT TYPE="checkbox" NAME="pagar" VALUE="giroPostal">Giro
postal.
<INPUT TYPE="checkbox" NAME="pagar" VALUE="cheque"> Cheque.
<INPUT TYPE="checkbox" NAME="pagar" VALUE="tarjeta">Tarjeta de
crédito.
```

Atributos a añadir a la marca INPUT para realizar checkboxes:

- **TYPE** = checkbox.
- **NAME** nombre simbólico. Es colocado junto con su valor cuando se envía el formulario.
- **VALUE** valor que se codifica junto con el nombre.
- **CHECKED** indica cuáles están marcados por defecto.

Radio botones (Radio Buttons)

Como desea pagar?

1. ☒ Giro postal.
2. ☐ Cheque.
3. *Tarjeta:*
 - o ☐ Mastercard.
 - o ☐ Visa.

Código HTML del ejemplo:

```
Como desea pagar?<OL>
<LI> <INPUT TYPE="radio" NAME="formaP" VALUE="giro" CHECKED>
Giro postal.
<LI> <INPUT TYPE="radio" NAME="formaP" VALUE="cheque">
```

```

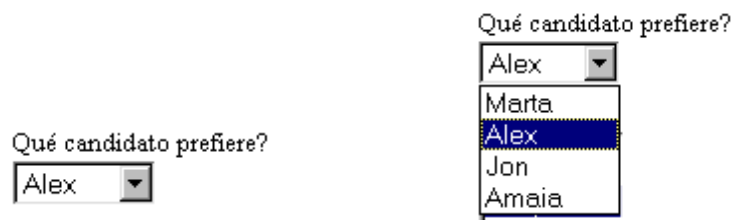
Cheque.
<LI><I>Tarjeta:</I>
<UL><LI> <INPUT TYPE="radio" NAME="formaP"
VALUE="mastercard"> Mastercard.
<LI> <INPUT TYPE="radio" NAME="formaP" VALUE="visa">
Visa.
</UL>
</OL>

```

Atributos que hay que añadir a la marca `INPUT` para realizar radiobotones:

- `TYPE = radio`
- `NAME` nombre simbólico. Es colocado junto con su valor cuando se envía el formulario.
- `VALUE` valor que se codifica junto con el nombre.
- `CHECKED` indica cuales están en "on" por defecto.

Cuadro combinado (Combo Boxes)



Código HTML del ejemplo

```

Qué candidato prefiere? <BR>
<SELECT NAME="candidato">
  <OPTION> Marta
  <OPTION SELECTED> Alex
  <OPTION> Jon
  <OPTION> Amaia
</SELECT>

```

Atributos que hay que añadir a la marca `SELECT`:

- `NAME` nombre simbólico que será unido a la opción cuando se envíe el formulario.
- `SIZE` por defecto es uno y aparecerá como un menú.
- `SELECTED` cuál aparece por defecto.

Cuadro de Listas (List Box)

Escoja las prendas que quiere llevarse:



Código HTML del ejemplo:

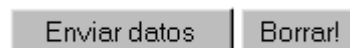
```
Escoja las prendas que quiere llevarse: <BR>
<SELECT NAME="ropa" MULTIPLE SIZE=5>
  <OPTION>Camisa
  <OPTION SELECTED>Vaqueros
  <OPTION>Jersey
  <OPTION SELECTED>Chaqueta
  <OPTION>Polo
</SELECT>
```

Atributos que hay que añadir a la marca `SELECT`:

- `NAME` nombre simbólico que será unido a la opción cuando se envíe el formulario.
- `SIZE` ($n > 1$). Para una lista de “n” elementos.
- `SELECTED` para que se muestren por defecto.
- `MULTIPLE` si está presente, especifica que `SELECT` debe permitir múltiples valores.

Botones de control

Al pie del formulario, se suele encontrar un botón de envío del formulario, y si se quiere, otro para limpiar el formulario.



Se declaran de la siguiente manera:

```
<INPUT TYPE="submit" VALUE="Enviar datos">
<INPUT TYPE="reset" VALUE="Borrar!">
```

Atributos:

- `TYPE = submit`, botón que causa que se envíe el formulario.
- `TYPE = reset`, botón que hace que se pongan los valores por defecto en el formulario.
- `VALUE`, etiqueta que muestra el botón.

Funcionamiento de los formularios

Cuando se pulsa el botón de `SUBMIT`, se codifican esos valores en el formulario antes de enviárselos al servidor de la siguiente forma:

Los formularios se codifican en una cadena de la forma `nombre = valor` (donde `nombre` es el valor del atributo `name`) separada por `&`. Los espacios blancos son convertidos en `+` y los caracteres especiales en seguido de dos dígitos hexadecimales `%hex hex`

Ejemplo: `marca=Renault&modelo=Clio+1.9i`

Form

```
<FORM ACTION="url" METHOD="POST|GET">
    ....
</FORM>
```

Atributos de `FORM`:

- `ACTION` es la URL del servidor de peticiones al cual los formularios deben ser enviados.
- `METHOD` es el método usado para enviar el formulario. Los métodos pueden ser:
 - `GET` los contenidos del formulario son añadidos a la URL.
 - `POST` los contenidos del formulario se añaden en un cuerpo en vez de en la URL
- `ENCTYPE` la codificación del formulario. Sólo se aplica si el método es `POST` y sólo puede tener un valor: `application/x-www-form-urlencoded`. Es el valor por defecto.

```
<FORM
    ACTION="/cgi-bin/programa"
    METHOD = "post">
<!-- Cualquier marca HTML menos otro FORM, ya que no pueden
anidarse. Aunque puede haber varios en un mismo
documento HTML-->

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

Envío de formularios

Pero hay diferencia donde van los datos según el método que se utilice:

Con el método `GET`:

Si se ha utilizado `METHOD="GET"`, en el formulario, cuando se envían, los contenidos son unidos a la pregunta URL de la forma:

`http://maquina/path/programa?nombre=valor&nombre=valor`

Ej.: `http://www.ehu.es/cgi-bin/programa?quien=pepe&...`

Con el método `POST`:

Si se ha utilizado `METHOD="POST"` en el formulario, cuando se envían, los contenidos van en un cuerpo separado de la URL. Esta opción es la recomendada.

Más información

En NCSA:

<http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/fill-out-forms/overview.html>

En W3:

<http://www.w3.org/>

Bibliografía / Referencias

"Guía Práctica para usuarios de HTML. Creación de páginas Web"
Alonso Alvarez. Editorial Anaya.

"Programación en Java para Internet"
Pedro Manuel Cuenca Jimenez. Editorial Anaya.

"Cómo elegir un proveedor de Internet"
Miguel Pardo Niebla. Editorial Anaya.

"HTML, CGI, JAVA, Servidores... Tecnología WWW"
Alonso Alvarez García. Editorial Anaya.

"Cliente Servidor y Bases de Datos en Internet"
Manuel de la Herrán Gascón. TSAI (Telefónica Servicios Avanzados de Información).
ALI Base Informática Nº 31, 1997

"Servidores de Internet"
Fernando García.
Internet (especial IDG) Suplemento PC World Nº 119

INDICE

CONCEPTOS GENERALES.....	1
Internet	1
Intranet	1
Direcciones IP y DNS	1
WWW	2
Arquitectura Cliente – Servidor	2
Navegadores / clientes / browsers	2
Servidor WWW o Web.....	2
PLUG-INS.....	3
CONSIDERACIONES EN LA INSTALACIÓN DE SERVICIOS DE INTERNET	4
REQUISITOS.....	4
<i>Acceso permanente a Internet.....</i>	<i>4</i>
La conexión con Internet correcta.....	4
Tipos de conexión con Internet.....	5
<i>Encaminadores / Router.....</i>	<i>5</i>
<i>Servidores de Internet (WWW, FTP, ...)</i>	<i>5</i>
PARÁMETROS A CONSIDERAR.....	6
<i>Servicios necesarios.....</i>	<i>6</i>
<i>Precio</i>	<i>6</i>
<i>Prestaciones</i>	<i>6</i>
<i>Facilidad de instalación y mantenimiento del sistema.</i>	<i>6</i>
<i>Versatilidad de manejo.....</i>	<i>6</i>
<i>Diseño de contenido.</i>	<i>6</i>
SOLUCIONES ACTUALES	7
<i>Sistemas UNIX genéricos sobre PC.....</i>	<i>7</i>
<i>Microsoft</i>	<i>7</i>
<i>MacIntosh.....</i>	<i>8</i>
<i>SUN Microsystems.....</i>	<i>8</i>
<i>Silicon Graphics.....</i>	<i>8</i>
PARÁMETROS EN LA CONFIGURACIÓN DE WWW	8
<i>Parámetros de red.</i>	<i>8</i>
<i>Directorios</i>	<i>9</i>
<i>Ficheros de registro.....</i>	<i>9</i>
<i>Seguridad</i>	<i>9</i>
<i>MIME Types.....</i>	<i>9</i>
<i>Navegar por directorios</i>	<i>9</i>
<i>Servidores Virtuales.....</i>	<i>9</i>
<i>Otros</i>	<i>9</i>
DESCRIPCIÓN DEL SERVICIO WWW	10
URL (UNIFORM RESOURCE LOCATOR)	10
<i>Referenciando un hipertexto</i>	<i>10</i>
<i>Referenciando un documento gopher.....</i>	<i>11</i>
<i>Referenciando un servicio FTP.....</i>	<i>11</i>
<i>Referenciando un archivo cualquiera.....</i>	<i>11</i>
<i>Referenciando un grupo de noticias</i>	<i>11</i>
<i>Referenciando una dirección e-mail.....</i>	<i>12</i>
HTTP (HYPERTEXT TRANSFER PROTOCOL)	12
MIME (MULTIPURPOSE INTERNET MAIL EXTENSIONS).....	14
<i>Fichero Mailcap</i>	<i>15</i>
<i>Fichero MIME.TYPES</i>	<i>15</i>
EJECUCIÓN DE APLICACIONES SOBRE WWW EN EL LADO DEL SERVIDOR	17

CGI (COMMON GATEWAY INTERFACE)	17
<i>Introducción</i>	17
<i>CGI. Entrada</i>	18
<i>Bibliotecas de funciones</i>	19
<i>CGI. Salida</i>	19
Non-parsed Header Output	19
Parsed Header Output	20
<i>Ejemplos</i>	20
<i>Variables de entorno</i>	20
<i>Seguridad en los CGIs</i>	22
<i>Problemas con los CGI</i>	22
<i>Más información</i>	22
ALTERNATIVAS A USO DE CGI. INTERFACES DE PROGRAMACIÓN DE APLICACIONES (API)	23
<i>Servlets (Sun Microsystems)</i>	23
<i>NSAPI, SSJ (Netscape)</i>	23
<i>ISAPI (Microsoft)</i>	23
EJECUCIÓN DE APLICACIONES SOBRE WWW EN EL LADO DEL CLIENTE	25
SCRIPTS EN EL CLIENTE	25
<i>VisualBasic Sripits</i>	25
<i>JavaScript</i>	26
Pequeños ejemplos de JavaScript	26
Más información	27
<i>Java</i>	27
MÁS INFORMACIÓN	29
CONEXIÓN WWW – BASES DE DATOS	30
MOTIVOS QUE LLEVAN A LA UNIÓN DE WWW CON BD	30
EJECUTANDO CGIS, API EN EL SERVIDOR	31
SCRIPTS EN EL SERVIDOR	31
CONEXIÓN WEB CON BASES DE DATOS ACCESS	33
<i>Ejemplo de una consulta</i>	33
<i>Descripción del archivo Sample.htx</i>	36
<i>Características del Conector de bases de datos de Internet</i>	37
Archivos del Conector de bases de datos de Internet (.idc)	38
Parámetros	38
<i>Campos de los archivos del Conector de bases de datos de Internet (.idc)</i>	40
Uso de cuadros de lista de selección múltiple en formularios HTML	41
Uso de consultas por lotes y consultas múltiples	42
Consultas por lotes	43
Consultas múltiples	43
<i>Archivos de extensión (htx) HTML</i>	43
<%begindetail%>, <%enddetail%>	43
<%if%>, <%else%>, <%endif%>	44
CurrentRecord, MaxRecords	45
Parámetros de los archivos del Conector de bases de datos de Internet	45
Variables HTTP	45
SEGURIDAD EN WWW	47
SERVIDORES SEGUROS	47
Funcionamiento de un servidor seguro	48
SECURE SOCKETS LAYER (SSL)	48
Instalación de SSL	49
Generación de un par de claves	49
Adquisición de un certificado	49
Instalación de un certificado con un par de claves	50
PROXY	50
Funcionamiento de un Proxy	51
Ventajas de un Proxy	51
CORTAFUEGOS / FIREWALL	52
<i>Inconvenientes de los Cortafuegos</i>	52
<i>Más información</i>	53

FICHEROS DE REGISTROS. ESTADÍSTICAS.....	54
ALMACENEN DE REGISTROS	54
<i>Lectura de los archivos de registro.....</i>	54
PROGRAMAS ANALIZADORES DE LOS FICHEROS DE REGISTROS.....	55
<i>Ejemplos.....</i>	55
SSI.....	59
<i>Formato SSI.....</i>	59
<i>Variables de entorno de SSI.....</i>	60
<i>Ejemplos.....</i>	61
Añadiendo ficheros.....	61
Inserción de Fecha y hora.....	61
Tamaños de fichero.....	62
Otros ejemplos de echo útiles	62
Nombres de Documentos y rutas.....	62
Web browser Utilizado.....	62
Información del Servidor.....	62
GLOSARIO	64
HTML	66
HTML (HYPERTEXT MARKUP LANGUAGE).....	66
<i>Los documentos HTML</i>	67
<i>Etiquetas.....</i>	67
HTML	67
HEAD	67
TITLE	68
BODY	68
Cabeceras	68
Párrafos	68
Listas	68
Listas no numeradas	68
Listas numeradas	69
Listas descriptivas o glosarios.....	69
Texto preformateado	69
Citas.....	70
Address.....	70
Líneas Horizontales 	70
Tablas	70
Comentarios.....	70
<i>Formato de caracteres.....</i>	71
Estilos lógicos	71
Estilos físicos	71
Caracteres especiales	71
<i>Enlaces / Referencias en documentos HTML.....</i>	72
Ruta relativa	72
Ruta absoluta.....	72
Enlaces a sección del mismo documento.....	72
<i>Multimedia en páginas HTML.....</i>	73
Imágenes	73
Imágenes externas	74
Imágenes de fondo.....	74
Audio.....	74
Vídeo.....	74
Mapas sensibles / Imágenes Activas.....	74
Server Side Imagemaps.....	74
Client-Side Imagemaps	75
<i>Más información</i>	75
FORMULARIOS.....	77
<i>Introducción de texto</i>	77
Campos de texto	77
Campos de Password.....	78
Áreas de texto.....	78

<i>Introducción de opciones</i>	<i>79</i>
CheckBoxes.....	79
Radio botones (Radio Buttons).....	79
Cuadro combinado (Combo Boxes).....	80
Cuadro de Listas (List Box)	81
<i>Botones de control</i>	<i>81</i>
<i>Funcionamiento de los formularios</i>	<i>82</i>
<i>Form</i>	<i>82</i>
<i>Envío de formularios</i>	<i>82</i>
<i>Más información.....</i>	<i>83</i>
BIBLIOGRAFÍA / REFERENCIAS	84
INDICE	85