

Introducción a Internet

Antes de comenzar a diseñar sitios Web se deben tener en cuenta algunas cosas que explicaré en esta introducción.

Para realizar sitios Web hay que saber de qué se tratan, diferenciar los elementos que los componen, conocer algunos lenguajes de diseño y programación -comenzando por HTML- y tener ciertas herramientas que permiten su realización.

Todo esto está al alcance de cualquiera de nosotros, ya que con una conexión a Internet se tiene acceso a las herramientas -todas gratuitas- y leyendo tutoriales, como éste, se aprende de qué se tratan los sitios Web así como los lenguajes que se requieren para desarrollarlos.

A pesar que usualmente utilizamos la palabra página para definir a un sitio Web, ésta solamente se refiere a lo que nos muestra en pantalla en un determinado momento el navegador; mientras que un sitio es el conjunto de estas páginas que forman parte del mismo.

Por ejemplo, lo que ves en pantalla en este momento es una página que forma parte del total del sitio Web llamado Youbioit.com, para poner otro ejemplo; Youtube sería un sitio mientras que al ver un video en particular se abre una página. Así como cada página forma parte de un libro; un sitio Web vendría a ser como un libro hecho por páginas Web.

Aclarado esto pasamos a ver cuáles son los componentes que hacen a una página y las herramientas que necesitamos para crearlos.

Hay muchos tipos de páginas, las hay estáticas únicamente con texto, otras presentan imágenes, otras presentan animaciones con efectos visuales, otras son interactivas y esperan una respuesta u orden nuestra -por ejemplo los buscadores-, etc. Basta con navegar un poco para darse cuenta del gran matiz de sitios que uno se encuentra en la red de redes.

Al principio, en la década de los '90s las páginas de los sitios Web solían ser estáticas con texto negro y fondo gris o blanco. Con el paso del tiempo comenzaron a agregarles imágenes, fondos artísticos y hasta alguna melodía que sorprendía al visitante del sitio -debo confesar que para nuestros tiempos de hoy todo eso resulta muy aburrido, pero en aquel entonces era lo último de lo último, una maravilla de la tecnología-. Luego, al ir creciendo Internet, llegaron los directorios y buscadores que nos facilitaban la

búsqueda de sitios e información a través de la red; más tarde hicieron su aparición los sitios vistosos con animaciones en Flash y finalmente lo que hoy conocemos como Web 2.0.

Web 2.0 es un término que define a la Internet de la actualidad en la que los sitios son interactivos y están hechos por los mismos usuarios.

Antiguamente, las páginas las hacían los diseñadores, las subían y luego los usuarios las visitaban. Hoy día son los mismos usuarios los que forman parte de comunidades virtuales como Facebook, Twitter, Wikipedia, Youtubioit, blogs, foros, sitios de ventas online, etc y las hacen crecer agregando contenido de todo tipo. Los usuarios comunes suben la información; escriben los artículos -como yo escribí este tutorial-; suben imágenes, videos, archivos, etc. y las páginas se construyen automáticamente con solamente tocar un botón (obviamente se requiere un programador haya desarrollado el mecanismo de los sitios Web para que los usuarios puedan luego crear las páginas de manera fácil, escribiendo en simples formularios y sin saber diseñar en lenguajes como HTML. Así que Web 2.0 es el período que comienza aproximadamente en el 2004 cuando todos comenzamos a participar de la Web activamente, sin necesidad de saber programación, y lo que permitió que tanta información esté al alcance de todos.

Historia de Internet

Otra cosa que necesitamos saber es de qué se trata Internet y por cuáles elementos está compuesta. Puede que a muchos les aburra leer temas de historia, pero esto puede resultarles muy interesante, además de abrirles la mente sobre ciertas dudas o términos que escuchamos diariamente pero no sabemos bien de qué se tratan. Además para poder hacer sitios Web en HTML es fundamental saber lo que estamos haciendo.

Voy a tratar de ser lo más sintético posible, pero a la vez informarlos bien acerca de cómo se desarrolló Internet y darles una idea de qué se trata, los sorprenderán algunas cosas que van a leer, créanme es así.

Básicamente la idea de Internet nació durante la denominada Guerra Fría entre Estados Unidos y la Unión Soviética, como un proyecto militar de Estados Unidos creado en 1958, el Advanced Research Projects Agency (Agencia de Proyectos de Investigación Avanzada) conocido como ARPA por sus siglas, cuando se lanzó el satélite ruso Sputnik y Estados

Unidos quería tomar la delantera en la carrera espacial -me pregunto qué hubiese pasado entonces si Estados Unidos no hubiera querido llegar a la Luna, hoy quizás no tendríamos Internet, o al menos no como la conocemos-.

La idea principal de ARPA era crear sistemas que permitiesen interconectar a los distintos organismos de defensa del país, como el Pentágono o la Estación de Fuerza Aérea de la Montaña Cheyenne, así como a todos los radares para mantenerse informados al instante de lo que ocurría en cada sector del sistema de defensa nacional.

Ya en 1957 se había realizado la primera demostración de Time-Sharing (uso compartido de recursos) y lo que sería la base de Internet. Gracias a esto, distintas computadoras se podían conectar a una computadora de mayor potencia y utilizar sus recursos al mismo tiempo -o sea lo que hacemos hoy día al navegar y bajar toda la información de distintos sitios, básicamente desde nuestra máquina tenemos acceso a una computadora remota-. Dicha supercomputadora a la que se conectaban otras y brindaba servicios, era un servidor. Nace así el concepto informático de redes e interconexión entre computadoras.

Fue así que bajo la supervisión del proyecto ARPA, el 29 de Octubre de 1969, se pone en funcionamiento la red que conectaba a la facultad de Ingeniería y Ciencias Aplicadas de la Universidad de UCLA con el Instituto Internacional de Investigaciones de Stanford, esta fue el acta de nacimiento de ARPANET -antiguo nombre de lo que hoy conocemos como Internet-.

En esta red se podían enviar paquetes de información y datos, así como muchas otras tareas que erróneamente pensamos que son desarrollos actuales; pero ya existían desde hace décadas; por ejemplo las teleconferencias, ventanas múltiples en pantalla, hipervínculos, publicación de medios, etc.

Más tarde se sumarían a la red en orden cronológico el Centro Interactivo de Matemáticas Culler-Fried de la Universidad de California en Santa Bárbara, California y el Departamento de Gráficos de la Universidad de Utah. Para finales de 1971 ya habían 15 sitios conectados a la naciente red de ARPANET.

Paralelamente en el Reino Unido se desarrolló otra red llamada International Packet Switched Service (Servicio Internacional de Intercambio de Paquetes

de Información). Fue creada en conjunto entre el Servicio Postal Británico, la empresa estadounidense Western Union y la red de telecomunicaciones estadounidense Tymnet.

Durante los próximos años se perfeccionaron las tecnologías que permitían el intercambio de información entre computadoras, incluyendo tecnologías como correo electrónico. Mientras tanto estas redes seguían creciendo al agregarse más y más sitios gubernamentales y de investigación.

Ya en 1985 la NSF United States National Science Foundation (Fundación Nacional de Ciencias de Estados Unidos), creó otra red, la NSFNET, la cual interconectaba universidades.

Finalmente en verano de 1989 tras la aprobación del Consejo de Redes del Gobierno de Estados Unidos, se vincularon las redes de NSFNET y la del sistema de correo electrónico de la empresa de telecomunicaciones MCI, la MCI Mail. Así quedaban las redes abiertas al público comercial y dejaban de ser de uso exclusivo del gobierno, universidades y entidades de investigación.

A partir de ese momento se sumaron otras redes de servicios comerciales de email (sí, todavía no era gratuito tener correo electrónico). Luego se sumarían más y más redes, que proporcionarían servicios varios; generando un conglomerado de redes que daba vida a una superred o red de redes, llamada Internet.

Sin embargo Internet no sería popular hasta la década de los años '90s cuando su uso fue facilitado gracias al sistema de navegación inventado por Tim Berners-Lee en el Centro Europeo de Investigaciones Nucleares CERN (Organisation européenne pour la recherche nucléaire) a finales del año 1990.

Berners-Lee, desarrolló el proyecto World Wide Web (Tejido o Telaraña Mundial) -de ahí lo de www-, inventando un sistema que facilitaba la lectura de información, mediante el uso de un programa de navegación, nació así el primer navegador Web, llamado WorldWideWeb (programado por el mismo Tim Berners-Lee durante la segunda mitad del año 1990 y que luego sería rebautizado como Nexus para evitar confusiones por su nombre que era igual al de la tecnología que representaba), seguido por el Line Mode Browser y el ViolaWWW, siendo este último el primero en popularizarse entre los primeros usuarios de la World Wide Web primitiva. Éste permitía a los

usuarios leer documentos que se bajaban con el navegador y pasar de un documento al otro mediante hipervínculos, también conocidos como links, enlaces o vínculos, entre otros nombres.



Tim Berners-Lee en 1994 en los laboratorios del CERN utilizando el navegador WorldWideWeb creado por él mismo en 1990

El lenguaje que se utilizaba para crear estos documentos pasó a llamarse HTML Hypertext Markup Language (Lenguaje de Marcado de Hipertexto), simplemente porque su componente básico eran los hipertextos (o sea los links o hipervínculos) que nos permitían pasar de un documento o página a otro. Por lo tanto llegamos a la conclusión que los cimientos de la popularización de Internet fueron los links o enlaces que nos permitían ir de una página a la otra fácilmente. Dicho de otra manera, los hipervínculos nos permitían navegar.

El 6 de agosto de 1991 se puso en línea el primer sitio Web de la historia, con dirección info.cern.ch

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

[What's out there?](#)

Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

[Help](#)

on the browser you are using

[Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#) , [X11 Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) , [Mail robot](#) , [Library](#))

[Technical](#)

Details of protocols, formats, program internals etc

[Bibliography](#)

Paper documentation on W3 and references.

[People](#)

A list of some people involved in the project.

[History](#)

A summary of the history of the project.

[How can I help ?](#)

If you would like to support the web..

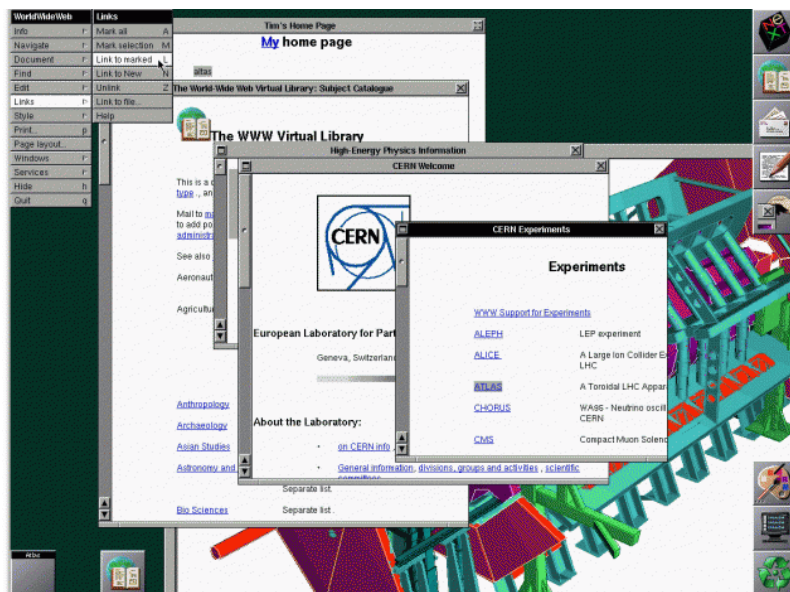
[Getting code](#)

Getting the code by anonymous FTP, etc.

Primer sitio Web de la historia - info.cern.ch

En 1993 se creó el navegador Mosaic, que muchos viejos usuarios de Internet recordarán, el cual originalmente fue desarrollado para el sistema operativo Unix, pero que pronto fue adaptado para sistemas operativos Windows, para los cuales nunca hubo versión del ViolaWWW, lo cual hizo que el Mosaic finalmente le quitara protagonismo. Desde entonces cada vez más personas se interesaron en Internet, creciendo a un ritmo de 100% anual y dando un salto explosivo entre 1996 y 1997. Con el tiempo saldrían otros navegadores, hacia 1994 se lanzaría el navegador Netscape, el cual durante la segunda mitad de la década del '90 sería el más utilizado. En 1995 nacería el Internet Explorer el cual durante la primera mitad de la década del 2000 sería el más popular. Hacia 2002 llegaría Mozilla Firebird (luego rebautizado Mozilla Firefox), el cual durante la segunda mitad de la década del 2000 sería el más popular. Finalmente llegaría Google Chrome el cual tomaría el reinado de la Web siendo el navegador más utilizado hacia 2013. Hoy día también comparten una porción de la torta de usuarios (en menor medida que Chrome, Internet Explorer y Firefox) otros dos navegadores: El Safari y el Opera.

Hacia junio de 2012 se calculaba que habían ya más de 2400 millones de usuarios de Internet en todo el mundo y no para de crecer.



Primer navegador de la historia el WorldWideWeb luego rebautizado Nexus - Foto tomada en 1993

Todos sabemos de qué se trata Internet, pero ¿sabemos exactamente qué es?

Si leíste la sección anterior de historia -lo cual recomiendo- ya te darás una idea. Básicamente se trata de un gigantesco sistema de redes (computadoras conectadas unas con otras), interconectadas. Por eso mismo se la denomina coloquialmente la Red de redes. En ella encontramos redes comerciales, gubernamentales, personales, empresariales, de telecomunicaciones, financieras, de entretenimiento, etc. Todas interconectadas entre ellas; formando un conglomerado mundial de redes, llamado Internet, término que proviene de la palabra Internetwork la cual a su vez proviene de Interconnected Computer Networks (Redes Interconectadas de computadoras).

Puesto que cada uno de nosotros tenemos acceso a pequeñas redes regionales de nuestros países que a su vez están conectadas a la super red, tenemos acceso así a todas las redes del planeta que estén conectadas a Internet.

¿Y que es lo que se transmite en Internet? La respuesta es simple, información. Paquetes de información que están alojados en computadoras, dado que cada red no es más que un conjunto de computadoras interconectadas, que al estar enlazadas a la red de redes son accesibles a cualquiera. O sea, que en otras palabras podemos entrar al disco rígido de esas computadoras donde está guardada la información que nos interesa.

Por ejemplo si una página está compuesta de un documento textual, tres imágenes y un video lo que hace nuestro navegador es entrar a la computadora donde se aloja esa página y bajarse el documento, las tres imágenes y el video, mostrándolas en el navegador. Otro ejemplo sería cuando entramos a algún sitio donde se alojan archivos, del tipo de Rapidshare, lo que hacemos es encontrar la dirección donde se aloja el archivo que queremos bajar a nuestro disco rígido -desde el disco rígido donde se aloja el sitio- y lo descargamos.

No hay ninguna diferencia entre bajarse un programa, imágenes, un video o un documento de texto; son todos archivos, algunos se ven en el navegador y otros necesitan aplicaciones especiales para ejecutarse -por ejemplo un juego-.

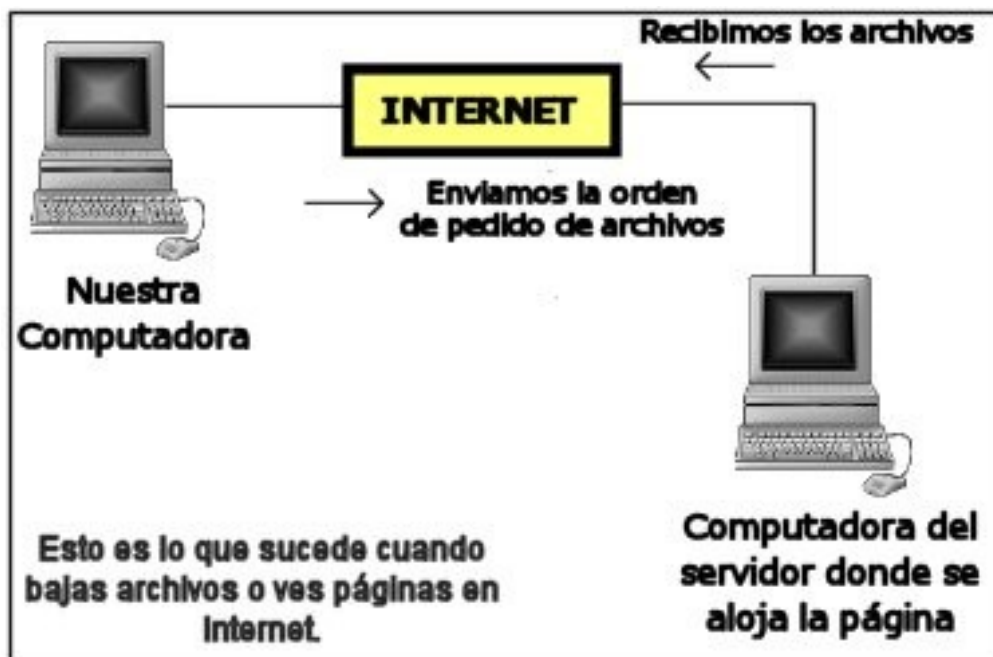
Las páginas que vemos están en formato .html las imágenes en .jpg , .png o .gif, los videos en .avi, la música en .mp3, etc. Todos estos archivos están alojados en computadoras llamadas servidores. Los sitios que visitamos y todo lo que bajamos de ellos se encuentran en estos servidores.

Pero no todo es bajar datos y archivos, también enviamos desde nuestras computadoras al enviar un mensaje de chat, al escribir un mail, al subir un video o simplemente al mandar una orden con nuestro navegador a una computadora de servidor para ver una determinada página y toda la información que ésta contiene.

Para graficar la situación de cuál es el proceso que ocurre al ver una página Web en nuestro navegador vamos a tomar el ejemplo de qué sucede al leer este tutorial de HTML:

Al bajar esta página Web en la que se ven texto e imágenes sucede más o menos lo siguiente:

- 1.Nuestra computadora se conecta con la computadora servidor donde se encuentra la página Web que queremos ver, entrando a la dirección Web de la barra de direcciones del navegador en la parte de arriba. En ese momento se establece lo que se denomina técnicamente una sesión entre nuestra computadora (técnicamente llamada computadora cliente) y la computadora del servidor. Una sesión es la comunicación o diálogo entre computadoras durante la cual se intercambian datos, información y órdenes.
- 2.Luego manda una orden a la computadora del servidor para que le envíe el paquete de información, o sea los archivos que conforman esta página del tutorial (documento HTML e imágenes).
- 3.Comienzan a bajarse los archivos a nuestra computadora cliente.
- 4.Una vez que el archivo de documento -o sea esto mismo que lees- y las imágenes se hayan descargado a nuestra computadora, el navegador las muestra en la pantalla del monitor.



Sesión entre una computadora cliente y una computadora servidor

Pero vamos a interiorizarnos un poco más en el tema. Como dijimos antes, cada una de nuestras computadoras están conectadas a una red que a su vez está conectada a la red de redes; en otras palabras, nuestra máquina está conectada a la red de nuestro ISP, siglas de Internet Service Provider (Proveedor de Servicios de Internet), a través de un Modem. El ISP es la empresa a la que le pagamos mensualmente un abono para que nos permita entrar a Internet haciendo uso de sus computadoras (que también son servidores) las cuales están conectadas a la red de redes. Todos los clientes de una compañía de ISP acceden a la red de esta empresa a través de lo que se llaman POP, siglas de Point of Presence (Punto de Presencia), o sea que los POP son la ubicación real (física) por la que se accede a la red del proveedor de servicios Internet. Los POP a su vez están conectados a la red de Internet a través de lo que se llaman NAP, siglas de Network Access Point (Punto de Acceso a la Red). Los NAP también son conocidos indistintamente como Internet Exchange Point (Punto de Intercambio de Internet) utilizando la siglas IXP.

Brevemente les describiré la arquitectura de las redes: En nuestra casa, escuela o trabajo puede que haya una o más computadoras, las cuales se conectan directamente a un modem, o a un router que interconecta a todas las computadoras presentes y que a su vez este router está conectado al modem formando entre todas las computadoras lo que se denomina una pequeña red local (LAN - siglas de Local Area Network lo que en inglés significa Red de Área Local). En una LAN cada computadora puede tener

acceso a las otras computadoras de la red pudiendo descargar y ejecutar sus programas de manera remota y también comunicarse con ellas.

También existen lo que se denominan WAN (Wide Area Network) lo que en inglés significa Red de Área Amplia. Las WAN son redes en las que se interconectan varias LAN a nivel zonal o regional. Las WAN a su vez se interconectan unas con otras formando lo que conocemos como Red de redes o Redes Interconectadas (Internetwork) o sea Internet.

Por ejemplo, en nuestra casa puede que hayan dos computadoras interconectadas a través de un router, formando una red local o LAN. El router a su vez nos permite conectarnos a la empresa proveedora de Internet o sea el ISP de la zona donde vivimos, al cual se conectan otros clientes con sus redes locales de una o más computadoras por cliente. Entre todos los clientes conectados al POP del ISP de la zona forman una red de área amplia o WAN. Esta WAN (la de la empresa proveedora de servicios de Internet y sus clientes) -y las WAN de otras empresas- a su vez está conectada a la NAP (como ya se dijo también conocida como IXP), o sea un router regional o nacional que interconecta a todas las WAN de la región o país con el resto de Internet. Cada país tiene uno o más NAP en su territorio.

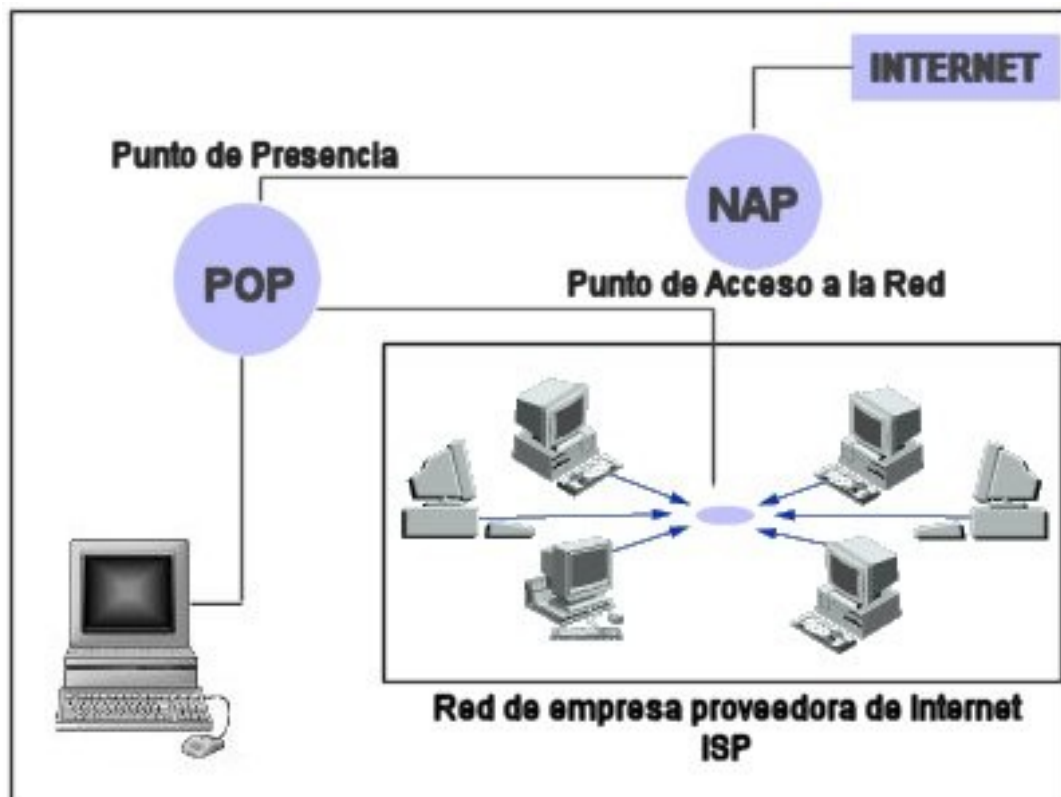
Veamos un ejemplo más práctico: Imaginemos una Compañía A de ISP, recordemos que significa Proveedor de Servicios de Internet, que en cada ciudad o barrio (si se trata de una ciudad muy grande) tiene un POP para que sus clientes accedan a la red de la empresa con sus Modems. A su vez la Compañía A le alquila a una empresa telefónica o de servicios de telecomunicaciones líneas de cables de fibra óptica para interconectar todos los POP de todas las localidades en las que presta servicio; así todas las computadoras de los clientes de esta empresa pueden estar interconectados entre ellas.

Por otro lado existe también la Compañía B que hace exactamente lo mismo y tiene su propia red WAN, interconectando a todos los clientes de esta.

Sin embargo los clientes de la red de la Compañía A no pueden conectarse a los de la Compañía B y vice versa. Es así como estas deciden conectarse al Punto de Acceso a la Red NAP y quedar de esta manera interconectadas una con la otra.

Es así que Internet está formada por innumerables redes de empresas, organismos y otras entidades que conectan sus redes a los NAP regionales o

nacionales que les dan acceso a Internet, que como dijimos antes no es más que una colección de miles y miles de redes interconectadas unas con otras y por lo tanto todas las computadoras que están conectadas a estas pequeñas redes también lo están a la red de redes, o sea Internet.



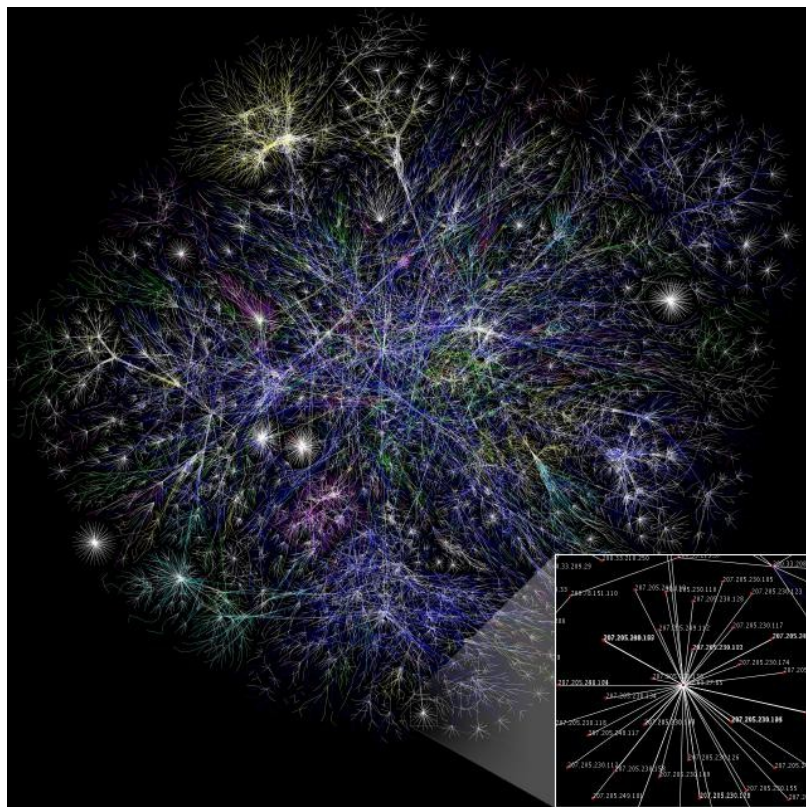
Las LAN de los clientes se conectan al POP del ISP formando una WAN y ésta a su vez se conecta al NAP que le da acceso a Internet

Los paquetes de información viajan de una red a la otra a través de Routers que cumplen la función de seleccionar qué camino o ruta tomará este paquete de datos e información de un router regional a otro y a través de qué redes lo hará. También cumplen la función de cerciorarse si los datos e información llegaron a destino así como evitar que se generen congestionamientos en el camino que harían colapsar el tráfico general de redes a lo largo del camino.

Los Routers de los NAP (o IXP como se los conoce también) son computadoras especiales que se encuentran a lo largo del camino que el paquete de información tomará a través de miles de senderos de cables de fibra óptica a nivel regional o mundial y todo esto en una fracción de segundo. Si el servidor del sitio Web de la página que se quiere descargar se encuentra en otro continente, la información deberá viajar miles de kilómetros de NAP en NAP a través de cables de fibra óptica subterráneos -y suboceánicos en las partes donde no hay tierra y que interconectan a los NAP en ambas orillas-.

Los cables de fibra óptica por los que transitan los datos e información por todo el planeta tienen distintas capacidades de velocidad, que van de 10 Gbps (Gigabits por segundo) hasta 100 Gbps en las instalaciones más modernas y regiones con mayor demanda. Por lo que el tiempo que tardará en descargarse una determinada página Web y todos los archivos que la conforman dependerá de la distancia desde nuestra computadora y las regiones por las que deberá pasar la información (influyendo el tráfico de Internet en cada una de éstas, la demanda que tienen los cables de fibra óptica de cada región por la que viajará la información y la capacidad de éstos).

Todos los cables de fibra óptica a nivel mundial que interconectan a cada red de Internet forman lo que se conoce como Internet Backbone (Columna Vertebral de Internet). Algunas de las mayores compañías del mundo que instalan el tendido de cables de fibra óptica y brindan los servicios de interconexión entre redes WAN dando forma al Internet Backbone son Verizon, AT&T y Sprint Nextel (en Estados Unidos), British Telecom en (Gran Bretaña), Hurricane Electric (en Norteamérica, Europa y Asia), France Télécom (en Francia), OVH (en Francia y otros países de Europa), IMPSAT y TELECOM (en Latinoamérica).



Mapa mundial de Internet - El Backbone o Columna Vertebral de Internet compuesto por cables de fibra óptica que recorren todo el planeta

Cada una de las computadoras conectadas a Internet cuentan con un identificador llamado Dirección IP, siglas de Internet Protocol (Protocolo Internet), que es el código identificador que usan las computadoras para comunicarse en Internet.

Una dirección de IP típica se parece a esto: 217.28.62.151

No vamos a profundizar en la estructura de los IP ya que está mas allá del objetivo de esta parte del tutorial, sin embargo realizaremos un breve repaso superficial para que se tenga una idea acerca de qué se trata más o menos.

Dijimos que cada computadora conectada a la red tiene su dirección IP, como vimos en la dirección anterior estas están compuestas por 4 números separados por puntos entre sí. Cada uno de estos puede contener 256 valores de 0 a 255. Por lo tanto si realizamos 256^4 -o sea $256 \times 256 \times 256 \times 256$ - obtenemos 4.294.967.296 distintas combinaciones de direcciones. Por lo tanto ese es el máximo número de conexiones que se pueden tener en Internet con el actual sistema denominado técnicamente IPv4 (Versión 4 del Protocolo de Internet).

Hacia enero de 2010 quedaban aproximadamente 400 millones de conexiones disponibles que se calcula estarían ocupadas hacia el 2014. Pero a no preocuparse, los ingenieros han desarrollado otro sistema que permite mayor cantidad de conexiones, el IPv6, el cual fue lanzado oficialmente el 6 de junio de 2012. De a poco las distintas empresas, servidores así como entidades gubernamentales, educativas y científicas están migrando sus redes al sistema IPv6 (Versión 6 de Protocolos Internet). Con IPv6 se podrán conectar hasta $3,4 \times 10^{38}$ computadoras, o sea $7,9 \times 10^{28}$ veces más que con el sistema IPv4.

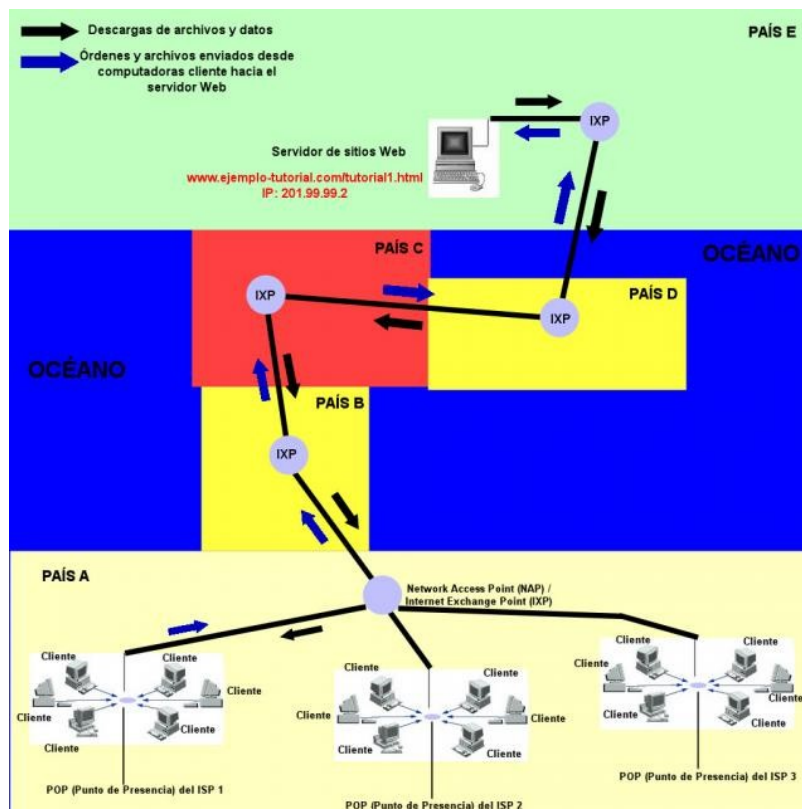
Obviamente el sistema de direcciones IP es más complicado de lo que se explica aquí, pero escaparía al enfoque de tutorial de HTML; simplemente es necesario tener una idea de qué se trata sin tener que ahondar en aspectos técnicos.

Cada dirección Web a la que entramos en realidad nos da acceso a una computadora remota que aloja el sitio que visitamos y por ende tiene su dirección IP única. Sin embargo como sería muy complicado recordar la dirección IP numérica de cada sitio, se inventó el DNS Domain Name System (Sistema de Nombre de Dominio) el cual asigna un nombre de

dominio como www.ejemplo.com a una dirección IP, siendo así más fácil de recordar.

Si quieres averiguar el número de IP de un sitio Web, hay varios sitios que brindan este servicio, simplemente googlea [website ip](http://websiteip.com) y ahí aparecerán varios que brindan este servicio, una vez ahí escribes el nombre de dominio del sitio Web cuyo número de IP quieres averiguar a lo que te responderá con toda la información que buscas.

También existen los Whois IP, que son servicios que te informan acerca de tu número de IP, que en realidad es el de la computadora del ISP a la que estás conectada, ya que cuando navegas lo haces a través de la computadora de tu proveedor de Internet. Simplemente googlea [my ip](http://myip.com) (que en inglés significa mi IP).



[Cómo viajan los datos en Internet - Clicar para ver en grande y leer toda la descripción de dicho proceso](#)

Cómo se hacen los sitios Web

Se calcula que en abril de 2017 habían en toda Internet aproximadamente 1.800 millones de sitios Web (de los cuales unos 169 millones estaban activos) y a su vez la mayoría de estos estaban compuestos por más de una página Web. Se calcula que hacia la misma fecha habían aproximadamente

4.480 millones de páginas Web que componen a esos sitios (o sea un promedio de alrededor de 2 páginas por sitio).

Ahora la pregunta es, ¿cómo fueron diseñadas estas páginas? o mejor dicho ¿qué necesitamos saber y tener para poder hacer nuestras propias páginas y por lo tanto sitios Web?

Para empezar; acceso a Internet, para poder bajar todas las herramientas y recursos que necesitemos así como para poder subir y probar las páginas de nuestro sitio.

Luego, para comenzar al menos, con el Bloc de notas nos alcanza para editar el código fuente de las páginas que diseñemos. Otra cosa necesaria es un navegador de Internet para ver los resultados, algo que obviamente ya tenemos, puesto que estamos leyendo este artículo.

Pero hay distintos tipos de páginas y sitios, que dan resultados diferentes o varían en su contenido y que requieren que conozcamos distintos lenguajes y herramientas acorde a los resultados que se quieran obtener.

Si lo que queremos obtener es una página estática simple, lo más básico que necesitamos saber es HTML y CSS (de esto hablaremos más adelante en capítulos más avanzados). Además HTML es la estructura fundamental, el esqueleto, de todas las páginas Web, desde la más simple a la más compleja.

Si ya queremos agregar algunos efectos dinámicos con botones, animaciones simples, aparición y desaparición de elementos de la página, etc. debemos estudiar el lenguaje de programación Javascript y JQuery (que es un derivado de Javascript).

Más adelante se puede estudiar Flash y Action Script para realizar efectos especiales, reproductores de películas, juegos online y otras cosas que tengan que ver con el mundo audiovisual.

Finalmente llegamos al punto avanzado de aprender a programar en PHP o ASP y usar bases de datos (como MySQL) para realizar sitios complejos interactivos como buscadores y redes sociales al estilo google, youtube, facebook, youbioit, etc.

Pero como en casi en todas las cosas de la vida debemos realizar cada paso, uno por uno, para llegar al destino esperado, en este caso el primer paso a dar es aprender HTML.

Entre las herramientas adicionales que necesitamos para realizar sitios Web, es conveniente saber crear y editar imágenes con algún editor de imágenes como Photoshop o Paint Shop Pro.

Y para subir nuestras páginas y archivos a la computadora del servidor donde se aloja nuestro sitio Web, necesitamos una herramienta llamada FTP de las siglas File Transfer Protocol (Protocolo de Transferencia de Archivos), que se puede conseguir de manera gratuita en la Web (como por ejemplo el [Filezilla Client](#)).

Eso es todo lo que se necesita saber y tener antes de empezar con el tutorial de HTML. Si se ha entendido todo lo anteriormente explicado en esta introducción, todo el resto del aprendizaje será más sencillo.

Antes de comenzar a aprender HTML, debemos saber con qué herramientas vamos a trabajar. Primero hay que entender que HTML es un lenguaje interpretado por los navegadores; eso significa que si escribimos correctamente los comandos en el código fuente de un archivo de formato .html o .htm, el contenido de este archivo será visualizado en cualquier navegador Web (por ejemplo Internet Explorer, Mozilla Firefox, Chrome, Safari o Opera).

Si queremos ver cómo está hecha una página Web, podemos hacerlo tocando sobre cualquier punto de ésta con el botón derecho de nuestro mouse, lo que abrirá un menú desplegable con varias opciones, en nuestro caso seleccionaremos Ver Código Fuente; y así se abrirá otra ventana con el código fuente de la página Web que estamos viendo. Quizá en un principio, si no sabemos HTML no entenderemos nada de lo que nos muestra la ventana del código fuente, pero con el tiempo a medida que aprendamos y practiquemos este lenguaje comenzaremos a comprender.

Ya sabemos cómo ver el código fuente de una página Web, ahora debemos saber con qué herramienta podremos hacer nuestras propias páginas. Hay muchos editores, como Dreamweaver por ejemplo, pero en mi opinión la mejor manera de aprender es utilizando una aplicación que todos tenemos en nuestras computadoras, el Bloc de notas. En realidad con esta simple herramienta podemos realizar sitios Web desde los más pequeños y simples, hasta portales de redes sociales de los más complejos, ya que con el Bloc de notas podemos programar en distintos lenguajes.

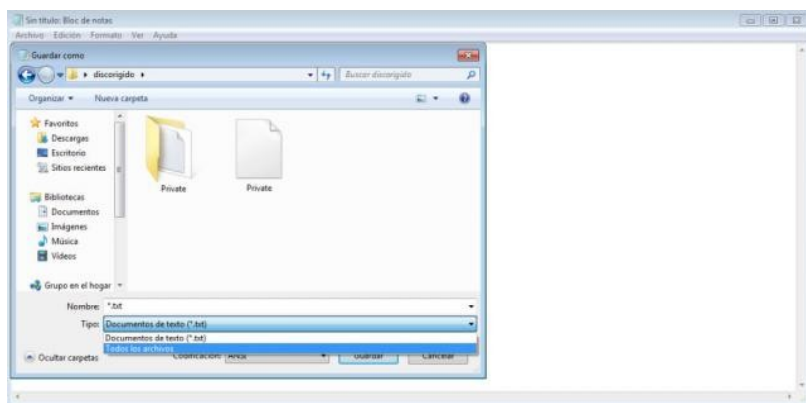
Bien, ahora es momento de comenzar con HTML. Primero hay que aclarar que su nombre proviene de las siglas de Hypertext Markup Language, que en inglés significa Lenguaje de Marcado de Hipertexto.

El hipertexto no es más que el texto remarcado que si es seleccionado nos redirige a otra página o documento Web; en otras palabras son los hipervínculos o links. Ya que al desarrollarse este lenguaje la característica principal que presentaba era la posibilidad de pasar de un documento a otro de maneja sencilla mediante el uso de estos textos remarcados en azul, siendo muy práctico lo que finalmente lo convirtió en el sistema de exploración Web estándar que hasta el día de hoy utilizamos en Internet.

Por lo tanto ya tenemos las herramientas para crear y visualizar nuestras páginas Web.

Los archivos de las páginas Web tienen que ser de tipo .htm o .html, los dos tipos son lo mismo, por lo tanto no habrán diferencias en los documentos que vayamos a crear, está en nosotros elegir cualquiera de los dos tipos.

Para guardar un archivo de página Web, vamos al menú Archivo de nuestro Bloc de notas, seleccionamos Guardar como, elegimos un nombre, luego en tipo de archivo cambiamos de Documentos de texto a Todos los archivos; esto último es muy importante, ya que si no seleccionamos el tipo de Todos los archivos, todo lo que hayamos escrito en el código fuente se guardará como archivo de tipo .txt; y nosotros queremos guardarlo como archivo .htm o .html.



Bloc de notas de Windows

Ahora pasaremos a los más importante, los comandos HTML que nos permiten hacer páginas Web.

ELEMENTOS Y ETIQUETAS DE HTML

Cada cosa que queremos mostrar en una página Web es un elemento, o sea que un elemento puede ser un párrafo, un título, una imagen, una lista, un hipervínculo, etc.

Cada elemento está definido por etiquetas y contenido. Las etiquetas son los comandos HTML que especifican lo que queremos mostrar y el contenido puede ser un texto, la dirección de una imagen, la dirección de un hipervínculo, etc.

En HTML las etiquetas deben ser abiertas y cerradas, o sea que se necesita una etiqueta de apertura que indica dónde comienza el elemento y una etiqueta de cierre que indica dónde termina el elemento.

La estructura de definición de elementos HTML es la siguiente:

`<nombre-de-etiqueta-de-elemento>`*Contenido***`</nombre-de-etiqueta-de-elemento>`**

Por ejemplo para escribir un párrafo debemos escribir el siguiente código fuente:

`<p>`*Este es un párrafo de ejemplo***`</p>`**

Las etiquetas HTML se forman con signos de menor y mayor los cuales rodean el nombre del comando HTML. Por ejemplo `<p>` es la etiqueta de párrafo. Las etiquetas de cierre utilizan una barra antes del nombre del comando; de esta manera en el caso del párrafo, la etiqueta de cierre es `</p>`.

Las etiquetas de apertura y cierre deben rodear al contenido del elemento, o sea que en el caso anterior el contenido viene a ser el texto, el cual debe estar rodeado por las etiquetas de apertura y cierre, `<p>` `</p>`.

Hay casos de elementos con contenido vacío, esto significa que no requieren etiqueta de apertura y cierre, éstos simplemente se cierran en la primera y única etiqueta; un ejemplo de este tipo de elementos son los saltos de línea:

`
`

El elemento anterior realiza un salto de línea al renglón inmediatamente inferior. El cierre del mismo se realiza con una barra al final de la etiqueta como se muestra en el ejemplo anterior.

Todas las páginas HTML requieren siempre de la presencia de ciertos elementos obligatorios, estos son:

`<html>`, `<head>`, `<title>` y `<body>`.

El elemento `<html>` define dónde comienza el documento o página HTML y dónde finaliza.

Entre las dos etiquetas de apertura y cierre `<html>` y `</html>` debemos poner el elemento `<body>` (que significa cuerpo); este elemento es donde se pondrá todo el contenido visual, o sea todo lo que se va a poder ver en la página; el contenido visual debe estar encerrado por una etiqueta de apertura `<body>` y una de cierre `</body>`; quedando así:

```
<html>  
<body>
```

Aquí deben ir todos los elementos visuales de la página, o sea párrafos, imágenes, hipervínculos, etc.

```
</body>  
</html>
```

Otro elemento necesario es `<head>`, que significa cabecera; dentro de este elemento se agregan todos los elementos no visuales de la página, o sea todo lo que no se ve en el cuerpo principal del explorador Web; por ejemplo información para servidores, para motores de búsqueda, etc. De eso hablaremos más adelante.

```
<html>  
<head>
```

Aquí deben ir todos los elementos no visuales de la página, como información para los motores de búsqueda entre otras cosas.

```
</head>  
<body>
```

Aquí deben ir todos los elementos visuales de la página, o sea párrafos, imágenes, hipervínculos, etc.

```
</body>  
</html>
```

Otro elemento necesario es el del título de página, el cual aparece en la barra de título de la parte de arriba del explorador o en los más modernos en la pestaña de la página activa. Por ejemplo el título de esta página es Curso de HTML - Primera Parte. El título de una página se define con el elemento `<title>` el cual debe ir dentro del elemento de cabecera `<head>`. El texto del título debe ir rodeado de una etiqueta de apertura `<title>` y otra de cierre `</title>`.

De esta manera el código fuente de una página Web simple, con lo mínimo indispensable, quedaría así:

```
<html>  
  
<head>  
<title>Este es el título de la página Web</title>  
</head>
```

```
<body>
```

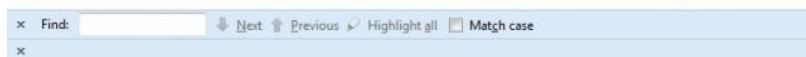
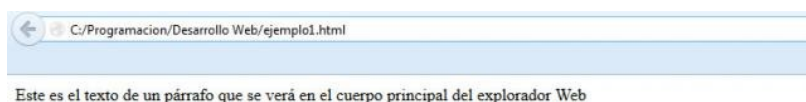
```
<p>Este es el texto de un párrafo que se verá en el cuerpo principal del explorador Web</p>
```

```
</body>  
</html>
```

Recuerden que el elemento `<p>` es el que define párrafos.

Pueden probarlo copiando y pegando el texto del código fuente de arriba en el Bloc de notas y guardando el archivo con el nombre de ejemplo1.html

El código fuente anterior debe hacer que la página Web creada se vea como en la siguiente imagen:



Una cosa que hay que recordar siempre es de cerrar los elementos, ya que a pesar que algunas veces pueden verse algunos de los elementos en ciertos navegadores aunque nos olvidemos de cerrarlos; no hay que confiarse; siempre hay que cerrarlos ya que muchos elementos no serán visualizados correctamente.

Otro consejo; las etiquetas de los elementos pueden escribirse en mayúscula o en minúscula; pero, se recomienda escribirlas en minúscula ya que en versiones modernas de HTML, como XHTML (ya se tratará más adelante el tema de las versiones de HTML), es obligatorio que las etiquetas estén escritas en minúscula; por lo tanto para tener compatibilidad en todos los navegadores del futuro es mejor que éstas sean escritas con letras minúsculas.

Las etiquetas HTML pueden tener también atributos, los cuales proveen información adicional acerca de los elementos que definimos. Los atributos siempre se escriben en la etiqueta de apertura, o sea la primera, de la siguiente manera:

```
<nombre-de-etiqueta atributo="valor">
```

Un ejemplo es la etiqueta <a> que se utiliza para crear hipervínculos:

```
<a href="www.youbioit.com">Link a Youbioit.com</a>
```

En este caso a es el nombre de la etiqueta del elemento utilizado para crear hipervínculos y href el atributo que provee información acerca de la dirección a la que nos enviará el link. Más adelante se describirá mejor el elemento de hipervínculos <a>, ya que en este momento simplemente lo utilizamos para explicar mejor qué son los atributos.

El valor del atributo siempre debe ser escrito entre comillas. Pueden utilizarse tanto comillas dobles " como comillas simples '; pero es recomendable utilizar las dobles. Las comillas simples se utilizan en aquellos raros casos en los que el valor incluye comillas dobles, por lo tanto para diferenciarlas se utilizan comillas simples, por ejemplo:

```
name="Tienda "La neuquina" - Indumentarias de todo tipo'
```

Si este no es ese el caso, lo recomendable es utilizar comillas dobles en este formato:

atributo="valor del atributo"

Como sucede con el nombre de las etiquetas, es recomendable que los nombres de atributos estén escritos en letras minúsculas.

Otra cosa acerca de la sintaxis de los atributos en HTML es que hay que escribir el nombre del atributo, el signo = y el valor entre comillas siempre pegados.

Sintaxis Correcta:

atributo="valor del atributo"

Sintaxis Incorrecta:

atributo = "valor del atributo"

Sintaxis Incorrecta:

atributo ="valor del atributo"

ELEMENTOS HTML

1 - ENCABEZADOS EN HTML

Los títulos o encabezados del cuerpo del documento en HTML se definen con las etiquetas <h1>, <h2>, <h3>, <h4>, <h5> y <h6> (la h es por header que en inglés significa encabezado). Cada una de estas etiquetas difieren en el tamaño del título que generan. Veamos algunos ejemplos:

```
<html>

<head>
<title>Ejemplo</title>
</head>

<body>

<h1>Título 1</h1>
<h2>Título 2</h2>
<h3>Título 3</h3>

</body>
</html>
```

El código fuente anterior daría como resultado los siguientes títulos.

Título 1

Título 2

Título 3

En la siguiente imagen se muestra cómo se vería el resultado en un navegador Web:



[Clicquear para ver en grande](#)

La etiqueta <h1> genera los encabezados más grandes mientras que la <h6> genera los títulos más pequeños. Cuanto mayor es el número, menor será el tamaño de fuente.

Las etiquetas <h1> a <h6> no deben ser utilizadas para resaltar grandes cantidades de texto o párrafos enteros, ya que para eso existen otras etiquetas especiales. Deben ser utilizadas solamente para definir títulos; ya que éstos son utilizados por los motores de búsqueda para indexar el contenido de nuestras páginas Web. Si utilizamos las etiquetas <h1> a <h6> con textos muy largos, corremos el riesgo de que los motores de búsqueda no indexen nuestras páginas Web, o las posicionen en puestos bajos de búsqueda.

Es bueno utilizar las etiquetas h para definir titulares o encabezados de artículos, páginas, textos, etc. Por ejemplo el título de una noticia o nota periodística.

2 - LÍNEAS HORIZONTALES EN HTML

La etiqueta <hr /> define líneas horizontales (la hr es por horizontal line que en inglés significa línea horizontal). Este elemento no requiere etiqueta de cierre y por lo tanto se cierra en la primera y única etiqueta que utiliza para crear líneas rectas.

Veamos un ejemplo:

```
<html>

<head>
<title>Ejemplo</title>
</head>
```

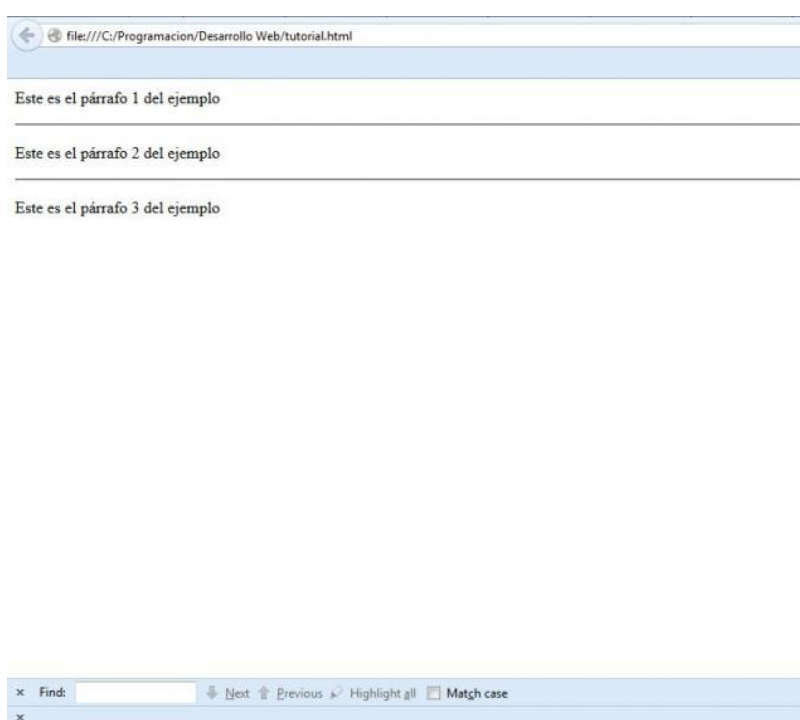
```
<body>

<p>Este es el párrafo 1 del ejemplo</p>
<hr />
<p>Este es el párrafo 2 del ejemplo</p>
<hr />
<p>Este es el párrafo 3 del ejemplo</p>

</body>
</html>
```

El código fuente anterior define tres párrafos separados por líneas horizontales. Pueden probarlo copiando y pegando el código fuente en el Bloc de notas, guardándolo como archivo de tipo .htm o .html y probándolo en un explorador Web.

El resultado debería verse como en la siguiente imagen:



[Cliquear para ver en grande](#)

3 - COMENTARIOS EN HTML

Si necesitamos guiarnos dentro del código fuente, explicando qué define cada pedazo de código fuente, podemos utilizar comentarios. Los comentarios son omitidos por el navegador Web y solamente son visibles en el código fuente de la página. Éstos pueden servir en códigos fuentes muy largos y confusos para guiarnos.

Los comentarios HTML se escriben entre medio de <!-- y -->

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

Pueden utilizarse todos los comentarios que sean necesarios, ya que como se mencionó anteriormente éstos no son visualizados por el navegador Web y no modifican el aspecto de nuestras páginas.

4 - PÁRRAFOS EN HTML

Como se mencionó en algunos ejemplos anteriores los párrafos de texto se definen con la etiqueta <p>. Cada párrafo de un documento debe estar encerrado por las etiquetas de apertura y cierre <p> y </p>.

Veamos un ejemplo:

```
<html>

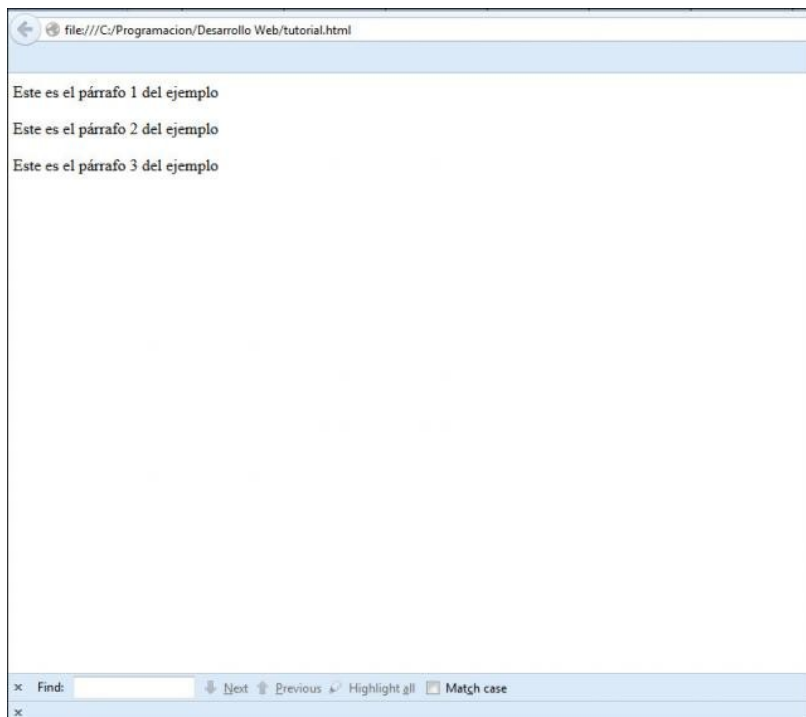
<head>
<title>Ejemplo</title>
</head>

<body>

<p>Este es el párrafo 1 del ejemplo</p>
<p>Este es el párrafo 2 del ejemplo</p>
<p>Este es el párrafo 3 del ejemplo</p>

</body>
</html>
```

Los párrafos se encuentran separados entre sí por un renglón intermedio. Recordemos que siempre debemos cerrar cada párrafo. El resultado del código fuente anterior debería verse en los navegadores Web como en la siguiente imagen:



[Clicar para ver en grande](#)

5 - SALTOS DE LÍNEA EN HTML

Si necesitamos realizar un salto de línea a un nuevo renglón sin salirnos del párrafo utilizamos la etiqueta `
`. La br es por break, que en inglés significa pausa, poner fin, quebrar o romper.

Veamos un ejemplo:

```
<html>

<head>
<title>Ejemplo</title>
</head>

<body>
```

```
<p>Este es el párrafo 1 del ejemplo.<br />
Este sigue siendo el párrafo 1.</p>

<p>Este es el párrafo 2 del ejemplo.<br />
Este sigue siendo el párrafo 2.</p>

<p>Este es el párrafo 3 del ejemplo.<br />
Este sigue siendo el párrafo 3.</p>

</body>
</html>
```

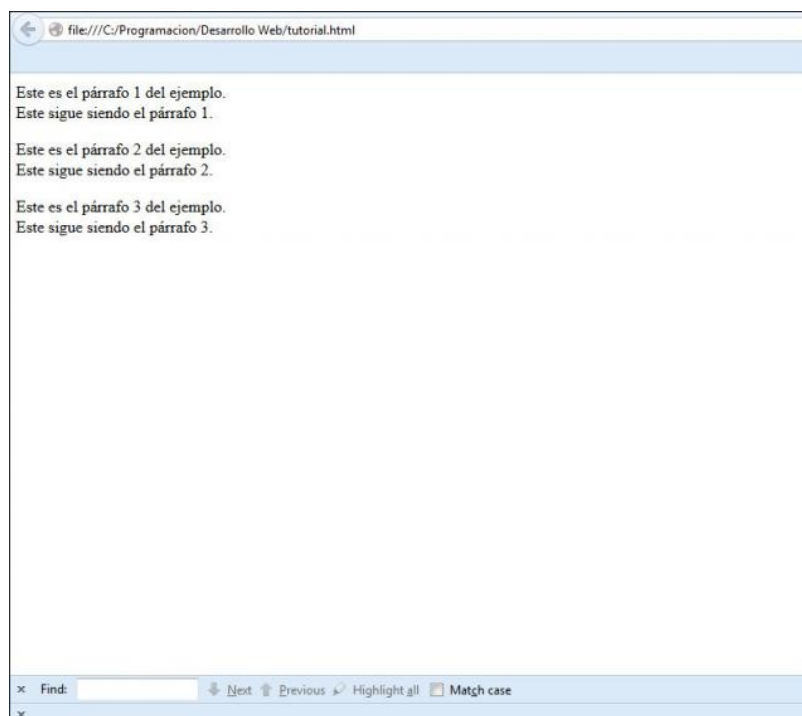
Esto dará como resultado lo siguiente:

Este es el párrafo 1 del ejemplo.
Este sigue siendo el párrafo 1.

Este es el párrafo 2 del ejemplo.
Este sigue siendo el párrafo 2.

Este es el párrafo 3 del ejemplo.
Este sigue siendo el párrafo 3.

Y que en los navegadores Web deberá verse como en la siguiente imagen:



[Clicar para ver en grande](#)

Cada vez que queramos agregar saltos de línea o nuevos renglones debemos utilizar la etiqueta `
`; todo nuevo renglón que agreguemos en el código fuente sin la etiqueta `
` será omitido por el explorador Web. Por lo tanto si requerimos un salto de por ejemplo 2 renglones, tenemos que incluir en el código fuente 2 etiquetas seguidas, de la siguiente manera: `

`. Si necesitamos 3 renglones ponemos `

`.

DARLE FORMATO AL TEXTO

A continuación veremos las etiquetas que nos permiten darle formato al texto; con estas etiquetas podemos presentar el texto en negrita, cursiva, texto grande, texto pequeño, etc.

1 - TEXTO EN NEGRITA

Para obtener texto en negrita debemos utilizar la etiqueta ``.

Por ejemplo:

```
<p><b>Este texto está en negrita.</b>Este texto no está en negrita.</p>
```

Esto daría como resultado lo siguiente:

Este texto está en negrita. Este texto no está en negrita.

2 - TEXTO EN CURSIVA

Para obtener texto en cursiva debemos utilizar la etiqueta `<i>`.

Por ejemplo:

```
<p><i>Este texto está en cursiva.</i>Este texto no está en cursiva.</p>
```

Esto daría como resultado lo siguiente:

Este texto está en cursiva. Este texto no está en cursiva.

3 - RESALTAR EL TEXTO

Podemos hacer resaltar el texto utilizando las etiquetas `` y ``.

La etiqueta `` (que significa fuerte en inglés) resalta más al texto que la etiqueta `` (su nombre proviene de emphasis que en inglés significa énfasis). Una cosa a tener en cuenta es que en los actuales navegadores Web, el texto resaltado con `` se ve en negrita, exactamente igual que con ``; y el texto resaltado con `` se ve en cursiva, exactamente igual que con `<i>`; sin embargo se recomienda que para obtener texto en negrita se utilice `` y para cursiva `<i>`; en lugar de `` y ``, ya que en un futuro, es posible que los navegadores Web muestren de manera distinta a los textos formateados con estas últimas dos etiquetas. Por lo tanto debemos utilizar `` y `` simplemente para resaltar una parte del texto, mientras dejamos el resto del mismo no resaltado.

Veamos un ejemplo:

```
<html>

<head>
<title>Ejemplo</title>
</head>

<body>

<p><b>Este texto está en negrita.</b></p>

<p><i>Este texto está en cursiva.</i></p>
```

```
<p><em>Este texto está resaltado.</em></p>
```

```
<p><strong>Este texto está resaltado más que el anterior.</strong></p>
```

```
</body>
```

```
</html>
```

El resultado sería el siguiente:



[Clicar para ver en grande](#)

Como se ve, los textos con `` y `` dan resultados similares al igual que `<i>` y ``.

Las etiquetas de formato pueden combinarse, así por ejemplo podemos tener texto en negrita y cursiva.

4 - OTRAS ETIQUETAS DE FORMATO DE TEXTO

Existen varias etiquetas que dan distinto formato al texto definido. Primero veremos las definiciones de cada una de estas etiquetas y luego veremos un ejemplo que incluye el código fuente de cada etiqueta:

La etiqueta `<big>` le da tamaño grande al texto definido.

La etiqueta `<small>` le da tamaño más pequeño al texto definido.

La etiqueta `<sub>` define texto en subíndice, o sea texto de tamaño más pequeño y ubicado ligeramente más abajo que el resto, como se suele utilizar en matemáticas por ejemplo: b_2 .

La etiqueta `<sup>` define texto en superíndice, o sea texto de tamaño más pequeño y ubicado ligeramente más arriba que el resto, como se suele utilizar en matemáticas para definir potencias; por ejemplo 2^2 .

Las etiquetas `` e `<ins>` se utilizan para definir un texto tachado y de corrección respectivamente; por ejemplo como se utiliza en casos de ofertas en las que figura el precio de antes y el de ahora. `` tacha el texto e `<ins>` muestra el texto subrayado como si fuese insertado para corregir el texto anterior; estas etiquetas suelen utilizarse en conjunto.

La etiqueta `<code>` define texto con estilo de código fuente de computadora.

La etiqueta `<kbd>` define texto de teclado o máquina de escribir.

La etiqueta `<tt>` define texto de estilo de teletipo.

La etiqueta `<pre>` permite definir texto preformateado con estilo similar al de la etiqueta `<code>`; esto significa que si al texto definido le agregamos varios espacios o saltos de línea en el código -sin utilizar `
`- éstos aparecerán tal cual se los escribió en el código fuente. Recordemos que en HTML el texto común acepta solamente un espacio entre palabras y por más que agreguemos dos o más espacios, siempre se mostrará uno; lo mismo sucede con los saltos de línea, ya que en el texto común la única manera de agregar saltos de línea es mediante la etiqueta `
`. Con la

etiqueta <pre> se acepta o muestra más de un espacio, mientras que los saltos de línea se pueden realizar sin
; el único detalle es que el estilo de texto es similar al definido por la etiqueta <code>.

La etiqueta <var> define texto en formato similar al utilizado para mostrar variables en el código fuente de editores de lenguajes de programación; en muchos navegadores Web el estilo resultante es similar al que define <code>.

La etiqueta <samp> define texto en estilo de ejemplo de código fuente; el resultado es similar al de <code>.

La etiqueta <abbr> sirve para definir abreviaturas, las cuales se muestran subrayadas por una línea de puntos y si se apoya el puntero del mouse por encima del texto abreviado aparece una leyenda, generalmente utilizada para explicar qué significa la abreviatura del texto definido. La leyenda se define con el atributo title.

Por ejemplo:

La <abbr title="Universidad Tecnológica Nacional">UTN</abbr> es líder en la enseñanza de carreras de tecnología.

La etiqueta <address> define texto en formato de dirección postal. En la mayoría de los navegadores Web aparece en cursiva con un espacio de un renglón antes y después del bloque de texto.

La etiqueta <blockquote> define texto en formato de cita; por ejemplo se puede utilizar para citar frases. En la mayoría de los navegadores Web, el bloque de texto definido con esta etiqueta aparece con un espacio de sangría adelante.

Veamos un ejemplo con todas las etiquetas explicadas anteriormente:

```
<html>

<head>
<title>Ejemplo</title>
</head>

<body>

<p><big>Este texto se verá más grande.</big></p>

<p><small>Este texto se verá más pequeño.</small></p>

<p>x<sub>1</sub> + x<sub>2</sub> = x</p>

<p>El resultado de la potencia de 2<sup>10</sup> es igual a 1024.</p>

<p>El precio de este pantalón es: Antes - <del>$100</del>. Ahora - <ins>$80</ins>.</p>

<p><code>Este texto tiene formato de código fuente de computadora.</code></p>

<p><kbd>Este texto tiene formato de letra de teclado o máquina de escribir.</kbd></p>

<p><tt>Este texto tiene formato de teletipo.</tt></p>

<pre>Este texto admite más de un      espacio
    así como saltos
    de línea.</pre>

<p><var>Este texto tiene formato de variable de código fuente.</var></p>

<p><samp>Este texto tiene formato de ejemplo de código fuente de programa.</samp></p>

<p>La <abbr title="Universidad Tecnológica Nacional">UTN</abbr> es líder en la
```

enseñanza de carreras de tecnología y de ciencias exactas tanto en la República Argentina como en América en general.

<address>

Dirección: Belgrano 2300 - Capital Federal

República Argentina

Tel: 4999-9999

</address>

<blockquote>Principio de la cita larga. Esta es una cita con espacios de sangrías agregados. Esta es una cita con espacios de sangrías agregados. Esta es una cita con espacios de sangrías agregados. Esta es una cita con espacios de sangrías agregados. Esta es una cita con espacios de sangrías agregados. Esta es una cita con espacios de sangrías agregados. Fin de la cita larga.**</blockquote>**

</body>

</html>

Esto se vería de la siguiente manera:



[Cliquear para ver en grande](#)

CÓMO AGREGAR IMÁGENES

El texto es muy útil para dar información, pero convengamos que lo que le da belleza estética a una página Web son las imágenes, así que a continuación veremos la etiqueta `` y sus atributos que nos permiten agregar imágenes.

Lo mínimo indispensable que hay que agregar en una etiqueta es lo siguiente:

```

```

Como se muestra en el ejemplo anterior, `` requiere solamente una etiqueta la cual se cierra al final de la misma. También requiere un atributo obligatorio, `src`, el cual proviene de "source" que significa fuente en inglés. Se trata del atributo que define el nombre de la imagen a agregar.

Es importante saber la ubicación de la imagen; si se encuentra en la misma carpeta que el archivo HTML que estamos editando, solamente hay que indicar el nombre del archivo, pero si se encuentra en otra carpeta hay que indicar la dirección completa con todos los nombres de carpetas.

Otro atributo opcional pero útil es alt, el cual sirve para definir un texto alternativo que aparece en caso que la imagen no se haya cargado. De esta manera, si por alguna razón la imagen no se ve o tarda mucho en cargar, en su lugar se

muestra este texto alternativo. La etiqueta `` con el atributo `alt` se ve de la siguiente manera:

```

```

Otros dos atributos opcionales, pero muy importantes, son los que definen el ancho y alto de la imagen que agregamos. El atributo `width` define el ancho y `height` define el alto.

Veamos un ejemplo:

```

```

La etiqueta anterior agrega una imagen llamada `imagen.jpg`, con texto alternativo "Imagen 1", un ancho de 400 pixels y un alto de 200 pixels.

Se puede especificar el mismo tamaño original de la imagen u otro distinto. Por ejemplo se pueden especificar dimensiones mayores y así estirar la imagen o se puede lograr el efecto contrario de mostrar una imagen con dimensiones inferiores a las originales.

Una cosa que hay que tener en cuenta es que los usuarios que visitarán nuestra página utilizarán distintas resoluciones de pantalla, por lo tanto no todos verán de la misma manera las imágenes que agreguemos a la página. Si se utilizan imágenes para hacer partes estructurales de una página, como un marco o una sombra; por ejemplo si queremos que éstas siempre se vean de la misma forma en todas las resoluciones de pantalla es conveniente utilizar porcentaje en lugar de pixels al definir el ancho de las imágenes.

```

```

La imagen anterior ocupará el 80% del ancho de la página en la que esté contenida.

Otra cosa que se puede hacer con las imágenes es alinearlas respecto a un texto adyacente. Por ejemplo podemos lograr que el texto aparezca a la derecha de un texto o a la izquierda, o arriba o en el medio; también podemos lograr que la imagen sea recorrida por los costados por el texto, etc. Esto se logra mediante el uso del atributo `align`; y algunos de los valores pueden ser `left`, `right`, `top`, `middle` o `bottom`, que significan izquierda, derecha, arriba, medio, fondo.

En el siguiente ejemplo sustituyamos el valor `imagen.jpg` del atributo `src` con el nombre de una imagen que poseamos.

```
<html>

<head>
<title>Ejemplo con imágenes</title>
</head>

<body>

<p>La imagen está alineada
a la izquierda de este texto, por lo tanto el texto recorrerá la imagen por el lado derecho de la
misma.La imagen está alineada a la izquierda de este texto, por lo tanto el texto recorrerá la
imagen por el lado derecho de la misma. La imagen está alineada a la izquierda de este texto, por
lo tanto el texto recorrerá la imagen por el lado derecho de la misma. La imagen está alineada a
la izquierda de este texto, por lo tanto el texto recorrerá la imagen por el lado derecho de la
misma.</p>

<p>La imagen está
alineada a la derecha de este texto, por lo tanto el texto recorrerá la imagen por el lado izquierdo
de la misma.La imagen está alineada a la derecha de este texto, por lo tanto el texto recorrerá la
imagen por el lado izquierdo de la misma.La imagen está alineada a la derecha de este texto, por
lo tanto el texto recorrerá la imagen por el lado izquierdo de la misma.La imagen está alineada a
la derecha de este texto, por lo tanto el texto recorrerá la imagen por el lado izquierdo de la
misma.La imagen está alineada a la derecha de este texto, por lo tanto el texto recorrerá la
```

imagen por el lado izquierdo de la misma.</p>

<p>La imagen está alineada arriba con respecto al texto, lo cual significa que el texto comenzará en la esquina superior derecha de la imagen.</p>

<p>La imagen está alineada en el medio con respecto al texto adyacente a la misma. Esto significa que el texto comenzará en el medio del lado derecho.</p>

<p>La imagen está alineada abajo con respecto al texto. La imagen está alineada abajo con respecto al texto. La imagen está alineada abajo.</p>

</body>

</html>

Esto se vería de la siguiente manera:



[Clicar para ver en grande](#)

Recordemos que debemos utilizar las imágenes con moderación, ya que son archivos aparte del documento HTML, y que cuantas más imágenes agreguemos más tardará en cargarse todo. Otra cosa a tener en cuenta es que procuremos que éstas no ocupen un tamaño grande de disco (en bytes), ya que utilizan espacio del ancho de banda. Por ejemplo si una página Web está compuesta por un documento HTML que ocupa 100 KB y cinco imágenes de 200 KB cada una; al cargar la página con un explorador deberemos bajar para verla 1.1 MB de datos (1 MB de las cinco imágenes y 100 KB del archivo .htm o .html)

Los formatos de imágenes más utilizados al crear páginas Web son .jpg, .png y .gif, siendo estos dos últimos más utilizados para crear botones o logos de sitios Web ya que permiten transparencias.

HIPERVÍNCULOS

Quizá uno de los elementos más importantes del lenguaje HTML son los hipervínculos, los cuales permiten a los usuarios saltar de una página a otra, de un sitio a otro o incluso de un punto a otro dentro de una misma página. Los hipervínculos pueden estar definidos en texto o imágenes, de esta manera si tocamos un texto o imágenes con hipervínculo saltaremos al punto que indica el mismo.

Los hipervínculos se definen mediante la etiqueta <a>. Para que un texto o imagen tenga un hipervínculo la etiqueta que lo define debe estar rodeada por las etiquetas de apertura y cierre <a>.

El atributo de <a> que define la dirección a la que apuntará el hipervínculo se llama href. El valor que se le introduce a href es una URL (nombre técnico que se utiliza para referirse a las direcciones Web, proveniente de las siglas de Uniform Resource Locator, que significa Localizador Uniforme de Recursos).

Veamos un ejemplo:

```
<html>

<head>
<title>Ejemplo con hipervínculo</title>
</head>

<body>

<p><a href="http://www.youbioit.com">Este texto es un hipervínculo</a></p>

</body>
</html>
```

El ejemplo anterior daría como resultado un hipervínculo hacia www.youbioit.com, como se ve a continuación:

[Este texto es un hipervínculo](#)

Hay dos maneras de definir la URL de un hipervínculo mediante una dirección absoluta o una dirección relativa. La dirección absoluta incluye la dirección completa y la relativa la dirección de un documento en relación a la ubicación del documento en el que nos encontramos.

Si por ejemplo nos encontramos en <http://www.ejemplo.com/carpeta1/carpeta2/carpeta3/documento1.html> y queremos definir un hipervínculo a <http://www.ejemplo.com/carpeta1/carpeta2/carpeta3/documento2.html>; hay dos maneras de hacerlo:

Utilizando la dirección absoluta: Texto
Utilizando la dirección relativa: Texto

Si nos encontramos en <http://www.ejemplo.com/carpeta1/documento1.html> y queremos definir un hipervínculo a <http://www.ejemplo.com/carpeta1/carpeta2/carpeta3/documento2.html>; hay dos maneras de hacerlo:

Utilizando la dirección absoluta: Texto
Utilizando la dirección relativa: Texto

Si nos encontramos en <http://www.ejemplo.com/carpeta1/carpeta2/carpeta3/documento1.html> y queremos definir un hipervínculo a <http://www.ejemplo.com/carpeta1/documento2.html>; hay dos maneras de hacerlo:

Utilizando la dirección absoluta: Texto
Utilizando la dirección relativa: Texto

Con ../ nos dirigimos a una carpeta atrás.

Un atributo opcional de <a> es target, el cual nos permite abrir el hipervínculo en una pestaña nueva del explorador, si le agregamos el valor "_blank".

Por ejemplo:

```
<html>

<head>
<title>Ejemplo con hipervínculo</title>
</head>

<body>

<p><a href="http://www.youbioit.com" target="_blank">Este texto define un hipervínculo que
se abre en una pestaña nueva</a></p>

</body>
</html>
```

El valor por omisión del atributo target en la etiqueta <a>, es "_self", (en inglés significa sí mismo), y lo que hace es abrir las páginas a las que nos dirigen los enlaces en la misma pestaña o ventana. Si no se indica dicho valor en la etiqueta mencionada, su link se abrirá en la misma pestaña, ya que es el valor por omisión, por lo que no es obligatorio ponerlo; sin embargo si por motivos de claridad se prefiere su presencia, puede incluirse.

Si queremos crear un hipervínculo hacia otro punto de la misma página en la que nos encontramos, por ejemplo para dirigirnos a una subsección del documento que estamos leyendo debemos utilizar un marcador. Los marcadores se definen con el atributo de <a> name. Estos marcadores son invisibles y simplemente definen un punto al que podemos saltar dentro de la página. Para dirigirnos al punto donde se encuentra el marcador utilizamos el atributo href y como valor el nombre del marcador anteponiendo un signo #.

Por ejemplo si queremos definir un marcador de nombre Marcador1 y un hipervínculo hacia el mismo debemos escribir el siguiente código fuente:

```
<html>

<head>
<title>Ejemplo con hipervínculo</title>
</head>

<body>

<p><a href="#Marcador1">Este texto define un hipervínculo que nos lleva al punto donde se
encuentra el marcador invisible llamado Marcador1.</a></p>

<p>Texto 1</p>

<p><a name="Marcador1">El hipervínculo anterior nos trae a este punto</a></p>

</body>
</html>
```

También podemos dirigirnos a un marcador ubicado en otra página; simplemente debemos poner en el atributo href la dirección URL de la página donde se encuentra el marcador, seguido por # y el nombre del marcador.

Por ejemplo si un marcador llamado Marcador1 se encuentra en la página ubicada en www.ejemplo.com/documento1.html, para definir el hipervínculo que nos lleve al mismo debemos escribir el siguiente código fuente.

``Este texto define un hipervínculo que nos lleva al punto donde se encuentra el marcador invisible llamado Marcador1.``

Para crear hipervínculos con imágenes en lugar de utilizar texto rodeado por la etiqueta `<a>`, utilizamos una imagen. En donde dice `nombredeimagen.jpg` deben reemplazarlo por el nombre verdadero de una imagen que tengan.

```
<html>

<head>
<title>Ejemplo con hipervínculo de imagen</title>
</head>

<body>

<a href="http://www.youbioit.com"></a>

</body>
</html>
```

También podemos crear hipervínculos a direcciones mail, los cuales al ser tocados, abren automáticamente la aplicación de mail de nuestro sistema operativo. Para crearlo debemos agregar en href como valor mailto: seguido de la dirección de mail.

```
<html>

<head>
<title>Ejemplo con hipervínculo a dirección de mail</title>
</head>

<body>

<a href="mailto:alguien@ejemplo.com">Esto abrirá el programa de mail de nuestro sistema
operativo con la dirección de mail agregada automáticamente</a>

<a href="mailto:alguien@ejemplo.com?Subject=Hola%20amigo">Esto abrirá el programa de
mail de nuestro sistema operativo con la dirección de mail y asunto agregados
automáticamente.</a>

</body>
</html>
```

En el segundo ejemplo vemos que agregamos el asunto del mail utilizando la palabra Subject separada de la dirección por un signo `?`. Cada palabra que forma el texto del asunto debe estar separada por `%20` y no por espacios, de lo contrario no funcionará correctamente, así que recordemos utilizar siempre `%20` en el asunto de mailto.

LISTAS

Otro elemento muy utilizado en el diseño de páginas Web es el de las listas. En HTML existen tres tipos de listas: las listas numeradas u ordenadas, las no numeradas o no ordenadas, y las listas de definición.

Las listas no ordenadas se ven como el siguiente ejemplo:

- Ítem 1
- Ítem 2
- Ítem 3
- Ítem 4
- Ítem 5

Para definir una lista no ordenada se utiliza la etiqueta y cada ítem se define con la etiqueta . Veamos el código fuente necesario para agregar un lista como la anterior.

```
<html>

<head>
<title>Ejemplo de lista</title>
</head>

<body>

<ul>
<li>Ítem 1</li>
<li>Ítem 2</li>
<li>Ítem 3</li>
<li>Ítem 4</li>
<li>Ítem 5</li>
</ul>

</body>
</html>
```

Como se ve en el código fuente anterior tanto la lista así como cada ítem requieren una etiqueta de apertura y de cierre separadas, es muy importante respetar esto al igual que con otras etiquetas HTML, siempre hay que cerrar cada ítem y la lista con la etiqueta de cierre.

Otra cosa a tener en cuenta es que las listas agregan sangría automáticamente a cada ítem para que la misma quede destacada.

Las listas ordenadas se ven como el siguiente ejemplo:

1. Ítem 1
2. Ítem 2
3. Ítem 3
4. Ítem 4
5. Ítem 5

Para definir una lista ordenada se utiliza la etiqueta y al igual que en las listas no ordenadas, cada ítem se define con la etiqueta . Veamos el código fuente necesario para agregar un lista como la anterior.

Como se ve en el código fuente anterior, la estructura de las listas ordenadas es igual a la de las no ordenadas y lo único que cambia es el nombre de la etiqueta que define qué tipo de lista queremos agregar.

Vale la pena aclarar que proviene de Unordered List que en inglés significa Lista no ordenada, mientras que proviene de Ordered List que significa Lista ordenada y proviene de List Item que significa Ítem de lista.

Hay un atributo que nos permite cambiar el tipo de listas ordenadas y no ordenadas, se trata del atributo type. Por ejemplo podemos tener una lista ordenada con letras en lugar de números o una lista no ordenada con cuadrados o círculos blancos en lugar de círculos negros.

Veamos un ejemplo:

```
<html>

<head>
<title>Ejemplo de lista</title>
</head>

<body>

<h4>Lista no ordenada con ítems de círculos negros</h4>
<ul type="disc">
<li>Ítem 1</li>
<li>Ítem 2</li>
<li>Ítem 3</li>
<li>Ítem 4</li>
<li>Ítem 5</li>
</ul>

<h4>Lista no ordenada con ítems de círculos blancos</h4>
<ul type="circle">
<li>Ítem 1</li>
<li>Ítem 2</li>
<li>Ítem 3</li>
<li>Ítem 4</li>
<li>Ítem 5</li>
</ul>

<h4>Lista no ordenada con ítems cuadrados</h4>
<ul type="square">
<li>Ítem 1</li>
<li>Ítem 2</li>
<li>Ítem 3</li>
<li>Ítem 4</li>
<li>Ítem 5</li>
</ul>

<h4>Lista ordenada con ítems de numerados</h4>
<ol>
<li>Ítem 1</li>
<li>Ítem 2</li>
<li>Ítem 3</li>
<li>Ítem 4</li>
<li>Ítem 5</li>
</ol>

<h4>Lista ordenada con ítems de letras mayúsculas</h4>
<ol type="A">
<li>Ítem 1</li>
<li>Ítem 2</li>
<li>Ítem 3</li>
<li>Ítem 4</li>
```

```
<li>Ítem 5</li>
</ol>

<h4>Lista ordenada con ítems de letras mayúsculas</h4>
<ol type="A">
<li>Ítem 1</li>
<li>Ítem 2</li>
<li>Ítem 3</li>
<li>Ítem 4</li>
<li>Ítem 5</li>
</ol>

<h4>Lista ordenada con ítems de letras minúsculas</h4>
<ol type="a">
<li>Ítem 1</li>
<li>Ítem 2</li>
<li>Ítem 3</li>
<li>Ítem 4</li>
<li>Ítem 5</li>
</ol>

<h4>Lista ordenada con ítems de números romanos en mayúscula</h4>
<ol type="I">
<li>Ítem 1</li>
<li>Ítem 2</li>
<li>Ítem 3</li>
<li>Ítem 4</li>
<li>Ítem 5</li>
</ol>

<h4>Lista ordenada con ítems de números romanos en minúscula</h4>
<ol type="i">
<li>Ítem 1</li>
<li>Ítem 2</li>
<li>Ítem 3</li>
<li>Ítem 4</li>
<li>Ítem 5</li>
</ol>

</body>
</html>
```

Esto se vería de la siguiente manera en un navegador Web:



[Cliquear para ver en grande](#)

Si no se especifica el tipo de ítem de lista, por omisión las listas no ordenadas contienen ítems con círculos negros y las ordenadas están enumeradas.

LISTAS DE DEFINICIÓN

Las listas de definición contienen ítems con una descripción de cada uno. Una lista de definición se ve así:

Ítem 1

Descripción del ítem 1

Ítem 2

Descripción del ítem 2

Ítem 3

Descripción del ítem 3

Ítem 4

	Descripción del ítem 4
Ítem 5	Descripción del ítem 5

```
<html>

<head>
<title>Ejemplo de lista</title>
</head>

<body>

<dl>
<dt>Ítem 1</dt>
<dd>Descripción de ítem 1</dd>
<dt>Ítem 2</dt>
<dd>Descripción de ítem 2</dd>
<dt>Ítem 3</dt>
<dd>Descripción de ítem 3</dd>
<dt>Ítem 4</dt>
<dd>Descripción de ítem 4</dd>
<dt>Ítem 5</dt>
<dd>Descripción de ítem 5</dd>
</dl>

</body>
</html>
```

La etiqueta que define una lista de definición es <dl>, la etiqueta que define un ítem es <dt> y <dd> es la descripción.

TABLAS HTML

Otro elemento muy utilizado en el diseño de páginas Web son las tablas. En HTML las tablas se definen en filas horizontales y columnas verticales. Para crear una tabla se requieren por lo menos tres etiquetas, la etiqueta <table> que define la tabla, <tr> para definir filas horizontales y <td> para definir las celdas de cada fila.

Por ejemplo:

Fila 1, Celda 1	Fila 1, Celda 2
Fila 2, Celda 1	Fila 2, Celda 2
Fila 3, Celda 1	Fila 3, Celda 2
Fila 4, Celda 1	Fila 4, Celda 2
Fila 5, Celda 1	Fila 5, Celda 2

Para crear la tabla anterior primero debemos definir la tabla con la etiqueta <table>, luego la primera fila con la etiqueta <tr>, seguido de las celdas 1 y 2 de la primera fila con la etiqueta <td>, continuamos con la segunda fila usando otra etiqueta <tr>, luego las dos celdas de la segunda fila con etiquetas <td> y finalmente la tercera, cuarta y quinta fila y sus celdas con las etiquetas <tr> y <td> respectivamente. Todo esto debe hacerse en orden y respetando etiquetas de apertura y cierre de cada etiqueta, o sea que al terminar las filas debemos cerrarlas con </tr> y al terminar cada celda cerramos con </td>.

Veamos el código fuente necesario para crear la tabla del ejemplo anterior:

```
<html>
```

```

<head>
<title>Ejemplo de tabla</title>
</head>

<body>

<table border="1">
<tr>
    <td>Fila 1, Celda 1</td><td>Fila 1, Celda 2</td>
</tr>
<tr>
    <td>Fila 2, Celda 1</td><td>Fila 2, Celda 2</td>
</tr>
<tr>
    <td>Fila 3, Celda 1</td><td>Fila 3, Celda 2</td>
</tr>
<tr>
    <td>Fila 4, Celda 1</td><td>Fila 4, Celda 2</td>
</tr>
<tr>
    <td>Fila 5, Celda 1</td><td>Fila 5, Celda 2</td>
</tr>

</table>

</body>
</html>

```

Como se ve en el código fuente anterior, primero debemos definir dónde comienza la tabla con la etiqueta <table>. Luego definimos en orden cada fila con la etiqueta <tr> y dentro de cada fila las celdas correspondientes con <td>. Recordemos que hay que cerrar cada celda con </td> y cada fila con </tr>. El contenido de cada celda debe estar incluido entre las etiquetas de apertura y cierre <td>. Todo debe estar colocado en el orden de aparición que deseamos lograr. Hay que tener mucho cuidado de colocar las etiquetas de celdas y filas correctamente; si alguna se llega a encontrar en una posición incorrecta pueden obtenerse resultados indeseados en el diseño de la página Web con la que estamos trabajando. Finalmente donde termina la tabla cerramos con la etiqueta de cierre </table>.

En el código fuente anterior, dentro de la etiqueta <table> también incluimos un atributo llamado border, el cual define el ancho del borde de la tabla. Por omisión si no incluimos ese atributo la tabla tendrá un borde invisible o de 0 píxeles. En el caso anterior la tabla tiene un borde de 1 píxel. El número del atributo border indica el ancho del borde en píxeles, o sea que cuanto más grande sea el número indicado dentro de border, más ancho será el borde de la tabla en píxeles.

Otro atributo opcional de la etiqueta <table> es cellpadding el cual sirve para indicar en píxeles el margen interno que tendrán las celdas de la tabla; cuanto más grande sea el número más alejado estará el contenido de las celdas de los bordes de las mismas. Por ejemplo cellpadding="10" indica que las celdas tendrán un margen interno de 10 píxeles.

También hay otro atributo de <table> llamado cellspacing el cual sirve para definir el espaciado entre celdas en píxeles, o sea que los bordes de las celdas tendrán un espacio vacío interno. Este atributo no afecta el borde externo de la tabla.

El atributo width es muy importante ya que define el ancho que tendrá la tabla. Por ejemplo si queremos que ésta ocupe cierto espacio del ancho de la página es conveniente indicarle un ancho que puede ser en píxeles o porcentaje del ancho de la página. Por ejemplo si dentro de la etiqueta <table> indicamos width="500" la tabla tendrá un ancho de 500 píxeles. Por otro lado si preferimos usar porcentajes e indicamos width="50%" la tabla tendrá un ancho del 50% de la página que la contiene. En casos que querramos que la tabla ocupe el mismo ancho en todo tipo de resoluciones de pantalla, es conveniente indicar el ancho de tabla en porcentaje en lugar de píxeles, ya que por ejemplo una tabla de 1000 píxeles de ancho en una pantalla de 1024 píxeles de ancho ocupará más espacio que en una pantalla de 1280 píxeles de ancho; sin embargo si indicamos para ésta un ancho de por ejemplo un 80%, la misma ocupará el mismo espacio en todo tipo de resoluciones, convirtiendo así a nuestro diseño en compatible para todo tipo de pantalla.

También podemos especificar el ancho de cada celda agregando el atributo width en las etiquetas <td>. Vale aclarar que si indicamos el ancho en pixeles en todas las celdas y también en la tabla, la sumatoria del ancho de las celdas debe tener el mismo valor que el ancho de la tabla para evitar resultados no deseados en el diseño final de la página. Igualmente se recomienda utilizar porcentaje en el ancho de las celdas; ya que el porcentaje será a razón del ancho total de la tabla y no de la página en que esté contenida la tabla. Así por ejemplo si una tabla está dentro de una página e indicamos un ancho de 50%, la tabla tendrá un ancho del 50% de la página, mientras que si indicamos para una celda de la tabla un ancho del 20%, la celda ocupará el 20% del ancho de la tabla; así si tenemos una tabla con 5 celdas por fila de 20% de ancho cada una, la suma total de las mismas dará un 100% del ancho de la tabla.

Veamos algunos ejemplos:

```
<html>
<head>
<title>Ejemplo de tabla</title>
</head>

<body>
<h4>Tabla 1</h4>
<table border="1" cellpadding="10" cellspacing="5" width="50%">
<tr>
  <td width="50%">Fila 1, Celda 1</td><td width="50%">Fila 1, Celda 2</td>
</tr>
<tr>
  <td width="50%">Fila 2, Celda 1</td><td width="50%">Fila 2, Celda 2</td>
</tr>
</table>

<br />

<h4>Tabla 2</h4>
<table border="5" cellpadding="5" cellspacing="0" width="50%">
<tr>
  <td width="50%">Fila 1, Celda 1</td><td width="50%">Fila 1, Celda 2</td>
</tr>
<tr>
  <td width="50%">Fila 2, Celda 1</td><td width="50%">Fila 2, Celda 2</td>
</tr>
</table>

<br />

<h4>Tabla 3</h4>
<table border="20" cellpadding="20" cellspacing="5" width="100%">
<tr>
  <td width="50%">Fila 1, Celda 1</td><td width="50%">Fila 1, Celda 2</td>
</tr>
<tr>
  <td width="50%">Fila 2, Celda 1</td><td width="50%">Fila 2, Celda 2</td>
</tr>
</table>

<br />

<h4>Tabla 4</h4>
```



```
<table border="0" cellpadding="5" cellspacing="0" width="50%">
<tr>
  <td width="50%">Fila 1, Celda 1</td><td width="50%">Fila 1, Celda 2</td>
</tr>
<tr>
  <td width="50%">Fila 2, Celda 1</td><td width="50%">Fila 2, Celda 2</td>
</tr>
</table>

<br />

<h4>Tabla 5</h4>
<table border="1" cellpadding="5" cellspacing="0" width="50%">
<tr>
  <td width="50%">Fila 1, Celda 1</td><td width="50%">Fila 1, Celda 2</td>
</tr>
<tr>
  <td width="50%">Fila 2, Celda 1</td><td width="50%">Fila 2, Celda 2</td>
</tr>
</table>

</body>
</html>
```

Este sería el resultado del código fuente anterior:

Tabla 1

Fila 1, Celda 1	Fila 1, Celda 2
Fila 2, Celda 1	Fila 2, Celda 2

Tabla 2

Fila 1, Celda 1	Fila 1, Celda 2
Fila 2, Celda 1	Fila 2, Celda 2

Tabla 3

Fila 1, Celda 1	Fila 1, Celda 2
-----------------	-----------------

Fila 2, Celda 1

Fila 2, Celda 2

Tabla 4

Fila 1, Celda 1

Fila 1, Celda 2

Fila 2, Celda 1

Fila 2, Celda 2

Tabla 5

Fila 1, Celda 1	Fila 1, Celda 2
Fila 2, Celda 1	Fila 2, Celda 2

CELDA DE TÍTULO

Existe una etiqueta que se comporta de la misma manera que la etiqueta de celdas <td>, se trata de <th>. Funciona de manera muy parecida a <td> como una celda más, con la diferencia de que el texto incluido en <th> aparece en negrita y centrado, ya que se trata de una celda de título.

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de tabla</title>
</head>

<body>

<table border="1" cellpadding="0" cellspacing="0" width="50%">
<tr>
  <th width="50%">Columna 1</th><th width="50%">Columna 2</th>
</tr>
<tr>
  <td width="50%">Fila 1, Celda 1</td><td width="50%">Fila 1, Celda 2</td>
</tr>

</table>

</body>
</html>
```

Esto se vería así:

Columna 1	Columna 2
Fila 1, Celda 1	Fila 1, Celda 2

TÍTULO DE TABLA

También se puede incluir un título para la tabla con la etiqueta <caption>, el cual aparece por omisión en la parte de arriba y afuera de la tabla, y de manera centrada. La etiqueta <caption> debe ir inmediatamente después de la etiqueta <table> y antes de la primera etiqueta <tr>, esto debe ser respetado para obtener resultados correctos.

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de tabla</title>
</head>

<body>

<table border="1" cellpadding="0" cellspacing="0" width="50%">
<caption>Título de la tabla</caption>
<tr>
    <td width="50%">Fila 1, Celda 1</td><td width="50%">Fila 1, Celda 2</td>
</tr>
<tr>
    <td width="50%">Fila 2, Celda 1</td><td width="50%">Fila 2, Celda 2</td>
</tr>

</table>

</body>
</html>
```

Esto se vería así:

Título de la tabla

Fila 1, Celda 1	Fila 1, Celda 2
Fila 2, Celda 1	Fila 2, Celda 2

CONFIGURANDO VARIAS CELDAS A LA VEZ

Hay una etiqueta que nos permite ahorrar trabajo cuando queremos agregar propiedades a varias celdas en tablas extensas. Por ejemplo supongamos que tenemos una tabla de ancho 100%, de 5 columnas por 4 filas, con un total de 20 celdas y si queremos que cada celda tenga un ancho de 20%, en lugar de agregar el atributo width en cada celda, podemos utilizar la etiqueta <colgroup> que nos permite agrupar celdas en columnas y agregar todos los atributos que querramos, los cuales serán aplicados a todas las celdas de la columna que el grupo correspondiente representa.

La etiqueta <colgroup> debe ir inmediatamente luego de la etiqueta <table> y antes de la primera etiqueta <tr>. El atributo span es el que define cuántas columnas agrupa cada etiqueta <colgroup> además del atributo span esta etiqueta debe incluir los atributos que se aplicarán en las celdas de la/s columna/s que el grupo representa. Se pueden tener mas de un grupo de columnas en una tabla y éstas deben estar ubicadas una después de la otra y además deben incluir etiqueta de cierre; entre la etiqueta de apertura y cierre no hace falta incluir nada, todo va en la de apertura.

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de tabla</title>
</head>

<body>

<table border="1" cellpadding="0" cellspacing="0" width="100%">
```

```

<colgroup span="5" width="20%"></colgroup>
<tr>
    <td>Fila 1, Celda 1</td><td>Fila 1, Celda 2</td><td>Fila 1, Celda 3</td><td>Fila 1,
Celda 4</td><td>Fila 1, Celda 5</td>
</tr>
<tr>
    <td>Fila 2, Celda 1</td><td>Fila 2, Celda 2</td><td>Fila 2, Celda 3</td><td>Fila 2,
Celda 4</td><td>Fila 2, Celda 5</td>
</tr>
<tr>
    <td>Fila 3, Celda 1</td><td>Fila 3, Celda 2</td><td>Fila 3, Celda 3</td><td>Fila 3,
Celda 4</td><td>Fila 3, Celda 5</td>
</tr>
<tr>
    <td>Fila 4, Celda 1</td><td>Fila 4, Celda 2</td><td>Fila 4, Celda 3</td><td>Fila 4,
Celda 4</td><td>Fila 4, Celda 5</td>
</tr>
<tr>
    <td>Fila 5, Celda 1</td><td>Fila 5, Celda 2</td><td>Fila 5, Celda 3</td><td>Fila 5,
Celda 4</td><td>Fila 5, Celda 5</td>
</tr>

</table>

```



```

<table border="1" cellpadding="0" cellspacing="0" width="100%">
<colgroup span="1" width="50%"></colgroup>
<colgroup span="4" width="12.5%"></colgroup>
<tr>
    <td>Fila 1, Celda 1</td><td>Fila 1, Celda 2</td><td>Fila 1, Celda 3</td><td>Fila 1,
Celda 4</td><td>Fila 1, Celda 5</td>
</tr>
<tr>
    <td>Fila 2, Celda 1</td><td>Fila 2, Celda 2</td><td>Fila 2, Celda 3</td><td>Fila 2,
Celda 4</td><td>Fila 2, Celda 5</td>
</tr>
<tr>
    <td>Fila 3, Celda 1</td><td>Fila 3, Celda 2</td><td>Fila 3, Celda 3</td><td>Fila 3,
Celda 4</td><td>Fila 3, Celda 5</td>
</tr>
<tr>
    <td>Fila 4, Celda 1</td><td>Fila 4, Celda 2</td><td>Fila 4, Celda 3</td><td>Fila 4,
Celda 4</td><td>Fila 4, Celda 5</td>
</tr>
<tr>
    <td>Fila 5, Celda 1</td><td>Fila 5, Celda 2</td><td>Fila 5, Celda 3</td><td>Fila 5,
Celda 4</td><td>Fila 5, Celda 5</td>
</tr>

</table>

```

```
</body>
</html>
```

Esto se vería así:

Fila 1, Celda 1	Fila 1, Celda 2	Fila 1, Celda 3	Fila 1, Celda 4	Fila 1, Celda 5
Fila 2, Celda 1	Fila 2, Celda 2	Fila 2, Celda 3	Fila 2, Celda 4	Fila 2, Celda 5
Fila 3, Celda 1	Fila 3, Celda 2	Fila 3, Celda 3	Fila 3, Celda 4	Fila 3, Celda 5
Fila 4, Celda 1	Fila 4, Celda 2	Fila 4, Celda 3	Fila 4, Celda 4	Fila 4, Celda 5

Fila 1, Celda 1	Fila 1, Celda 2	Fila 1, Celda 3	Fila 1, Celda 4	Fila 1, Celda 5
Fila 2, Celda 1	Fila 2, Celda 2	Fila 2, Celda 3	Fila 2, Celda 4	Fila 2, Celda 5
Fila 3, Celda 1	Fila 3, Celda 2	Fila 3, Celda 3	Fila 3, Celda 4	Fila 3, Celda 5
Fila 4, Celda 1	Fila 4, Celda 2	Fila 4, Celda 3	Fila 4, Celda 4	Fila 4, Celda 5

En la primera tabla tenemos un grupo de 5 columnas en el que se aplicó el atributo width de ancho con un valor de 20%, haciendo que no fuera necesario incluir este atributo en cada celda de la tabla. En la segunda tabla tenemos dos grupos de columnas, el primero incluye una columna, indicado en el atributo span con el valor 1, mientras que el segundo grupo incluye cuatro columnas, indicado en el atributo span con el valor 4. En el primer grupo de una columna se aplicó el atributo width con valor 50% y en el segundo de cuatro columnas un width con valor 12.5% de ancho para cada celda.

ALINEAMIENTO DEL CONTENIDO DE CELDAS

Por omisión el contenido de las celdas aparece justificado a la izquierda, pero puede suceder que querramos centrar dicho contenido, para eso existe el atributo align, con el cual podemos justificar el contenido de las celdas a la izquierda, centro o derecha. También existe el atributo valign que nos permite alinear verticalmente el contenido de las celdas. Dichos atributos pueden ser utilizados tanto en la etiqueta <table> como en las etiquetas <td> de una tabla. Si los utilizamos en la etiqueta <table> tendrán validez en toda la tabla, mientras que si los utilizamos en una etiqueta <td> solamente tendrán validez en la celda donde se encuentran. También hay que tener en cuenta que si tenemos atributos de alineamiento tanto en <table> como en una celda, prevalecerán los valores del atributo de la celda. Por ejemplo, si queremos que en toda la tabla el contenido esté centrado salvo en una celda; debemos poner un atributo de alineamiento con valor centrado, align="center", en la etiqueta <table>, para que todas las celdas tengan contenido centrado mientras que en la celda que queremos que el contenido esté justificado a la izquierda ponemos align="left".

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de tabla</title>
</head>

<body>

<table border="1" cellpadding="0" cellspacing="0" width="50%" align="center"
valign="middle">
<tr>
<td width="50%">Contenido de la celda 1 de la fila 1 el cual se encuentra centrado y
alineado verticalmente en el medio</td><td width="50%" align="left" valign="top">Fila 1,
Celda 2</td>
```

```

</tr>
<tr>
  <td width="50%">Contenido de la celda 1 de la fila 2 el cual se encuentra centrado y
  alineado verticalmente en el medio</td><td width="50%" align="right" valign="bottom">Fila
  2, Celda 2</td>
</tr>

</table>

</body>
</html>

```

Esto se vería así:

Contenido de la celda 1 de la fila 1 el cual se encuentra centrado y alineado verticalmente en el medio.	Fila 1, Celda 2
Contenido de la celda 1 de la fila 2 el cual se encuentra centrado y alineado verticalmente en el medio.	Fila 2, Celda 2

En la etiqueta <table> indicamos en el atributo align que el contenido de las celdas de la tabla deben estar centradas con el valor center (centro), mientras que con el atributo valign indicamos que debe estar verticalmente alineado al medio con el valor middle (medio). En el <td> de la segunda celda de la primera fila especificamos que el contenido debe estar alineado a la izquierda con el valor left (izquierda) y verticalmente arriba con el valor top (tope). En el <td> de la celda 2 de la fila 2 indicamos que el contenido debe estar alineado a la derecha con el valor right (derecha) mientras que en el atributo valign indicamos que queremos el contenido verticalmente alineado abajo con el valor bottom (fondo).

Como se puede ver en el ejemplo anterior los valores de los atributos align y valign de las celdas de la derecha se superponen a los valores predeterminados de la tabla, indicados en los atributos align y valign de la etiqueta <table>.

CELDA QUE OCUPAN MÁS DE UNA FILA DE ALTO O COLUMNA DE ANCHO

Hasta ahora hemos creado tablas en formato de grilla regular con celdas que ocupan una fila de alto y una columna de ancho; sin embargo se puede hacer que una celda llegue a tener un ancho mayor al de una columna o un alto mayor al de una fila. Para lograr eso se utiliza en el <td> de la celda que queremos redimensionar, el atributo colspan para hacer que una celda tenga un ancho mayor a una columna o el atributo rowspan para que tenga un alto mayor a una fila.

Veamos un ejemplo:

```

<html>
<head>
<title>Ejemplo de tabla</title>
</head>

<body>
<h4>Tabla 1</h4>
<table border="1" cellpadding="5" cellspacing="0" width="100%">

<tr>

```

```

    <td colspan="4" align="center"><b>Celda de 4 columnas de ancho</b></td>
</tr>
<tr>
    <td width="25%">Fila 2, Columna 1</td><td width="25%">Fila 2, Columna 2</td><td
width="25%">Fila 2, Columna 3</td><td width="25%">Fila 2, Columna 4</td>
</tr>

</table>

<h4>Tabla 2</h4>
<table border="1" cellpadding="5" cellspacing="0" width="50%">

<tr>
    <td rowspan="4" width="50%" align="center" valign="middle"><b>Celda de 4 filas de
alto</b></td><tdwidth="50%"><b>Fila 1, Columna 2</b>
</tr>
<tr>
    <td width="50%">Fila 2, Columna 2</td>
</tr>
<tr>
    <td width="50%">Fila 3, Columna 2</td>
</tr>
<tr>
    <td width="50%">Fila 4, Columna 2</td>
</tr>

</table>

</body>
</html>

```

Esto se vería así:

Tabla 1

Celda de 4 columnas de ancho			
Fila 2, Columna 1	Fila 2, Columna 2	Fila 2, Columna 3	Fila 2, Columna 4

Tabla 2

Celda de 4 filas de alto	Fila 1, Columna 2
	Fila 2, Columna 2
	Fila 3, Columna 2
	Fila 4, Columna 2

Cuando diseñamos una tabla debemos ser muy cuidadosos y respetar el orden y la cantidad de celdas y filas, o sea que la cantidad de celdas en una fila deben coincidir con la cantidad de celdas en las demás filas de la tabla; pero cuando diseñamos una tabla con celdas de más de una fila de alto o más de una columna de ancho debemos ser aún más cuidadosos. Como se ve en el código fuente anterior en la primera fila de la primera tabla, donde la celda ocupa cuatro columnas de ancho, solamente agregamos una celda a diferencia de la segunda fila donde sí agregamos las cuatro

celdas. En la segunda tabla, donde tenemos una celda de cuatro filas de alto; esta celda pertenece a la primera fila, por lo tanto dentro del primer <tr> incluimos dos celdas, la primera con el rowspan de 4 filas y la de la segunda columna de tamaño normal; ya en las otras tres filas <tr> solamente debemos incluir una celda por fila pertenecientes a la segunda columna, ya que la primera columna está totalmente ocupada por la celda que tiene una altura de cuatro filas.

CABECERA, CUERPO Y PIE DE TABLAS

Otra posibilidad que nos da HTML es la de dividir una tabla en tres secciones, la de cabecera, cuerpo y pie de la tabla. Quizá no se utiliza mucho, pero a veces puede llegar a ser útil. Las etiquetas que definen estas secciones son <thead> para la cabecera de la tabla que va en la parte de arriba, <tbody> para el cuerpo de la misma y <tfoot> para el pie.

Tanto <thead>, <tfoot>, como <tbody> tienen que contener una o más etiquetas <tr>, las tres deben ir siempre dentro de la tabla, o sea que deben ir colocadas después del <table> de apertura, así como siempre deben estar colocadas después de las etiquetas <caption> o <colgroup> en caso de que la tabla contuviera alguna de estas etiquetas.

Otra cosa a tener en cuenta es que las secciones deben estar en este orden, primero se coloca la sección de cabecera <thead> con todo su contenido de filas <tr> y celdas <td> o <th>; luego la sección de pie de tabla <tfoot> con todas las filas <tr> y celdas <td> y finalmente la sección de cuerpo <tbody> con sus filas <tr> y celdas <td>.

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de tabla</title>
</head>

<body>

<table border="1" cellspacing="0" cellpadding="5">
<caption>Tabla con cabecera, cuerpo y pie</caption>
<thead>
<tr>
    <td>Fila 1, Celda 1</td><td>Fila 1, Celda 2</td>
</tr>
</thead>
<tfoot>
<tr>
    <td>Fila 5, Celda 1</td><td>Fila 5, Celda 2</td>
</tr>
</tfoot>
<tbody>
<tr>
    <td>Fila 2, Celda 1</td><td>Fila 2, Celda 2</td>
</tr>
<tr>
    <td>Fila 3, Celda 1</td><td>Fila 3, Celda 2</td>
</tr>
<tr>
    <td>Fila 4, Celda 1</td><td>Fila 4, Celda 2</td>
</tr>
</tbody>
</table>

</body>
</html>
```


Tabla con cabecera, cuerpo y pie

<i>Fila 1, Celda 1</i>	<i>Fila 1, Celda 2</i>
<i>Fila 2, Celda 1</i>	<i>Fila 2, Celda 2</i>
<i>Fila 3, Celda 1</i>	<i>Fila 3, Celda 2</i>
<i>Fila 4, Celda 1</i>	<i>Fila 4, Celda 2</i>
<i>Fila 5, Celda 1</i>	<i>Fila 5, Celda 2</i>

Como vemos en el ejemplo anterior primero se colocó la sección <thead>, luego la <tfoot> y finalmente la <tbody>, sin embargo en la tabla aparecen primero la sección de cabecera, luego la del cuerpo de la tabla y finalmente la del pie.

Mediante la división de una tabla en secciones de cabecera, cuerpo y pie se pueden agregar atributos distintos a los contenidos de cada una de estas. Por ejemplo si se quiere que las filas y celdas de cabecera tengan un cierto color de fondo y de fuente con texto centrado, las celdas de cuerpo tengan otros colores y texto justificado a la izquierda y las de pie tengan color de fondo y de fuentes distinto al de las otras secciones, la división de la tabla en secciones puede llegar a ser muy útil, ya que los atributos de estilo se aplicarían en <thead>, <tfoot> y <tbody> y todas las etiquetas de adentro heredarían estos atributos. Vale mencionar que los atributos de estilo se definen mediante un sublenguaje llamado CSS que se utiliza en conjunto con HTML, y que veremos más adelante en capítulos siguientes del curso.

MARCOS DE TABLA

Existe un atributo de <table> que permite tener bordes de tabla completos o parciales, o sea que solamente agregan bordes en uno o más lados de la tabla. Por ejemplo puede que se quiera tener borde solamente en el lado de arriba.

El atributo se llama frame que significa marco en inglés y los valores disponibles del mismo son: border (borde), box (caja), void (vacío), above (arriba), below (abajo), hside (que proviene de lados horizontales), vside (que proviene de lados verticales), lhs (para tener borde solamente en el lado izquierdo) y rhs (para borde solamente en el lado derecho).

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de tabla</title>
</head>

<body>

<table frame="border">
<caption>Marco de borde completo</caption>
<tr>
<td>Fila 1, Celda 1</td><td>Fila 1, Celda 2</td>
</tr>
<tr>
<td>Fila 2, Celda 1</td><td>Fila 2, Celda 2</td>
</tr>
<tr>
<td>Fila 3, Celda 1</td><td>Fila 3, Celda 2</td>
</tr>
<tr>
<td>Fila 4, Celda 1</td><td>Fila 4, Celda 2</td>
</tr>
<tr>
<td>Fila 5, Celda 1</td><td>Fila 5, Celda 2</td>
</tr>
</table>
```

```
<table frame="box">
<caption>Marco de caja (box) - Mismo resultado que border -</caption>
<tr>
  <td>Fila 1, Celda 1</td><td>Fila 1, Celda 2</td>
</tr>
<tr>
  <td>Fila 2, Celda 1</td><td>Fila 2, Celda 2</td>
</tr>
<tr>
  <td>Fila 3, Celda 1</td><td>Fila 3, Celda 2</td>
</tr>
<tr>
  <td>Fila 4, Celda 1</td><td>Fila 4, Celda 2</td>
</tr>
<tr>
  <td>Fila 5, Celda 1</td><td>Fila 5, Celda 2</td>
</tr>
</table>
```

```
<table frame="void">
<caption>Marco vacío</caption>
<tr>
  <td>Fila 1, Celda 1</td><td>Fila 1, Celda 2</td>
</tr>
<tr>
  <td>Fila 2, Celda 1</td><td>Fila 2, Celda 2</td>
</tr>
<tr>
  <td>Fila 3, Celda 1</td><td>Fila 3, Celda 2</td>
</tr>
<tr>
  <td>Fila 4, Celda 1</td><td>Fila 4, Celda 2</td>
</tr>
<tr>
  <td>Fila 5, Celda 1</td><td>Fila 5, Celda 2</td>
</tr>
</table>
```

```
<table frame="above">
<caption>Marco de borde solamente arriba</caption>
<tr>
  <td>Fila 1, Celda 1</td><td>Fila 1, Celda 2</td>
</tr>
<tr>
  <td>Fila 2, Celda 1</td><td>Fila 2, Celda 2</td>
</tr>
<tr>
  <td>Fila 3, Celda 1</td><td>Fila 3, Celda 2</td>
</tr>
```

```
<tr>
  <td>Fila 4, Celda 1</td><td>Fila 4, Celda 2</td>
</tr>
<tr>
  <td>Fila 5, Celda 1</td><td>Fila 5, Celda 2</td>
</tr>
```

```
</table>
```

```
<table frame="below">
<caption>Marco de borde solamente abajo</caption>
<tr>
  <td>Fila 1, Celda 1</td><td>Fila 1, Celda 2</td>
</tr>
<tr>
  <td>Fila 2, Celda 1</td><td>Fila 2, Celda 2</td>
</tr>
<tr>
  <td>Fila 3, Celda 1</td><td>Fila 3, Celda 2</td>
</tr>
<tr>
  <td>Fila 4, Celda 1</td><td>Fila 4, Celda 2</td>
</tr>
<tr>
  <td>Fila 5, Celda 1</td><td>Fila 5, Celda 2</td>
</tr>
```

```
</table>
```

```
<table frame="hsides">
<caption>Marco de borde solamente en los lados horizontales</caption>
<tr>
  <td>Fila 1, Celda 1</td><td>Fila 1, Celda 2</td>
</tr>
<tr>
  <td>Fila 2, Celda 1</td><td>Fila 2, Celda 2</td>
</tr>
<tr>
  <td>Fila 3, Celda 1</td><td>Fila 3, Celda 2</td>
</tr>
<tr>
  <td>Fila 4, Celda 1</td><td>Fila 4, Celda 2</td>
</tr>
<tr>
  <td>Fila 5, Celda 1</td><td>Fila 5, Celda 2</td>
</tr>
```

```
</table>
```

```
<table frame="vsides">
<caption>Marco de borde solamente en los lados verticales</caption>
<tr>
```

```

    <td>Fila 1, Celda 1</td><td>Fila 1, Celda 2</td>
</tr>
<tr>
    <td>Fila 2, Celda 1</td><td>Fila 2, Celda 2</td>
</tr>
<tr>
    <td>Fila 3, Celda 1</td><td>Fila 3, Celda 2</td>
</tr>
<tr>
    <td>Fila 4, Celda 1</td><td>Fila 4, Celda 2</td>
</tr>
<tr>
    <td>Fila 5, Celda 1</td><td>Fila 5, Celda 2</td>
</tr>
</table>

```

```

<table frame="lhs">
<caption>Marco de borde solamente en el lado izquierdo</caption>
<tr>
    <td>Fila 1, Celda 1</td><td>Fila 1, Celda 2</td>
</tr>
<tr>
    <td>Fila 2, Celda 1</td><td>Fila 2, Celda 2</td>
</tr>
<tr>
    <td>Fila 3, Celda 1</td><td>Fila 3, Celda 2</td>
</tr>
<tr>
    <td>Fila 4, Celda 1</td><td>Fila 4, Celda 2</td>
</tr>
<tr>
    <td>Fila 5, Celda 1</td><td>Fila 5, Celda 2</td>
</tr>
</table>

```

```

<table frame="rhs">
<caption>Marco de borde solamente en el lado derecho</caption>
<tr>
    <td>Fila 1, Celda 1</td><td>Fila 1, Celda 2</td>
</tr>
<tr>
    <td>Fila 2, Celda 1</td><td>Fila 2, Celda 2</td>
</tr>
<tr>
    <td>Fila 3, Celda 1</td><td>Fila 3, Celda 2</td>
</tr>
<tr>
    <td>Fila 4, Celda 1</td><td>Fila 4, Celda 2</td>
</tr>
<tr>

```

```
<td>Fila 5, Celda 1</td><td>Fila 5, Celda 2</td>
</tr>

</table>

</body>
</html>
```

Marco incorporado

Otra cosa que se puede hacer en HTML es incluir una página dentro de otra página. La etiqueta que permite hacer esto es `<iframe>` la cual mediante el uso de atributos nos permite elegir qué página queremos incluir, el tamaño que ésta tendrá dentro de la página donde la incluimos, si tendrá barra de desplazamiento o no, si tendrá borde.

Los atributos de `<iframe>` son: `src` para definir la dirección de la página que queremos incluir dentro de la página donde estamos trabajando; `width` (ancho) para definir el ancho de la página incorporada en píxeles o porcentaje; `height` (alto) para definir el alto de la página incorporada en píxeles o porcentaje; `scrolling` para indicar si se quiere que incluya barra de desplazamiento, los valores posibles son `yes`, `no`, y `auto`; y finalmente el atributo `frameborder` para definir si se quiere que el marco incorporado tenga borde o no, los valores posibles son `1` para agregar un borde y `0` para que no tenga borde. Una cosa a tener en cuenta es que el atributo `frameborder` en algunos exploradores Web no funciona.

Para los navegadores Web que no soportan marcos incorporados hay que agregar un texto entre la etiqueta `<iframe>` de apertura y la de cierre. En los casos de no ser soportados aparecerá el texto, sino se verá la página incluida.

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de marco incorporado</title>
</head>

<body>
<iframe src="http://www.youbioit.com" width="50%" height="500" scrolling="no"
frameborder="1"> Este texto aparece si el explorador Web utilizado no soporta marcos
incorporados</iframe>
<br /><br />
<iframe src="http://www.youbioit.com" width="50%" height="500"
scrolling="yes" frameborder="0"> Este texto aparece si el explorador Web utilizado no soporta
marcos incorporados</iframe>

</body>
</html>
```

El código fuente anterior debe mostrar en un navegador Web lo siguiente:



iframes en HTML - Clickear para ver en grande

Formularios HTML

Un elemento muy importante en el diseño y desarrollo de páginas Web es el de ingreso de datos por parte de los usuarios para que éstos sean procesados por algún programa y se devuelvan resultados. Desde formularios de contacto, pasando por buscadores, formularios de inscripción, hasta campos de texto en foros de discusión y comunidades virtuales, el ingreso de datos es esencial para el funcionamiento de los sitios Web que los utilizan.

Sin la capacidad de ingreso de datos no existiría lo que hoy conocemos como Web 2.0; antes de la cual los sitios Web estaban armados únicamente por los administradores o webmasters de los mismos y los usuarios solamente podían ver el contenido sin poder participar. Con la modalidad de la Web 2.0, los usuarios son principalmente los que hacen crecer a los sitios Web mediante la información y datos que ingresan. Las comunidades virtuales (como Youbioit.com en donde soy un simple usuario y pude escribir este tutorial de HTML) donde son los usuarios los que agregan temas, páginas, videos, imágenes, comentarios, mensajes, etc. dependen del uso de formularios de ingreso de datos para que dichos datos puedan ser introducidos en los sitios Web, almacenados y luego puedan ser vistos por otros usuarios.

Por ejemplo cuando uno realiza una búsqueda en un buscador, debe ingresar los datos en un campo de texto y luego presionar un botón para que ésta se realice. Cuando uno deja un mensaje, artículo o comentario en una comunidad virtual como Youbioit, Twitter, etc. la información se ingresa en un área de texto y se envía tocando un botón. Cuando uno se inscribe en un sitio Web para convertirse en usuario del mismo, los datos deben ser ingresados mediante el uso de un formulario y enviados tocando un botón.

Por lo tanto la importancia que tienen los formularios es muy grande, ya que permiten a los usuarios comunicarse, agregar archivos, información y haciendo crecer a Internet, sin necesidad de ser programadores o desarrolladores Web.

El funcionamiento de los formularios cuenta con dos partes, la estética o estructural y la funcional. La parte estética o estructural es la del diseño del formulario, con sus campos de texto, menús de opciones, botones, etc.; mientras que la funcional es la del programa que procesa los datos introducidos en el formulario devolviendo una respuesta.

Este proceso funciona de la siguiente manera: cuando un usuario (técnicamente se lo denomina cliente) entra a una página con un formulario su computadora la descarga temporalmente (como a todas las páginas de Internet) en la memoria de la misma; el usuario completa el formulario, supongamos que se trata de un buscador con un campo de texto de búsqueda y un botón de Enviar. Una vez que el usuario presiona el botón, los datos son enviados a un programa que se encuentra en el servidor del sitio Web, donde los datos ingresados son procesados; en este caso el proceso realizado por el programa ubicado en el servidor es el de buscar los términos ingresados en el campo de texto a través de una base de datos que también se encuentra en el servidor; en caso de coincidir las palabras ingresadas o al menos ser similares a las encontradas en la base de datos, el programa automáticamente genera una página con las respuestas más cercanas a la búsqueda, la cual es enviada a la computadora del usuario que realizó la búsqueda.

Otro ejemplo sería el de un formulario de contacto; primero el usuario abre una página la cual se baja a la memoria de la computadora, en la página hay un formulario de contacto el cual cuenta con una serie de campos de texto donde el usuario escribe sus datos y el área de texto donde deja un mensaje, luego presiona el botón de enviar y los datos así como el mensaje del área de texto son enviados a un programa ubicado en el servidor del sitio Web, el cual procesa los datos generando automáticamente un mail que es enviado a una dirección especificada en el programa, en este caso sería la dirección de mail del Webmaster o administrador del sitio, quien recibe todos los datos ingresados por el usuario así como el mensaje en formato de correo electrónico. Simultáneamente el programa envía a la computadora del usuario una página de agradecimiento, con un texto que dice que el mensaje enviado será leído por el administrador.

La parte estructural o estética es la del formulario y sus elementos visuales, la cual tiene un funcionamiento de tipo client side (del lado del cliente), ya que se ejecuta o baja en la memoria de la computadora del usuario; mientras que la parte funcional del formulario, o sea el programa que procesa los datos ingresados y enviados por el usuario, tiene un funcionamiento de tipo server side (del lado del servidor) ya que se ejecuta en la memoria de la computadora del servidor. La parte estructural se diseña con lenguaje HTML mientras que la parte funcional debe programarse con un lenguaje lógico como PHP, ASP u otro. HTML no tiene comandos lógicos, por lo tanto solamente sirve para diseñar las partes estéticas y estructurales de las páginas Web. Sin embargo antes de aprender otros lenguajes de desarrollo Web hay que aprender HTML y CSS, los cuales son los principales lenguajes de diseño Web, y que serán cubiertos plenamente en este tutorial. A continuación veremos cómo se diseñan los formulario HTML.

ELEMENTOS DE FORMULARIOS HTML

La etiqueta que nos permite agregar un formulario a una página es <form>, por lo tanto todos los elementos que forman parte del mismo (campos de texto, menús, botones, etc.) deben estar entre la etiqueta <form> de apertura y la de cierre.

La etiqueta <input> nos permite agregar distintos tipos de elementos de ingreso de datos, como campos de texto, botones de opción, casillas de selección y botones simples. El tipo de elemento de ingreso de datos se elige mediante el uso del atributo de <input> type y los distintos valores posibles de este atributo son: text (para campo de texto), checkbox (para casilla de selección), radio (para botón de opción), submit (para botón de envío de datos), y reset (para botón de restablecer). La etiqueta <input> no utiliza etiqueta de cierre así que se cierra al final como sucede con

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de formulario</title>
</head>

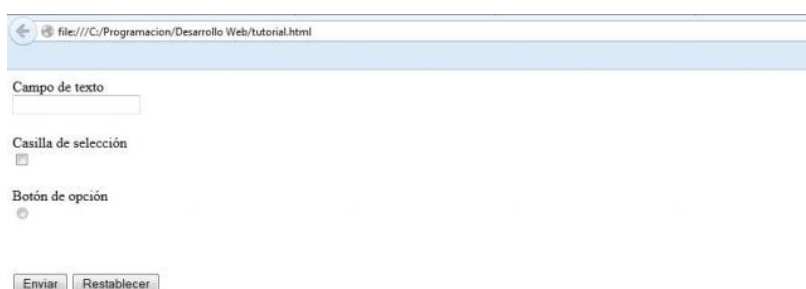
<body>
<form>
Campo de texto<br />
<input type="text" />
<br /><br />
Casilla de selección<br />
<input type="checkbox" />
```

```
<br /><br />
Botón de opción<br />
<input type="radio" /><br /><br />
<br /><br />
<input type="submit" value="Enviar" /> <input type="reset" value="Restablecer" />

</form>

</body>
</html>
```

El código fuente anterior debe mostrar lo siguiente en el navegador Web



[Formularios Web - Clickear para ver en grande](#)

El ejemplo anterior es muy simple y simplemente agrega los distintos elementos que se pueden tener en un formulario.

A continuación vamos a analizar cada elemento con más detalles. También vamos a ver los otros atributos de `<input>`.

Un atributo muy importante de `<input>` que le da verdadera funcionalidad en un formulario, es `name` (nombre); el cual representa al nombre de la variable donde se almacenará el dato ingresado en el elemento. A pesar que este es un tema de programación algo más avanzada; es importante al menos tener idea de lo que es una variable.

Cuando trabajamos con una computadora todos los datos y órdenes, el funcionamiento en general, se almacenan temporalmente en la memoria de la máquina. Para graficarlo de una manera fácil de entender, la memoria está subdividida en celdas de memoria de un byte cada una y un número de dirección que la identifica; la totalidad de todas las celdas dan como resultado la cantidad de memoria que tiene una computadora. Los programas utilizan la memoria para almacenar temporalmente, mientras se ejecutan, los datos ingresados en celdas de memoria. Para identificarlas se requiere el uso de un nombre, este nombre es la variable. Las variables ocupan un cierto espacio de la memoria, acorde al tipo de datos que se almacenan. Por lo tanto, los datos se almacenan temporalmente en variables que son utilizadas por los programas para identificar la posición de memoria donde estas se encuentran.

Supongamos que en un campo de texto de un buscador ingresamos una frase para que el programa de la computadora del servidor del motor de búsqueda compare a la frase de los datos ingresados con los datos que tiene en la base de datos y si encuentra una coincidencia devuelva una página con la lista de respuestas o links de páginas relacionadas; para que todo esto ocurra la frase ingresada debe ser enviada a la computadora del servidor y almacenada en una variable que representa un espacio de la memoria de esa computadora, para que el programa de búsqueda la utilice.

Cuando definimos el nombre de la variable donde se almacenarán los datos ingresados tenemos que fijarnos que éste coincida con el nombre de variable requerido por el programa que se ejecuta en el servidor.

Un ejemplo sería `<input type="text" name="nombre-de-variable" />`, esta etiqueta define un campo de texto, cuyos valores ingresados serán almacenados en la variable "nombre-de-variable" del programa que se encuentra en la memoria de la computadora del servidor donde se realizará el procesamiento de los datos introducidos.

Otro atributo muy importante, esta vez de la etiqueta `<form>` es `action` (acción), el cual define que acción realizará el formulario una vez enviados los datos ingresados. Por lo general la acción más utilizada es la de especificar la dirección del programa ubicado en el servidor que procesará los datos ingresados. Si la etiqueta `<form>` no tiene un atributo `action` con un valor, el formulario no realizará ninguna acción; ya que debemos recordar que con HTML solamente se diseña la parte estructural del formulario y el programa de procesamiento lógico server-side (ubicado en el servidor) se desarrolla en otro lenguaje como PHP o ASP. Así que en `action` se pone la dirección del programa de procesamiento.

Un ejemplo sería `<form action="www.ejemplo.com/buscador.php">`, el cual al enviar los datos ingresados ejecuta el programa ubicado en "" para que éstos sean procesados.

Hasta ahora vimos que hay que especificar en `<form>`, mediante el uso del atributo `action`, la dirección donde se encuentra el programa de procesamiento de datos y mediante el uso de `name` en las etiquetas `<input>` del formulario, los nombres de las variables donde se almacenarán los datos. Pero falta especificar el atributo que define la forma o método de envío de los datos. El atributo que define la manera de enviar los datos se llama `method` (método). Hay dos maneras de enviar los datos desde la computadora cliente (la del usuario) a la computadora del servidor: uno es el método GET y el otro el método POST.

Para entender plenamente la diferencia entre GET y POST hay que saber más de programación Web, sin embargo por ahora para los fines requeridos basta con conocer algunos aspectos de estos dos métodos.

Con el método de envío GET los datos ingresados son añadidos a la dirección URL del programa de procesamiento.

Supongamos que tenemos un buscador cuyo programa de procesamiento se encuentra en la dirección `www.ejemplo.com/buscador.php` (dirección que debemos especificar en el atributo `action` así: `<form action="www.ejemplo.com/buscador.php">`); la variable que almacena los datos del campo de texto se llama "palabra-de-busqueda" (que debe ser definida así `<input type="text" name="palabra-de-busqueda" />`); y finalmente el método de envío de datos es GET, la etiqueta `<form>` quedaría así: `<form action="www.ejemplo.com/buscador.php" method="GET">`. Si por ejemplo queremos buscar la palabra "tutoriales", esta sería enviada a la variable "palabra-de-busqueda" del programa de procesamiento a través de la dirección URL de la siguiente manera: `palabra-de-busqueda=tutoriales`; por lo tanto eso sería lo que veríamos en la barra de direcciones del explorador Web.

Mediante GET la información enviada se encuentra a la vista ya que se ve en la barra de direcciones; también la cantidad de datos incluidos es limitada. Nunca hay que usar el método GET para enviar datos que no se quiere que sean vistos, por ejemplo los datos de un formulario de inscripción el cual incluye datos personales o un nombre de clave.

Para enviar muchos datos, de manera más segura es conveniente utilizar el método de envío POST; ya que no presenta límites en la cantidad de información que se puede enviar y estos viajan ocultos en un paquete de información el cual no queda añadido a la dirección del programa de procesamiento.

El método de envío GET puede llegar a ser más conveniente para casos en los que no hay que ocultar la información enviada, como un motor de búsqueda; mientras que el método POST es más conveniente para casos como formularios de contacto o formularios de inscripción.

Veamos un ejemplo (Atención: la dirección que aparece en el atributo `action` no es real, por lo tanto debe ser reemplazada por una real; también hay que tener en cuenta que si no se tiene un programa de procesamiento en un servidor, el formulario no realizará ninguna acción y solamente tendrá su parte estructural o estética).

```
<html>
<head>
<title>Ejemplo de formulario</title>
</head>

<body>
<form action="www.ejemplo.com/formulario-de-contacto.php" method="POST">
<p>Nombre<br /><input type="text" name="nombre" /></p>

<p>Teléfono<br /><input type="text" name="telefono" /></p>
```

```

<p>Email<br /><input type="text" name="email" /></p>

<p>Dirección<br /><input type="text" name="direccion" /></p>

<p><input type="submit" value="Enviar" /> <input type="reset" value="Restablecer" /></p>

</form>

</body>
</html>

```

Esto muestra lo siguiente en un navegador Web:



The screenshot shows a web browser window with the address bar displaying 'file:///C:/Programacion/Desarrollo Web/tutorial.html'. Below the address bar is a form with four text input fields labeled 'Nombre', 'Teléfono', 'Email', and 'Dirección'. At the bottom of the form are two buttons: 'Enviar' and 'Restablecer'.

[Formulario Web - Clickear para ver en grande](#)

Otro atributo de `<input>` es `value`, mediante el cual se agrega un valor a la variable indicada en el atributo `name` directamente desde el código fuente HTML de la página que estamos diseñando. Por ejemplo si se le agrega un atributo `value` a un campo de texto, el valor especificado en el mismo aparecerá por omisión en el campo de texto cada vez que se descargue la página con el formulario; obviamente que los usuarios podrán reemplazar este valor predeterminado con tan sólo tocar dentro del campo de texto y borrarlo suplantándolo con otro texto de su preferencia.

Entre los distintos tipos de elementos de ingreso de datos también están las casillas de selección, las cuales se utilizan para seleccionar una o varias opciones de un grupo. Por ejemplo se puede elegir en un grupo de géneros de películas cuales son los géneros favoritos, o entre un grupo de marcas de autos, las marcas favoritas. Para agregar una casilla de selección hay que poner en el atributo `type` el valor `"checkbox"`.

Una cosa que hay que tener en cuenta es que al definir un grupo de casillas de selección deben especificarse los valores distintos en el atributo `value` de cada casilla, estos valores serán agregados en la variable de la casilla. Son necesarios ya que al seleccionar una casilla el usuario, el valor de esa casilla es enviado a la variable que utiliza el programa que procesa los datos en el servidor.

Otra cosa que hay que tener en cuenta es que el nombre de la variable de un grupo de casillas de selección debe ser el mismo para que los valores de las distintas casillas sean agregados a la misma variable; por ejemplo si se tienen cinco casillas en un grupo en el que hay que elegir marcas de autos favoritas, los valores de las casillas deben ser enviados a una variable de nombre `"marcas"` donde serán agregados para que el programa ubicado en el servidor los procese.

Veamos un ejemplo:

```

<html>
<head>
<title>Ejemplo de formulario</title>
</head>

<body>

```

```

<form action="www.ejemplo.com/formulario-de-contacto.php" method="POST">

<p>Nombre<br /><input type="text" name="nombre" /></p>

<p>Teléfono<br /><input type="text" name="telefono" /></p>

<p>Dirección<br /><input type="text" name="direccion" /></p>

<p>Email<br /><input type="text" name="email" value="escribe tu email" /></p>

<p><u>Géneros de película favoritos</u><br />
<input type="checkbox" name="peliculas" value="Accion" />Acción<br />
<input type="checkbox" name="peliculas" value="Ciencia Ficción" />Ciencia Ficción<br />
<input type="checkbox" name="peliculas" value="Comedia" />Comedia<br />
<input type="checkbox" name="peliculas" value="Intriga" />Intriga<br />
<input type="checkbox" name="peliculas" value="Terror" />Terror<br /></p>

<p><input type="submit" value="Enviar" /> <input type="reset" value="Restablecer" /></p>

</form>

</body>
</html>

```

El código fuente anterior muestra lo siguiente en un navegador Web:

[Cliquear para ver en grande](#)

Otro elemento de formulario son los botones de opción; a diferencia de las casillas de selección en las que se pueden elegir una o más opciones en estos últimos solamente se puede seleccionar una opción del grupo, por eso mismo también se los conoce como botones de selección excluyente, ya que al tocar una opción se deselecta la que ya estaba elegida antes. El valor que se debe especificar en el atributo type de la etiqueta <input> es "radio".

Cuando se arma un grupo de botones de opción excluyente, cada uno de los elementos del mismo debe tener el mismo valor en el atributo de variable name; para que de esta manera se consideren todos partes del mismo grupo y de esta manera al seleccionar una nueva opción se deselecte cualquier otra del grupo con el mismo nombre de variable.

Veamos un ejemplo:

```

<html>
<head>
<title>Ejemplo de formulario</title>
</head>

<body>
<form action="www.ejemplo.com/formulario-de-contacto.php" method="POST">

<p>Nombre<br /><input type="text" name="nombre" /></p>

<p>Teléfono<br /><input type="text" name="telefono" /></p>

<p>Email<br /><input type="text" name="email" /></p>

<p>Dirección<br /><input type="text" name="direccion" /></p>

<p><u>Nivel de estudios alcanzado</u><br />
<input type="radio" name="estudios" value="Primario" />Primario<br />
<input type="radio" name="estudios" value="Secundario" />Secundario<br />
<input type="radio" name="estudios" value="Terciario" />Terciario<br />
<input type="radio" name="estudios" value="Universitario" />Universitario<br />
<input type="radio" name="estudios" value="Postgrado" />Postgrado<br /></p>

<p><u>Géneros de película favoritos</u><br />
<input type="checkbox" name="peliculas" value="Accion" />Acción<br />
<input type="checkbox" name="peliculas" value="Ciencia Ficción" />Ciencia Ficción<br />
<input type="checkbox" name="peliculas" value="Comedia" />Comedia<br />
<input type="checkbox" name="peliculas" value="Intriga" />Intriga<br />
<input type="checkbox" name="peliculas" value="Terror" />Terror<br /></p>

<p><input type="submit" value="Enviar" /> <input type="reset" value="Restablecer" /></p>

</form>

</body>
</html>

```

El código fuente anterior muestra lo siguiente en los navegadores Web:

file:///C:/Programacion/Desarrollo Web/tutorial.html

Nombre

Teléfono

Email

Dirección

Nivel de estudios alcanzado

☐ Primario
☐ Secundario
☐ Terciario
☐ Universitario
☐ Postgrado

Géneros de película favoritos

☐ Acción
☐ Ciencia Ficción
☐ Comedia
☐ Intriga
☐ Terror

[Clicar para ver en grande](#)

Como se ha visto en los ejemplos anteriores los botones de envío de datos se definen poniendo el valor "submit" (enviar o presentar) en el atributo type; y que el valor especificado en el atributo value es el rótulo que aparece en el botón.

Por otro lado, también vimos en los ejemplos anteriores que para definir un botón de restablecer (que al presionarse devuelve todos los campos de texto y casilleros a sus estados originales) hay que poner el valor "reset" (restablecer) en el atributo type; mientras que el valor especificado en el atributo value será el rótulo que aparecerá en el botón.

BOTON DE IMAGEN

Los botones de envío de datos que se definen con el valor "submit" tienen un diseño preestablecido por HTML, en caso de necesitar algún diseño más personalizado de botón existe la posibilidad de utilizar una imagen con funcionalidad de botón de envío de datos de formulario. Esto se logra agregando en el atributo type de <input> el valor "image" (imagen); y el atributo src (diminutivo de "source", que en inglés significa fuente), donde se debe especificar la dirección de la imagen que se utilizará como botón.

Otros dos atributos que se pueden utilizar al definir un botón de imagen son width (ancho) y height (alto), los cuales funcionan de la misma manera que en la etiqueta , indicando el ancho y el alto en pixeles o porcentaje.

Al igual que en , cuando definimos un botón de imagen podemos utilizar en <input> el atributo alt, en el que se puede especificar un texto alternativo el cual aparece en casos de que la imagen no pudiera ser cargada.

BOTONES CON DISTINTAS FUNCIONALIDADES

También se pueden definir botones que no tienen funcionalidad de envío de datos ingresados en un formulario y que se suelen utilizar para activar alguna acción programada en un lenguaje de programación Web como Javascript, el cual sirve para darle dinamismo a las páginas Web (efectos especiales, animaciones, etc). Por ejemplo se puede definir un botón con el cual se puede cambiar el color de fondo de una página, o se puede realizar algún efecto especial, etc).

Este tipo de botón no tiene mucha utilidad en formularios. Para definir un botón así hay que utilizar el atributo de <input> button (botón). Como se mencionó antes por lo general se lo debe utilizar en combinación de otro lenguaje de programación Web como Javascript, el cual ahora no veremos en este tutorial pero sí en uno próximo más avanzado.

CAMPOS DE ENVIO DE ARCHIVOS

En muchos sitios Web existe la posibilidad de subir o enviar archivos; por ejemplo cuando se quiere adjuntar archivos en un mail, o cuando se sube una imagen o un video se utilizan campos de envío de archivos, los cuales incluyen un botón que al presionarse nos da la posibilidad de elegir un archivo de nuestro disco rígido.

Para definir un campo de envío de archivos se debe poner el valor "file" en el atributo type de la etiqueta <input>.

CAMPOS DE TEXTO CON CARACTERES OCULTOS

Un tipo de campo de texto que se suele utilizar por lo general en formularios de inscripción o cuando se deben ingresar datos personales o secretos como una clave son los campos de caracteres ocultos, en los cuales al ingresar un texto se muestran puntos o asteriscos que ocultan la información introducida.

Para definir un campo de texto con caracteres ocultos se debe poner el valor "password" (clave) en el atributo type de la etiqueta <input>.

ENVIO DE INFORMACION OCULTA

A veces para que el programa de procesamiento ubicado en el servidor funcione correctamente se necesita enviar ciertos datos ocultos, los cuales no son ingresados por el usuario sino que los envía la misma página Web automáticamente cada vez que se toca el botón de envío de datos.

Por ejemplo en un formulario de contacto los datos ingresados por los usuarios son enviados a un programa de procesamiento ubicado en el servidor, el cual arma un mail con toda la información ingresada en los elementos del formulario y lo envía a una dirección de correo electrónico que por lo general es la del Webmaster o administrador del sitio Web. Para que esto ocurra al diseñar la página de formulario debemos especificar la dirección a la que siempre serán enviados los mail que crea el programa de procesamiento ubicado en el servidor.

Para definir datos que serán enviados automáticamente de manera oculta cada vez que se presione el botón de envío de datos, se debe poner el valor "hidden" (oculto) en el atributo type de la etiqueta <input>; mientras que como en otros casos la variable y el valor a enviar se definen con los atributos name y value respectivamente.

Veamos un ejemplo (Atención: los valores introducidos en la dirección de la imagen del botón de envío, el valor de la dirección de email donde se envían los datos y la del programa de procesamiento deben ser reemplazados con valores reales. En el caso del programa de procesamiento, se requiere tener uno en un servidor para que los datos puedan ser procesados; sin embargo si no se tiene uno no importa para los fines de este tutorial, ya que estamos aprendiendo diseño de páginas Web con HTML, y para los fines visuales de un formulario no se requiere un programa de procesamiento en un servidor el cual solamente cumple funciones funcionales.)

```
<html>
<head>
<title>Ejemplo de formulario</title>
</head>

<body>

<input type="button" value="Puedes presionarme igual no hago nada" />

<form action="www.ejemplo.com/formulario-de-contacto.php" method="POST">
<input type="hidden" name="mail" value="mimail@ejemplo.com" />

<p>Nombre<br /><input type="text" name="nombre" /></p>

<p>Teléfono<br /><input type="text" name="telefono" /></p>

<p>Email<br /><input type="text" name="email" /></p>

<p>Dirección<br /><input type="text" name="direccion" /></p>

<p>Nombre de usuario<br /><input type="text" name="usuario" /></p>

<p>Clave<br /><input type="password" name="clavesecreta" /></p>

<p>Subir imagen de usuario<br /><input type="file" name="foto-de-usuario" /></p>

<p><u>Nivel de estudios alcanzado</u><br />
<input type="radio" name="estudios" value="Primario" />Primario<br />
```

```

<input type="radio" name="estudios" value="Secundario" />Secundario<br />
<input type="radio" name="estudios" value="Terciario" />Terciario<br />
<input type="radio" name="estudios" value="Universitario" />Universitario<br />
<input type="radio" name="estudios" value="Postgrado" />Postgrado<br /></p>

<p><u>Géneros de película favoritos</u><br />
<input type="checkbox" name="peliculas" value="Accion" />Acción<br />
<input type="checkbox" name="peliculas" value="Ciencia Ficción" />Ciencia Ficción<br />
<input type="checkbox" name="peliculas" value="Comedia" />Comedia<br />
<input type="checkbox" name="peliculas" value="Intriga" />Intriga<br />
<input type="checkbox" name="peliculas" value="Terror" />Terror<br /></p>

<p><input type="image" src="imagendeboton.jpg" alt="Enviar" width="170"
height="40" /> </p>

</form>

</body>
</html>

```

El código fuente anterior muestra lo siguiente en los navegadores Web:

[Cliquear para ver en grande](#)

Hay otros cinco atributos de `<input>` que funcionan únicamente con ciertos tipos de elementos.

El atributo `checked` funciona únicamente con las casillas de selección y los botones de opción, tipos `"checkbox"` y `"radio"` respectivamente; el mismo sirve para que al cargarse la página Web con el formulario, ya haya un botón de opción tildado o seleccionado o una o varias casillas de selección tildadas. El valor que debe agregarse también es `"checked"`.

El atributo disabled (deshabilitado) sirve para deshabilitar elementos del formulario; este atributo se puede utilizar en todos los elementos <input> y el valor que hay que poner en el atributo disabled también es "disabled".

El atributo maxlength (longitud máxima) se utiliza únicamente con los campos de texto (tipo "text") y sirve para indicar la máxima cantidad de caracteres que se pueden ingresar en el mismo. El valor que hay que poner en maxlength es un número que indica la cantidad máxima de caracteres que se pueden ingresar en el campo de texto.

Otro atributo que se utiliza con los campos de texto es size (tamaño), el cual sirve para indicar el ancho del campo de texto en cantidad de caracteres. El valor que hay que poner en el atributo size es un número que indica la cantidad de caracteres de ancho que tendrá el campo de texto; por ejemplo un valor de "40" indica 40 caracteres de ancho. El atributo size también sirve para definir el ancho de los campos de envío de archivos con valor "file" en el atributo type.

No hay que confundir maxlength con size, el primero indica la cantidad máxima de caracteres que se pueden escribir dentro del campo de texto, mientras que el segundo indica el ancho en caracteres que tendrá el campo en sí.

Otro atributo que también funciona con los campos de texto, es readonly (sólo lectura), el cual se utiliza para indicar que un campo de texto es de sólo lectura; esto significa que si por ejemplo ya contiene un texto predeterminado al cargar la página (el cual se define en el atributo value como ya hemos visto) y además tiene un atributo readonly con valor "readonly"; no se podrá escribir nada en el mismo, solamente se podrá leer el texto predeterminado de value.

Veamos un ejemplo con los atributos anteriores:

```
<html>
<head>
<title>Ejemplo de formulario</title>
</head>

<body>

<input type="button" value="No puedes presionarme estoy deshabilitado"
disabled="disabled" />

<form action="www.ejemplo.com/formulario-de-contacto.php" method="POST">
<input type="hidden" name="mail" value="mimail@ejemplo.com" />

<p>Nombre<br /><input type="text" name="nombre" size="40" /></p>

<p>Email<br /><input type="text" name="email" size="40" /></p>

<p>Nombre de usuario<br /><input type="text" name="usuario" value="Nombre de usuario"
readonly="readonly" size="40" /></p>

<p>Clave<br /><input type="password" name="clavesecreta" size="40"
maxlength="5" /></p>

<p>Subir imagen de usuario<br /><input type="file" name="foto-de-
usuario" size="40" /></p>

<p><u>Nivel de estudios alcanzado</u><br />
<input type="radio" name="estudios" value="Primario" />Primario<br />
<input type="radio" name="estudios" value="Secundario"
checked="checked" />Secundario<br />
<input type="radio" name="estudios" value="Terciario" />Terciario<br />
<input type="radio" name="estudios" value="Universitario" />Universitario<br />
<input type="radio" name="estudios" value="Postgrado" />Postgrado<br /></p>

<p><u>Géneros de película favoritos</u><br />
```



```

<input type="checkbox" name="peliculas" value="Accion" />Acción<br />
<input type="checkbox" name="peliculas" value="Ciencia Ficción" />Ciencia Ficción<br />
<input type="checkbox" name="peliculas" value="Comedia" />Comedia<br />
<input type="checkbox" name="peliculas" value="Intriga" checked="checked" />Intriga<br />
<input type="checkbox" name="peliculas" value="Terror" checked="checked" />Terror<br />
/> </p>

<p><input type="image" src="imagendeboton.jpg" alt="Enviar" width="170"
height="40" /> </p>

</form>

</body>
</html>

```

El código fuente anterior muestra lo siguiente en los navegadores Web:

[Cliquear para ver en grande](#)

LISTAS Y MENUS

Hasta ahora hemos visto los elementos de formulario que se definen con la etiqueta `<input>`, sin embargo también existen otros elementos de formulario que se definen con otras etiquetas; éstos son los menús y listas de opciones, así como las áreas de texto que permiten ingresar comentarios más largos de varios renglones o párrafos.

Para agregar un menú o lista de opciones se debe utilizar la etiqueta `<select>` para definir el menú o la lista y la etiqueta `<option>` para definir cada una de las opciones, que deben colocarse entre las etiquetas `<select>` de apertura y cierre.

Además, para definir el valor de cada opción que será enviado al programa de procesamiento ubicado en el servidor del sitio Web, hay que utilizar el atributo value en cada etiqueta <option> de la lista o menú. El texto que se debe escribir entre la etiqueta <option> de apertura y cierre tiene funciones visuales para que los usuarios puedan leer cada opción del menú o lista que se diseña, sin embargo el verdadero valor enviado al servidor es el definido en el atributo value.

El nombre de la variable que se utiliza en el programa de procesamiento ubicado en el servidor y donde se almacenarán los valores especificados en cada atributo value de las etiquetas <option> debe definirse en el atributo name de la etiqueta <select>. En aquella variable serán añadidos el o los valores seleccionados por los usuarios.

Veamos un ejemplo simple:

```
<html>
<head>
<title>Ejemplo de formulario</title>
</head>

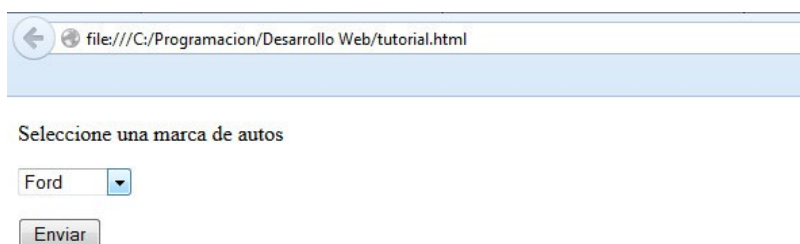
<body>
<form action="www.ejemplo.com/formulario-de-contacto.php" method="POST">
<p>Seleccione una marca de autos</p>
<select name="marcas">
<option value="Ford">Ford</option>
<option value="Chevrolet">Chevrolet</option>
<option value="Chrysler">Chrysler</option>
<option value="Buick">Buick</option>
<option value="Cadillac">Cadillac</option>
</select>

<p><input type="submit" value="Enviar" /> </p>

</form>

</body>
</html>
```

El código fuente anterior muestra lo siguiente en navegadores Web:



The screenshot shows a web browser window with the address bar displaying 'file:///C:/Programacion/Desarrollo Web/tutorial.html'. Below the address bar, the text 'Seleccione una marca de autos' is displayed. Underneath this text is a dropdown menu with 'Ford' selected. To the right of the dropdown menu is a button labeled 'Enviar'.

[Cliquear para ver en grande](#)

En el ejemplo anterior agregamos un menú desplegable; para definir una lista de opciones hay que agregar el atributo size (tamaño) en la etiqueta <select> donde se debe especificar la cantidad de renglones que serán visualizados entre la cantidad total de opciones que la lista contiene; por ejemplo si la lista contiene 20 opciones y se especifica un tamaño de 5 renglones, para ver todas las opciones la lista tendrá una barra de desplazamiento vertical ubicada a la derecha.

Para definir una lista de opciones en la que se pueden seleccionar más de una opción se debe utilizar el atributo multiple en la etiqueta <select> y especificar en el mismo el valor "multiple".

Veamos un ejemplo que contiene un menú y una lista de opciones:

```

<html>
<head>
<title>Ejemplo de formulario</title>
</head>

<body>
<form action="www.ejemplo.com/formulario-de-contacto.php" method="POST">
<p>Seleccione una marca de autos</p>
<select name="marcas-menu">
<option value="Ford">Ford</option>
<option value="Chevrolet">Chevrolet</option>
<option value="Chrysler">Chrysler</option>
<option value="Buick">Buick</option>
<option value="Cadillac">Cadillac</option>
</select>

<br /><br />

<select size="4" name="marcas-lista">
<option value="Ford">Ford</option>
<option value="Chevrolet">Chevrolet</option>
<option value="Chrysler">Chrysler</option>
<option value="Buick">Buick</option>
<option value="Cadillac">Cadillac</option>
</select>

<br /><br />

<select name="marcas-lista-multiple" multiple="multiple" size="5">
<option value="Ford">Ford</option>
<option value="Chevrolet">Chevrolet</option>
<option value="Chrysler">Chrysler</option>
<option value="Buick">Buick</option>
<option value="Cadillac">Cadillac</option>
</select>

<p><input type="submit" value="Enviar" /> </p>

</form>

</body>
</html>

```

El código fuente anterior muestra lo siguiente en los navegadores Web:

file:///C:/Programacion/Desarrollo Web/tutorial.html

Seleccione una marca de autos

Ford

Ford
Chevrolet
Chrysler
Buick

Ford
Chevrolet
Chrysler
Buick
Cadillac

Enviar

[Cliquear para ver en grande](#)

Se puede tener una opción preseleccionada de menú al cargar la página Web con el formulario o variar opciones preseleccionadas en el caso de listas que permiten elegir selecciones múltiples. Esto se logra mediante el uso del atributo `selected` (seleccionado) en la etiqueta `<option>` y en el cual debe especificarse el valor "selected".

Otra posibilidad es la de tener opciones de menú o lista deshabilitadas al cargar la página Web; también se pueden tener menús o listas deshabilitadas por completo; esto se logra mediante el atributo `disabled` (deshabilitado), en el cual se debe especificar el valor "disabled". El mismo puede ser utilizado tanto en la etiqueta `<select>` como en `<option>`.

Veamos un ejemplo que utilizan estos atributos:

```
<html>
<head>
<title>Ejemplo de formulario</title>
</head>

<body>
<form action="www.ejemplo.com/formulario-de-contacto.php" method="POST">
<p>Seleccione una marca de autos</p>
<select name="marcas-menu">
<option value="Ford">Ford</option>
<option value="Chevrolet">Chevrolet</option>
<option value="Chrysler">Chrysler</option>
<option value="Buick" disabled="disabled">Buick</option>
<option value="Cadillac" selected="selected">Cadillac</option>
</select>

<br /><br />

<select size="4" name="marcas-lista">
<option value="Ford">Ford</option>
<option value="Chevrolet">Chevrolet</option>
<option value="Chrysler">Chrysler</option>
<option value="Buick" disabled="disabled">Buick</option>
<option value="Cadillac" selected="selected">Cadillac</option>
</select>

<br /><br />

<select name="marcas-lista-multiple" size="5" disabled="disabled">
```

```

<option value="Ford">Ford</option>
<option value="Chevrolet">Chevrolet</option>
<option value="Chrysler">Chrysler</option>
<option value="Buick">Buick</option>
<option value="Cadillac">Cadillac</option>
</select>

<p><input type="submit" value="Enviar" /> <input type="reset" value="Restablecer" /></p>

</form>

</body>
</html>

```

Esto se vería de la siguiente manera en los navegadores Web:

file:///C:/Programacion/Desarrollo Web/tutorial.html

Seleccione una marca de autos

Ford

Ford
Chevrolet
Chrysler
Buick

Ford
Chevrolet
Chrysler
Buick
Cadillac

Enviar Restablecer

[Clickear para ver en grande](#)

Dentro de las listas o menús se pueden agregar títulos para separar las opciones en grupos, la etiqueta que permite hacer eso es <optgroup>; cada grupo de opciones debe estar encerrado entre la etiqueta <optgroup> de apertura y de cierre. El título que representa cada grupo de la lista o menú se define dentro del atributo label de esta etiqueta.

Veamos un ejemplo:

```

<html>
<head>
<title>Ejemplo de formulario</title>
</head>

<body>
<form action="www.ejemplo.com/formulario-de-contacto.php" method="POST">
<p>Seleccione una marca de autos</p>
<select name="marcas-menu">
<optgroup label="Marcas de General Motors">
<option value="Buick">Buick</option>
<option value="Chevrolet">Chevrolet</option>
</optgroup>
<optgroup label="Otra marca">
<option value="Ford">Ford</option>
</optgroup>
</select>

```

```

<br /><br />
<select name="marcas-lista" multiple="multiple" size="5">
<optgroup label="Marcas de General Motors">
<option value="Buick">Buick</option>
<option value="Chevrolet">Chevrolet</option>
</optgroup>
<optgroup label="Otra marca">
<option value="Ford">Ford</option>
</optgroup>
</select>

<p><input type="submit" value="Enviar" /> <input type="reset" value="Restablecer" /></p>

</form>

</body>
</html>

```

El código fuente anterior muestra lo siguiente en los navegadores Web:

file:///C:/Programacion/Desarrollo Web/tutorial.html

Seleccione una marca de autos

Ford

Marcas de General Motors

- Buick
- Chevrolet
- Otra marca**
- Ford

Enviar Restablecer

[Clicar para ver en grande](#)

Hay una etiqueta que no agrega ningún cambio visual, pero que facilita el uso de formularios con el mouse al seleccionar un elemento de formulario. Se trata de la etiqueta `<label>` la cual le confiere funcionalidad de selección a un texto o rótulo. Esto significa que si por ejemplo tenemos un texto junto a un campo de texto o a una casilla de selección o botón de opción, si aquel texto se encuentra entre etiquetas `<label>` de apertura y cierre y se lo relaciona con un elemento de formulario; al tocarse con el botón del mouse el texto, el elemento de formulario relacionado con el mismo se activará; en el caso de un campo de texto el cursor del mismo se activa, en el caso de una casilla ésta se selecciona.

Para que un texto se relacione con un elemento de formulario, el elemento debe tener un nombre de identificación el cual se define mediante el atributo `id` y la etiqueta `<label>` debe tener un atributo `for` (para) en el cual se debe colocar el mismo nombre especificado en el atributo `id` del elemento de formulario, así al tocar el texto se activará dicho elemento.

Veamos un ejemplo:

```

<html>
<head>
<title>Ejemplo de formulario</title>
</head>

<body>
<form action="www.ejemplo.com/formulario-de-contacto.php" method="POST">

```

```

<p><label for="nombre">Nombre</label><br /><input type="text" name="nombre"
id="nombre" /></p>

<p><label for="tel">Teléfono</label><br /><input type="text" name="telefono" id="tel"
/></p>

<p><label for="mail">Email</label><br /><input type="text" name="email" id="mail"
/></p>

<p><label for="direc">Dirección</label><br /><input type="text" name="direccion"
id="direc" /></p>

<p><u>Nivel de estudios alcanzado</u><br />
<input type="radio" name="estudios" value="Primario" id="prima" /><label
for="prima">Primario</label><br />
<input type="radio" name="estudios" value="Secundario" id="secun" /><label
for="secun">Secundario</label><br />
<input type="radio" name="estudios" value="Terciario" id="terci" /><label
for="terci">Terciario</label><br />
<input type="radio" name="estudios" value="Universitario" id="univ" /><label
for="univ">Universitario</label><br />
<input type="radio" name="estudios" value="Postgrado" id="post" /><label
for="post">Postgrado</label><br /></p>

<p><u>Géneros de película favoritos</u><br />
<input type="checkbox" name="peliculas" value="Accion" id="acci" /><label
for="acci">Acción</label><br />
<input type="checkbox" name="peliculas" value="Ciencia Ficción" id="cienc" /><label
for="cienc">Ciencia Ficción</label><br />
<input type="checkbox" name="peliculas" value="Comedia" id="come" /><label
for="come">Comedia</label><br />
<input type="checkbox" name="peliculas" value="Intriga" id="intri" /><label
for="intri">Intriga</label><br />
<input type="checkbox" name="peliculas" value="Terror" id="terr" /><label
for="terr">Terror</label><br /></p>

<p><input type="submit" value="Enviar" /> <input type="reset" value="Restablecer" /></p>

</form>

</body>
</html>

```

Esto muestra lo siguiente en navegadores Web:

A screenshot of a web browser window showing a local file named 'tutorial.html'. The form contains the following elements:

- Text input fields for 'Nombre', 'Teléfono', 'Email', and 'Dirección'.
- A radio button group for 'Nivel de estudios alcanzado' with options: Primario, Secundario, Terciario, Universitario, and Postgrado.
- A checkbox group for 'Géneros de película favoritos' with options: Acción, Ciencia Ficción, Comedia, Intriga, and Terror.
- 'Enviar' and 'Restablecer' buttons at the bottom.

[Clicquear para ver en grande](#)

Hay otro elemento muy utilizado en formularios, el área de texto, `<textarea>`; el cual sirve para realizar comentarios largos, temas de foros de discusión, escribir texto para formularios de contacto, etc. A diferencia de los campos de texto, las áreas de texto permiten ingresar más de un renglón formando párrafos. También se pueden especificar las dimensiones de un área de texto tanto en cantidad de renglones como en cantidad de caracteres visibles de ancho.

A diferencia de la etiqueta `<input>`, la etiqueta `<textarea>` utiliza etiquetas de apertura y de cierre. Para definir las dimensiones de un área de texto se utiliza el atributo `cols` (columnas) para el ancho en cantidad de caracteres y el atributo `rows` (filas o renglones) para el alto en cantidad de renglones.

Al igual que en otros elementos de formulario el atributo `name` sirve para definir el nombre de variable que utilizará el programa de procesamiento en el servidor y en el que se almacenará el texto introducido en el área de texto.

También como se vio anteriormente con otros elementos, se pueden diseñar áreas de texto deshabilitadas, con el atributo `disabled` (deshabilitado), donde se pone el valor "disabled" como sucede con `<input>`, `<select>` y `<option>`.

A veces se requiere la utilización de áreas de texto de sólo lectura en las cuales se puede leer el texto ya escrito que aparece al cargar la página pero no se puede ingresar ninguno nuevo. Por ejemplo se los utiliza en formularios de inscripción de sitios Web que incluyen una especie de contrato con normas o términos de uso y políticas de privacidad. Para definir un área de texto de sólo lectura se debe utilizar el atributo `readonly` (sólo lectura) con el valor "readonly".

Veamos un ejemplo con áreas de texto:

```
<html>
<head>
<title>Ejemplo de formulario</title>
</head>

<body>
<form action="www.ejemplo.com/formulario-de-contacto.php" method="POST">
<p>Comentario</p>
<textarea cols="40" rows="10" name="comentario"> </textarea>
<br /> <br />

<p>Comentario 1</p>
<textarea cols="40" rows="10" name="comentario-1">Escribe un comentario</textarea>
```



```
<br /><br />
```

```
<p>Area de texto deshabilitada</p>  
<textarea cols="40" rows="10" name="comentario-2" disabled="disabled"></textarea>  
<br /><br />
```

```
<p>Area de texto de solamente lectura</p>  
<textarea cols="40" rows="10" name="comentario-3" readonly="readonly">Esta es un area de  
texto de solamente lectura</textarea>  
<br /><br />
```

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

```
</form>
```

```
</body>
```

```
</html>
```

Lo anterior muestra lo siguiente en los navegadores Web:

file:///C:/Programacion/Desarrollo Web/tutorial.html

Veamos un ejemplo con áreas de texto:

Comentario

Comentario 1

Escribe un comentario

Area de texto deshabilitada

Area de texto de solamente lectura

Esta es un area de texto de solamente lectura

Enviar Restablecer

[Cliquear para ver en grande](#)

Como se ve, se puede agregar texto previamente ingresado en el código fuente, el cual será cargado en el área de texto al abrir la página; el mismo debe ser escrito entre las etiquetas <textarea> de apertura y de cierre.

RECUADRO DE FORMULARIO

Ya sabemos casi todo acerca del diseño de formularios en HTML; solamente nos falta conocer una etiqueta de carácter más bien estético; su función es simplemente la de agregar un recuadro alrededor de un formulario. Se trata de la etiqueta <fieldset>, cuya etiqueta de apertura debe estar ubicada luego de la etiqueta <form> y antes de todos los elementos que forman parte del formulario, mientras que la etiqueta de cierre </fieldset> debe estar colocada inmediatamente antes de la etiqueta de cierre </form>.

Además, opcionalmente, se le puede agregar un título al recuadro de formulario; para eso junto a la etiqueta <fieldset> se debe utilizar la etiqueta <legend> la cual debe ir ubicada inmediatamente después de la etiqueta <fieldset> de apertura.

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de formulario</title>
</head>

<body>
<form action="www.ejemplo.com/formulario-de-contacto.php" method="POST">
<fieldset>
<legend>Formulario de contacto</legend>
<p><label for="nombre">Nombre</label><br /><input type="text" name="nombre"
id="nombre" /></p>

<p><label for="tel">Teléfono</label><br /><input type="text" name="telefono" id="tel"
/></p>

<p><label for="mail">Email</label><br /><input type="text" name="email" id="mail"
/></p>

<p><label for="direc">Dirección</label><br /><input type="text" name="direccion"
id="direc" /></p>

<p><u>Nivel de estudios alcanzado</u><br />
<input type="radio" name="estudios" value="Primario" id="prima" /><label
for="prima">Primario</label><br />
<input type="radio" name="estudios" value="Secundario" id="secun" /><label
for="secun">Secundario</label><br />
<input type="radio" name="estudios" value="Terciario" id="terci" /><label
for="terci">Terciario</label><br />
<input type="radio" name="estudios" value="Universitario" id="univ" /><label
for="univ">Universitario</label><br />
<input type="radio" name="estudios" value="Postgrado" id="post" /><label
for="post">Postgrado</label><br /></p>

<p><u>Géneros de película favoritos</u><br />
<input type="checkbox" name="peliculas" value="Accion" id="acci" /><label
for="acci">Acción</label><br />
<input type="checkbox" name="peliculas" value="Ciencia Ficcion" id="cienc" /><label
for="cienc">Ciencia Ficción</label><br />
```

```

<input type="checkbox" name="peliculas" value="Comedia" id="come" /><label
for="come">Comedia</label><br />
<input type="checkbox" name="peliculas" value="Intriga" id="intri" /><label
for="intri">Intriga</label><br />
<input type="checkbox" name="peliculas" value="Terror" id="terr" /><label
for="terr">Terror</label><br /></p>

<p>Comentario</p>
<textarea cols="40" rows="10" name="comentario"></textarea>
<br /><br />

<p><input type="submit" value="Enviar" /> <input type="reset" value="Restablecer" /></p>
</fieldset>
</form>

</body>
</html>

```

El código fuente anterior mostraría lo siguiente en los navegadores Web:

Múltiples hipervínculos en una imagen

En la primera parte del tutorial de HTML aprendimos a cómo agregar hipervínculos a imágenes, o en otras palabras como utilizar imágenes como hipervínculos, simplemente rodeando la etiqueta `` con las etiquetas de hipervínculo de apertura y cierre. Por ejemplo:

```
<a href="http://www.youbioit.com"></a>
```

Pero HTML nos da la posibilidad de agregar múltiples hipervínculos en una sola imagen; simplemente definiendo áreas de la imagen para distintos links. Supongamos que tenemos una imagen con cuatro rectángulos, y queremos que cada rectángulo de la imagen al ser presionado con el botón izquierdo del mouse nos envíe a una página distinta; lo único que deberíamos hacer para que esto ocurra es agregar áreas para cada rectángulo indicando las coordenadas superior izquierda e inferior derecha de cada uno e indicar la dirección URL a la que nos dirigirá dicho hipervínculo al ser presionado.

Las etiquetas que nos permiten definir áreas de hipervínculos para imágenes son `<map>` y `<area>`. La etiqueta `<map>` se utiliza para indicar cuál es la imagen que será utilizada para hipervínculos múltiples, mientras que `<area>` sirve para definir cada área y agregarle un hipervínculo a cada una.

Para utilizar una imagen para hipervínculos múltiples hay que agregar en la imagen a usar el atributo usemap, dentro del cual agregamos cualquier nombre que querramos, precedido por un signo numeral; de la siguiente forma:

```

```

Luego colocamos debajo de la imagen la etiqueta <map>, con un atributo name (nombre), donde agregamos el mismo nombre que declaramos en el atributo usemap de la imagen. De esta manera las áreas a definir quedarán vinculadas a la imagen en cuestión; siguiendo el ejemplo anterior sería:

```
<map name="hipervinculosmultiples">
```

Entre las etiquetas de apertura y cierre <map> se deben colocar las etiquetas <area> que representan cada una de las áreas a utilizar como hipervínculos.

Las áreas pueden tener distintas formas; rectangulares, circulares y en forma de polígono. Para indicar la forma que utilizaremos en un área debemos indicar en el atributo shape (forma) de la etiqueta <area>, el tipo de forma que queremos que tenga. Los posibles valores de shape son "rect" (rectangular), "circle" (circular) o "poly" (poligonal).

Una vez elegida la forma, debemos indicar las coordenadas de los vértices de cada área; así en el caso de la forma rectangular debemos poner los valores de las coordenadas superior izquierda e inferior derecha del rectángulo que define el área. En el caso del área circular se deben indicar las coordenadas del centro del círculo que definirá el área y la longitud del radio del mismo. Finalmente para el área poligonal se deben agregar las coordenadas de cada uno de los vértices que definen el polígono; si por ejemplo se tiene un pentágono se deben agregar las coordenadas de cada uno de los cinco vértices.

Los valores de las coordenadas se deben agregar en el atributo coords de la etiqueta <area> y se miden en píxeles, así por ejemplo si tenemos una imagen de 500x500 píxeles y en la misma hay un rectángulo cuyo vértice superior izquierdo se encuentra en el píxel con coordenadas (x,y) 50,50 y el vértice inferior derecho se encuentra en el píxel con coordenadas 200,200; en coords debemos agregar los valores 50,50,200,200.

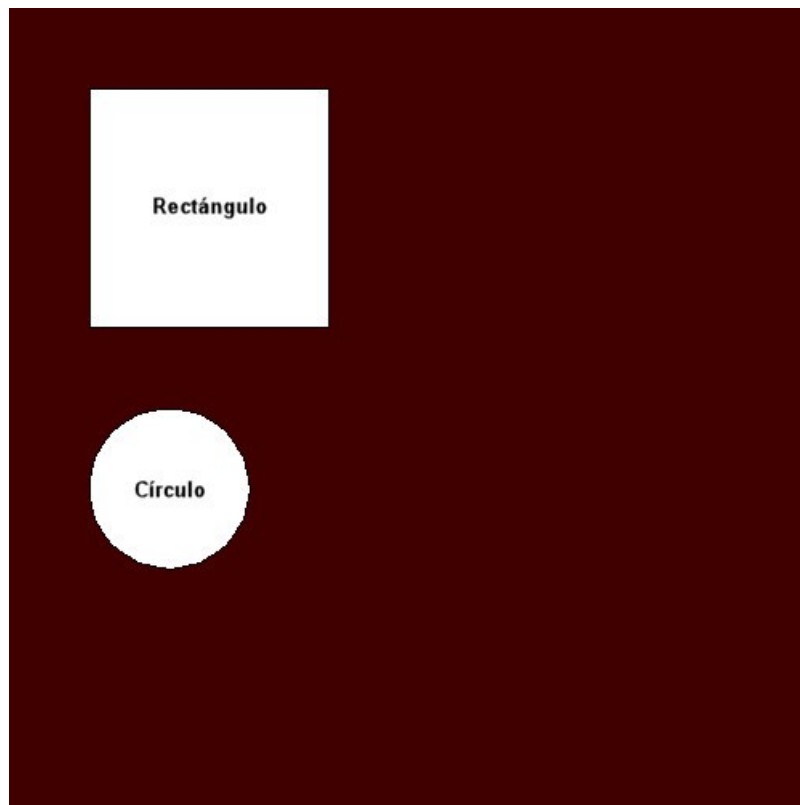
Si queremos definir un área circular debemos indicar las coordenadas (x,y) del centro del círculo y la longitud del radio en píxeles. Entonces si por ejemplo tenemos un círculo con centro en el píxel con coordenadas 100,300 y el mismo tiene un radio de 50 píxeles; en coords debemos agregar los valores 100,300,50,50.

Para definir el hipervínculo de cada área de la imagen hay que agregar el atributo href en cada etiqueta <area>, con la URL a la que nos enviará el hipervínculo.

También al igual que en la etiqueta <a> podemos utilizar el atributo target en <area> para hacer que los hipervínculos se abran en pestañas nuevas.

También se puede incluir un atributo alt en cada etiqueta <area> con un nombre cualquiera, el cual aparecerá en un rótulo o leyenda cada vez que posemos el puntero sobre dicha área. Es bueno aclarar que esto no funciona en todos los exploradores, ya que en algunos no dará resultados; por ejemplo uno de ellos es Firefox; sin embargo en Internet Explorer sí funciona.

Veamos un ejemplo. (Para probar dicho código simplemente debemos guardar en la carpeta que utilizamos para almacenar las páginas creadas durante el tutorial la siguiente imagen). En el ejemplo la imagen está en la misma carpeta del archivo .html.



A continuación sigue el código:

```
<html>
<head>
<title>Ejemplo de imagen con hipervinculos multiples</title>
</head>

<body>



<map name="multi">
<area shape="rect" coords="50,50,200,200" alt="Rectangulo" target="_blank"
href="http://www.youbioit.com" />
<area shape="circle" coords="100,300,50,50"
alt="Circulo" target="_blank" href="http://www.youtube.com" />
</map>

</body>
</html>
```

Caracteres especiales HTML

Cuando escribimos algún texto en HTML existen ciertos caracteres especiales que por distintos motivos no pueden ser representados simplemente como una letra o número común. Por ejemplo si queremos representar un símbolo > no podemos agregarlo simplemente en el texto y listo; ya que eso causaría un conflicto con la etiqueta precedente; rompiendo la estructura del código fuente y derivando en resultados indeseados en el diseño de la página Web en la que estamos trabajando.

Veamos un ejemplo de lo que no hay que hacer:

<p>La etiqueta sirve para agregar imágenes en la página HTML.</p>

El ejemplo anterior dará resultados indeseados, ya que nosotros queremos que aparezca la etiqueta en el texto, pero el explorador se confunde creyendo que queremos mostrar una imagen; dando como resultado un cuadrado vacío de imagen no cargada.

Para solucionar esto se deben utilizar códigos de escape, también conocidos como secuencias de escape, los cuales nos permiten agregar símbolos especiales mediante el uso de códigos especiales. Cada símbolo tiene su propio código; el cual puede ser el código ASCII que le corresponde o un código especial de HTML; cualquiera de los dos da el mismo resultado esperado.

Los códigos especiales HTML deben ir siempre precedidos por un símbolo & y sucedidos por un ; . Así, siguiendo con el ejemplo anterior:

Para representar el símbolo < debemos escribir: <

Para representar el símbolo > debemos escribir: >

Si en lugar de utilizar los códigos HTML se prefieren los códigos ASCII; podemos hacerlo agregando el número correspondiente a cada símbolo. El mismo debe estar precedido por la combinación de símbolos &# y sucedido por ; (punto y coma).

Volviendo al ejemplo anterior; para obtener los resultados esperados del código fuente en el que queríamos representar debemos poner:

<p>La etiqueta sirve para agregar imágenes en la página HTML.</p>

Las secuencias o códigos de escape también son muy importantes para asegurar la compatibilidad del texto en todas las computadoras. Cada idioma tiene su juego de caracteres y códigos; así mientras que el inglés no tiene la letra ñ o no usa acentos, el idioma castellano sí. Por lo tanto si una computadora tiene su sistema operativo en un idioma distinto al que está escrito la página; puede que algunos símbolos, caracteres o signos de acentuación u ortografía en general no se visualicen correctamente o sean reemplazados por otros signos que en aquel idioma si existan, perdiendo el texto así su claridad.

Por eso cada vez que se escriba una palabra con acento, o tilde (como la ñ); estas letras no deben ser escritas directamente como van a ser representadas pero con el código o secuencia de escape que les corresponde. Por ejemplo.

Para representar á: ´

Para representar é: é

Para representar í: í

Para representar ó: ó

Para representar ú: ú

Para representar Á: Á

Para representar É: É

Para representar Í: Í

Para representar Ó: Ó

Para representar Ú: Ú

Para representar ñ: ñ

Para representar Ñ: Ñ

Para representar ä: ä

Para representar ë: ë

Para representar ï: ï

Para representar ö: ö

Para representar ü: ü

Para representar Ä: Ä

Para representar Ê: Ë

Para representar Ï: Ï

Para representar Ö: Ö

Para representar Ü: Ü

Al final del tutorial de HTML, debajo de todo, hay una lista con muchos más códigos o secuencias de escape para símbolos.

Marcos HTML

Un elemento de diseño Web que estaba muy de moda en los primeros años del siglo 21 eran los marcos de página (frames en inglés); pero a partir de mediados de la década comenzaron a caer en desuso. Con los marcos se puede dividir una página Web en varias subpáginas separadas por marcos. Por ejemplo se pueden tener dos columnas, cada una mostrando una página; o se puede dividir la página en dos franjas horizontales cada una mostrando el contenido de un documento .html o .htm distinto. Cuando estaban de moda, por lo general se solían dividir las páginas en cuatro marcos; uno superior horizontal ocupando todo el ancho de la pantalla donde se solía colocar un logo o título; uno vertical en la izquierda para la botonera; un marco central que mostraba una subpágina con el contenido del documento y un marco horizontal inferior.

Uno de los inconvenientes de los marcos, es que se debe crear una página para cada marco más la página que contiene a los marcos. Así por ejemplo si se tiene una página con 4 marcos se deben crear 5 páginas Web, y unir las en una principal.

El conjunto de etiquetas que se utilizan para armar una página con marcos son <frameset> (conjunto de marcos) y <frame> (marco).

Supongamos que queremos dividir una página en 4 marcos horizontales. Lo primero que debemos hacer es crear las cuatro páginas que ocuparán los marcos. Luego tenemos que crear la quinta página que contendrá los distintos marcos con las páginas.

```
<html>
<head>
<title>Ejemplo de Marcos</title>
</head>

<frameset rows="25%, 25%, 25%, 25%">

<frame src="pagina1.html" />
<frame src="pagina2.html" />
<frame src="pagina3.html" />
<frame src="pagina4.html" />

</frameset>
</html>
```

El código fuente anterior daría un resultado como el siguiente:



Como se puede ver en el código fuente anterior, al crear una página con marcos no se utiliza `<body>`, ya que el contenido de la página está en el conjunto de marcos incluidos, por lo tanto en su lugar se utiliza la etiqueta `<frameset>`, entre cuyas etiquetas de apertura y cierre se colocan cada una de las etiquetas `<frame>` que representan los marcos con subpáginas.

En el atributo `rows` de `<frameset>` se define la cantidad de marcos horizontales que tendrá la página, especificando el tamaño de cada uno. Los valores pueden ser en píxeles o porcentaje del alto de la página. En los atributos `src` de cada `<frame>` se indican las ubicaciones de las páginas que ocuparán cada marco del conjunto. Los nombres de las páginas `.html` del ejemplo pueden ser reemplazados por los nombres de las páginas Web que queramos o incluso por URLs de páginas de otros sitios.

También se pueden crear páginas con marcos verticales, los cuales se dividen en columnas cuyas dimensiones se indican en el atributo `cols`.

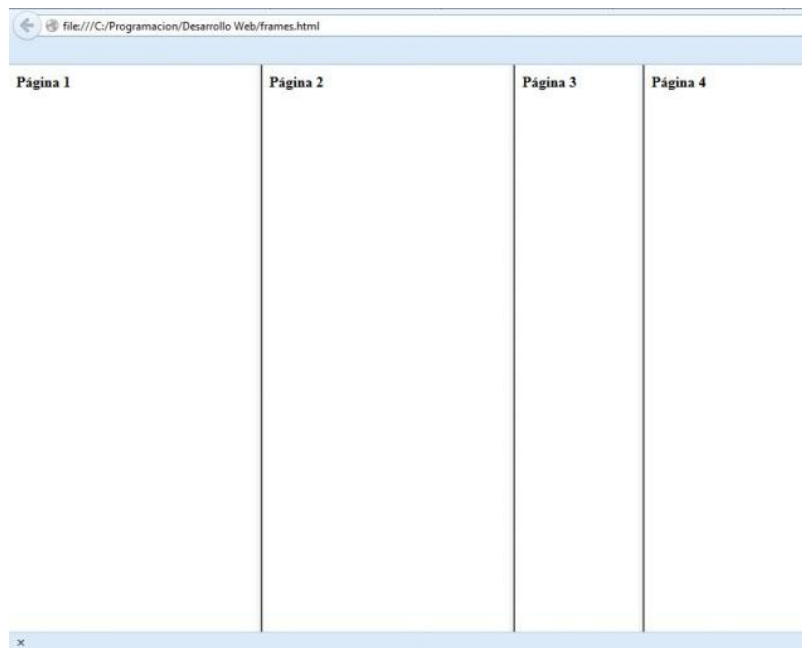
```
<html>
<head>
<title>Ejemplo de Marcos</title>
</head>

<frameset cols="20%, 20%, 10%, *">

<frame src="pagina1.html" />
<frame src="pagina2.html" />
<frame src="pagina3.html" />
<frame src="pagina4.html" />

</frameset>
</html>
```

El código fuente anterior mostraría lo siguiente en los navegadores Web:



[Clicar para ver en grande](#)

En el ejemplo anterior dividimos la página en cuatro marcos verticales. Un detalle que llama la atención es el asterisco * de la cuarta posición del atributo cols de `<frameset>`. El asterisco se utiliza cuando no sabemos el tamaño que resta en una página y representa "lo sobrante de la página". En el ejemplo anterior definimos una primera columna a la izquierda que ocupa el 20% del ancho de la página, otra del 20%, otra del 10% y finalmente a la derecha una que ocupa el espacio sobrante, que en este ejemplo equivale al 50% del ancho de la página. Por lo general el asterisco se utiliza cuando trabajamos en píxeles, ya que podemos definir tamaños específicos para algunas columnas pero desconocemos el espacio sobrante que habrá en las pantallas de los distintos usuarios. Supongamos que creamos una página con dos columnas, una izquierda para la botonera con 400 píxeles de ancho y una segunda que ocupa el espacio sobrante, el cual variará de pantalla en pantalla según la resolución horizontal de cada una. El asterisco se puede utilizar para cualquiera de las columnas.

Como se puede ver en los ejemplos anteriores, la posición de cada `<frame>` coincide con la posición de cada valor de los atributos cols y rows. Así el primer `<frame>` tendrá el alto o el ancho indicado en el primer valor de rows o cols respectivamente; y así ocurre con cada uno de los marcos restantes. Por ejemplo una modificación en el cuarto valor de cols, modificará al cuarto `<frame>`.

Si se quiere quitar el borde de los marcos hay que agregar el atributo `frameborder` en cada uno de los `<frame>`, indicando un valor "0" que representa "sin borde". Si se quieren bordes en los marcos se debe poner un valor de "1" en cada `<frame>`. El valor por omisión de `frameborder` es "1"; ya que por default los marcos tienen bordes, entonces si no se indica nada, tendrán bordes.

Por omisión los marcos pueden ser redimensionados con el mouse, sin embargo si no se quiere este efecto y se buscan marcos estáticos se debe agregar en la etiqueta `<frame>` que queremos mantener en su tamaño original, el atributo `noresize` (no redimensionar) cuyo valor también debe ser "noresize". Este atributo puede llegar a ser útil si se quiere hacer un marco conteniendo una botonera de dimensiones fijas o un marco que contiene el logo del sitio; para evitar modificar el aspecto visual del mismo.

Otro atributo de `<frame>` que puede llegar a ser útil en el diseño de una página con marcos, es `scrolling`, mediante el cual se puede especificar si se quiere incluir una barra de scroll o deslizamiento en dicho marco. Los valores posibles de este atributo son "yes" (que siempre haya barra de deslizamiento), "no" (que nunca haya barra de deslizamiento) y "auto" (que solamente aparezca la barra de deslizamiento si es necesario); el último es el valor por omisión, que equivale a no usar el atributo.

A cada marco también se le pueden agregar márgenes horizontales y verticales; lo cual se logra con los atributos de `<frame>` `marginheight` (alto de margen) y `marginwidth` (ancho de margen). Los valores de los mismos deben ser especificados en píxeles.

Finalmente el atributo de `<frame>` `name` (nombre), sirve para denominar a cada marco con un nombre que permite que cada uno pueda ser identificado por otros elementos HTML. Dicho atributo es principalmente útil si se quiere diseñar

una página con marco de botonera y marco de contenido. Así cada vez que se pulsa un hipervínculo de la botonera del marco de la izquierda, la página a la cual dicho hipervínculo nos envía, se abrirá en el marco de contenido principal. Para lograr eso, cada etiqueta de hipervínculo `<a>` ubicada en la página que aparecerá en el marco de la izquierda, deberá contener el atributo `target` (objetivo), del que ya hemos hablado en la [primera parte](#) del tutorial cuando aprendimos a crear hipervínculos. En `target` debemos colocar como valor el nombre del marco en el cual queremos que las páginas se abran.

A continuación veremos un ejemplo, de una página de marcos dividida en dos columnas, una izquierda para la botonera de tres hipervínculos y una principal a la derecha para el contenido. Para realizarla se deberá crear una página de botonera, tres de contenido y una que contenga el conjunto de marcos (frameset). Primero se mostrará el código fuente de la página de la botonera, luego las tres páginas de contenido, y finalmente en el último recuadro, el código fuente de la página de marcos. En total tendremos cinco recuadros de código fuente para el siguiente ejemplo de página con marcos.

1-Página de botonera (botonera.html):

```
<html>
<head>
<title>Ejemplo de marcos</title>
</head>

<body>

<p><a href="pagina1.html" target="marcoprincipal">BOTON 1</a></p>
<p><a href="pagina2.html" target="marcoprincipal">BOTON 2</a></p>
<p><a href="pagina3.html" target="marcoprincipal">BOTON 3</a></p>

</body>
</html>
```

2- Página de contenido 1 (pagina1.html):

```
<html>
<head>
<title>Ejemplo de marcos - Pag. 1</title>
</head>

<body>
<p>Sección 1 de la página de marcos</p>

</body>
</html>
```

3- Página de contenido 2 (pagina2.html):

```
<html>
<head>
<title>Ejemplo de marcos - Pag. 2</title>
</head>

<body>
<p>Sección 2 de la página de marcos</p>

</body>
</html>
```

4- Página de contenido 3 (pagina3.html):

```
<html>
<head>
<title>Ejemplo de marcos - Pag. 3</title>
</head>

<body>
<p>Sección 3 de la página de marcos</p>

</body>
</html>
```

5- Página del conjunto de marcos (marcos.html):

```
<html>
<head>
<title>Ejemplo de Marcos</title>
</head>

<frameset cols="200, *">

<frame src="botonera.html" noresize="noresize" scrolling="no" frameborder="0"
marginheight="10" marginwidth="10" />
<frame src="pagina1.html" noresize="noresize" scrolling="auto"
frameborder="0" marginheight="10" marginwidth="10" name="marcoprincipal" />

</frameset>
</html>
```

Resultado del ejemplo anterior:



En este ejemplo hemos creado tres páginas de contenido, una para la botonera con tres hipervínculos y finalmente una para el conjunto de marcos.

En la página del conjunto de marcos (marcos.html); definimos en la etiqueta <frameset> dos columnas, una a la izquierda con 200 pixeles de ancho (para la página de botonera) y otra a la derecha (para las páginas de contenido) con

un ancho que ocupa lo sobrante de la pantalla. En cada <frame> indicamos que queremos bordes de 1 pixel, que los marcos deben ser fijos y que cada uno debe tener un margen a los cuatro lados de 10 pixeles. En cuanto a la barra de deslizamiento se especifica que el marco de la izquierda no tendrá barra, mientras que la de la derecha solamente si es necesario.

En el segundo <frame> agregamos un atributo name para que dicho marco pueda ser identificado por los hipervínculos de la página de la botonera que tienen como marco objetivo (target) para abrir las páginas, el del contenido. Así cada vez que se pulse uno de los botones, la página se abrirá en el marco denominado "marcoprincipal". También hay que aclarar que como en el atributo src del <frame> de contenido se indica la "pagina1.html", cada vez que se abra por primera vez el archivo marcos.html; por omisión será cargado con dicha página 1. Luego al pulsar los otros botones se cargarán las otras.

Ya hemos aprendido a crear marcos de página, sin embargo es muy importante dejar claro que se han vuelto casi obsoletos, al menos para el diseño de sitios Web. Quizá se sigan utilizando en manuales, libros o tutoriales de productos diseñados en formato HTML; pero se desaconseja su utilización para el diseño sitios Web; ya que son complicados de armar; no siempre se visualizan bien en todos los exploradores Web, dando resultados variados y no son muy amigables a los buscadores; lo que significa que les cuesta indexar en su base de datos, sitios Web desarrollados con marcos.

Además resultan estéticamente desagradables para muchos usuarios, ya que dan un aspecto de sitio viejo o desactualizado.

En lugar de los marcos de página se recomienda utilizar los marcos incorporados <iframe> de los cuales hablamos en la [tercera parte](#) del tutorial, ya que resultan más fáciles de diseñar, más amigables a los buscadores y además más atractivos visualmente. Por ejemplo se los suele utilizar para incorporar páginas de otros sitios dentro de la página que diseñamos, sin la necesidad de que los usuarios tuvieran que abrirlas tras pulsar un hipervínculo. Los usos de los marcos incorporados son variados.

Versiones de HTML

Desde la creación de la World Wide Web (Red informática mundial) HTML estuvo presente, ya que se trata del lenguaje en el que los documentos Web están escritos. Al principio las páginas Web solamente podían mostrar texto, pero con el tiempo se fueron agregando nuevos elementos, como las imágenes. Pero ninguna de las primeras versiones llegó a considerarse un estándar.

Luego se creó el consorcio de la World Wide Web, llamado W3C (de WWW Consortium); que se encarga de dar recomendaciones para la estandarización de comandos y elementos de diseño y desarrollo Web. Se supone que los creadores de exploradores Web deben adherir a estos estándares; no incluyendo la reproducción de etiquetas viejas o de versiones anteriores de HTML, ya desaprobadas; y siguiendo normas de diseño Web. La primera versión de HTML que fue organizada por el consorcio W3C fue HTML 3.0. Entre algunas de las nuevas capacidades más destacables de HTML 3.0, se incluían facilidades para la creación de tablas, la posibilidad de hacer fluir el texto alrededor de imágenes, o la capacidad de agregar elementos matemáticos para la representación de fórmulas matemáticas de todo tipo. Sin embargo dadas las complejidades tecnológicas que requerían esos elementos, esta versión no fue apoyada por los desarrolladores de exploradores Web de la época; expirando el estándar al poco tiempo. Más tarde se organizó la versión HTML 3.1 pero no llegó a ser oficializada. Pero la siguiente versión HTML 3.2 fue adoptada; ya que retiraba muchas de las funcionalidades de la versión HTML 3.0, para adaptarse a las tecnologías de la época; una de las capacidades abandonadas era la de representación de fórmulas matemáticas; las cuales pasaron a formar un lenguaje separado llamado MathML. Además incorporaba varios elementos desarrollados por Mosaic y Netscape, que eran los exploradores Web más populares.

Ya para 1997 la W3C creó un nuevo estándar, la versión HTML 4.0; la cual adoptaba elementos usados por los exploradores Web populares, pero también comenzó a limpiar el lenguaje, dando cierto orden y organización al mismo; desaprobando la utilización de algunas etiquetas antiguas o formas de diseño, a favor de otras más prácticas desde el punto de vista de diseño.

En el año 1999 se publicó una revisión de la versión HTML 4.0; denominada HTML 4.01, la cual incluía ciertas modificaciones sobre la anterior.

A pesar de la existencia del estándar, al cual los variados exploradores Web utilizados hoy día adhieren; muchas etiquetas viejas HTML desaprobadas por el W3C siguen funcionando en la mayoría de los exploradores Web; pero esto no significa que en futuras versiones sigan funcionando; por lo tanto por motivos de compatibilidad con futuros

exploradores Web se sugiere respetar las normas del W3C y no utilizar antiguas etiquetas. Es importante aclarar que para un correcto aprendizaje del lenguaje HTML, en este tutorial no se utilizan etiquetas desaprobadas por el W3C y todo lo aprendido cumple con las normas del estándar establecido en las versiones HTML 4.01 y XHTML de la cual hablaremos a continuación.

El lenguaje HTML está basado en el sistema de definición y organización de etiquetas llamado SGML (Standard Generalized Markup Language - en inglés Estándar de Lenguaje de Marcado Generalizado). SGML no es un lenguaje en sí, sino que se trata de un sistema de etiquetado de documentos, o sea que a través de la aplicación de un sistema de etiquetas se puedan definir los distintos elementos que forman parte de un documento. El lenguaje HTML, se creó a partir de la sintaxis de este sistema de estructuración o creación de documentos. Por otro lado, posteriormente se creó una versión más perfeccionada y simplificada de SGML, para la definición o creación de lenguajes de estructuración de documentos llamada XML (EXtensible Markup Language - que en inglés significa Lenguaje de Marcado Extensible). En base a XML se creó XHTML.

El XHTML no es otra cosa que HTML pero basado en XML, siendo por lo tanto casi idénticos. XHTML es soportado por la mayoría de los exploradores Web populares de hoy día. XHTML presenta leves diferencias con el HTML simple; las cuales se describirán aquí.

- Las etiquetas y los atributos en XHTML deben estar escritos en letras minúsculas; de lo contrario no se obtendrán los resultados deseados.
- En HTML a pesar que es un error no cerrar los elementos con etiquetas de cierre, pueden ser reproducidos por muchos exploradores Web. Sin embargo en XHTML es obligatorio que todos los elementos estén correctamente cerrados; incluso los elementos vacíos como `` que no llevan etiquetas de cierre, siendo cerrados al final de los mismos. Algunos ejemplos incluyen `
`, `<hr />` entre tantos otros. Por lo tanto es importante ser cuidadoso y cerrar todos los elementos.
- Todos los valores de los atributos de etiquetas deben estar entre comillas.
- Los elementos en XHTML deben estar correctamente anidados. En HTML simple puede que algunos exploradores Web reproduzcan elementos que fueron incorrectamente anidados como por ejemplo: `<p>Hola</p>`, pero en XHTML esto no funcionaría así que habría que corregirlo anidando bien las etiquetas así: `<p>Hola</p>`.

Como se puede observar en la lista anterior las diferencias entre HTML simple y XHTML son mínimas, tratándose nada más de una versión más estricta en cuanto al respeto de la sintaxis de codificación se trata. De hecho es recomendable adherir a las normas de XHTML, ya que respetar las normas de sintaxis al pie de la letra es una buena costumbre para todo diseñador Web. De hecho todos los ejemplos de este tutorial adhieren a las normas de XHTML; ya que tanto las etiquetas como los atributos están escritos con letras minúsculas, todos los elementos están cerrados así como correctamente anidados.

Entonces surge la pregunta, de cuál es la ventaja de XHTML por sobre el HTML simple. En realidad si se trata de un documento Web que incluye además de los elementos del lenguaje HTML, elementos de otros lenguajes de marcado como MathML (elementos para la representación de fórmulas matemáticas como el símbolo de raíz cuadrada, integrales, etc) o SVG (Scalable Vector Graphics, en inglés Gráficos Vectoriales Escalables; para la representación de valores vectoriales bidimensionales), entre otros; se recomienda trabajar en XHTML; de lo contrario si se trata de una página Web convencional se puede trabajar en la versión HTML 4.01. Sin embargo, dada la ventaja anteriormente mencionada de XHTML, así como la falta de diferencias estructurales en lo que al lenguaje HTML respecta entre HTML 4.01 y XHTML a veces es mejor trabajar en este último. Además en XHTML estamos obligados a escribir el código fuente con la sintaxis correcta.

Actualmente se encuentra en etapa de experimentación la versión de sintaxis HTML 5 y paralelamente XHTML 5; incorporando algunos elementos nuevos como las etiquetas `<audio>` y `<video>`, de las cuales hablaremos más adelante. Sin embargo hay que aclarar que muchos elementos de HTML 5 todavía no pueden ser representados por todos los exploradores Web, incluso en sus versiones más nuevas; haciendo que mientras puedan funcionar en algunos no lo harán en otros. Por eso si se necesita utilizar algún elemento de HTML 5, es recomendable chequear que funcione en varios exploradores Web.

Cómo declarar la versión de HTML del documento Web

Cuando estamos diseñando una página Web es muy importante declarar en qué versión de HTML está codificada para que el explorador Web la pueda reproducir acorde a los elementos que contiene. Hay etiquetas que ya no se utilizan en la versión de HTML 4.01 y que fueron desaprobadas por el W3C; sin embargo cuando se declara la versión en la que está escrita la página Web, se puede especificar que a pesar de todo se permitirán elementos desaprobados por el

consorcio W3C; o también lo contrario, declarando que se trata de un código fuente estricto que no permite reproducir elementos viejos desaprobados.

La declaración de la versión HTML en el que está escrita la página Web, siempre debe estar ubicada en la primera línea o renglón del código fuente antes de <html>. La declaración de la versión de HTML se realiza con la instrucción <!DOCTYPE>. No está de más reiterar que siempre debe ser incluida en el código fuente y que debe estar ubicada en la primera línea o renglón del código fuente.

A continuación se muestran las distintas declaraciones <!DOCTYPE> utilizadas para indicar en qué versión está escrita una página.

Versión HTML 4.01 Transicional: A pesar de estar en versión 4.01, permite la reproducción de etiquetas viejas o desaprobadas.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

Versión HTML 4.01 Estricta: Está escrita en versión HTML 4.01; pero prohíbe estrictamente la reproducción de elementos viejos.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

Versión XHTML 1.0 Transicional: A pesar de estar en versión XHTML 1.0, permite la reproducción de etiquetas viejas o desaprobadas.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Versión XHTML 1.0 Estricta: Está escrita en versión XHTML 1.0; pero prohíbe estrictamente la reproducción de elementos viejos.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Versión HTML 5: En la versión HTML 5 no hay declaraciones en estado transicional que permite el uso de elementos viejos o desaprobados; ya que esta versión es estricta en cuanto al uso de las etiquetas viejas desaprobadas; así que hay sólo una versión.

```
<!DOCTYPE html>
```

Para describir brevemente esta instrucción; DOCTYPE proviene de Document Type (en inglés Tipo de Documento); PUBLIC indica que es de tipo público accesible a todos los usuarios de la Web; W3C que respeta las normas del estándar de dicho consorcio; DTD proviene de Document Type Definition (Definición de Tipo de Documento) luego de lo cual se define la versión de HTML en la que está escrito el documento. La dirección del archivo de formato dtd que figura al final de la instrucción DOCTYPE de cada versión (salvo la de HTML 5); indica la ubicación del archivo donde se definen todos los comandos y elementos que componen una versión de HTML. Para cada versión hay un archivo dtd diferente.

Las declaraciones de versión de HTML siempre se escriben exactamente como se ha indicado en los párrafos anteriores; así, cada vez que se necesiten; acorde a la versión seleccionada; podemos venir a esta página del tutorial y copiar la instrucción de declaración requerida; o también podemos crear un archivo de texto donde podemos copiarlas para tenerlas a mano.

Una cosa más que hay que aclarar es que cuando una página Web está en versión XHTML, hay que agregar un atributo en la etiqueta <html> de nombre xmlns (XML Namespace - Espacio de nombres XML) cuyo valor es la dirección donde se encuentra el espacio o conjunto de nombres de elementos que componen la versión XHTML. La dirección siempre es "<http://www.w3.org/1999/xhtml>". Atención, el atributo xmlns de <html> solamente debe ser utilizado en la versión XHTML.

Veamos un ejemplo de declaración de versión HTML en el código fuente de uno de los ejemplos que utilizamos anteriormente:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Ejemplo de formulario</title>
</head>
```

```

<body>
<form action="www.ejemplo.com/formulario-de-contacto.php" method="POST">
<p>Seleccione una marca de autos</p>
<select name="marcas">
<option value="Ford">Ford</option>
<option value="Chevrolet">Chevrolet</option>
<option value="Chrysler">Chrysler</option>
<option value="Buick">Buick</option>
<option value="Cadillac">Cadillac</option>
</select>

<p><input type="submit" value="Enviar" /> </p>

</form>

</body>
</html>

```

En el ejemplo anterior se declaró un código fuente XHTML 1.0 Transicional, el cual si llega a contener alguna etiqueta vieja ya desaprobadada de versiones anteriores, igualmente podrá ser visualizada por los exploradores Web sin ningún inconveniente.

No está de más reiterar que todos los ejemplos de este tutorial respetan las normas del W3C al pie de la letra y en ningún momento se han enseñado etiquetas viejas de versiones anteriores a HTML 4.01, ya desaprobadadas; todas son etiquetas vigentes.

Tampoco está de más reiterar que es importante siempre declarar la versión del código fuente de la página. Así que a declarar siempre la version!

CSS en HTML

En los primeros años de la World Wide Web (Red Informática Mundial) el lenguaje HTML se utilizaba para definir tanto la parte estructural de las páginas Web (inclusión de distintos elementos) así como la estética (colores, posiciones, diseño y trazado).

Pero con el paso del tiempo y la suma de más elementos que componían las páginas Web, cada vez más complejas y estéticamente ricas, se requirió la creación de un lenguaje aparte, dedicado exclusivamente al diseño estético de las páginas Web, llamado CSS.

El nombre CSS proviene de las siglas Cascading Style Sheets (Hojas de Estilo en Cascada); un lenguaje dedicado exclusivamente a la presentación estética de la página. De esta manera la definición de la parte estructural de las páginas Web, o sea la incorporación de elementos (texto, imágenes, hipervínculos, tablas, formularios, etc) se realiza con HTML, mientras que la definición de la parte estética (colores de texto, posiciones de elementos, distribución o trazado del diseño de la página Web, etc) se realiza con CSS. Una analogía arquitectónica válida sería comparar a HTML con la parte estructural de un edificio (cimientos, vigas, paredes, techos) y a CSS con la parte de diseño del mismo (pintura, empapelado, decoración interior y exterior). Así, el trabajo combinado de HTML y CSS da como resultado páginas visualmente ricas.

Antes, las páginas Web no tenían tanta riqueza visual como ahora; y toda la parte estética del diseño dependía de HTML. Claro que la variedad de elementos estéticos de las páginas Web en aquellos tiempos era inferior a los que se pueden encontrar hoy día.

Por aquel entonces se podía cambiar el color, el tamaño y el tipo de fuente del texto; cambiar el color de fondo de la página así como incluir una imagen de fondo; alinear los elementos de la página (texto, imágenes, tablas, etc) a la izquierda, centro o derecha; y para distribuir los elementos en posiciones específicas se utilizaban tablas con bordes invisibles de tamaño "0".

La mayoría de las etiquetas y atributos HTML que se utilizaban para mejorar estéticamente las páginas, quedaron desaprobados (deprecated en inglés) en la versión HTML 4.01; sin embargo hasta el día de hoy la mayoría de ellos todavía pueden ser visualizados en los exploradores Web si se especifica en el `<!Doctype>` del código fuente una versión de HTML transicional (no estricta). A pesar que funcionen no se recomienda incluir estos elementos y atributos; ya que mientras algunos pueden funcionar en ciertos exploradores Web puede que no lo hicieran en otros; sin mencionar que es probable que en un futuro todos los browsers dejen de visualizarlos, convirtiendo a la página en incompatible con exploradores Web futuros. Además es importante destacar que los beneficios visuales ofrecidos por estas etiquetas y atributos de HTML son limitados en comparación con la capacidad de diseño estético con que cuenta CSS, el cual brinda un sin fin de combinaciones visuales que permiten el desarrollo de páginas Web mucho más bonitas que las de antes.

Algunas de las etiquetas desaprobadas:

`<center>`: Centra todo tipo de elementos como texto, imágenes, tablas, etc.

``: Utilizada para indicar el color, tamaño y tipo de fuente de un texto.

`<basefont>`: Servía para indicar el color, tamaño y tipo de fuente de todo el texto de la página. Esta etiqueta se colocaba fuera de `<body>` en `<head>`.

Algunos atributos ya desaprobados por el W3C:

Atributos de `<body>`:

`link`: Define el color de los hipervínculos de la página.

`vlink`: Define el color de los hipervínculos ya visitados.

`alink`: Define el color de los hipervínculos activos.

`bgcolor`: Define el color de fondo de la página.

`background`: Define una imagen de fondo para la página.

Atributos de `<p>`:

`align`: Indica el alineamiento de un texto en la página.

Atributos de `<table>` y `<td>`:

`bgcolor`: Color de fondo de una tabla o celda.

Hoy día en lugar de utilizar etiquetas y atributos HTML para agregar belleza estética a las páginas se utiliza CSS; el cual facilita mucho el diseño ya que cuenta con una gran variedad de aditivos estilísticos específicos para cada elemento HTML.

Los comandos o elementos de estilo CSS pueden incorporarse dentro del código fuente HTML de la página, así como también pueden estar en un archivo separado de formato .css el cual se vincula o importa en la página mediante la etiqueta `<link>` (vincular), la cual debe ir ubicada en la sección de cabecera del código fuente, fuera de `<body>`, entre las etiquetas de apertura y cierre `<head>`.

Si en lugar de crear un archivo separado de tipo .css se prefiere incorporar el código CSS dentro del código HTML, se debe utilizar la etiqueta `<style>` (estilo), entre cuyas etiquetas de apertura y cierre se coloca el código en lenguaje CSS. `<style>` al igual que `<link>` debe colocarse fuera de `<body>`, en la sección de cabecera entre las etiquetas de apertura y cierre de `<head>`.

Una de las ventajas de utilizar archivos CSS separados del código fuente de las páginas, es que si se quiere un diseño homogéneo y general para todas las páginas que componen un sitio Web; el mismo puede ser especificado en dicho archivo CSS; donde el estilo visual de cada elemento HTML de las páginas es definido. Por ejemplo, si se quiere que todas las páginas del sitio tengan un cierto color de fondo, en lugar de especificarlo en cada una de ellas; simplemente hay que hacerlo una vez en el archivo CSS, ya que los comandos estilísticos de aquel archivo se aplican a todas las páginas que componen el sitio Web. Si luego se quiere cambiar el color de fondo de cada página, en lugar de tener que modificar el código fuente de cada página, simplemente se modifica el color de fondo desde el archivo CSS; ahorrando así tiempo y trabajo.

Si se prefiere tener los comandos de estilo dentro del código fuente de la página Web, los cambios realizados entre las etiquetas de apertura y cierre `<style>` se aplicarán a toda la página; pero no afectarán al resto de los documentos del sitio Web.

Si por otro lado, si sólo se requiere aplicar los cambios de estilo en un elemento HTML específico, se puede agregar el atributo `style` en cualquiera de las etiquetas que forman parte del código fuente de la página Web. Los cambios sólo se aplicarán en dicho elemento.

El potencial de diseño estético que tiene CSS es muy grande, ofreciendo una infinidad de combinaciones estilísticas que pueden ser aplicadas en las páginas Web. Por eso al terminar de aprender HTML es indispensable estudiar CSS para complementar los conocimientos de desarrollo Web adquiridos. Hay que tener claro que si no se sabe CSS, no se pueden diseñar páginas y por ende sitios Web que cubran las expectativas visuales de los usuarios de hoy día. Sería como construir casas y dejarlas en su estado estructural; sin ventanas ni puertas, pisos y techos descubiertos de cemento; sin pintar, etc.

El lenguaje HTML sirve para definir los elementos que forman parte de la página, mientras que con CSS se organiza la distribución de los elementos de la página, colocándolos en posiciones precisas elegidas por el diseñador; se da estilo a dichos elementos (colores de fondo, formato de texto, colores de hipervínculos, efectos especiales, etc). Por lo tanto HTML sin CSS es incompleto.

En este tutorial nos dedicamos solamente a HTML, por lo tanto es recomendable que al finalizarlo busquemos un buen tutorial de CSS para completar el curso de diseño Web. Obviamente que para convertirnos en buenos diseñadores y desarrolladores Web, también se necesita saber utilizar algunas aplicaciones como Flash; editores de imágenes como Paint Shop Pro o Photoshop; así como otros lenguajes entre los que se cuentan Javascript, PHP, ActionScript y el manejo de bases de datos como MySQL. Con todos esos conocimientos más el uso de otras pequeñas herramientas útiles de administración de archivos como los FTP (que sirven para subir al servidor los archivos que forman parte de un sitio); se pueden crear sitios Web de gran calibre; con lo cual podremos incluso dedicarnos profesionalmente al diseño y desarrollo Web.

Aunque este tutorial está dedicado exclusivamente al aprendizaje completo de HTML y no de CSS u otros lenguajes, veremos un ejemplo de CSS para darnos una idea de qué se trata y cómo se utiliza. En el siguiente ejemplo daremos estilo al texto.

```
<html>
<head>
<title>Ejemplo de estilo CSS</title>

<style type="text/css">
h1 {color: red; font-family: arial;}
h2 {color: blue; font-family: verdana;}
p {color: fuchsia; font-family: courier; font-size: 4em;}
</style>

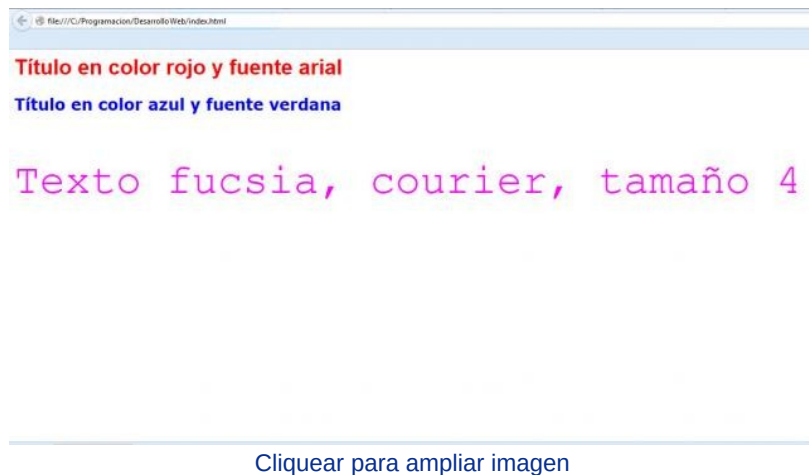
</head>

<body>

<h1>¡Título en color rojo y fuente arial!</h1>
<h2>¡Título en color azul y fuente verdana!</h2>
<p>Texto fucsia, courier, tamaño 4</p>

</body>
</html>
```

El código fuente anterior mostraría lo siguiente en los navegadores Web:



Javascript en HTML

El lenguaje HTML sirve para diseñar la estructura de una página, sin embargo los elementos que se agregan mediante este lenguaje son estáticos. Para darle dinamismo a la página; por ejemplo botones dinámicos con efectos especiales, imágenes animadas y en general cambios dinámicos del contenido; se requiere otro lenguaje de programación llamado Javascript. Una de las ventajas de Javascript es que el código fuente puede ser agregado entre medio del código fuente de la página HTML; siento estos dos lenguajes completamente compatibles. Sin embargo hay que tener en cuenta que la sintaxis de estos lenguajes es muy distinta una de la otra; ya que HTML es un lenguaje de diseño estructural de páginas Web y Javascript es un lenguaje de programación lógico que le da dinamismo a los elementos de la página Web.

Para agregar código fuente Javascript entre medio del código HTML se debe utilizar la etiqueta `<script>`. El código fuente deberá ser codificado entre medio de las etiquetas de apertura y cierre `<script>`. A partir de la etiqueta de apertura, todo lo que haya dentro de aquel espacio será código fuente Javascript hasta la etiqueta de cierre, tras la cual se retorna a HTML.

La etiqueta `<script>` sirve para agregar elementos dinámicos a las páginas Web programados en distintos tipos de lenguajes compatibles a HTML, por eso mismo se debe indicar en el atributo `type` de la misma, el nombre del lenguaje de programación; a pesar de esto, el lenguaje de programación compatible con HTML más utilizado es Javascript, sin embargo es importante aclarar en `type` el nombre del lenguaje.

En este tutorial no se explicarán los comando de Javascript, ya que se trata de otro lenguaje; sin embargo a continuación, a modo de ejemplo, dentro del espacio de `<script>` se muestra una línea que contiene un comando Javascript.

```
<html>
<head>
<title>Ejemplo de Javascript</title>
</head>

<body>
<script type="text/javascript">

document.write("Hola, esto es simplemente un ejemplo");

</script>
</body>
</html>
```

El ejemplo anterior simplemente sirve para demostrar cómo se utiliza la etiqueta HTML `<script>` para agregar código fuente Javascript.

Maquetación de páginas Web

Uno de los puntos más importantes que hay que tener en cuenta al diseñar un sitio Web, es el de la distribución visual de las páginas Web que lo componen; dónde estarán ubicadas las opciones o botones de hipervínculos, el contenido principal, etc.

Por lo general lo más recomendable, así como también lo más acostumbrado, es situar los botones de hipervínculos en la parte superior o en la parte izquierda de la página Web, mientras que en el área central se sitúa el contenido principal de la página.

Uno de los motivos principales de ese tipo de organización o distribución de los elementos de la página se debe a que en la mente humana las cosas por lo general se perciben en un cierto orden que obedece a mecanismos psicológicos que hacen que observemos o analicemos nuestro alrededor siguiendo un patrón específico. Por lo tanto el usuario se dirige inmediatamente hacia lugares específicos de la página para realizar ciertas actividades; si los elementos que busca no se encuentran en las posiciones donde habitualmente se localizan en la mayoría de los sitios Web, se pierde eficiencia.

Es muy importante también mantener suavidad y uniformidad en los colores de la página para hacer que el usuario tenga una sensación de tranquilidad y pueda concentrarse en el contenido. Los colores muy llamativos pueden distraer la atención del usuario.

Hay dos maneras de organizar y distribuir los elementos de una página, el método viejo mediante la utilización de tablas y el método actualmente más utilizado que depende de CSS. Las tablas son elementos HTML que fueron específicamente creados para tabular información y no como herramientas de diseño; sin embargo en los primeros tiempos cuando no se utilizaba CSS para el diseño visual, se debía recurrir plenamente a HTML para lograr páginas lo más atractivas posible.

Dado que no existen etiquetas especiales o atributos en HTML para ubicar los distintos elementos que forman parte de una página, mediante la utilización de tablas de borde invisible o valor "0" se pueden situar dichos elementos en las distintas celdas de dimensiones determinadas para lograr un efecto de posicionamiento libre en cualquier punto de la página.

Por ejemplo con una tabla de borde "0" o invisible, de 2 columnas y 2 filas; se puede diseñar una página con una barra horizontal en la parte superior para colocar el logo, una columna a la izquierda para la botonera y un espacio para el contenido principal. Se podría utilizar la celda superior izquierda para el logo, la inferior izquierda para la columna de botones y la inferior derecha para el contenido central. Dentro de la celda del contenido pueden insertarse tablas anidadas para organizar los elementos del contenido en posiciones específicas mediante el ajuste de los tamaños de las celdas.

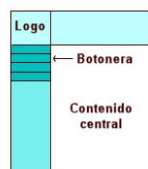


Tabla de 4 celdas (2 columnas x 2 filas) y borde 0. En la celda superior izquierda se encuentra el logo, en la celda inferior izquierda la botonera de hipervínculos y en la celda inferior derecha el contenido central de la página.

A pesar de obtenerse los resultados buscados, utilizar tablas para organizar y distribuir elementos de HTML no es correcto. Lo que se debe hacer es utilizar CSS, dadas las mayores capacidades de diseño que ofrece. Mediante CSS se pueden organizar y distribuir los elementos de las páginas con mayor facilidad y flexibilidad que con las tablas. Otra desventaja de utilizar tablas para la distribución de páginas Web en lugar de CSS, es que dados los mecanismos utilizados por los motores de búsqueda, se les complica indexar las páginas cuyos elementos están distribuidos en tablas, resultando en puestos más bajos en búsquedas. Por lo tanto no sólo por motivos de diseño sino también de posicionamiento en buscadores, es más conveniente utilizar CSS en lugar de tablas para la distribución de las páginas Web.

Las etiquetas que permiten aprovechar atributos CSS en la distribución de páginas son `<div>` y ``. Dichas etiquetas definen áreas individuales dentro de una página. De por sí solas `<div>` y `` no dan resultados visuales, ya que lo único que hacen es definir zonas o áreas dentro de la página; pero si se les agregan modificadores de estilo CSS para definir las posiciones y estilos de diseño de dichas áreas, se puede apreciar el gran potencial que tienen estas dos etiquetas.

A continuación se van a dar dos ejemplos de la misma distribución de una página Web realizados con una tabla en el primer ejemplo y con la combinación de `<div>` y CSS en el segundo. Hay que aclarar que en el primer ejemplo que

utiliza una tabla para la distribución, se utiliza un poco de CSS dentro de algunas etiquetas pero solamente para definir los colores de fondo de cada celda, los cuales facilitan la diferenciación visual de cada una; en ese ejemplo no se utiliza CSS en absoluto para la distribución.

Ejemplo 1 (tablas):

```
<html>
<head>
<title>Ejemplo de distribución de página</title>
</head>

<body>

<table width="100%" style="height: 100%;" border="0" cellspacing="0">
<tr>
<td colspan="2" style="background-color:magenta;">
<h1>Logo de la página Web</h1>
</td>
</tr>

<tr valign="top">
<td width="20%" valign="top" style="background-color:grey;">
<b>Botonera</b><br />
Botón 1<br />
Botón 2<br />
Botón 3<br />
Botón 4
</td>
<td valign="top" style="background-color:white;">Contenido principal de la
página.</td>
</tr>

<tr>
<td colspan="2" align="center" style="background-color:darkgrey;">Copyright © 2012
Página de Ejemplo</td>
</tr>
</table>
</body>

</html>
```

El ejemplo de maquetación anterior mostraría lo siguiente:



[Clicar para ampliar imagen](#)

Ejemplo 2 (<div> y CSS):

```

<html>
<head>
<title>Ejemplo de distribución de página</title>
</head>

<body style="margin: 0px;">

<div style="width:100%; height:100%; position: absolute; margin-left: 0px; margin-top: 0px;
padding: 0px; top: 0px;">

<div id="header" style="background-color:magenta; height: 100px; position: relative; margin-
top: 0px; top: 0px;">

<h1 style="margin-bottom:0;">Logo de la página Web</h1></div>

<div id="menu" style="background-color:grey;height:100%; width:20%;float:left;">
<b>Botonera</b><br />
Botón 1<br />
Botón 2<br />
Botón 3<br />
Botón 4</div>

<div id="content" style="background-color:white;height:100%; width:80%; float:left;">
Contenido principal de la página.</div>

<div style="background-color:darkgrey; clear:both; text-align:center; height: 40px;">
Copyright © 2012 Página de Ejemplo</div>

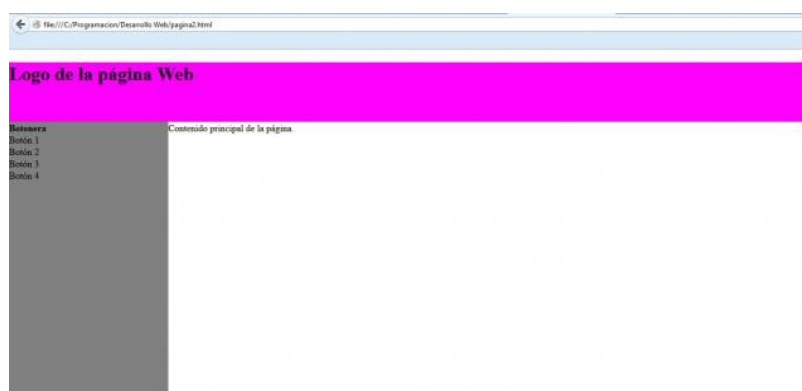
</div>

</body>

</html>

```

El ejemplo de maquetación anterior mostraría lo siguiente:



[Clicar para ver en grande](#)

No es objetivo del actual curso de HTML explicar los modificadores de estilo CSS que se incluyen en las etiquetas <div> del ejemplo anterior; la idea es simplemente comparar los dos métodos. Pero lo que sí se debe aclarar es que el segundo método es el más recomendado para organizar la distribución de una página Web. Por lo tanto es un requerimiento estudiar CSS luego de haber aprendido HTML, para poder lograr el diseño de sitios Web de calidad profesional.

Por lo tanto, las tablas son importantes pero para lo que fueron proyectadas, tabular datos y no para fines de diseño estilístico.

Multimedia en HTML

Los archivos multimedia vienen en muchos formatos. Multimedia es cualquier cosa que se puede oír o ver como texto, sonido, música, grabaciones, videos, películas, etc. Hoy en día ya es muy usual encontrar elementos de multimedia en páginas Web.

Los distintos exploradores Web soportan un cierto número de formatos multimedia, pudiendo así reproducir videos, música, etc. Los primeros exploradores Web solamente soportaban texto y no sólo eso sino que contaban únicamente con un tipo de fuente. Más adelante comenzaron a soportar colores, distintas fuentes y estilos de letras; y ya luego imágenes.

El soporte de audio, video y animaciones es manejado de distintas formas en cada tipo de explorador Web. Mientras que algunos elementos multimedia pueden ser reproducidos directamente por los componentes básicos del explorador, otros, para poder funcionar, requieren el uso de ciertos programas especiales que se añaden complementariamente, llamados plug-ins.

Hay distintos tipos de formatos multimedia, y cada uno realiza distintas tareas. A continuación se listan algunos de los más utilizados.

- Archivos AVI: Tienen extensión de formato .avi (Audio Video Interleave - En inglés Audio Video Intercalado); este formato fue desarrollado por Microsoft, siendo por lo tanto compatible en todas las computadoras con sistema operativo Windows. Además es soportado por la mayoría de los exploradores Web o browsers. En otros sistemas operativos a veces no funciona.
- Archivos WMV: Tienen extensión de formato .wmv (Windows Media); y como el anterior también ha sido desarrollado por Microsoft, siendo compatible con todos los sistemas operativos Windows. Sin embargo para poder ser reproducidos en otros sistemas operativos requieren la instalación de un programa o componente especial el cual es gratuito.
- Archivos MPEG: Tienen extensión de formato .mpg o .mpeg (Moving Pictures Expert Group - En inglés Grupo de Expertos de Imágenes en Movimiento). Este formato de video es el más popular en Internet y es soportado por todos los sistemas operativos y exploradores Web; por lo tanto es uno de los más recomendables a la hora de agregar videos.
- Archivos Quicktime: Tienen extensión de formato .mov . Este formato ha sido desarrollado por Apple; por lo tanto es soportado por las computadoras de aquella empresa como las Mac; mientras que para poder ser ejecutados en sistemas operativos Windows, requieren que se instale un plug-in o aplicación componente adicional gratuito.
- Archivos Flash o Shockwave: Tienen extensión de formato .swf y .flv . El formato Flash ha sido desarrollado por Macromedia. Este formato es utilizado para poder reproducir películas Flash (usadas para representar animaciones, interfaces dinámicas, juegos online, etc.). Hoy día son muy populares en Internet, dados los resultados estéticos positivos que posee. Por otro lado el formato .flv es muy utilizado en sitios de videos como Youtube.

Insertar objetos en HTML

Una de las capacidades de los exploradores Web es la de poder ejecutar algunos tipos de aplicaciones externas dentro de los mismos; dichas aplicaciones son conocidas como programas ayudantes o plug-ins (enchufados o conectados). Los plug-ins enriquecen a las páginas Web tanto estéticamente como funcionalmente, ya que permiten la ejecución de efectos especiales o tareas que no podrían funcionar en documentos Web si no fuera por ellas; por ejemplo los reproductores de películas o juegos online. Algunas de las aplicaciones incrustadas o enchufadas en documentos Web incluyen reproductores de archivos de Flash (.swf y .flv), reproductores Quicktime, etc. Una de las aplicaciones más utilizadas hoy día en páginas Web son los reproductores Flash. Desde interfaces visualmente ricas y con efectos especiales hasta reproductores de películas (como los de Youtube) y juegos online son en realidad aplicaciones o programas independientes enchufados en las páginas Web. La etiqueta que nos permite incluir estos objetos externos es <object>.

La etiqueta <object> va acompañada de otra etiqueta llamada <param>. Mientras la primera sirve para enchufar los programas ayudantes o plug-ins, la segunda es para ajustar los parámetros de funcionamiento específicos del programa introducido.

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de programa insertado</title>
</head>

<body>
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/
swflash.cab#version=8,0,0,0" width="708" height="137">

<param name="movie" value="http://www.lookpink.com/games/swfs/0.11.swf" />
<param name="quality" value="high" />
<param name="allowScriptAccess" value="sameDomain" />

</object>
</body>
</html>
```

Con el código fuente anterior se inserta una aplicación Flash de formato .swf. En el mismo se pueden ver diversos atributos en la etiqueta <object>.

Comencemos con los más fáciles; width y height que se utilizan para especificar el ancho y el alto que ocupará la aplicación dentro de la página.

En cuanto al extraño atributo classid el cual contiene un aún más extraño valor, se trata de una herramienta que sirve únicamente en Internet Explorer (IE). Ese valor le indica a IE qué tipo de programa se quiere ejecutar, por ejemplo el valor "d27cdb6e-ae6d-11cf-96b8-444553540000" declara que se quiere utilizar una aplicación Shockwave-Flash, requiriendo por lo tanto un reproductor de archivos de tipo .swf. A continuación vemos el atributo codebase en el cual se indica la ubicación del código fuente que permite la ejecución del objeto insertado; en el ejemplo anterior al tratarse de un archivo de tipo Flash, el código o aplicación que permite la ejecución del objeto se encuentra en la dirección "<http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=8,0,0,0>"; por lo tanto si en IE no tenemos instalado el plug-in que permite ejecutar programas Flash, mediante este atributo IE nos sugerirá bajarlo e instalarlo desde la dirección indicada.

El motivo de la existencia de classid es que en IE existe la posibilidad de insertar aplicaciones especiales de Microsoft o programadas para ejecutarse en sistemas operativos Microsoft Windows; dichas aplicaciones se denominan componentes ActiveX y pueden ser insertadas en todo tipo de herramientas de Microsoft como el Word, Excel o Visual Studio entre otros. El Internet Explorer, no se encuentra exento de dicha capacidad. Los componentes ActiveX son programas que se instalan y pueden ser controlados desde otras aplicaciones; por eso mismo pueden ser insertados en páginas Web como objetos externos. En classid lo que se indica mediante un código es el tipo de aplicación que se quiere enchufar. Cada tipo de aplicación tiene un código de identificación, por lo tanto no será el mismo para el reproductor Media Player que para el reproductor de películas Flash. Hay que aclarar que Flash no es un producto de Microsoft, pero igual por temas de compatibilidad cuenta su propio código de identificación para poder funcionar en Internet Explorer.

Con las etiquetas <param> se ajustan parámetros específicos del objeto insertado; dichos parámetros varían según el programa; no siendo los mismos en aplicaciones Flash, que en aplicaciones Quicktime o de otro tipo. El nombre del parámetro se indica en el atributo de <param> name, mientras que el valor de dicho parámetro se declara en el atributo value.

En aplicaciones o películas Flash se utiliza el parámetro movie (película) para indicar la dirección URL o ubicación de dicha película. También se ve el atributo quality (calidad) en el que indicamos un valor de "high" (alta). El parámetro allowScriptAccess permite a la película Flash tener acceso al código fuente de la página HTML e interactuar con la misma; los valores posibles son "sameDomain" (mismo dominio) y "always" (siempre); el primer valor indicando que solamente puede ser accedido el código fuente de la página si el archivo .swf se encuentra en el mismo dominio del documento Web; mientras que el segundo valor especifica que puede ser accedido por archivos .swf ubicados en otros servidores; el valor por omisión es "sameDomain". Hay que aclarar que los parámetros recién mencionados son específicos de Flash.

Sin embargo es muy posible que algunos exploradores Web como Firefox el código fuente anterior no funcione. Eso se

debe a que el atributo classid necesario para que Internet Explorer pueda insertar la película Flash, confunde a Firefox resultando en un espacio vacío donde debería estar el objeto. La solución a este problema nos lleva al concepto de condiciones de IE.

Las condiciones de Internet Explorer funcionan únicamente en aquel explorador Web, pero pueden llegar a ser útiles para solucionar conflictos como el que tenemos aquí. Se trata de un tipo de comando específico de IE cuya sintaxis se encuentra incluida dentro comentarios. Primero veremos cómo quedaría el código fuente anterior y luego lo analizaremos con cuidado.

```
<html>
<head>
<title>Ejemplo de programa insertado</title>
</head>

<body>
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/
swflash.cab#version=8,0,0,0" width="708" height="137">

<!--[if !IE]><!-->
<object type="application/x-shockwave-flash" codebase="http://fpdownload.macromedia.c
om/pub/shockwave/cabs/flash/swflash.cab#version=8,0,0,0" width="708" height="137"
data="http://www.lookpink.com/games/swfs/0.11.swf">
<!--<![endif]>>

<param name="movie" value="http://www.lookpink.com/games/swfs/0.11.swf" />
<param name="quality" value="high" />
<param name="allowScriptAccess" value="sameDomain" />

</object>
</body>
</html>
```

En el código fuente se puede observar la etiqueta <object> sin ninguna modificación respecto al código fuente anterior, y además una segunda versión de la misma etiqueta con algunas modificaciones, ubicada entre unos extraños comandos. La primera versión es para que el IE pueda insertar el objeto, mientras que la segunda versión es para que lo hagan otros exploradores Web como Firefox. Primero analizaremos la sintaxis de los comandos condicionales de IE y a continuación los atributos.

Estos comandos no son parte de HTML y tampoco son validados por el consorcio W3C; por lo tanto son incluidos como comentarios, ya que solamente pueden ser interpretados por IE y los otros exploradores Web pueden dar resultados no deseados.

La estructura de las condiciones son una parte esencial de la programación y de la informática en general, en otras palabras, sin comandos de condiciones, la computación como la conocemos hoy día no existiría. Sin embargo son elementos pertenecientes a lenguajes de programación lógicos como Javascript, PHP, Visual Basic, C/C++, etc; los cuales son más avanzados que HTML, un lenguaje de marcado estático que se utiliza para definir estructuras de documentos Web; siendo así, los comandos de condiciones escapan al objetivo del presente tutorial dedicado exclusivamente a HTML. No obstante, dada la necesidad de aplicarlos para hacer funcionar objetos insertados en exploradores Web como Firefox, se dará una breve descripción del funcionamiento o mecanismo que hay detrás de las condiciones.

En el código fuente hay una etiqueta <object> (la primera) con atributos que pueden ser interpretados por IE pero no por Firefox; a continuación le sigue una condición la cual ordena al explorador Web que si no es IE ejecute la segunda etiqueta <object> con atributos interpretados por otros exploradores Web que no son IE. El truco detrás de dicha movida es que si tenemos un explorador Web IE el mismo interpretará y ejecutará la primera etiqueta <object>, mientras que al leer la condición de "si no es IE ejecute la segunda etiqueta <object>", como se trata efectivamente de IE el resultado de dicha condición será falso, o sea que se saltará la segunda etiqueta <object> y no la ejecutará. Por el contrario, Firefox

no podrá interpretar la primera etiqueta <object> y tampoco interpretará los comandos condicionales de IE, ya que los tomará como simple texto; pero sí podrá entender la segunda etiqueta <object>, logrando insertar así finalmente la película Flash. A continuación analizaremos la sintaxis de los comandos condicionales de IE.

El condicional [if !IE]> se puede descomponer de la siguiente manera: La palabra if significa "si" en inglés; el símbolo ! significa "no es"; mientras que IE, como ya sabemos, son las siglas de Internet Explorer; de esta manera se arma la condición "si no es IE". El comando [endif], simplemente señala el fin del condicional "si" ("end" significa final), por lo tanto significa "finalizar si".

En el medio, entre el principio de la condición y el final se coloca la etiqueta a ejecutar por otros exploradores Web y a ignorar por IE.

Un detalle al que hay que prestarle atención es que los comandos condicionales deben estar dentro de comentarios, ya que los exploradores Web como Firefox, al no interpretarlos los considerarán como simple texto. Por lo tanto el código de los comandos condicionales de IE quedaría así: <!--[if !IE]><!--> y <!--<![endif]-->. Al figurar como comentarios, Firefox se lo saltará leyendo directamente la segunda etiqueta <object> que podrá interpretar dadas unas modificaciones en los atributos.

Ahora pasemos a analizar los atributos de <object> que permiten visualizar el objeto insertado en otros exploradores Web que no sean IE. Una de las razones por las que no se puede insertar correctamente el objeto en otros browsers como Firefox, es por el atributo classid que es incompatible con estos últimos. Por lo tanto debemos reemplazar a classid con el atributo type (tipo), en el que se indica el tipo de aplicación que se quiere insertar; siendo en el ejemplo nuestro "application/x-shockwave-flash" (aplicación Shockwave Flash). Otro atributo que debemos agregar es data, en donde se debe declarar la URL donde se encuentra el objeto a insertar. No importa que en el atributo value de <param> haya sido también declarado, se trata de una redundancia necesaria. Con las modificaciones recién descritas finalmente se podrá reproducir la película Flash en todos los exploradores Web. Sin embargo no todo es tan simple, hay unos puntos que hay que aclarar antes.

Microsoft ha anunciado que a partir de Internet Explorer 10, los condicionales IE no funcionan más en páginas Web diseñadas en versión HTML 5; por lo tanto la implementación de dichos condicionales no garantiza su funcionamiento en futuras versiones de IE.

Otra manera de hacer que el objeto insertado pueda ser visualizado en todos los exploradores Web es con la etiqueta <embed>; no obstante hay que aclarar que dicha etiqueta se había desaprobado en versiones HTML 4 y XHTML 1.0; pero volvió a ser aprobada en HTML 5. Para utilizarla en versiones HTML 4 y XHTML hay que declarar en <!DOCTYPE> que se trata de una versión de tipo transicional; en HTML 5 la etiqueta <embed> funcionará sin ningún problema.

Veamos el código fuente:

```
<html>
<head>
<title>Ejemplo de programa insertado</title>
</head>

<body>
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/
swflash.cab#version=8,0,0,0" width="708" height="137">

<param name="movie" value="http://www.lookpink.com/games/swfs/0.11.swf" />
<param name="quality" value="high" />
<param name="allowScriptAccess" value="sameDomain" />

<embed src="" quality="high" width="708" height="137"
allowScriptAccess="sameDomain" type="application/x-shockwave-flash"
pluginspage="" />

</object>
```

```
</body>
</html>
```

En el código fuente anterior se puede ver que `<embed>` se coloca entre las etiquetas `<object>` de apertura y cierre, reemplazando los comandos condicionales de IE y la etiqueta `<object>` para browsers como Firefox. Los parámetros del objeto en vez de ser definidos en etiquetas separadas `<param>`, son definidos en atributos dentro de la etiqueta `<embed>`.

Los atributos utilizados son: `src`, donde se declara la dirección del objeto a insertar; `allowScriptAccess` que cumple la misma función que en su versión de valor en la etiqueta `<param>`, `type` para indicar el tipo de aplicación que se inserta y `pluginspage` (página del plug-in) para indicar la dirección de descarga del plug-in que permite ejecutar el objeto si aún no está instalado en el browser.

El tipo de solución a aplicar depende de nosotros y del tipo de finalidad que tendrá el sitio Web. Es interesante mencionar que cuando se crean películas o archivos `.swf`, el editor de Flash crea además automáticamente un archivo HTML con un código fuente que aplica el último método de inserción de objetos que vimos, combinando las etiquetas `<object>` y `<embed>`. Dicho código fuente puede ser copiado y pegado en la página Web que estamos diseñando, así como también se pueden modificar los valores de algunos parámetros y atributos ajustándolos a los requerimientos nuestros.

Cada tipo de objeto utiliza una serie de parámetros propios del mismo, por lo tanto pueden variar de un tipo de aplicación a otra.

Agregar audio en HTML

Agregar audio a una página Web no es tan simple, ya que se requieren algunos trucos para lograr que todos los exploradores Web y los distintos sistemas operativos puedan reproducir sin problemas los archivos de audio de la página.

La forma más fácil de agregar archivos de audio en una página es utilizando el reproductor gratuito de Yahoo. Lo único que hay que hacer es agregar en la página un corto código de Javascript, que se encuentra en el servidor de Yahoo mediante la etiqueta `<script>`. Se debe apuntar hacia la dirección URL donde está ubicado el reproductor para que pueda ser añadido en la página. Luego se agregan unos simples enlaces, con la etiqueta `<a>` que apuntan hacia los archivos de audio que queremos agregar en la página. Al tocar cualquiera de los enlaces, el reproductor comenzará a reproducir el audio.

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de Audio</title>
</head>

<body>

<p><a href="cancion1.mp3">Canci&oacute;n 1</a></p>
<p><a href="cancion2.mp3">Canci&oacute;n 2</a></p>
<p><a href="cancion3.mp3">Canci&oacute;n 3</a></p>
<p><a href="cancion4.mp3">Canci&oacute;n 4</a></p>

<script type="text/javascript" src="http://mediaplayer.yahoo.com/js"></script>

</body>

</html>
```

Como se ve en el código fuente anterior, la etiqueta `<script>` agrega el reproductor de Yahoo, ya que en el atributo `src` se declaró la dirección donde se encuentra el código fuente del reproductor; haciendo que de esta manera el mismo sea incrustado en la página. De esta manera si en la misma página hay hipervínculos hacia archivos de audio, estos se abrirán y serán ejecutados automáticamente por el reproductor añadido de Yahoo. Los nombres de archivo del

ejemplo anterior deben ser reemplazados por nombres de archivos de audio reales, ya que esos son solamente para ejemplificar.

Otra manera de agregar audio en una página Web es mediante la etiqueta <audio>, entre cuyas etiquetas de apertura y cierre se pone la etiqueta <source>. La etiqueta <audio> se utiliza para introducir el reproductor del explorador Web en la página; para ello hay que incluir el atributo controls cuyo valor también es "controls". Pero eso solamente coloca el reproductor en la página; para agregar un archivo de audio se debe incluir entre las etiquetas de apertura y cierre de <audio> la etiqueta <source>, en cuyo atributo src se debe indicar la dirección de la ubicación del archivo de audio, mientras que en el atributo type se indica de qué tipo de archivo o formato se trata. Estos dos atributos son necesarios y siempre deben utilizarse.

Por temas de compatibilidad en distintos browsers o exploradores Web es recomendable que los archivos de audio se suban en formato .mp3 y .ogg, ya que si un formato no puede ser reproducido por un cierto browser otro formato sí podrá.

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de Audio</title>
</head>

<body>

<audio controls="controls">
<source src="cancion.mp3" type="audio/mpeg" />
<source src="cancion.ogg" type="audio/ogg" />
</audio>

</body>

</html>
```

Hay que aclarar que la etiqueta <audio> solamente está validada en la versión de HTML 5; por lo tanto solamente funcionará en exploradores Web nuevos. Tampoco funcionará en códigos fuente que han sido declarados en <!DOCTYPE> como escritos en versiones XHTML o HTML 4. Así, al utilizar esta etiqueta hay que aclarar que se trabaja en la versión HTML5.

Otro reproductor similar al de Yahoo es el reproductor de Google; pero a diferencia del anterior que está basado en un código Javascript, el reproductor de Google se incrusta en la página con la etiqueta <embed>; en cuyo atributo type (tipo) se debe indicar que se trata de una aplicación shockwave-flash mediante el siguiente valor "application/x-shockwave-flash"; mientras que en el atributo src se debe indicar la dirección del reproductor en el servidor de Google así como la dirección del archivo de audio a reproducir de la siguiente manera: "<http://www.google.com/reader/ui/3523697345-audio-player.swf?audioUrl=cancion4.mp3>", donde "<http://www.google.com/reader/ui/3523697345-audio-player.swf>" es la ubicación del reproductor de Google y "cancion4.mp3" es la dirección URL del archivo de audio a reproducir. También se pueden agregar los atributos width y height para indicar el ancho y el alto que ocupará el reproductor.

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de Audio</title>
</head>

<body>

<embed type="application/x-shockwave-flash"
src="http://www.google.com/reader/ui/3523697345-audio-player.swf"
```

```
audioUrl=cancion4.mp3" height="200" width="200"></embed>

</body>

</html>
```

Como se ve en el código fuente anterior, no se pone nada entre las etiquetas de apertura y cierre de <embed>. Además en src se indican los valores de la ubicación del reproductor y luego de la parte que dice "audioUrl" se indica la dirección del audio. En este caso se agregó un archivo de sonido llamado cancion4.mp3 el cual se encuentra en el mismo directorio en que se encuentra la página html que estamos diseñando, por eso solamente se indica el nombre del archivo; por lo tanto este nombre de archivo puede ser reemplazado por cualquier otra dirección Web.

Agregar video en HTML

Así como con el audio, agregar videos en una página Web no es tan simple por motivos de compatibilidad. Los comando existentes pueden funcionar en algunos browsers mientras que en otros no; así como hay formatos de video que no pueden ser reproducidos en ciertos sistemas operativos. También hay que tener en cuenta que hay etiquetas no compatibles en versiones anteriores de HTML, haciendo que los browsers más viejos no puedan reproducir los videos agregados.

El primer método que veremos para agregar videos es el que utiliza la etiqueta <video>. Esta etiqueta es nueva y está validada únicamente a partir de HTML 5; por lo tanto los usuarios de exploradores Web más viejos que no soportan HTML 5 no podrán ver estos videos, por suerte sin embargo la mayoría de los usuarios hoy día utilizan browsers actualizados. También si en <!DOCTYPE> se indica que el código fuente está escrito en versión XHTML, tampoco funcionará.

Al igual que <audio>, la etiqueta <video> agrega el reproductor en la página, mientras que la etiqueta acompañante <source>, que debe ir colocada entre las etiquetas de apertura y cierre <video>, es la que indica la dirección del video a reproducir.

Los atributos de <video> width y height definen el ancho y el alto del reproductor respectivamente; mientras que el atributo controls incluye los controles del reproductor, el valor de este atributo debe ser también "controls"; este atributo es muy recomendable incluso se puede decir que es obligatorio, ya que sin controles el reproductor el video no podrá ser reproducido.

Con la etiqueta <source> se indica la fuente del video a reproducir mediante el atributo src; así como el tipo y formato del mismo en el atributo type. Es recomendable convertir el video a varios formatos con una aplicación especial, para que pueda ser reproducido en browsers o sistemas operativos que no soportan ciertos formatos; por lo tanto entre las etiquetas de apertura y cierre <video> se tendrán que agregar varias etiquetas <source> con los archivos del video en distintos formatos.

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de Video</title>
</head>

<body>

<video width="200" height="200" controls="controls">
<source src="videodeejemplo.ogg" type="video/ogg" />
<source src="videodeejemplo.mp4" type="video/mp4" />
<source src="videodeejemplo.webm" type="video/webm" />
Su browser no soporta la etiqueta video
</video>
```

```
</body>
```

```
</html>
```

En el ejemplo anterior se agregaron varios formatos del mismo video por las dudas si no es compatible con algún browser o sistema operativo. Además se puede ver un texto antes de la etiqueta de cierre `</video>` que aparecerá si el reproductor no funciona; por distintos motivos como tener una página escrita en versión HTML 4.01 o XHTML estricta; o por ser un browser viejo.

Para agregar videos también se puede utilizar la etiqueta `<embed>` en cuyo atributo `src` se indica la dirección del video y en los atributos `width` y `height` el ancho y el alto respectivamente. Pero es importante aclarar que no se incluyen controles de reproductor y el video se reroducirá automáticamente. Este método es más conveniente para reproducir videos Flash en formato swf que videos en otros formatos. Uno de los inconvenientes de la etiqueta `<embed>` es que no está validado en la versión HTML 4, por lo tanto solamente funcionará en códigos fuente declarados como HTML 5 o HTML 4 no estricto y por lo tanto en nuevos exploradores Web. Otra cosa a tener en cuenta es que los videos Flash no pueden ser reproducidos en iPad y iPhone.

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de Video</title>
</head>

<body>

<embed src="pelicula.swf" width="200" height="200" />

</body>

</html>
```

Otra etiqueta similar a `<embed>` para agregar videos es `<object>`; la cual también es más utilizada para agregar videos Flash de formato swf. En el atributo `data` se debe indicar la dirección del video; mientras que con los atributos `width` y `height` se establecen el ancho y el alto del video respectivamente. La etiqueta `<object>` funciona tanto en versiones HTML 4, como XHTML y HTML 5. Tambien hay que tener en cuenta que los videos Flash no pueden ser reproducidos en iPad y iPhone.

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de Video</title>
</head>

<body>

<object data="pelicula.swf" width="200" height="200" />

</body>

</html>
```

Agregar videos de Youtube en HTML

Hoy día está muy de moda agregar videos de Youtube en páginas Web, esto se logra usando un marco incorporado con la etiqueta `<iframe>`, en cuyo atributo `src` se indica la dirección de la página Youtube del video. Sin embargo no se trata de solamente copiar y pegar la URL, ya hay que hacer una pequeña modificación en la dirección. Por ejemplo en un video de Youtube ubicado en la página "<http://www.youtube.com/watch?v=K0c2HG2sBjs>" hay que hacer una pequeña modificación que deja la dirección con el siguiente formato: "<http://www.youtube.com/v/K0c2HG2sBjs>". En los atributos `width` y `height` se indican el ancho y el alto respectivamente con el que el video Youtube aparecerá en la página que diseñamos.

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de Video Youtube</title>
</head>

<body>

<iframe src="http://www.youtube.com/v/K0c2HG2sBjs" width="50%" height="500"
scrolling="no" frameborder="1"></iframe>

</body>

</html>
```

También se puede utilizar la etiqueta `<embed>` para agregar videos de Youtube en una página Web, en cuyo atributo `src`, al igual que con `<iframe>`, hay que indicar la dirección de la página Youtube donde se encuentra el video a incluir; también con el formato de la dirección modificado, como se hace con `<iframe>`. Además hay que incluir un atributo `type` (tipo), cuyo valor debe indicar el tipo de aplicación que se agrega; que sería "application/x-shockwave-flash" (aplicación Shockwave-Flash).

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de Video Youtube</title>
</head>

<body>

<embed src="http://www.youtube.com/v/K0c2HG2sBjs" width="500" height="500"
type="application/x-shockwave-flash"></embed>

</body>

</html>
```

Recordemos que la etiqueta `<embed>` quedó desaprobada en HTML 4.01 y luego volvió a ser aprobada por la W3C en la versión de HTML 5. Por lo tanto si se quiere utilizar `<embed>` hay que declarar en `<!DOCTYPE>` un código fuente transicional o de HTML 5.

Etiquetas de cabecera en HTML

Hasta ahora hemos trabajado con elementos HTML visibles, o sea que son visualizados en el explorador Web, dado que han sido declarados dentro de la sección `<body>` (cuerpo) del código fuente, que es donde se insertan los

elementos estructurales de la página.

Pero todavía nos falta estudiar la sección de la cabecera del código fuente <head>, donde se ubican las etiquetas de elementos no visibles, o sea que no son visualizado en la página y solamente cumplen funciones informativas acerca del documento Web.

La información que brindan las etiquetas de la sección de la cabecera a pesar de no ser directamente perceptible por los usuarios es de gran importancia, puesto que es interpretada por los exploradores Web, motores de búsqueda y otros servicios de Internet.

De las etiquetas de la cabecera solamente aprendimos a utilizar una, <title>, de hecho fue una de las primeras que vimos en el curso de HTML, que sirve para definir el título de la página que aparecerá en la barra de título o en la pestaña. En realidad se trata de la única etiqueta de cabecera que brinda resultados visibles a los usuarios, aunque se encuentren fuera del cuerpo del contenido de la página, se pueden ver. El resto de las etiquetas brindan únicamente información.

Comenzaremos con la etiqueta <base>, la cual brinda información a los exploradores Web. Se utiliza para indicar la URL base de todas las URL relativas de los hipervínculos, imágenes y otros elementos que hay en la página. Por ejemplo si la URL base que se indica en <base> es "<http://www.ejemplo.com/paginas/>" y en el atributo src de una etiqueta de imagen indicamos la dirección "imagen1.jpg", el explorador Web cargará la imagen ubicada en "<http://www.ejemplo.com/paginas/imagen1.jpg>". También se utiliza para definir el target (objetivo) de todos los hipervínculos.

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de URL y target base</title>
<base href="http://www.ejemplo.com/paginas/" target="_blank" />
</head>

<body>



<a href="inicio.html">Inicio</a>
<a href="servicios.html">Servicios</a>
<a href="productos.html">Productos</a>
<a href="ayuda.html">Ayuda</a>
</body>

</html>
```

En el ejemplo anterior la URL base es "<http://www.ejemplo.com/paginas/>", por lo tanto la imagen de la etiqueta deberá estar en "<http://www.ejemplo.com/paginas/imagen1.jpg>" para poder ser visualizada; mientras que los hipervínculos abrirán las páginas que se encuentran en "<http://www.ejemplo.com/paginas/inicio.html>", "<http://www.ejemplo.com/paginas/servicios.html>", "<http://www.ejemplo.com/paginas/productos.html>" y "<http://www.ejemplo.com/paginas/ayuda.html>". Todos los hipervínculos se abrirán en ventanas o pestañas nuevas según lo indicado en <base>.

Los valores posibles del atributo target de <base> son los mismos que los de la etiqueta <a>; "_blank" para abrir las páginas en ventanas o pestañas nuevas; "_top" abre la página en la misma ventana de la página abierta (es el valor por omisión); "_self" abre la página en el mismo marco del hipervínculo (muy poco utilizado, solamente sirve si se trabaja con "frames"; y finalmente se puede declarar el nombre del marco (frame) donde se quiere abrir la página del hipervínculo.

En realidad la función de <base> no es estructuralmente muy importante, simplemente ofrece comodidad al diseñador, para evitar tener que indicar la URL absoluta de cada imagen e hipervínculo; o el target de cada hipervínculo. Por ejemplo si primero declaramos target "_blank" y después nos arrepentimos queriendo que los hipervínculos se abran en

la misma ventana o pestaña, solamente tenemos que cambiar el target de <base> en lugar de tener que modificar el target de cada <a>.

ELEMENTO DE CABECERA LINK

La etiqueta <link> de la sección de cabecera sirve para vincular un recurso externo a una página Web. Su utilización más común es para vincular archivos de hojas de estilo CSS externos con la página Web. Como ya hablamos antes, los modificadores de estilo CSS pueden insertarse individualmente dentro de las etiquetas HTML con el atributo style; pueden declararse dentro de la sección <head> mediante la utilización de la etiqueta <style> de la cual hablaremos más tarde; y también pueden declararse en un archivo CSS externo el cual es insertado o importado en la página Web con la etiqueta <link>.

Para graficarlo de otra manera cuando el explorador Web carga el código fuente de la página HTML, además le adjunta el código fuente del archivo de hojas de estilo CSS externo dentro de la sección de cabecera, resultando un sólo código fuente.

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de archivo CSS insertado</title>
<link rel="stylesheet" type="text/css" href="estilo.css" />
</head>

<body>

<p>Página Web con archivo de estilo insertado.</p>

</body>

</html>
```

El código fuente anterior no genera ninguna diferencia visual, ya que para eso se necesitan tener algunos conocimientos de CSS que van más allá del objetivo del actual curso de HTML. El ejemplo anterior simplemente muestra cómo vincular un archivo o recurso externo con el documento HTML utilizando la etiqueta <link>. A continuación analizaremos cada atributo.

Primero tenemos el atributo rel (por relationship que en inglés significa relación), en donde se indica qué tipo de relación tiene la página con el archivo o recurso con el que se vincula; en otras palabras, se indica para qué se quiere vincular el recurso con el documento HTML. En el ejemplo la relación indicada es "stylesheet" (hoja de estilo CSS); lo que significa que se trata de un archivo CSS con estilos que serán insertados en el código fuente HTML en el momento que se cargue la página.

En type se indica el tipo o formato de archivo a insertar; en el ejemplo se declara el valor "text/css", significando que se trata de un archivo en formato texto y de tipo css. Finalmente en el atributo href se indica la dirección del archivo o recurso a insertar.

Aunque se pueden vincular muchos tipos de archivos, la etiqueta <link> es más frecuentemente utilizada para vincular hojas de estilo CSS.

En la sección de cabecera se pueden incluir múltiples etiquetas <link>, vinculando al documento HTML con varios archivos de recurso. Así por ejemplo se pueden tener varios archivos de hoja de estilo CSS vinculados o insertados en la página HTML.

ELEMENTO DE CABECERA STYLE

Como ya se mencionó antes, se pueden incluir los modificadores de estilo CSS dentro de la sección de cabecera del archivo HTML, en lugar de insertar un archivo CSS conteniendo a los mismos. Dichos modificadores de estilo serán aplicados en los elementos HTML de la página. Para lograr eso se requiere la utilización de la etiqueta <style>, dentro de la cual (entre la etiqueta de apertura y cierre) se debe escribir el código fuente en CSS. Los modificadores de estilo

incluidos se aplicarán a los elementos HTML de la sección de contenido visible o cuerpo <body>. El atributo que se utiliza en <style> es type, y sirve para indicar el tipo de estilo que se incluye; siendo el valor siempre "text/css" (tipo de texto css).

Veamos un ejemplo:

```
<html>
<head>
<title>Ejemplo de estilo CSS</title>

<style type="text/css">
h1 {color: red;}
h2 {color: green;}
p {color: blue;}
</style>

</head>

<body>

<h1>Título 1 de color rojo.</h1>
<h2>Título 2 de color verde.</h2>
<p>Párrafo de color azul.</p>

</body>

</html>
```

En el ejemplo anterior se puede ver el código fuente CSS entre las etiquetas de apertura y cierre <style>, con modificadores de estilo que definen los colores de los elementos <h1>, <h2> y <p> del cuerpo <body> de la página Web.

ELEMENTOS META HTML

Uno de los tipos de elementos de cabecera más utilizados son los <meta>, los cuales proveen datos o información acerca del documento o página Web para los browsers, motores de búsqueda y otros servicios Web. Veamos como funcionan.

Los recursos o archivos que se transmiten entre los servidores y los exploradores Web, lo hacen cumpliendo ciertas normas o protocolos llamados HTTP (Hypertext Transfer Protocol - que en inglés significa Protocolo de Transferencia de Hipertexto). Por eso mismo en las URL de las direcciones Web se antepone el tipo de protocolo al que responde el documento a descargar por el browser "http://". Los tipos de recursos que se descargan son documentos HTML, imágenes, videos, etc. Cuando se abre una página Web ocurren una serie de actividades que el usuario no ve entre el explorador Web (denominado cliente en el protocolo HTTP) y el servidor; los cuales intercambian un cierto número de mensajes.

Antes de descargar los recursos que forman parte de una página Web, el browser (cliente) envía un mensaje de petición al servidor para descargar la página Web y los recursos que la componen; como respuesta el servidor le envía otro mensaje y a continuación los recursos de la página. El mensaje de petición del browser tiene la siguiente primera línea o renglón:

```
GET /archivo.html HTTP/1.1
```

El mensaje anterior significa "dame el archivo del directorio raíz llamado archivo.html, soy compatible con la versión de protocolo HTTP 1.1".

El mensaje de respuesta del servidor tiene en la primera línea o renglón (status line - línea de estado en inglés) lo siguiente:

HTTP/1.1 200 OK

El mensaje anterior significa "los recursos han sido enviados satisfactoriamente respetando el protocolo HTTP 1.1". El número 200 es un código que significa que lo pedido se ha enviado exitosamente. Existen distintos códigos de mensaje de respuesta, por ejemplo el código significa 401 "no autorizado", 402 "se requiere pago", 403 "prohibido", 404 "no se encuentra", etc.

Además de la primera línea o línea de estado, el mensaje de respuesta puede contener más líneas con información para el browser acerca de la página. Dichas líneas pueden ser agregadas por nosotros mediante la utilización de la etiqueta <meta>.

La estructura de las etiquetas <meta> y sus atributos para proveer información a los browsers acerca de la página es la siguiente:

```
<meta http-equiv="tipo-de-información" content="valor-información" />
```

Como se puede ver hay dos atributos, uno es http-equiv, cuyo valor es el tipo de información que se va a pasar, y content en el cual se indica el valor del tipo de información declarado en http-equiv. A continuación veremos distintos tipos de información meta que se pueden especificar en la cabecera del código fuente de la página Web que estamos diseñando.

Mensaje meta content-type

El tipo de información meta "content-type" se utiliza para que en el mensaje de respuesta del servidor, cuando se descarga una página, se especifique el tipo de juego de caracteres que utiliza la página. En la gran mayoría de los sitios Web se utiliza el juego de caracteres UTF-8, que responde al estándar de codificación de caracteres Unicode, con el cual cada carácter y símbolo tiene un código numérico que lo representa. Se recomienda que siempre se agregue este mensaje meta.

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />
```

En http-equiv se indica el nombre de la información meta "content-type" (tipo de contenido) y en el atributo content (contenido) se indica que es texto de html que obedece al juego de caracteres UTF-8 "text/html; charset=UTF-8". Se recomienda agregar la información meta "content-type" en todas las páginas que diseñemos como se mostró recién en el ejemplo.

Mensaje meta date

La información meta "date" se utiliza para especificar a los exploradores Web la fecha y la hora en que la página fue creada:

```
<meta http-equiv="date" content="Wed, 20 Apr 2011 22:00:00" />
```

Mensaje meta last-modified

El mensaje meta "last modified" indica a los exploradores Web y buscadores cuándo fue modificada por última vez la página Web:

```
<meta http-equiv="last-modified" content="Mon, 20 Dec 2011 20:20:00" />
```

Mensaje meta cache-control

Los exploradores Web tienen la capacidad de almacenar las páginas que son visitadas frecuentemente, para no tener que descargar del servidor, los recursos que la componen (imágenes, videos, página HTML, archivo de estilo CSS, etc) cada vez que se visita. En lugar de descargarla del servidor, la descarga de una carpeta de archivos temporales dentro del disco duro de nuestra computadora, pudiendo así acceder a ella y visualizarla mucho más rápido. Dicho mecanismo se llama cache, y las ventajas que ofrece es que se puede acceder más rápido a páginas que visitamos frecuentemente; sin embargo una de las desventajas es que si la página se actualiza frecuentemente, es posible que los usuarios que la tienen guardada en cache, no vean las modificaciones inmediatamente. Por ejemplo en portales de diarios o periódicos no se recomienda permitir el cache, ya que dichos sitios Web se actualizan frecuentemente y no se verían las modificaciones.

Existen dos tipos de chache, los privados y los públicos. Los caches privados son los provistos por el explorador Web como vimos recién. Los caches públicos por otro lado, pertenecen a los proveedores de servicios de Internet ISP (Internet Service Provider), o sea la empresa que nos permite conectarnos a Internet. Ese tipo de cache se denomina "cache de proxy". Como es utilizado por mucha gente, y el cache del proveedor de Internet es compartido por todos ellos, cuando una página es muy vista por algunos usuarios, la misma es cacheada y puede ser descargada por otros usuarios más rápidamente desde el servidor del ISP. El proxy del ISP se puede habilitar desde la sección de configuración de proxy del browser. A continuación veremos los distintos tipos de información meta "cache-control" que podeos utilizar.

```
<meta http-equiv="cache-control" content="no-cache" />
```

Por ejemplo en la línea anterior se indica que la página no debe ser cacheada por los exploradores Web y tiene que ser descargada. Sin embargo hay que aclarar que en muchos browsers lo que hace es guardarla y cada vez que se quiere ver la página, primero chequea en el servidor si hay alguna modificación o actualización de la misma y si no la hay la abre directamente de la carpeta de archivos temporales de Internet. Para que directamente no se guarde la página hay que escribir:

```
<meta http-equiv="cache-control" content="no-store" />
```

La línea anterior indica que no debe cachear la página ni almacenarla; por lo tanto cada vez que se quiere abrir la página, la debe descargar del servidor. Sin embargo la mejor opción para no cachear una página es utilizar los dos valores:

```
<meta http-equiv="cache-control" content="no-cache, no-store" />
```

Por el contrario si lo que queremos es que la página sea cacheada, tenemos las siguientes opciones que se dan a continuación:

```
<meta http-equiv="cache-control" content="private" />
```

En la línea anterior se indica que la página puede ser cacheada en los browsers de los usuarios. Si se quiere que sea cacheada tanto por los browsers como por los proxys de los ISP de manera pública, hay que escribir la siguiente línea meta:

```
<meta http-equiv="cache-control" content="public" />
```

Mensaje meta pragma

En versiones más viejas de browsers, se utilizaba para indicar que la página no debía ser cacheada el mensaje meta "pragma".

```
<meta http-equiv="pragma" content="no-cache" />
```

Sin embargo en nuevas versiones de browsers ya no es tan utilizada y se la pasa por alto, siendo más útil "cache-control". También como otra opción se pueden utilizar de manera combinada tanto el meta "pragma" así como "cache-control".

Mensaje meta refresh

Con el mensaje meta "refresh" se puede hacer que la página se actualice cada cierta cantidad de segundos automáticamente, o sea que se recarga sola sin necesidad de que el usuario presione F5 o el botón del browser de "actualizar". Además se puede hacer que después de pasados los segundos indicados en lugar de recargarse la misma página se abra una página en otra dirección. Por lo general dicha modalidad es utilizada por administradores de sitios que se han mudado de dirección, para que los usuarios que entran a la dirección vieja sean redireccionados después de pasados unos segundos. Pero hay que tener en cuenta que a los buscadores no les gustan mucho las páginas que se actualizan o recargan automáticamente, resultando en puestos más bajos de búsqueda, por lo tanto hay que tratar de evitar "refresh".

```
<meta http-equiv="refresh" content="40" />
```

En la línea anterior se indica que la página debe ser refrescada (refresh) o recargada cada 40 segundos. La cifra siempre debe indicarse en segundos. Por el contrario si se quiere redireccionar al usuario después de pasados unos segundos:

```
<meta http-equiv="refresh" content="40; URL=http://www.youbioit.com" />
```

En la línea anterior se indica que después de pasados 40 segundos se debe redireccionar al browser a la dirección indicada.

Otra cosa por la que se desaconseja la utilización de "refresh" es porque a muchos les molesta que la página se recargue sola.

Hasta ahora hemos visto mensajes meta definidos con el atributo http-equiv, ahora veremos meta definidos con el atributo name.

Mensaje meta expires

El mensaje meta "expires" (expira) se utiliza para indicar en que fecha expira o caduca la página Web. Eso es para que los buscadores dejen de indexar la página y por lo tanto desaparezca de los resultados de las búsquedas que hagan los usuarios. En realidad no se recomienda utilizar el mensaje meta "expires" ya que tanto los buscadores como los exploradores Web de hoy día no le prestan atención. Para que los buscadores no indexen la página, se recomienda utilizar "cache-control".

```
<meta name="expires" content="Fri, 30 Dec 2011 20:00" />
```

Mensaje meta "robots"

El mensaje meta "robots" se utiliza para indicarles a los buscadores que acciones pueden tomar en la página. Entre las acciones se incluyen permitir indexar la página, permitir entrar a otras direcciones a través de los hipervínculos incluidos en la página, prohibir indexar la página, prohibir entrar a los hipervínculos de la página. Veamos los distintos mensajes meta "robots".

Permitido hacer todo (opción default):

```
<meta name="robots" content="all" />
```

Permitido indexar:

```
<meta name="robots" content="index" />
```

Permitido seguir hipervínculos:

```
<meta name="robots" content="follow" />
```

Prohibido indexar:

```
<meta name="robots" content="noindex" />
```

Prohibido indexar y seguir links:

```
<meta name="robots" content="noindex, nofollow" />
```

Como se puede ver en la última línea, también se pueden combinar las distintas opciones. El meta "robots" es muy útil si por ejemplo tenemos alguna página del sitio que no queremos que se vea en buscadores ya que puede pertenecer a una sección privada del sitio, como por ejemplo la página privada de perfil de usuario en una red social. De lo contrario es conveniente permitir a los buscadores tanto indexar las páginas como también permitir que entren a los links para que puedan indexar otras páginas del sitio, resultando en mejores posiciones de búsqueda, lo cual permite más visitas al sitio.

Mensaje meta googlebot

Google tiene su propio mensaje meta, similar a "robots", con el cual se puede indicar a Google qué se le permite hacer en la página.

```
<meta name="googlebot" content="index, follow" />
```

La línea anterior indica a Google que la página puede ser indexada y sus links pueden ser visitados por el buscador para indexar también las páginas hacia las que apuntan dichos hipervínculos. Así podrán estar en resultados de búsquedas de Google. Igualmente es bueno mencionar que dado que por omisión Google puede indexar la página así como seguir los links de la misma, indicarle que puede hacer dichas acciones es redundante, agregando una línea innecesaria al código.

En la siguiente línea le indicamos a Google que no puede indexar la página ni tampoco seguir los hipervínculos de la misma.

```
<meta name="googlebot" content="noindex, nofollow" />
```

Cada vez que el buscador de Google, también conocido como la araña de Google, visita un sitio; en lo posible pasa por todas las páginas que forman parte del mismo, utilizando los hipervínculos presentes en cada una. Además de indexar

las páginas, en el momento de cada visita toma una copia de la página, para que en caso de no estar disponible en algún momento por problemas técnicos o de servidor entre otros, el usuario podrá ver igualmente la página como se encontraba en el momento en que Google hizo una copia cache de la misma. Obviamente un mensaje indica en la parte de arriba, que se trata de una versión cache y no de la página actual. Si no queremos que Google tome una copia de la página cada vez que la visita, podemos indicarle que no lo haga mediante el siguiente mensaje meta de "googlebot" con valor "noarchive".

```
<meta name="googlebot" content="noarchive" />
```

Cada vez que realizamos una búsqueda, Google muestra en los resultados un fragmento del texto de la página que contiene las palabras incluidas en nuestra búsqueda, marcadas en negrita. Si no queremos que Google muestre en los resultados de búsquedas un fragmento del texto de la página hay que incluir el "googlebot" con valor "nosnippet" (no fragmento).

```
<meta name="googlebot" content="nosnippet" />
```

Como sucede con el meta "robots", en "googlebot" también se pueden combinar varios valores con indicaciones al buscador.

```
<meta name="googlebot" content="nofollow, nosnippet" />
```

Por ejemplo en la línea anterior se indica a Google que no puede seguir los links de la página ni tampoco dar fragmentos de texto en los resultados de búsqueda. Las combinaciones pueden ser variadas dependiendo de las necesidades del administrador.

Mensaje meta "keywords"

Quizá uno de los mensajes meta más utilizados por los desarrolladores Web es "keywords" (palabras clave). Con "keywords" se pueden agregar distintas palabras o frases que tengan relación con la temática del texto incluido en la página, las cuales pueden ser utilizadas o tomadas en cuenta por los distintos buscadores para promover la página a mejores posiciones de resultados de búsqueda. Así, si un usuario hace una búsqueda que coincida con la frase incluida en "keywords", es posible que nuestra página aparezca en una mejor posición de resultado de búsqueda, dado que dicha frase fue indexada por el motor de búsqueda. Por ejemplo si en una página se habla de electrónica podemos escribir:

```
<meta name="keywords" content="cursos de electronica, tutorial de electronica, transistores, capacitores, fabricación de circuitos integrados" />
```

En el ejemplo anterior se incluyen palabras y frases que pueden ser utilizadas por usuarios en búsquedas y que posiblemente por coincidencia hagan que nuestra página pueda ser incluida en posiciones de resultado de búsqueda más altas, siendo la primera página de resultados la más favorable, resultando en mayor cantidad de visitas. Las palabras y frases siempre tienen que estar relacionadas a la temática de la página, y preferentemente que sean exactamente iguales a palabras que aparecen en el texto de la página Web. Las palabras clave deben estar separadas por coma.

Mensaje meta "description"

Otro meta muy utilizado, es "description" el cual sirve para darles una descripción acerca de la página Web a los buscadores; para que dicha descripción aparezca en los resultados de búsquedas, como texto descriptivo sintético de la página.

```
<meta name="description" content="Página de datos acerca de componentes y dispositivos electrónicos; desde transistores hasta circuitos integrados." />
```

En el mensaje meta anterior se dio una descripción ficticia acerca de una supuesta página de electrónica. Se supone que en los resultados de búsqueda de algunos buscadores debería aparecer el texto del meta "description" debajo del título.

Mensaje meta "revisit-after"

Con "revisit-after" (revisitar después de) se le indica a los buscadores cada cuánto tienen que volver a visitar la página para indexar las actualizaciones. El período se marca en días, por lo tanto si se indican 20 días, los buscadores visitarán, revisarán e indexarán la página cada 20 días (days). Obviamente se trata de una sugerencia a los buscadores, lo que no significa que visitarán la página exactamente con la frecuencia que se indica en "revisit-after", algunos lo harán más seguido, otros lo harán con menor frecuencia y también habrán buscadores que sí respetarán el período indicado por nosotros.

```
<meta name="revisit-after" content="20 days" />
```

Mensaje meta "author"

Con "author" simplemente se especifica quién es el autor de la página Web, se trata de una especie de crédito al desarrollador.

```
<meta name="author" content="Mi nombre" />
```

Mensaje meta "generator"

Con "generator" se indica qué programa o herramienta se utilizó para generar la página. Si la diseñamos a mano sin la ayuda de algún diseñador visual, como Dreamweaver, no hace falta especificar nada, si por el contrario utilizamos un editor de páginas Web, automáticamente se agregará una línea con el meta "generator" con el nombre del programa utilizado.

```
<meta name="generator" content="Dreamweaver" />
```

Mensaje meta "copyright"

Con "copyright" se especifican los derechos de autor y protección contra copias. Hay que tener en cuenta que es importante siempre mostrar los derechos de autor en la parte visible de la página, sin embargo no está de más incluir el meta "copyright".

```
<meta name="copyright" content="© 2012 - El nombre de nuestra empresa" />
```

Mensaje meta "owner"

Con "owner" se indica el nombre del propietario del sitio Web y por ende de la página que se visita, a veces es el mismo que el autor.

```
<meta name="owner" content="Mi nombre" />
```

Veamos un ejemplo tomado de uno de los códigos fuente vistos anteriormente en el que agregamos algunos mensajes meta:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

<meta http-equiv="content-type" content="text/html; charset=UTF-8" />
<title>Automotores - El Cuervo de Boedo</title>
<meta name="description" content="Venta de automotores nuevos y usados. Somos la
concesionaria de automóviles más grande de la ciudad." />
<meta name="keywords" content="marcas de autos, automoviles, vehiculos automotores,
ford, chrysler" />
<meta http-equiv="cache-control" content="no-cache, no-store" />
<meta name="revisit-after" content="20 days" />
<meta name="owner" content="Automotores - El Cuervo de Boedo" />

</head>

<body>
<form action="www.ejemplo.com/formulario-de-contacto.php" method="POST">
<p>Seleccione una marca de autos</p>
<select name="marcas">
<option value="Ford">Ford</option>
<option value="Chevrolet">Chevrolet</option>
<option value="Chrysler">Chrysler</option>
<option value="Buick">Buick</option>
<option value="Cadillac">Cadillac</option>
</select>
```

```

<p><input type="submit" value="Enviar" /> </p>

</form>

</body>
</html>

```

Ya hemos estudiado todo HTML y analizamos todos los elementos y etiquetas del lenguaje. Pero como ya se comentó anteriormente, para complementar los conocimientos de diseño y desarrollo Web adquiridos en el actual curso es necesario continuar aprendiendo otros lenguajes, comenzando con CSS y continuar con Javascript y aplicaciones como Flash, Photoshop y Paint Shop Pro. Después de aprender bien esos lenguajes y herramientas; si estamos convencidos de querer dedicarnos profesionalmente al desarrollo Web, podríamos continuar estudiando un lenguaje para la creación de sitios más complejos; como redes sociales, sitios de ventas online, buscadores, etc. El lenguaje más popular para la creación de sitios de ese tipo es PHP el cual debe ir acompañado de algún tipo de base de datos que debemos aprender a administrar. Sabiendo todo lo anteriormente mencionado podemos dedicarnos profesionalmente al desarrollo Web.

ANEXO - CARACTERES ESPECIALES CODIGOS O SECUENCIAS DE ESCAPE HTML

Símbolo	Número de Secuencia de Escape	Secuencia de escape	Nombre
"	"	"	Comillas
'	'	'	Apóstrofe
&	&	&	Ampersand
<	<	<	Menor a
>	>	>	Mayor a
	 	 	Espacio
¡	¡	¡	Signo de exclamación invertido
¢	¢	¢	Centavo
£	£	£	Pound
¤	¤	¤	Moneda, signo monetario genérico
¥	¥	¥	Yen
	¦	¦	Barra vertical separada
§	§	§	Sección
¨	¨	¨	Diéresis
©	©	©	Copyright - Derechos de autor
ª	ª	ª	Indicador de género femenino
«	«	«	Comillas bajas angulares (izq)
¬	¬	¬	Negación

®	®	®	Marca registrada
—	¯	¯	Guión alto
°	°	°	Grados
±	±	±	Signo mas menos
²	²	²	Superíndice 2 (al cuadrado)
³	³	³	Superíndice 3 (al cubo)
´	´	´	Acento agudo
μ	µ	µ	micro
¶	¶	¶	Párrafo
·	·	·	Punto medio
¸	¸	¸	Cedilla
¹	¹	¹	Superíndice 1
º	º	º	Indicador de género masculino
»	»	»	Comillas bajas angulares (derecha)
¼	¼	¼	Fracción 1/4
½	½	½	Fracción 1/2
¾	¾	¾	Fracción 3/4
¿	¿	¿	Signo de interrogación invertido
×	×	×	Multiplicación
÷	÷	÷	División
À	À	À	A mayúscula, acento grave
Á	Á	Á	A mayúscula, acento agudo
Â	Â	Â	A mayúscula, acento circunflejo
Ã	Ã	Ã	A mayúscula, con tilde
Ä	Ä	Ä	A mayúscula, con diéresis
Å	Å	Å	A mayúscula, con anillo
Æ	Æ	Æ	Diptongo ae en mayúscula
Ç	Ç	Ç	C mayúscula, con cedilla
È	È	È	E mayúscula, acento grave
É	É	É	E mayúscula, acento agudo
Ê	Ê	Ê	E mayúscula, acento circunflejo
Ë	Ë	Ë	E mayúscula, con

			diéresis
Ì	Ì	Ì	I mayúscula, acento grave
Í	Í	Í	I mayúscula, acento agudo
Î	Î	Î	I mayúscula, acento circunflejo
Ï	Ï	Ï	I mayúscula, con diperesis
Ð	Ð	Ð	Letra eth mayúscula
Ñ	Ñ	Ñ	Letra Ñ mayúscula
Ò	Ò	Ò	O mayúscula, acento grave
Ó	Ó	Ó	O mayúscula, acento agudo
Ô	Ô	Ô	O mayúscula, acento circunflejo
Õ	Õ	Õ	O mayúscula, con tilde
Ö	Ö	Ö	O mayúscula, con diéresis
Ø	Ø	Ø	O mayúscula, con barra inclinada
Ù	Ù	Ù	U mayúscula, acento grave
Ú	Ú	Ú	U mayúscula, acento agudo
Û	Û	Û	U mayúscula, acento circunflejo
Ü	Ü	Ü	U mayúscula, con diéresis
Ý	Ý	Ý	Y mayúscula, acento agudo
Þ	Þ	Þ	Letra Thorn mayúscula
ß	ß	ß	Letra eszett
à	à	à	Letra a, acento grave
á	á	á	Letra a, acento agudo
â	â	â	Letra a, acento circunflejo
ã	ã	ã	Letra a, con tilde
ä	ä	ä	Letra a, con diéresis
å	å	å	Letra a, con anillo
æ	æ	æ	Diptongo ae
ç	ç	ç	Letra c, con cedilla
è	è	è	Letra e, acento grave
é	é	é	Letra e, acento agudo
ê	ê	ê	Letra e, acento

			circunflejo
ë	ë	ë	Letra e, con diéresis
ì	ì	ì	Letra i, acento grave
í	í	í	Letra i, acento agudo
î	î	î	Letra i, acento circunflejo
ï	ï	ï	Letra i, con diéresis
ð	ð	ð	Letra eth
ñ	ñ	ñ	Letra ñ
ò	ò	ò	Letra o, acento grave
ó	ó	ó	Letra o, acento agudo
ô	ô	ô	Letra o, acento circunflejo
õ	õ	õ	Letra o, con tilde
ö	ö	ö	Letra o, con diéresis
ø	ø	ø	Letra o, con barra inclinada
ù	ù	ù	Letra u, acento grave
ú	ú	ú	Letra u, acento agudo
û	û	û	Letra u, acento circunflejo
ü	ü	ü	Letra u, con diéresis
ý	ý	ý	Letra y, acento agudo
þ	þ	þ	Letra thorn
ÿ	ÿ	ÿ	Letra y, con diéresis

<https://web.archive.org/web/20160314210215/http://www.youbioit.com/es/article/shared-information/2601/curso-de-html-primera-parte>

<https://web.archive.org/web/20160314164530/http://www.youbioit.com/es/article/shared-information/4129/curso-de-html-segunda-parte>

<https://web.archive.org/web/20160314235807/http://www.youbioit.com/es/article/shared-information/4971/curso-de-html-tercera-parte>

<https://web.archive.org/web/20160315000311/http://www.youbioit.com/es/article/shared-information/5340/curso-de-html-cuarta-parte>