Redes de ordenadores

WWW

Grupo de sistemas y comunicaciones
Juan Jesús Muñoz Esteban
jjmunoz@gsyc.inf.uc3m.es

8. WWW

World Wide Web (la telaraña mundial) es un sistema de información **hipermedia** que ha revolucionado Internet en los años 90.

Se basa en un protocolo de trasferencia de información (**HTTP**), que utilizan los programas cliente (**navegador** o **browser**) para recuperar datos en forma de páginas en un formato normalizado (**HTML**) provenientes de servidores dispersos por Internet. El usuario puede seleccionar otras páginas definidas mediante **URLs** que aparecen resaltadas en las páginas como **hiperenlaces** mediante un interfaz amigable.

La publicitación es un método rápido (porque la actualización tiene efectos inmediatos en quienes acceden a la información) y económico de poner información a disposición de cualquiera(en todo el mundo) que tome la iniciativa (igualdad de oportunidades) de pedirla.

El modelo inicial de navegar de página en página buscando información se ha enriquecido con la proliferación de herramientas de búsqueda (**spiders, crawlers, robots**) así como el acceso a páginas generadas dinámicamente como respuesta a consultas interactivas (**CGI**).

Las capacidades iniciales de visualización se han extendido con la evolución de HTML y la aparición de otros lenguajes para representación de objetos tridimensionales (**VRML**), ejecución en local de aplicaciones (**JAVA**)... aumentando la interactividad.

La riqueza del entorno ha provocado una demanda para incorporarle características de seguridad mejorada (**SSL**) que permitan extender el modelo de acceso a la información a aspectos que pueden cambiar tanto el mundo como el comercio electrónico (**SET**).

8.1 Historia

Surge en Europa, en el CERN (Tim Berners-Lee fue el líder del proyecto, y ahora dirige el consorcio W3C (http://www.w3.org, coliderado en INRIA-MIT, creado a mediados de 1994), un foro abierto para el desarrollo tecnológico sin rupturas de Web al que pertenecen multitud de empresas y universidades.

La versión en modo carácter (telnet://info.cern.ch) dio pronto paso a un prototipo con ventanas desarrollado en 1990 sobre NeXT. Durante 1991 se difundió, comenzó a enlazarse a otros sistemas (WAIS), y en 1992 terminaron siendo 50 servidores Web. Aparecen los primeros interfaces gráficos populares (primero para X y Mac) hasta que en Febrero de 1993, NCSA Mosaic (National Center of Supercomputing Applications) se convierte en el primer navegador ampliamente difundido.

El tráfico web en NFSnet pasó de ser 0,1% al 2.2 % solo en ese año.

En 1994 se crea Netscape (Marc Andreessen de NCSA y Jim Clark de Silicon Graphics). Su navegador copa el mercado y el espectacular crecimiento en bolsa de la compañía hace que Microsoft reaccione al fenómeno Internet. A mitad de año eran 1500 los servidores y a fin de año el 16% del tráfico de NFSnet.

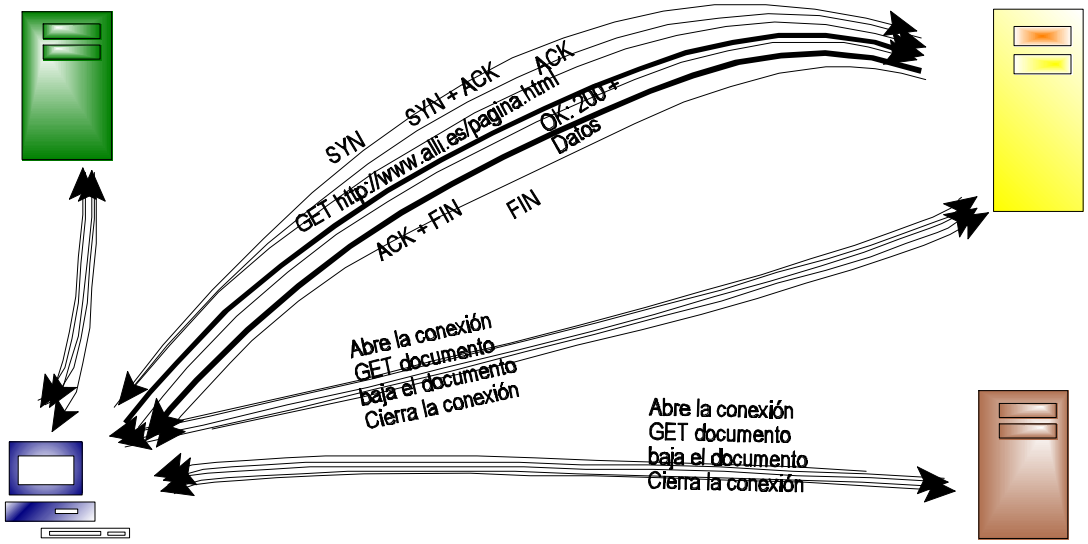
Los grandes servicios comerciales (AOL, Prodigy, CompuServe) ofrecen acceso al web a sus clientes y ante el abrumador éxito de Internet sobre otras redes privadas (MSN) el mundillo de la informática se centra en soluciones basadas en WWW (Intranets, multitud de productos orientados a Internet, Java)...

El éxito frente a otras tecnologías coetáneas (gopher) se puede resumir en:

hipertexto + multimedia = hipermedia

8.2 Arquitectura cliente-servidor

Se basa en un protocolo **sin estado**. El navegador analiza la URL, se conecta al servidor y le pide el documento deseado. A partir de él puede necesitar otras conexiones al mismo u otro servidor para obtener otros objetos de la misma página (por defecto se realizan hasta 4 conexiones simultáneas para traerse elementos gráficos una vez analizado el texto HTML). Al seleccionar un hiperenlace, vuelve a empezar.



Para obtener una página, aunque todas las imágenes estén en el mismo servidor, hacen falta tantos inicios y cierres de conexión como elementos la compongan.

Tampoco permite que el servidor pueda llevar una traza de las actividades del cliente (tan solo hits a objetos). En ocasiones esto puede ser molesto: si el cliente va seleccionando productos y metiéndolos en la cesta de la compra, en cada conexión habrá de pasarle al servidor la lista de todo lo comprado, y éste a su vez volver a enviar la lista al cliente en cada página, que generará dinámicamente (CGI). Resulta más adecuado que estos datos se almacenen en el cliente, en forma de un fichero que sólo puede ser utilizado por el servidor que lo creó (por seguridad). Netscape los bautizó como *cookies* http://search.netscape.com/newsref/std/cookie_spec.html. Los servidores tienen extensiones que facilitan su generación y gestión.

8.3 Servidores

Los servidores originales surgen en el CERN y posteriormente en NCSA.

Los servidores son programas sencillos (mucho más que los clientes). Su función es recuperar la información almacenada y devolverla en mensajes según el protocolo HTTP. El servidor espera solicitudes y localiza y devuelve la página almacenada en el fichero o la genera en ese momento ejecutando un programa (cgi-bin).

Existen muchos (Sun, Oracle...), en general derivados de los originales. Algunos destacaron por mejoras en su rendimiento (Netscape), otros por su fácil instalación: Microsoft tiene versiones personales para Windows 9x y versiones de NT (IIS) que integran su sistema de generación dinámica de páginas.

El más extendido con diferencia es Apache (http://www.apache.org), una derivación del de NCSA con parches (de ahí su nombre), para UNIX (y ya también para NT), de dominio público (mantenido por grupos de voluntarios de USA, Canada, Italia... en muchos casos administradores de *web sites*) y con multitud de extensiones (módulos de seguridad...), multihoming y http 1.1, y capacidad de actuar como proxy (algunos comerciales necesitan un producto aparte. También hay solo-proxies, como squid).

Las extensiones de seguridad (SSL), APIS alternativas a CGI (NSAPI, ISAPI...) son aspectos que se tienen en cuenta en su elección, así como la integración con herramientas de groupware (Notes 4.5) o de generación y gestión de sites (FrontPage).

Jigsaw, del consorcio w3 (http://www.w3.org) formado por INRIA y el MIT, es un servidor escrito en Java (corre en cualquier plataforma que tenga una máquina virtual). Soporta http 1.1, actúa como proxy, tiene un mecanismo (Servlet API) para invocar programas Java que se ejecuten en el servidor para crear páginas dinámicamente (alternativa a CGI, también soportado), es extensible (como Apache) y hace caché de sus propias páginas.

8.4 Clientes

Conocidos como *navegadores o browsers*, son el interfaz de usuario para el acceso a WWW y otras aplicaciones de Internet. Permiten recuperar todos los componentes de una página html y visualizarlos en pantalla. El usuario seleccionará hiperenlaces y el cliente recuperará las páginas, almacenará las direcciones organizadamente, etc.

El desarrollo de WWW incluye desde un primer momento un navegador, en modo carácter y al poco en modo gráfico sobre Next y Macintosh. Pero es en febrero de 1993 cuando se produce el gran despegue de este sistema de información, gracias al NCSA Mosaic, el primer visor gráfico de gran difusión para PC (Windows).

Al poco tiempo el más popular era Navigator (70%), de Netscape Communications, creada por iniciativa de Marc Andreessen, líder del proyecto Mosaic. Con versiones para multitud de sistemas operativos, ha estado perdiendo cuota de mercado frente a Microsoft Internet Explorer, con el que rivaliza en la incorporación de novedades incompatibles. Tras la integración del IE en el GUI de Windows98 se desencadenan una serie de litigios legales apelando a la libre competencia, en los que participan el gobierno federal americano y varios estados, y que coinciden en el tiempo con otros pleitos sobre Java. La situación comercial de Netscape ha provocado también en 1998 que su cliente vuelva a ser gratuito y que su código fuente pase al dominio público.

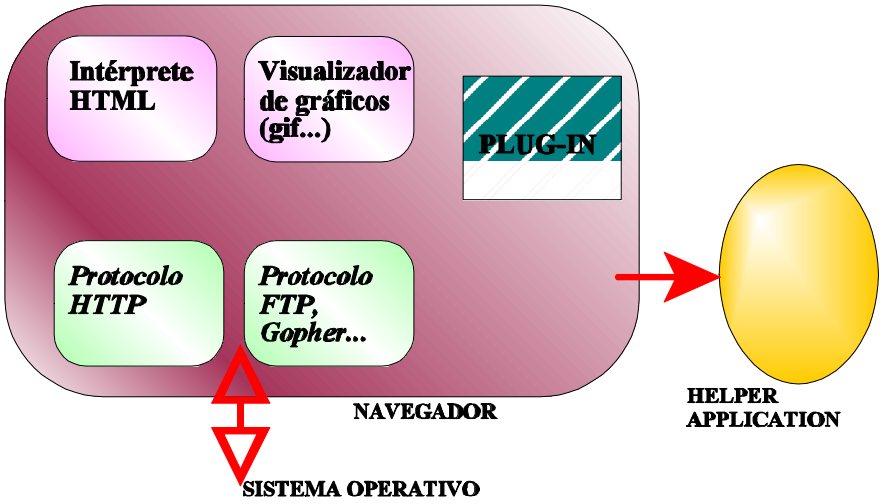
Tras la sorpresa del enorme éxito de Internet con Mosaic en 1993, muchas empresas intentaron entrar en el negocio. Las de software como Quarterdeck sacaron programas que sus usuarios pudiesen tener instalados con facilidad (Internet Suite). Las grandes redes privadas compraron sus browsers (AOL compró Booklinks Internet Works browser, Compuserve compró Spry Air Mosaic Browser) y conectaron sus servicios y luego a sus usuarios a Internet. Las primeras han tenido que ir dedicándose (como siempre) a productos añadidos ante el tirón de las dos grandes. Las segundas se han ido concentrando y se han convertido en proveedores de acceso a Internet aunque con un componente importante de Web-sites de éxito (a donde acude mucha gente y encuentra lo que necesita).

Existen decenas de otros navegadores de los que prácticamente nadie ha oído hablar y a los que les cuesta competir en un mercado claramente bipolarizado. Son más sencillos (pequeños y rápidos en máquinas no muy potentes) al no verse sobrecargados con tanta opción puntera.

- NCSA Mosaic 3.0 (http://www.ncsa.uiuc.edu/SDG/Software/Mosaic) es mucho más simple que Navigator y Explorer. Fué el primer navegador gráfico de gran difusion en Internet (de hecho las primeras versiones del navegador de Microsoft no eran más que un Mosaic ligeramente modificado, y lo mismo ocurre con el de Oracle). Sigue siendo el tercero de la lista de los más usados, aunque a una considerable distancia de IE. Mientras navegas va creando un mapa del recorrido.Tiene correo y news. De él también se deriva Spyglass Enhanced Mosaic (http://www.spyglass.com)
- Lynx: En modo texto, es simple pero rápido en una red lenta.
- Emacs: El editor que sirve para todo, permite la visualización incluso de gráficos, además de tener incorporada la gestión de correo (vm de xemacs)
- Arena: Era un navegador experimental de w3c para HTML 3 (tablas, hojas de estilo), que corre en X/unix. Su sustituto se ha llamado Amaya.
- Hotjava: Escrito en Java por SUN, es extensible y corre sobre cualquier ordenador para que que exista una máquina virtual Java.
- Opera (http://www.operasoftware.com) es un navegador europeo (Oslo) de cierto éxito por su pequeño tamaño y por correr sobre PCs poco potentes. En verano de 1998 la versión 3.2 ofrece cientos de URLs recomendadas organizadas en 13 categorías. Además de funcionar como navegador, Opera ofrece correo (facilitando la organización y envío) acceso a los grupos de noticias de Usenet.
- 1st Choice Browse98 (http://www.ftppro.com) configura muchas URLs en el bookmark (esta iniciativa la tuvieron también los grandes, aunque éstos tienen a llevarte a su Site (Netcenter)) y se centra en su gestión (permite importar desde el IE, añadir sobre la marcha...), de manera que navegar ya no es sólo seleccionar y retroceder, lo que debería ahorrar tiempo ya que la base de datos que crea además de la URL y el título incorpora una breve reseña.
- NeoPlanet (http://www.neoplanet.com) es un añadido a IE o Navigator, para acceder con facilidad a los “mejores sitios” y personalizar la navegación. Tiene una revista asociada con noticias...
- Tango Multilingual Browser (http://www-ar.alis.com/internet_products/index.en.html) soporta 90 idiomas en su interfaz. También va incorporando correo.
- ChiBrow (http://www.chibrow.com) está dirigido a los niños (interfaz muy sencillo, con los bonotes imprescindibles y no más) y permite filtrarles ciertas direcciones prohibidas.
- HexaBit Junior (http://www.hexabit.com/junior/index.htm) también permite la censura mediante contraseñas, y ya tiene más opciones (algo menos simple).

No todos los visores muestran igual un mismo documento. Algunos no muestran la página si no es correcto HTML (tendencia general a validar el ML). Otros no son capaces de interpretar las nuevas extensiones (**script) o las tratan de forma diferente.

Las últimas versiones incorporan la posibilidad de extender los tipos de objetos que pueden manejar, aunque tienen el problema de ser específicos para cada sistema:



! Helper Applications: Programas independientes que son lanzados automáticamente por el navegador cuando éste detecta un tipo concreto de objeto MIME.

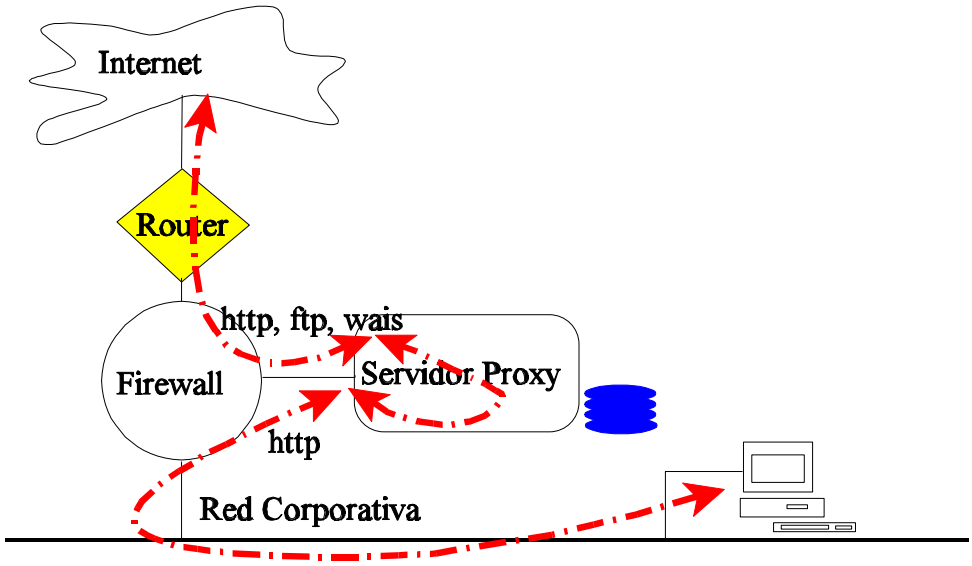
! Plug-ins: Extensiones integradas en el propio navegador que le permiten visualizar información de un determinado tipo en su propia ventana.

Además de estas facilidades genéricas, que permiten a las empresas difundir sus formatos propietarios (acrobat, envoy, autocad, Mathlab...), los nuevos navegadores rivalizan en dar soporte a nuevas normas (DHTML, XML), incluyen intérpretes de Java y **Script, enlaces con ActiveX... e incorporan programas adicionales: clientes de correo y news, ayudas para organizarse los bookmarks y address-books, chats, telefonía, visualización de imágenes tridimensionales, agendas, calendarios, editores HTML...

8.5 Proxies

Se trata de servidores intermedios que analizan las peticiones, las lanzan a sus destinos reales y redirigen lo obtenido al cliente que originó la transacción, almacenando una copia en una cache compartida por otros clientes (con el beneficio que a todos puede aportar el descenso de tráfico por su conexión a Internet cuando una página puede entregarse desde el propio proxy).

Permiten implementar seguridad (dual-homed host) y eliminan la necesidad de que los clientes tengan que tener acceso a DNS.



Podrían aprovecharse para simplificar el cliente (basta que hable http y el proxy hablará el resto de los protocolos (ftp...) por él y le devolverá los resultados). También es el punto adecuado para poner filtros (a qué direcciones no se debe ir).

Los caches pueden relacionarse en una jerarquía. Existen proyectos que intentan así paliar la sobrecarga de los enlaces intercontinentales.

8.6 Identificación de los documentos: URLs

Los hiperenlaces son referencias a otras páginas que pueden estar en cualquier servidor del mundo. Hace falta un mecanismo para distinguir y nombrar cada página de cada máquina, y se llama **URL** (Uniform Resource Locator)

La sintaxis y semántica de las URL está en las RFC 1738 y RFC 1808

protocolo://maquina[:puerto]/path[[#parte]|[parametros]]

- Protocolos: http, ftp, telnet, wais, gopher, news, mailto, https, file
- Máquina: puede ser su nombre (para DNS) o la dirección. La RFC 1900 recomienda evitar direcciones IP como referencia al host (hay programas que no las aceptan).
- Puerto: por defecto es el 80, pero puedes especificar que el servidor escucha en otro (Ej: 800)
- Path : El directorio y nombre del fichero. Las mayúsculas son distintas de las minúsculas pues lo interpreta el S.O. del servidor (no como en DNS, que no es case-sensitive).No tienen límite en su longitud, pero está previsto un error si el servidor no puede manejar uno muy largo.
- Parte: Referencia a una marca definida dentro de esa página. El navegador deberá descargar la página entera, pero en lugar de visualizarla desde el principio lo hará desde donde aparezca ese tag.

El path puede ser absoluto (especificando el servidor, puerto y camino: http://www.w3.org/directorio/fichero.extension) o relativo (a la página html que lo referencia, dentro del mismo servidor: ./directorio/fichero o fichero.htm), lo que resulta cómodo cuando hay que mover las páginas sin ayuda de herramientas, aunque cuando un cliente se hace una copia se pierden las referencias (si no usa wget), y cuando el servidor es accedido desde intranets utilizando nombres ficticios de servidores.

El directorio raíz, origen de los documentos, no es el raíz del ordenador, sino el directorio que se ha configurado como tal en el servidor web (httpd.conf). Si se permite a los usuarios tener páginas, por seguridad, se asume que sus páginas comienzan en $HOME/public_html (si $HOME tiene permiso de acceso y public_html también, y las páginas permiso de lectura. Public_html también puede cambiarse a otro valor).

Es frecuente intentar acceder a páginas que se han movido de sitio, y que no quede una indicando el nuevo lugar (para que cambies tu bookmark) o que directamente te redirija a él. En este caso el navegador muestra un error (una página que envía el servidor indicando que no encontró lo pedido). Esta problemática está abordada por W3C, con un modelo generalizado de **URI**s (Universal Resource Identifiers, RFC 1630):

URN: (UR Naming) Dado el nombre de la página, y el cliente la localiza ayudado por servidores y la trae de donde esté (si hay replicas, del lugar más cercano). Es como si hubiese un servicio global de directorio de páginas, y éstas pudiesen migrar de servidor de forma transparente (y no a base de poner redirecciones en sus anteriores ubicaciones).

URC: (UR Citation) Mediante palabras clave se accede a la información, no indicando dónde residía físicamente inicialmente. Equivale a ocultar completamente la localización de la página, que una red semántica localice la información por su contenido.

La definición formal de estos identificadores más conocidos es:

- URI = (absoluteURI | relativeURI) ["#" fragment]
- absoluteURI = scheme ":" *(uchar | reserved)
- relativeURI = net_path | abs_path | rel_path
- net_path = "//" net_loc [abs_path]
- abs_path = "/" rel_path
- rel_path = [path] [";" params] ["?" query]
- path = fsegment *("/" segment)

Ejemplos:
mailto:cuenta@direccion.de.correo.internet
news:es.comp.delphi
telnet://ordago.gsyc.inf.uc3m.es
ftp://ftp.gsyc.inf.uc3m.es
gopher://gopher.umn.edu
file:///C|/TMP/fichero.htm

8.7 HTTP

HyperText Transfer Protocol corresponde al nivel 7 (aplicación) OSI.

Es un protocolo sin estado: el servidor no conoce anteriores consultas del cliente. No sabe qué ha pedido antes ni qué puede pedir en el futuro. Para guardar el estado hay que usar cookies. El cliente navega libremente y en cada transacción trae un documento: inicia una conexión TCP, se trae (si puede) el documento, y cierra la conexión. Si la página tiene varios componentes(gráficos), cada uno supone una conexión (por defecto hacen 4 concurrentes) aunque sea contra el mismo servidor.

Está pensado para sistemas de información hipermedia distribuido y colaborativo, aunque su uso se extiende a otras aplicaciones. Surge con WorldWideWeb en 1990.

La primera versión (HTTP 0.9) permitía simplemente transferencia de datos en bruto por Internet. HTTP 1.0 (RFC 1945) mejora al incorporar formato tipo MIME (RFC 1521) con metainformación respecto de los datos y modificadores en la semántica petición/respuesta. Pero causa problemas de prestaciones en el servidor: backoff de cada conexión, una conexión por elemento de la página en lugar de una única.

El cliente envía un método, URI y versión del protocolo, y después pueden ir modificadores de la petición en formato MIME con información sobre las capacidades del cliente y el posible contenido del cuerpo. Acaba la petición una línea en blanco.

El servidor devuelve una línea de estado, que incluye la versión del protocolo y un código de error (o éxito) seguido por un mensaje MIME con metainformación del servidor y la respuesta (Content-Type: tipo/subtipo).

text	**multipart**	**message**	**appplication**	**image**	**audio**	**video**	**model**
plain richtext html sgml	mixed voice-mesage encripted signed	rfc822 news http	postscript pdf zip msword	jpeg gif tiff png	basic 32kadpcm	mpeg quicktime	vrml mesh iges

Separado por una línea en blanco, a continuación aparece el contenido.

8.8 Métodos (peticiones | respuestas)

Hay una línea inicial con el comando (son case-sensitive: distingue mayúsculas de minúsculas) la URL y la versión, una o varias cabeceras MIME, una línea en blanco (nada delante del CRLF) que indica el final de las cabeceras, y espera los datos:

GET	Recupera un objeto identificado por el URI (Se puede usar para invocar un programa cgi-bin). Acepta modificadores (como: If-Modified-Since)
HEAD	Es igual que GET pero devuelve cabeceras HTTP y no el cuerpo. (Para verificar si ha cambiado respecto de la última copia descargada o si sigue siendo válida o ha sido borrado y mantener actualizadas páginas de referencias)
PUT	Especifica que los datos en el cuerpo deben ser almacenados bajo la URL . El archivo debe existir previamente (para crear uno nuevo usar POST y REQUEST)
POST	Crea un nuevo objeto vinculado al objeto especificado. Se considera que es subordinado al objeto especificado (como un fichero y su directorio o un artículo y su newsgroup).
DELETE	Pide que el servidor borre el objeto de la URL, que deja de ser válida.
CHECKOUT	Parecido a GET pero protege el objeto contra la actualización de otras personas. El lock o protección se puede romper por el administrador o pasado un tiempo.
CHECKIN	Parecido a PUT, pero libera el lock (cerradura) colocado sobre el objeto con checkout.
SHOW METHOD	Devuelve una descripción breve del comando
VINCULE / LINK	Vincula un objeto existente al objeto especificado.
UNLINK	Desvincula información desde un objeto.
TEXTSEARCH	El objeto puede ser preguntado con una ristra(string) de texto.
SPACEJUMP	El objeto acepta coordenadas de un punto dentro de el objeto. El comando se usa con GET y un URL derivado.
SEARCH	Busca un URL equiparando parte del mensaje adjuntado.

8.9 Obtención de una página

1. El navegador analiza la URL, y se determina la máquina y el puerto.

2. Se establece una conexión TCP al servidor. Previamente habrá resuelto el nombre con DNS y habrá puesto el puerto 80 si no se especificó otro alternativo.

3. Se envía la petición(HTTP). La primera línea contiene el comando, el identificador del objeto y la versión del protocolo:

GET http://www.gsyc.inf.uc3m.es/~jjmunoz HTTP/1.0

Después se envía información adicional sobre el cliente mediante cabeceras como la RFC822: < HTRQ Header > = < Fieldname >: < Value > < CrLf >

El modelo inicial no enviaba cabeceras MIME ni esperaba como respuesta el tipo.

4. Se espera la respuesta (código de error y si es el 200, contenido de la página o cabecera si es lo pedido), y se cierra la conexión.

La respuesta es: (Línea de Estado) ::= (Version HTTP) (Código de Estado) (Explicación)(CrLf)
habitualmente: **HTTP/1.0** **Status 200: Document Follows** (MIME: version...)
Las cabeceras de respuesta (en formato RFC 822) sirven para identificar sobre todo el tipo, pero también el servidor. Después va el contenido, de esta manera:

Línea en blanco (CrLf)
Contenido (en formato ISO Latin-1 8859-1).
Línea en blanco (CrLf)

5. Si la página tiene otros componentes (imágenes), se repite el proceso para cada uno

El servidor está esperando la petición, la analiza (algunos no entregan sus páginas si el cliente no es lo suficientemente avanzado), localiza la página (o la genera), comprueba permisos de acceso y la devuelve (precedida de cabeceras).

8.10 Ejemplos

telnet www 80

GET http://www/~jjmunoz HTTP/1.0

HTTP/1.0 302 Found
MIME-Version: 1.0
Server: CERN/3.0
Date: Monday 24-Nov-97 16:45:47 GMT
Location: http://lareal.gsyc.inf.uc3m.es/~jjmunoz
Content-Type: text/html
Content-Length: 397

<HTML>
...
</HTML>

telnet otro.lugar 80

GET http://otro.lugar

<TITLE>..... (En este caso sin cabeceras)

telnet otro.lugar 80

GET http://otro.lugar.mas HTTP/1.0

HTTP/1.0 404 Not Found
Server: Netscape-Enterprise/2.0d
Date:
Content-Length:
Content-Type: text/html

<!DOCTYPE HTML PUBLIC...

telnet y.otro.mas 8099

GET / HTTP/1.1
Server: Apache/1.3b3-dev

8.11 HTTP 1.1

La última versión (tras 2 años) en este momento es la 1.1 (RFC 2068). Permite:

- Conexiones persistentes: en lugar de establecer una conexión, traer un objeto y luego cerrar, cuando muy probablemente tendrán que establecerse y cerrarse otras varias, no terminar esa conexión y aprovecharla para bajar más objetos. Tiene un gran efecto sobre las prestaciones (percibible). Antes se usaba (Netscape 2.0 y Apache 1.1) la cabecera Connection: Keep-Alive pero no funcionaba a través de proxies, ni servía para páginas creadas dinámicamente.
- hosts virtuales sin dirección IP: un mismo servidor responde páginas de una organización o de otra según lo pedido en la URL, pero en una misma dirección IP y puerto TCP.
- tipado y negociación de la representación de los datos: en caso de que esté disponible en varios formatos, automáticamente que lo traiga en el preferido (en pdf o en html)
- recuperación de un documento desde donde se había cortado: como ha ocurrido con ftp, esta facilidad se ha hecho necesaria por la congestión de la red y los ficheros muy grandes. Supone un beneficio también para el resto de la comunidad (ahorra ancho de banda).
- Chunked transfer encoding : ir generando fragmentos de respuesta con sus respectivas longitudes.

La evolución es constante, incorporando cosas como que el cliente no tenga que esperar a recibir una respuesta del servidor para enviar una nueva petición, que las cookies se normalicen (RFC 2109, ligeramente distintas de las de Netscape) o que se propongan un mecanismo de autenticación mediante extracto (digest): RFC 2069

Aunque no ha tenido mucha difusión, SUN propuso **WebNFS,** un protocolo alternativo para descargar bajo demanda sólo partes de la página a visualizar, de manera que según el usuario avanza va descargando más partes del documento (con HTTP han de bajarse completas, y si los gráficos no tienen su tamaño definido, los visores no visualizan nada hasta tenerlos. Por todo ello, se recomienda que las páginas sean pequeñas y que para documentos grandes se emplee Acrobat o Envoy).

HTTP es un protocolo rico y complejo que sigue en evolución. HTTPng tendrá características que permitan utilizarlo para aplicaciones distribuidas de otros tipos , con redirección a otros protocolos para audio/video (no necesariamente sobre transporte TCP) y con nivel de presentación (tipo XDR).

8.12 HTML

(HyperText MarkUp Language) es un lenguaje basado en marcas (tags) para identificar las partes del documento. Define la estructura del documento, no su apariencia (layout). El programa cliente interpreta el documento recibido (y recupera todos sus componentes gráficos si no se ha optado por prescindir de las imágenes) y lo visualiza, por lo que es el *navegador* quien decide la apariencia final.

No todos los navegadores muestran igual un mismo documento. Es más, algunos si detectan algún defecto en la secuencia de códigos HTML de la página optan por no visualizarla.

La documentación se genera con editores (basta uno de texto plano, aunque conforme se añade riqueza van siendo necesarios otros especializados y con capacidad de gestionar el conjunto de las páginas). Se almacena en los servidores y los clientes solicitan, recuperan y muestran el documento elegido. (Además también pueden recuperarse para otros fines, como indexar la información). Para el texto usa ISO 8859 (Latin-1), que tiene los caracteres de los idiomas europeos (á = á ...).

HTML se parece a una particularización (DTD) del standard ISO 8879:1986, Information Processing- Text and Office Systems. **SGML**: Standard Generalized Markup Language http://www.sgmlopen.org que era lo que se utilizaba en el CERN para que la documentación generada por tantos grupos disjuntos fuese integrable y compartible.

Es un meta-lenguaje formal sin semántica, que se concreta mediante DTDs (Document Type Definitions) que definen la información de estilo para formatear su presentación. Los datos y la estructura son independientes (esto es bueno cuando grupos de personas colaboran en introducir información que debe tener apariencia similar, a diferencia de los procesadores WYSIWYG.)

http://www.ietf.cnri.restos.va.us/html.charters/html-charter.html

http://www.w3.org/hypertext/WWW/MarkUp/MarkUp.html.

8.13 Estructura de un documento HTML

```
<HTML>
<!-- Un comentario -->
        <HEAD>
            <TITLE>Título del documento</TITLE>
        </HEAD>

        <BODY>
            Contenido del documento
        </BODY>

        <ADDRESS>
            Parte final del documento (referencia). Habitualmente:
            <a href="http://www..."> Mi página Web </a>
            <a href="mailto:yo@aqui.es>Escríbeme </a>
        </ADDRESS>
    </HTML>
```

Los tags son directivas entre corchetes angulares < y >. Si existe la correspondiente directiva de terminar la zona de influencia, se llama igual precedida de /. No hay diferencia entre mayúsculas y minúsculas en las directivas.

Los tags van anidados unos dentro de otros. Hay navegadores que comprueban la corrección de la versión de HTML utilizada (se puede "certificar" que una página cumple con un determinado estándar, en la red (http://validator.w3.org) o mediante ciertos editores). Hay visores que no muestran las páginas que no están correctamente escritas. Por ejemplo, Air Mosaic no mostraría nada del documento si se encuentra algo (incluso un comentario) antes de <HTML>. La tendencia a esa verificación automática es para usuarios quieren más posibilidades y acuden a SGML (ahora a XML) para aplicaciones concretas (expresión de fórmulas matemáticas, moléculas...)

8.14 Elementos HTML

Los más significativos en la definición de la estructura del documento son:

- Las cabeceras(headings) y párrafos, que definen el nivel dentro de la estructura: <H1> <H2>...<P>
- Las listas, ordenadas o no, con sus items: <UL>, <OL>, <MENU>, <DIR>, <LI>
- Los hiperenlaces (anchor) con referencias a otros documentos:

  <A HREF="URL donde ir">texto que aparece</A>
- El texto que aparece puede ser una imagen

  <IMG ALIGN=TOP ALT="Marc's Picture" SRC="marcpic.gif"></IMG>
- Texto preformateado. <PRE>

Y además se incorporan:

- Mapas: imagenes con asociaciones de zonas de la imagen a URLs
- Frames: partes de la ventana tratadas como zonas independientes.
- Tablas
- Formularios

Los tags relativos a la apariencia (fonts, blink, italic) se consideran dañinos por parte del consorcio w3 porque dificultan el mantenimiento de la información (Tidy Program). Por ello se tiende a dejar esos detalles a las hojas de estilo.

http://www.ietf.reston.va.us/home.html
http://www.w3.org/hypertext/WWW/Consortium/Prospectus
http://www.w3.org/hypertext/WWW/Markup/Markup.html

8.15 Versiones de HTML

0	texto
1	Incorpora imágenes, hiperenlaces y listas
2	Incorpora formularios, mapas de imágenes activas
Mozilla	blink, backgrounds, fonts, figure, alignment, centering http://www.netscape.com/home/services_docs/html-extensions.html
3.0	(propuesta el 28 de marzo y aceptada a finales de 1995): tablas, ecuaciones, toolbars, footnotes http://www.hpl.hp.co.uk/people/dsr/html/CoverPage.html
3.2	La gran (otra más?) esperanza de unificación de las extensiones: frames... http://www.w3.org/Markup/Wilbur
4.0	Definida en Julio de 1997: hojas de estilo, internacionalización (atributo LANG),accesibilidad (Braille, sintetizadores de voz), novedades en tablas y formularios, inserción de scripts, multimedia (tag OBJECT)... http://www.w3.org/TR/REC_html40

8.16 Referencia de HTML

CÓDIGO	*ATRIBUTOS DEL CÓDIGO*	*VERSIÓN HTML*
A	HREF=“URL”	HTML2
	HREF="nombrefichero"	HTML2
	NAME="nombre”	HTML2
ADDRESS	NINGUNO	HTML2
B	NINGUNO	HTML2
BASE	HREF=“URL”	HTML2
BASEFONT	SIZE=n (1-7)	NETSCAPE
BLOCKQUOTE	NINGUNO	HTML2
BODY	NINGUNO	HTML2
	BACKGROUND=“URL”	HTML3
	BGCOLOR=#rrggbb	NETSCAPE
	LINK=#rrggbb	NETSCAPE
	TEXT=#rrggbb	NETSCAPE
	VLINK=#rrggbb	NETSCAPE
BR	NINGUNO	HTML2
	CLEAR=LEFT, RIGHT 0 ALL	HTML3
CENTER	NINGUNO	NETSCAPE
CITE	NINGUNO	HTML2
CODE	NINGUNO	HTML2
COMMENT	NINGUNO	HTMl2
DD	NINGUNO	HTML2
DFN	NINGUNO	HTML2
DIR	NINGUNO	HTML2
DL	NINGUNO	HTML2
DT	NINGUNO	HTML2
EM	NINGUNO	HTML2
FONT	SIZE=n; SIZE=+n O -n	NETSCAPE
FORM	ACTION=“URL”	HTML2
	METHOD=GET 0 POST	HTML2
Hn	NINGUNO	HTML2
	ALIGN=CENTER	HTML3
HR	NINGUNO	HTML2
	ALIGN=LEFT, RIGHT 0 CENTER	HTML3
	NOSHADE	NETSCAPE
	SIZE	NETSCAPE
	WIDTH=n%	NETSCAPE
	WIDTH=n%	NETSCAPE

CÓDIGO	*ATRIBUTOS DEL CÓDIGO*	*VERSIÓN HTML*
I	NINGUNO	HTML2
IMG	ALIGN=TOP, MIDDLE O BOTTOM	HTML3
	ALIGN=LEFT O CENTER	HTML3
	ALT="texto"	HTML2
	BORDER=n	NETSCAPE
	HEIGHT=n	HTML3
	HSPACE=n	NETSCAPE
	ISMAP	HTML2
	SRC	HTML2
	VSPACE=n	NETSCAPE
	WIDTH=n	HTML3
INPUT	ALIGN=TOP, MIDDLE 0 BOTTOM	HTML2
	CHECKED=TRUE 0 FALSE	HTML2
	MAXLENGTH=longitud”	HTML2
	NAME="nombre"	HTML2
	SIZE="tamaño"	HTML2
	SIZE="ancho,largo"	HTML2
	SRC=“dirección"	HTML2
	TYPE=“TEXT, PASSWORD, CHECKBOX, RADIO, IMAGE,HIDDEN, SUBMIT, RESET	HTML2
	VALUE	HTML2
ISINDEX	NINGUNO	HTML2
	ACTION=“nombrefichero”	NETSCAPE
	PROMPT=“texto”	HTML3
KBD	NINGUNO	HTML2
LI	NINGUNO	HTML2
	TYPE=A,a,I,i 0 1	NETSCAPE
	VALUE=n	NETSCAPE
LISTING	NINGUNO	HTML2
MENU	NINGUNO	HTML2
META	HTTP-EQUIV="REFRESH"	NETSCAPE
	CONTENT=“n;URL=URL”	NETSCAPE
NOBR	NINGUNO	NETSCAPE
OL	NINGUNO	HTML2
	START=n	NETSCAPE
	TYPE=A,a,I,i 0 1	NETSCAPE

CÓDIGO	*ATRIBUTOS DEL CÓDIGO*	*VERSIÓN HTML*
OPTION	NINGUNO	HTML2
	SELECTED	HTML2
	VALUE	HTML2
p	NINGUNO	HTML2
	ALIGN=CENTER	HTML3
PLAINTEXT	NINGUNO	HTML2
PRE	NINGUNO	HTML2
S	NINGUNO	HTML2
SAMPLE	NINGUNO	HTML2
SELECT	NINGUNO	HTML2
	MULTIPLE	HTML2
	NAME	HTML2
TABLE	NINGUNO	HTML3
	BORDER="N"	HTML3
	WIDTH="N"	HTML3
	HEIGHT="N"	HTML3
	CELLSPACING="N"	HTML3
	CELLPADDING="N"	HTML3
TD	NINGUNO	HTML3
	ALIGN="LEFT,CENTER, RIGHT"	HTML3
	VALIGN="TOP, MIDDLE, BOTTOM, BASELINE"	HTML3
TH	NINGUNO	HTML3
	ALIGN="LEFT,CENTER, RIGHT"	HTML3
	VALIGN="TOP, MIDDLE, BOTTOM, BASELINE"	HTML3
	COLSPAN="N"	HTML3
	ROWSPAN="N"	HTML3
TR	NINGUNO	HTML3
	ALIGN="LEFT,CENTER, RIGHT"	HTML3
	VALIGN="TOP, MIDDLE, BOTTOM, BASELINE"	HTML3
	ROWSPAN="N"	HTML3
TR	ALIGN="LEFT,CENTER, RIGHT"	HTML3
	SIZE	HTML2
TITLE	NINGUNO	HTML2
TT	NINGUNO	HTML2
u	NINGUNO	HTML2
UL	NINGUNO	HTML2
VAR	NINGUNO	HTML2
WBR	NINGUNO	NETSCAPE

8.17 Extendiendo HTML

Los documentos se pueden clasificar en:

Estáticos: Solo cambian cuando el autor modifica el fichero.
Dinámicos: Páginas creadas por el servidor en el momento que le llega la petición, por programa que se ejecutan en el servidor (CGI), y que toman información de una base de datos y la muestran mediante una página HTML.
Activos: Una aplicación en el cliente actualiza la presentación. La aplicación se descarga desde el servidor, con el que interactúa.

Hay un tag para forzar que se refresque el contenido periódicamente (el browser vuelva a cargar las páginas). Puede utilizarse para recuperar páginas estáticas que se sabe que el servidor cambia cada cierto tiempo pero que no se generan dinámicamente en cada petición (por ejemplo, un servidor de teletipos). Alto gasto de ancho de banda.

La generación de documentos activos se basa en que al descargarla se bajan instrucciones interpretadas por el navegador (scripts) o aplicaciones ejecutables (nativas Activex o para máquina virtual Java).

Los scripts permiten crear sencillas aplicaciones que respondan a ciertos eventos que captura el navegador (el ratón se posa sobre un objeto, se pulsa clic sobre otro, hay un tic de reloj...) y tienen control sobre lo que visualiza o invoca.

Netscape incorporó a su navegador *LiveScript*, que junto con Sun hizo que evolucionara comercialmente a *JavaScript*. Sólo se parece a Java en la sintaxis y el nombre. Las instrucciones vienen en la propia página HTML en tags que solo interpretan algunos navegadores. La alternativa de Microsoft es VBScript. ECMAscript parece ser la vía hacia la normalización de esta rápida forma de aportar interactividad.

Las imágenes animadas aprovechando el formato GIF y el PNG (Portable Network Graphics) dan también sensación de movimiento: el fichero con la imagen realmente contiene varias, que se van mostrando sucesivamente captando la atención humana.

8.18 DHTML

Dynamic HTML es el último elemento en la “guerra” entre los dos principales fabricantes de navegadores, que han realizado implementaciones distintas aunque terminarán convergiendo a las especificaciones de w3consortium.

Se caracteriza por:

1. **CSS** (Cascading Style Sheets): A nivel 1 permite que el creador de las páginas defina las fuentes y colores con los que desea se visualice (es como definir tags indicando qué apariencia tiene que tener el texto afectado), a nivel posicionado permite el movimiento de objetos (en 3D con la extensión <layer> propietaria de Netscape).

Los estilos permiten a los creadores indicar cómo han de representarse los distintos elementos sintácticos del documento (cabeceras, párrafos), de manera que la definición genérica de documentos por parte de los usuarios permita una apariencia u otra (y en general una unificada para todas las de un servidor) sin intervención humana.

El que sea en cascada significa que puede haber características definidas para todos los documentos, pero dentro de un grupo de trabajo pueden definir otras propiedades que se apliquen a continuación de esas.

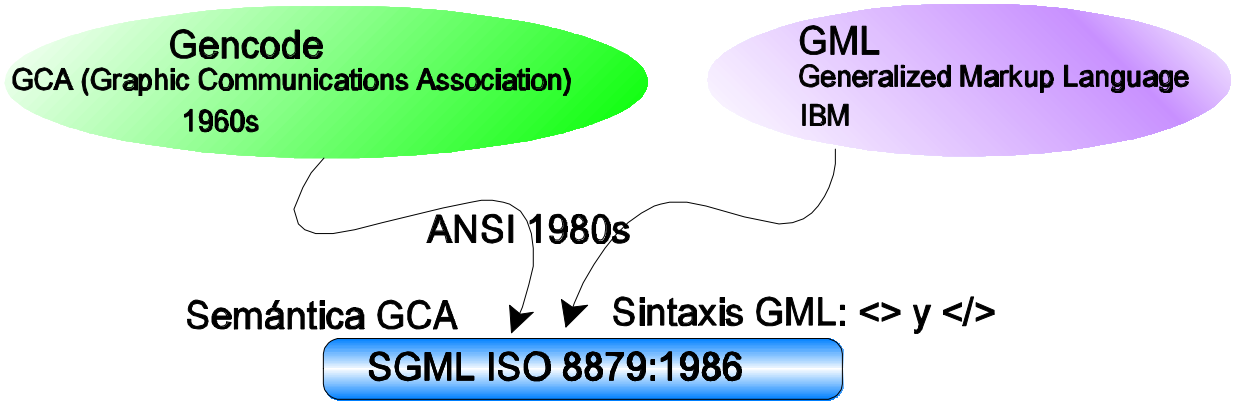
2. Lenguajes de **scripting**:

JavaScript (Netscape)
VBScript (Microsoft)
ECMAscript (European Computer Manufacturer Asociation)

DOM (Document Object Model): API de W3C que define el acceso de los scripts para el control de los elementos del documento.

Las páginas serán más dinámicas, permitiendo que haya imágenes que se muevan de sitio por la pantalla (pueden a su vez ser animadas), que se oculten parcialmente unas a otras al cruzarse, según su profundidad...

8.19 XML (eXtensible Markup Language)



En 1990 Berners-Lee tenía unas hojas de estilo y conceptos de SGML (se usaba en el CERN). Aunque lo usa para formalizar, los documentos HTML no exigen validación (SGML es impracticable: no aprovecha la teoría de compiladores, no permite parsing interactivo ni incremental...)

En 1996 **WC3** se plantea definir **XML** como un alternativa/subconjunto de SGML (con sus cualidades, pero no todas sus características) más fácil de implementar (la especificación es la décima parte), pensado para la red: DTDs simples y validables por los navegadores. Los autores puede diseñar la apariencia con CSS o DSSL (Document Style Semantic and Specification Language).

Características de XML

- formal: pruebas automáticas de validez (como SGML pero sin su complejidad)
- estructurado: permite grandes y complejos documentos (definiendo la DTD: el conjunto de tags y su interpretación)
- extensible: incorporación de nuevas características y tipos de documentos, por lo que puede gestionar grandes repositorios: Con él se pueden definir otros lenguajes (CML: química, representación de moléculas, MathML, HDML para hand-helds, ROML...)

8.20 VRML

Pronunciado [vermul], Virtual Reality Modeling Language, es una manera de describir mundos virtuales. El cliente recupera descripciones textuales y muestra los gráficos tridimensionales en pantalla. En principio los servidores son los mismos, y los clientes también con el correspondiente módulo de ampliación.

Surge en 1994, en la primera conferencia WWW, y en otoño existía una especificación (aprovechando Open Inventor ASCII File Format de SGI). Sus autores, Pesce, Gavin Bell y Tony Parisi tuvieron como criterios de diseño:

- Que fuera independiente de la plataforma
- Que funcionase bien en conexiones de baja velocidad
- Que fuese ampliable

El lenguaje permite definir objetos ilimitadamente complejos, y en la versión 2.0 interactuar con ellos (se abren las puertas poco a poco), asociar comportamientos (animación), sesiones multiusuarios (puntos de encuentro en realidad virtual) y sonido envolvente 3D (el sonido parece proceder de lugares distintos).

El trabajo de representar los gráficos lo hace el cliente, que debe mostrar texturas de los materiales, fuentes de luz, efectos de niebla... Se denomina rendering. Además puede integrar imágenes (jpg) proyectadas, hiperenlaces...

Ej:(Objeto MIME x-world/x-vrml)

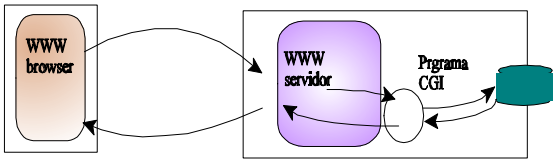
```
Cylinder {
      radius  4
      Height  2
{
```

Cosmos, de SGI, es el entorno más difundido.

8.21 CGI: Common Gateway Interface

Las páginas dimáticas son generadas cuando el cliente selecciona hiperenlaces del tipo <a href=http://www.xxx.es/programa.pl?argumentos>.

Desarrollado en NCSA, CGI es una pasarela que permite que un servidor WWW (httpd) invoque un programa que puede no haber sido diseñado para red. Equivale a que una persona ejecute remotamente un comando en el servidor, y su resultado se convierte en una amigable página HTML.



Constituye un problema de seguridad no solucionable mediante cortafuegos. Toda petición (malintencionada o no) supone que su autor ejecuta un programa en el servidor con los permisos del propietario de httpd: no debe ser root y debe restringirse qué hay en los directorios cgi-bin para que no pueda ejecutar cualquier cosa. Incluso ejecutado sin privilegios, hay que saber exactamente lo que puede hacer cada programa.

Se define un método por el que el servidor obtiene unos parámetros, realiza el parsing (cuidado con la interpretación de sus valores) y ejecuta (en un entorno definido por unas variables) un programa escrito en cualquier lenguaje (shell script, PERL: Practical Extraction & Reporting Language, TCL: Tool Command Language, C, Java...)

El resultado será habitualmente una página Web (también puede ser una imagen o cualquier otra cosa) que se devuelve como respuesta.

Se suele emplear para permitir consultas a bases de datos (que pueden estar en otro servidor de una intranet): Una página HTML contiene formularios que se rellenan. Cada variable toma su valor y se envía por HTTP al servidor, que realiza la consulta y devuelve una página construida específicamente para esa consulta: unifica el interfaz de cualquier aplicación al amigable web.

El resultado del programa invocado (CGI) debe comenzar con una cabecera, que puede ser de 2 tipos:

Content-type: text/html
↩
(A lo que sigue el contenido en HTML en este caso)

ó

Location: /path/file
↩
(con lo que se redirige la consulta hacia la página mencionada)

El mecanismo es independiente del sistema operativo, y existen facilidades para desarrollar con rapidez, librerías de perl como cgi-lib/cgi.pm

Los parámetros se pueden pasar de dos formas:

1. Como variables (QUERY_STRING y PATH_INFO): método GET

<a href=http://www.x.es/programa.pl?variable=valor&variable=valor>
(los blancos de convierten en + y algunos caracteres se ponen en hexadecimal)

Y mediante formularios el cliente se encarga de recoger los valores

```
<FORM METHOD=GET ACTION="http://.../programa.pl">
Texto <INPUT VARIABLE = "Texto" ><BR>
INPUT TYPE=submit VALUE="submit">
</FORM>
```

Nota sobre seguridad: tiene el peligro de que si no se hace correctamente el parsing, el servidor puede ejecutar lo que no pretendía (un ; en unix separa comandos: evitar)

2. Como entrada estándar <FORM METHOD=POST...

Recomendado, sin límite de tamaño (lo especifica con la variable de entorno CONTENT_LENGTH), y utilizable junto con variables.

8.22 JAVA

JAVA es un lenguaje (tipo C++ pero fuertemente tipado) y un modelo de sistema basado en una máquina virtual que interpreta pseudocódigo (bytecode).

El objetivo es aumentar la interactividad de las páginas Web, y permitir que los clientes ejecuten siempre software actualizado. Esta idea se ha mantenido en los lenguajes de script, pero ha evolucionado hacia los NC (network computers) como puestos de trabajo independientes volviendo a los orígenes de ésta tecnología que surgió para poner un microordenador en cualquier electrodoméstico. Aunque Java se ha difundido como añadido al web, pretende ser una alternativa a windows con puestos de trabajo ligeros y aplicaciones que corren en máquinas de cualquier arquitectura.

Las páginas HTML incorporan un nuevo tag que invoca una pequeña aplicación (applet) que es cargada desde la red e intrepretada en el cliente (si el navegador tiene activada esa opción): las instrucciones pasan a un procesador software (máquina virtual java) que las ejecuta como si fuera uno de silicio.

Existen compiladores para otros lenguajes que producen bytecode, pero el lenguaje ha tenido éxito y aunque los intérpretes son lentos hay chips que ejecutan bytecode en silicio y compiladores JIT (just in time: en lugar de interpretar java, se compila justo tras la descarga y se ejecuta luego código máquina nativo)

Entre las características del lenguaje, orientado a objetos, destacan

- Alta productividad: solo se emplea el 10% del tiempo depurando, interfáz gráfico sencillo, poco código (reutilización de objetos de librerías)
- Simplicidad (frente a C++): tiene herencia múltiple restringida: hereda declaraciones de métodos pero no su implementación
- Manejo de excepciones y eventos
- gestión de memoria dinámica automática (garbage collect) y sin punteros
- Threads y exclusión mútua
- AWT (Abstract Windowing Toolkit): Interfaz gráfico básico, que se extiende con IFC (Internet Foundation Classes de Netscape)

Tiene librerías de acceso a bases de datos (JDBC), invocación remota de métodos (RMI), pero hay una fuerte iniciativa para el uso de CORBA. Incluso hay una plataforma de comercio electrónico: JECF (Java Electronic Commerce Framework: http://java.sun.com/products/commerce/doc.white.papers.html). La jerarquía de clases se pretende estandarizar: JavaBeans Component Model

El modelo de seguridad se basa en la verificación formal previa a la ejecución (el entorno restringido garantiza los efectos locales) que incluye que las referencias sean correctas, que no se acceda a otros servidores salvo el que descargó el applet, que no se acceda al PC, etc. El applet se ejecuta en una zona estanca del ordenador.

Existen multitud de entornos de desarrollo, siendo el básico el JDK. También existen varias máquinas virtuales: en verano de 1998 HP ha anunciado una de 500Kb.

Microsoft pese a licenciar JAVA tiene estrategia diferente: ActiveX, que combina su tecnología OCX con las facilidades de red, según el modelo DCOM (Distributed Component Object Model, que es un superconjunto de OLE para integración de aplicaciones): descargas parte de aplicaciones Windows dinámicamente y las ejecutas en el NetPC. El modelo de seguridad se basa en la certificación de autoría del código.

El modelo de seguridad de ActiveX es que la aplicación va firmada digitalmente (muchos servidores ftp de software libre ponen junto al ejecutable, su firma en PGP).

En un intento de crear un estándar de facto se ha remitido a OSI y se ha creado la iniciativa de certificación 100% JAVA, para evitar cosas como que Microsoft, que licenció esta tecnología, modifique librerías básicas de manera que lo generado con ellas no se pueda ejecutar en plataformas no Wintel, y que ha terminado en los tribunales de justicia porque va contra el objetivo de SUN (*write once, run everywhere*).

8.23 Webcasting: Push Technology.

Descargarse todas las veces todos los programas que se usan con asiduidad es lento (salvo en LANs). Una alternativa sería dejar una copia en local que se actualice por la red cuando sufra cambios. Esto puede hacerse de dos maneras:

- POP: el cliente toma la iniciativa de descargarse una serie de archivos que almacena localmente para utilizarlos en modo desconectado.
- PUSH: el servidor envía automáticamente cualquier novedad a sus "afiliados". De esta manera el cliente no tiene que estar pendiente y las actualizaciones le llegan en cuanto salen.

Si el navegador está escrito en Java, es extensible en el lenguaje. Con unas clases básicas y un procesador que entienda bytecode se tiene un cliente extensible potencialmente capaz de ejecutarlo todo y entender todo formato, ya que se actualiza sobre la marcha. Pero esto mismo se consigue con un ordenador normal al que se le cargue remotamente el software, como hace Marimba Castanet

OSD (Open software description) permite configurar remotamente un parque de ordenadores que tengan que tener una serie de aplicaciones y ficheros de configuración.

La tecnología PUSH se utiliza mucho para difusión de información: un usuario que se apunte a un canal recibirá (en su salvapantallas) cualquier noticia según se vaya publicitando. El servidor más famoso es Pointcast

CDF: Formato de Definición del Canal (Microsoft y Pointcast)
MCF: metaContent File Format (Netscape) para contenidos web
DRP: Distribution & Replication Protocol (Marimba y Netscape)

Entre sus características destacan su consumo de ancho de banda (uso de multicast) y su capacidad de adecuarse al perfil del usuario.

- [Sunsite.unc.edu/boutell/faq/www_faq.html]
- [www.w3.org/hypertext/WWW/Clients.html]
- [www.yahoo.com//Computers/World_Wide_Web/Browsers]

8.24 Localización de la información requerida

Navegar por la enorme cantidad de servidores y páginas supone requiere una enorme cantidad de tiempo y no garantiza localizar información sobre un tema concreto (y además hay riesgo de verse enganchado por encontrar por casualidad algo atractivo pero no buscado). Se critica que en Internet hay muchos datos, pero no información.

Los spiders, crawlers, wanderers, knowbots o robots son programas que ojean cuan persona sin otra cosa que hacer, e indexan la información de la red. Sobre sus índices atienden posteriormente consultas interactivas basadas en palabras a cotejar con el contenido de las páginas visitadas: ¿qué documentos contienen la palabra XXX?

Hay normas para que un indexador no sature a un servidor mientras lo indexa (<META VAR="Robots" VALUE="noindex">). Cada servidor podría crear un índice (quizá con wais) y dejar disponible(como en gopher): es mejor usar la propia CPU para indexar cada cierto tiempo que ver consumido el ancho de banda del enlace mientras las páginas son solicitadas sistemáticamente por los robots.

El orden de recorrido (depth-first vs. Breath-first) y la evitación de bucles es un tema importante, así como la parte del cyberespacio que se indexa (Altavista y Hotbot, que son los que más páginas visitan, sólo cubren el 30% de la red, y a 10.000.000 de páginas al día tardan un mes en actualizarse).

Estas herramientas se financian por la publicidad que se va actualizando cada vez que alguien quiere hacer una consulta. Cada una de estas herramientas pretende indexar más páginas, mejor (tesauros, proximidad de las palabras, etc) y en consecuencia dar respuestas más afinadas que atraigan a más usuarios haciendo más vista su publicidad (cobran por hits).

En España, Olé, Ozú, Magallanes... http://www.rediris.es/doc/buscadores.es.html

8.25 Buscadores

En primavera de 1996 se calculaba que había 20.000000 de páginas en Internet. En verano del año siguiente, 150.000.000, duplicándose cada 4 meses. Tantos datos no se convierten en información si no hay una forma rápida de localizar lo que se necesita. Y como no hay ninguna organización en la información(surge anárquicamente), sólo con la ayuda de aplicaciones de la propia red es posible abordar esta función: webrobots, conocidos como crawlers, spiders, worms, walkers...

Se necesitan dos fases:

1. Construcción de un índice: un robot navega por la red y va anotando qué palabras significativas encuentra en cada página, de la que anota su URL.

- Se parte de un conjunto de URLs (semilla) y se aplica la recursión (breath-first o depth-first)
- Pueden predefinirse particiones del ciberespacio (solo buscar en el dominio .es, por ejemplo)
- La frecuencia de actualización puede alterar la calidad de las respuestas
- El número de lugares indexados no es garantía de buenos resultados, porque suele suponer menor actualización y más ruido.
- Algunos anotan un bit que indica que aparece la palabra, otros ponderan con números que indican "cuanto" aparece en el documento (o en sus primeras líneas) y en ocasiones se busca mayor conocimiento semántico (una palabra significa una cosa u otra según las palabras que la acompañan).

2. Localización de documentos según parámetros de búsqueda

- Capacidad de búsquedas complejas (+ & " " ...)
- Relevance feedback: ir afinando a base de indicar si un documento recuperado satisface
- Incluso pueden usarse tesauros para identificar sinónimos, o traducir vocablos al inglés...

Tras la búsqueda se obtienen referencias de documentos por los que habrá que navegar (tener 15.000 referencias es como no tener nada) para ver si pueden ser útiles.

Los usuarios pueden pedir que se indexen sus páginas (conviene hacerlo tras cualquier actualización importante), y pueden también indicar si no debe indexarse (fichero robots.txt) por tratarse de información cambiante o para no saturar su enlace.

8.26 Clasificación de buscadores

Para clasificar un documento, pueden usarse datos objetivos (autor, fecha), u otros más interesantes pero menos objetivos, relacionados con el contenido. Aunque existen tags (<META> de HTML 3) para especificar palabras significativas, el ruido (páginas encontradas que realmente no sirven) es un problema grave. Si el objetivo del buscador es ser exhaustivo, su calidad vendrá dada por el cociente relevantes/encontradas, y si es la especificidad, la precisión será relevantes/recuperadas.

Ante esta situación hay dos estrategias claramente diferenciadas:

1. Registration sites: pides que te indexen y si lo consideran oportuno ponen una referencia en su índice organizado por materias:

- Yahoo (http://www.hayoo.com) Yet Another Hierarchically Obstreperous Oracle ha ganado su reputación por la selección que realiza su experta plantilla: aunque el criterio sea subjetivo, buscan "lo mejor de lo mejor", teniendo en cuenta además de la calidad de la página su respeto a la propiedad intelectual y la adecuada categorización.
- W3 Virtual Library (http://vlib.stanford.edu/Overview.html)
- http://www.w3.org/hypertext/DataSources/bySubject/overview.html
- Magellan, Galaxy

2. Indexadores de fuerza bruta: automáticamente navegan y crean un índice donde su programa localiza las referencias. Necesitan muchos recursos (CPU en la indexación y recuperación, disco para el índice y ancho de banda tanto para navegar como para atender a los usuarios concurrentes). Se diferencian por la frecuencia de actualización, la amplitud del espacio de búsqueda y la sofisticación de las búsquedas:

- Altavista (http://www.altavista.digital.com) tiene un robot llamado scooter que indexa texto completo (asume que las primeras líneas son el abstract). Se le puede pedir que indexe una página (anticipando así la visita que tenga para ella programada). Surgió como demostración de la capacidad del procesador alpha, pero ha supuesto para Compaq una división de software de Internet.
- Webcrawler (http://webcrawler.cs.washington.edu/WebCrawler/WebQuery.html)
- Hotbot (www.hotbot.com). Su robot se llama slurp. Se basa en una red de wokstations en paralelo.
- Inktome: de Berkeley, basado en superdordenadores escalares.

También existen modelos híbridos

- InfoSeek (http://www.infoseek.com) (Cobraba $10 al mes por una búsqueda mejorada que incluye proximidad de palabras...) Indexa html y pdf, y tiene búsqueda de imágenes
- Lycos (http://www.lycos.com) construye un índice donde buscar usando heurísticos. Indexa títulos, cabeceras... y se queda con los 100 términos que más han aparecido. Tiene fama de poco ruidoso. http://lycos.cs.cmu.edu
- Excite (http://www.excite.com) indexa decenas de millones de páginas.

Aunque hay muchos estudios comparativos sobre las bondades y defectos de estos buscadores, hay que resaltar que los resultados dependen mucho de la destreza del usuario para afinar las búsquedas, aprovechando las opciones avanzadas que permiten especificar que aparezcan conjuntos de palabras relacionadas entre si (and, " "...)

Además existen metabuscadores: buscan en otros, ponderan lo obtenido tras un plazo prefijado y dan a elegir. Ofrecen como valor añadido un interfaz uniforme aunque puede ser difícil afinar las consultas:

- Metacrawler, (http://www.metacrawler.com) surge en la Universidad de Wasington (WU). Busca en Lycos, Infoseek, WebCrawler, Excite, Altavista y Yahoo.
- StartingPoint, NetLocator, InfoMarket

Hay programas para PC que realizan esta función.

Para construir un buscador, basta poner un programa que indexe (excite) en un ordenador con disco, acceso a los servidores a indexar y tiempo para emplear en ello, y un servidor web con un interfaz (CGI) para realizar las consultas. Si lo que se pretende es indexar páginas del propio servidor, puede utilizarse waisindex (sgml.src, sighyper.src). Para más información, http://osiris.wu_wien.ac.at/infocenter/searchwww.html

http://www.searchinsider.com, http://searchenginewatch.internet.com

8.27 Agentes (Knowbots).

La recopilación automática de información por parte de servidores es también posible desde los puestos de trabajo del los usuarios. Los agentes inteligentes podrán hacer esta operación de forma adecuada, al tener entrenamiento sobre los gustos y costumbres de sus usuarios humanos. Indexarán sólo las partes que les interesen, lanzarán consultas a varios robots para agrupar y refinar automáticamente las respuestas... Podrán programar la replicación de partes de otros servidores “al gusto de su humano” para éste pueda luego ojearlos con más rapidez.

Esto puede potenciarse con el uso de XML, que está pensado para que los ordenadores puedan manejar semántica y por tanto la red se utilice más para que los ordenadores trabajen por los humanos y no sean estos quienes gasten su tiempo.

El asistente automático que organizará nuestro trabajo (ya hay iniciativas para que las agendas se sincronicen mediante correo electrónico a fin de concertar entrevistas entre personas sin la intervención de estas, buscando huecos comunes) y se comunicará por nosotros con el resto de recursos de la red, simplificándonos el acceso a la información en un modelo en el que la información va hacia el usuario en función de su perfil y se actualiza según su gusto.

Los ordenadores deben mejorar la calidad de vida, y en la sociedad de la información, en lugar de esclavizar a los humanos, deben potenciar sus capacidades ayudándoles a absorber la cantidad y variedad de datos.

http://www.w3.org/hypertext/WWW/FAQ/Bootstrap.html

8.28 Publicitación vs publicación

Se está acuñando la palabra publicitación para denominar la disposición de información para que sea accesible desde Internet. La rapidez de la puesta a disposición de la información (no hay que imprimir y repartir las copias) supone que cualquier persona del mundo puede obtener la misma información prácticamente en el mismo momento (máxima sensación de igualdad).

Para quien publicita es más cómodo (instantáneo desde que se produce la información hasta que se pone a disposición) y barato (basta un espacio en un servidor). El modelo cambia respecto de la publicación ya que es quien quiere acceder a la información quien tiene que tomar la iniciativa de pedirla.

La publicitación de cierto tipo de información está siendo objeto de debate por el escándalo que ha supuesto la existencia de personas que utilizan la red para fines "dudosamente éticos". Si bien es cierto que cualquiera podría publicar información "peligrosa" y difundirla, la limitación de su propia capacidad para dar publicidad a esa información acota los efectos y además existen mecanismos tradicionales para perseguir la acción. La globalización genera una inseguridad jurídica e imposibilidad de que ningún gobierno pueda actuar (cursos de fabricación de armamento, pornografía...)

En Internet la publicitación es instantánea, de ámbito mundial, y además existen herramientas automáticas de tratamiento de la información que potencian la facilidad para acceder a este tipo de información. La persecución de la información en origen es complicada porque puede residir en cualquier país o paraíso fiscal... El filtrado de la información en destino (padres que quieren evitar que sus hijos vean ciertas cosas, ver www.aui.es e iniciativa P3P) supone un problema de actualización de la configuración: la información puede localizarse en multitud de réplicas.

Hay que prestar atención no sólo a la calidad de la información, o a su apariencia, sino también a su estructura (el acceso actual a Internet no es tan "rápido" como a un CD-ROM, por lo que todo atajo para llegar a lo realmente deseado es interesante).

La apariencia de la primera página puede ser fundamental para despertar el interés del visitante o que pase a otra de las millones de ofertas (nunca hay una segunda oportunidad de producir una buena primera impresión). Pero si solo es un PoP, nadie querrá volver a pasar por ahí más adelante.

Dado que muchos servidores web se financian mediante banners publicitarios, los estudios sobre convivencialidad de sus interfaces cada vez afinan más. Unos sencillos consejos a tener en cuenta a la hora de crear un servidor son:

- Nadie espera más de 10 segundos para ver una página
- Conviene definir siempre el tamaño de las imágenes (para que los visores muestren el texto según se va cargando)
- Es bueno tener un site-map para que el cliente pueda conocer la organización de la información.
- No hay que asumir que el usuario conoce cómo está organizado el servidor
- Los frames con un índice permanente e imagen corporativa se están imponiendo
- Las páginas deben ser cortas (usar pdf para documentos voluminosos): solo el 10% de los usuarios hacen scroll para ver la parte inferior de la página
- Toda página debe quedar identificada (para poder ponerse en contacto con el autor)
- Si se utilizan las opciones más novedosas, puede que haya usuarios que se queden fuera
- El exceso de animación y parpadeo tiene efectos negativos para la atención del usuario
- Define la página para un tamaño de visualización razonable, pero vigila su apariencia en otros
- La apariencia de la primera página puede evitar que el usuario ahonde en el servidor
- La falta de actualización o de contenido provoca también la falta de popularidad.

8.29 Publicitación y propiedad intelectual

La manera más rápida de escribir una página que se parezca a una que has visto en Internet es copiartela y editar su contenido. Los navegadores tienen opciones para ver el código e incluso editarlo.

Pero el contenido (el texto) puede también ser copiado y publicitado con o sin alteración, en el mismo u otros idiomas, y lo mismo se aplica a las imágenes

El trabajo de digitalización de las obras de un museo podría ser aprovechado por otras personas para obtener beneficios comerciales por la venta de un trabajo que no han realizado. El marcado digital de la información de manera que se pueda determinar si ha habido reproducción ilegal es un tema importante si queremos que las personas que producen documentos de calidad los publiciten en lugar de publicarlos.

Es necesario que ciertas entidades actúen de certificadoras para que un autor pueda registrar su creación y que las herramientas gráficas automáticamente se conecten a estos lugares y verifiquen. Las leyes nacionales de protección de la propiedad intelectual necesitan una globalización acorde a la de la red.

El negocio de la música sobrevive bien en los países occidentales pese a la facilidad que existe para piratear. Pero ya hay problemas con China por no respetar los derechos de autor. Si no hay gente que pueda vivir de la producción de información en Internet, no habrá más dedicación que la de los voluntarios sin mayor compromiso.

En el caso de aplicaciones en Java, el modelo de distribución de software es todavía más complejo. Pagar por uso puede no ser una solución sencilla, porque puedes guardar una copia localmente y utilizarla cuanto quieras. Proteger las aplicaciones con fechas de expiración puede ser una solución.

8.30 Comercio electrónico

El modelo es bueno para la distribución de información comercial, de la administración, de la comunidad científica.

Pero cuando tener un punto de presencia publicitario no es suficiente, surge la necesidad de realizar transacciones con efectos económicos. Y entonces surge el problema de la seguridad, ya que enviar un número de tarjeta de crédito por la red puede ser tan fácil como que alguien obtenga ese número y lo utilice para comprar a tu costa. Y las reclamaciones son difíciles, en especial cuando se cruzan las fronteras (inseguridad jurídica: ¿Dónde reclamar? ¿Qué legislación se aplica? ¿Es delito allí?)

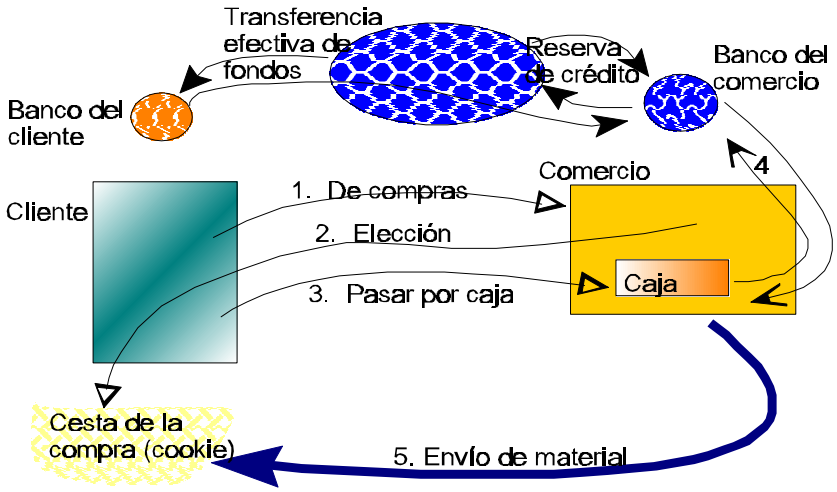
Se puede navegar por ciber-malls, con atractivos entornos VRML y chats, IRCs o MOOs donde “charlar” con los amigos. Mediante micropagos (monedero electrónico) se pueden encargar bienes o servicios de menor cuantía sin demasiada complejidad y sin que se pueda extraer un perfil de lo que adquiere cada usuario.

Cuando hay que pagar cantidades mayores, se pueden emplear tarjetas de crédito. Aunque el número de delitos con las tarjetas físicas (¿quien vigila qué hacen con ellas cuando las llevan a la trastienda?) es mucho mayor que los problemas acaecidos en la red, será difícil cambiar la cultura social si no existe una garantía de que si alguien usa correctamente su tarjeta nada puede pasarle.

SSL, (en el que se basa https) permite enviar datos por Internet de forma cifrada (tras una negociación inicial se acuerda una contraseña para cifrar el flujo de información empleado para enviar el número de la tarjeta), de manera que alguien que capture tramas en un enlace no pueda hacerse con el número de la tarjeta y utilizarla.

Aunque pueda parecer simple, es necesario un modelo completo que vincule al titular de la tarjeta, a su banco, a la sociedad de la tarjeta, al vendedor y a su banco, de manera que cualquier transacción pueda trazarse y determinar quien no ha cumplido.

Visa, MasterCard (y American Express) han acordado un estándar (SET) para la utilización de dinero electrónico, de manera que los paseos por los centros de ventas que permiten desde hace tiempo comprar hasta zapatillas de deporte tengan una garantía para compradores y vendedores:



Existen ya muchas experiencias exitosas (Amazon, CISCO, Dell), y aunque las espectativas (sobre todo en Europa) prevén una implantación relativamente lenta, los efectos sobre los precios de la eliminación de intermediarios y los beneficios de la distribución masiva sin costes de exposición terminarán teniendo su influencia en el cambio que se está produciendo gracias a Internet y del que todos somos cada vez mas conscientes.

ÍNDICE