

00324

25



UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO

FACULTAD DE CIENCIAS

HTML Y JAVASCRIPT UNA HERRAMIENTA PARA LA CREACIÓN DE PÁGINAS WEB

T E S I S

QUE PARA OBTENER EL TÍTULO DE

ACTUARIA.

P R E S E N T A :

JASMIN ELIZABETH GAMAS VELASCO

DIRECTOR DE TESIS: Act: RICARDO HUMBERTO SEVILLA AGUILAR



FACULTAD DE CIENCIAS UNAM

DIVISION DE ESTUDIOS PROFESIONALES



FACULTAD DE CIENCIAS SECCION ESCOLAR

TESIS CON FALLA DE ORIGEN

6
1



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

PAGINACIÓN

DISCONTINUA



GOBIERNO NACIONAL
SECRETARÍA DE EDUCACIÓN PÚBLICA

DRA. MARÍA DE LOURDES ESTEVA PERALTA
Jefa de la División de Estudios Profesionales de la
Facultad de Ciencias
Presente

Comunicamos a usted que hemos revisado el trabajo escrito:

"HTML y JAVASCRIPT una herramienta para la creación de páginas web."

realizado por JASMIN ELIZABETH GAMAS VELASCO

con número de cuenta 9533507-2 ,quien cubrió los créditos de la carrera de ACTUARÍA

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis
Propietario

Act. RICARDO HUMBERTO SEVILLA AGUILAR

Propietario

Act. FERNANDO ELEAZAR VANEGAS CHAVEZ

Propietario

Dr. HERMILO SANCHEZ CRUZ

Suplente

Act. DAVID LOPEZ SERVIN

Suplente

M. en C. MARIA GUADALUPE ELENA
IBARGÜENGOITIA GONZALEZ

Consejo Departamental de MATEMÁTICAS

M. en C. JOSE ANTONIO FLORES DIAZ

FACULTAD DE CIENCIAS

CONSEJO DEPARTAMENTAL

DE MATEMÁTICAS

1-A

INDICE

Introducción

1. Capítulo 1 Aspectos Generales

1.1 Antecedentes

1.1.1 ¿Qué es Internet y cómo funciona?

1.1.2 El Internet en México

1.2 ¿Que es html?

1.3 Hardware

1.4 Accesibilidad

1.5 ¿Cómo publicar una página?

1.5.1 ¿Cómo encontrar un host?

1.5.2 ¿Cómo conseguir su propio nombre de dominio?

1.5.3 ¿Qué hacer para que visiten su página?

2. Capítulo 2 Html

2.1 Html

2.2 Creación de párrafos

2.3 Formato del texto

2.4 Fondos

2.5 Color del texto

2.6 Caracteres especiales

2.7 Imágenes

2.8 Comentarios

2.9 Creación de enlaces

2.10 Listas

2.11 Tablas

2.12 Formularios

2.13 Texto en movimiento

3. Capítulo 3 JavaScript

3.1 ¿Qué es JavaScript?

3.2 Diferencias entre java y JavaScript

3.3 JavaScript y html

3.4 Elementos básicos

3.5 Tipos de datos

3.6 Operadores

3.7 Estructuras de control

3.8 Funciones

- 3.9 Funciones predefinidas
- 3.10 Objetos
- 3.11 Eventos
- 3.12 Jerarquía
- 3.13 Objetos especiales

4. Capítulo 4 Creación De La Tienda Virtual

- 4.1 Creación De La Tienda Virtual Arma Tu Pc

Conclusiones

Apéndice Manual de la tienda virtual

Referencias

Bibliografía

AGRADECIMIENTOS

**GRACIAS A DIOS, A MIS PADRES, A HARIM QUE
ME INICIO EN ESTE TEMA Y A TI RICARDO QUE
ERES COMO MI ANGEL DE LA GUARDA.**

INTRODUCCIÓN

Desde hace ya algunos años, se viene comentando lo relevante que resulta la utilización de la computadora como herramienta en distintos ámbitos de la vida cotidiana, ahora ese poder que se brinda se multiplica por medio de Internet al poder conectar una computadora a millones de computadoras en el mundo, sin importar donde estén localizadas y utilizar la información almacenada en todas éstas.

Actualmente, Internet cuenta con más de 350 millones de usuarios en todo el mundo y el número sigue aumentando diariamente, debido a la facilidad que brinda Internet para obtener información desde la comodidad de su casa u oficina. Este número creciente de usuarios puede estar formado por clientes potenciales para una empresa o negocio.

En Internet, cada día también aumenta la cantidad de productos y servicios que se comercializan por este medio, facilitando conseguir mejores precios y nuevos proveedores, mejorando así los márgenes de utilidad.

Es sabido el auge y la proliferación de las computadoras personales, los centros de Internet, los cibercafé, todo con dedicación a navegar e informarse de lo que sucede en cualquier lugar del mundo.

Internet, nos da acceso a la información en forma rápida, de primera fuente y en cualquier momento, sin importar día, hora o lugar del planeta.

HTML y JavaScript son lenguajes con los cuales se crean páginas para Internet, son lenguajes fáciles de aprender y no necesitan que el lector tenga conocimientos previos de programación.

Este trabajo tiene como propósito auxiliar al usuario de Internet para elaborar páginas web, se presenta un manual didáctico de HTML (Hypertext Markup Language) y JavaScript. Didáctico porque por lo regular se presentan manuales de HTML separados de los de JavaScript por ejemplo los manuales de HTML como "diseño creativo html.2 manual de diseño práctico para Internet" [38] y "HTML creación y difusión de páginas web" [32] dan la definición de lo que es JavaScript, mencionan en qué parte del código HTML se recomienda insertar el código de JavaScript, refieren a JavaScript como una manera de realizar páginas web con mayor interacción con los usuarios y dan algunos ejemplos sencillos de JavaScript; "Páginas web mágicas" [33], menciona algo sobre las estructuras de control, y da ejemplos de algunos eventos, los manuales de JavaScript como "Aprendiendo JavaScript en una semana" [28] y "Netscape Developer's guide to JavaScript 1.2" [30] se limitan a mencionar en qué parte del código HTML va el código de JavaScript y simplemente usan el código dando por hecho que el lector ya sabe HTML; "JavaScript manual of style" [39] y "Diseño de

páginas web interactivas con JavaScript" [37] aparte de mencionar en donde va el código, explican de manera básica la manera de realizar una página con HTML.

Ésta es una tesis de Actuaría que puede ser útil para cualquier actuario o persona que desee tener su propia página web, ya sea personal o viéndola como una manera de facilitarle la vida. Como aplicación práctica, se crea una tienda virtual utilizando los lenguajes ya mencionados en donde se describen los pasos para su manejo, se subió a Internet y se encuentra en <http://www.angelfire.com/amiga/actuaría/armapc2.html>.

En el capítulo 1 se dan algunos antecedentes de Internet y de HTML, se definen las características necesarias que debe tener una página web, se explica cómo publicarla y se mencionan algunas características que debe tener el hardware, ya que éste afecta la manera en la que cargan las páginas en la computadora.

En el capítulo 2 se construye un manual sobre HTML, describe cómo crear una página, además se explica cómo llevar a cabo las siguientes acciones: cambiar el color de fondo, cambiar el color y tipo de fuente, insertar imágenes, crear vínculos, hacer tablas, formularios, listas, caracteres especiales y demás.

En el capítulo 3 se diseña un manual de JavaScript, las estructuras de control, los eventos, funciones y cómo se combinan HTML y JavaScript.

En el capítulo 4 se explica paso a paso como fue creada la aplicación práctica, que es una tienda virtual llamada "Arma tu PC".

¿Por qué crear páginas web? Porque la web se está convirtiendo en una parte fundamental del mundo laboral y de entretenimiento.

CAPITULO UNO

ASPECTOS GENERALES

1.1 ANTECEDENTES

A fines de los años 60 el Departamento de Defensa de los EE.UU. crea Arpanet la primera red de datos del mundo, la cual enlazaba a varios centros de investigación. El gran éxito de Arpanet fue cuando en 1980 se consolidó el *protocolo* TCP/IP. Arpanet completó su transición al TCP/IP en 1983, además en este año Arpanet se dividió en dos sistemas diferentes llamados Arpanet y Milenet. Arpanet es entregada para uso de la comunidad y Milenet para uso estrictamente militar.

En 1986 la National Science Foundation (NSF) de Estados Unidos, comenzó el desarrollo de NSFnet, para conectar los centros de súper computación de la nación. Al poco tiempo se fue requiriendo de un mayor ancho de banda, de esta forma la NSF dio origen a la Merit, una unión entre MCI e IBM, con el fin de desarrollar un cerebro de 1,5 Mbps para NSFnet, desplazando de esa forma a la Arpanet. Luego se creó una nueva corporación llamada ANS, la cual sirvió como encargada del desarrollo de la banda ancha de NSFnet, llegando a 45Mbps (mega bites por segundo), la red TCP/IP más veloz del mundo. Posteriormente en 1990 nace la Internet con el fin de llegar a todo el mundo y compartir conocimientos e investigaciones. [3]

Desde entonces, se han desarrollado varios lenguajes y protocolos evolucionados dentro de la estructura creciente de Internet. Un **protocolo** es un conjunto de reglas que la computadora utiliza para comunicarse a través de las redes. Por ejemplo, telnet, ftp, http, gopher, tcp/ip, ppp, slip, smtp y uucp [7].

1.1.1 ¿Qué es Internet y cómo funciona?

Internet es, simplemente, un grupo de computadoras que están conectadas entre sí, el cual tiene como fin, prestar servicios a los usuarios ya sean comerciales o gratuitos.[34] Entre los principales y más conocidos servicios podemos encontrar el correo electrónico (e-mail), la llamada web y el chat.

Internet no es más que una acción denominada cliente/servidor, en donde el cliente es una computadora común y el servidor es una computadora que aloja información, la cual es solicitada por el cliente. Una computadora cliente es conectada principalmente a Internet por medio de un modem y una línea telefónica, de tal forma que se comunica con su respectivo ISP, es decir su proveedor de servicios de Internet, una vez conectado al proveedor es quien manja el servicio de Internet, haciendo posible las solicitudes de información a los servidores de todo el mundo, los ISP'S se comunican con otros ISP'S alrededor del mundo, antes pasando por *routers*, los cuales elegirán cual es el camino más apropiado por el cual tienen que pasar las peticiones. Los ISP'S se comunican entre sí por medio de conexiones de fibra óptica llamados *backbones*, éstos están conectados alrededor del mundo a través de cables submarinos o

vía conexiones satelitales. De esta forma los servidores conectados a un ISP pueden darse a conocer por todo el planeta.

Para ser un ISP se necesita un equipo que soporte el software necesario ya que hay que instalar en el equipo un servidor WWW (Por ejemplo: AnalogX, Apache, PHP, Mysql), un servidor FTP (Por ejemplo: WarFTP), un servidor de correo (Ej. Agsmail, smtpBeamer), un servidor de noticias (nDaemon), un servidor de IRC (Hyperchat), un servidor proxy (AnalogX), un servidor telnet (telnetXQ); estos servidores son el software que necesita una computadora para ser servidor [9].

El servidor **WWW** es una aplicación que procesa las peticiones de un navegador WWW, es decir, entrega páginas WWW a través de Internet.

El servidor **FTP** sirve para transferir archivos.

El servidor de **Correo** consiste en tramitar su nombre de dominio propio para Internet de esta manera todas las cuentas de correo se pueden configurar con su dominio. Por ejemplo: ventas@sucmpresa.com.

El Servidor de **Noticias** es capaz de construir un periódico individualizado para cada lector basándose en las peticiones que éste formula.

El servidor de **IRC** permite comunicarse en tiempo real a dos o más personas a través de sus computadoras.

Un Servidor **Proxy** permite que varios ordenadores conectados a una misma red local puedan compartir un mismo acceso a Internet o conexión a Internet de manera simultánea.

Telnet: Es otra de las aplicaciones de Internet. Permite conectarse a ordenadores remotos para ejecutar sus programas y consultar su información.

La forma en que viajan las peticiones e información es gracias a los protocolos, el más usado es el TCP/IP, el cual se divide en dos partes:

TCP: Este tiene la misión de dividir los datos en paquetes, entregándoles un nombre de cabecera (en este caso es un número) y el denominado *suma de comprobación*, que coincide con el número total de datos que contiene el paquete, estas dos características sirven para que cuando los paquetes lleguen a su destino se vuelvan a juntar correctamente.

IP: Cada paquete de información cuenta con la dirección IP tanto de la computadora que lo envió como de la que recibe el paquete. Una dirección IP son 4 números separados por puntos.

Los cuatro números en la dirección IP se llaman octetos porque pueden tener valores entre el 0 y el 255 (2⁸ posibilidades por octeto).

Una vez que la información pedida llega a nuestra computadora, el protocolo TCP, vuelve a la acción con el objetivo de ordenar los paquetes, realizando una nueva suma de comprobación y la compara con la suma original. Si alguna de ellas no coincide, detectándose así pérdida de información en el trayecto, se solicita de nuevo el envío del paquete desde el origen.

Cada computadora conectada a Internet cuenta con una dirección IP, las cuales pueden ser estables o dinámicas. Los usuario comunes y corrientes por lo general tienen una IP dinámica, esto quiere decir que cada vez que se conecta a Internet su ISP le otorgará una nueva IP; Las IP estables son aquellas asignadas a una computadora y no cambian. Normalmente son usadas por servidores web o máquinas que deban estar conectadas a la red de forma permanente.

Pero el único problema de estas direcciones IP'S son lo complicado de memorizar, por esa razón se crearon los nombres de dominio que son mucho más fácil de recordar que las IP'S, un ejemplo de nombre de dominio es: www.0bytes.cl, la dirección IP de este dominio es 216.64.77.89.

Un nombre de dominio se divide en tres partes:

www --> Tipo de servicio (telnet, ftp, gopher, http).

0bytes --> Nombre del Dominio.

cl --> El tipo de entidad [3].

1.1.2 El Internet en México

1989 México se conecta al Internet por primera vez; el Instituto Tecnológico y de Estudios Superiores de Monterrey (Campus Monterrey) ITESM se conecta hacia la Universidad de Texas en San Antonio (UTSA), por medio de una línea dedicada analógica a 9600 bps; siendo el primer nodo de Internet en este país.

Se conecta la Universidad Nacional Autónoma de México (UNAM) vía satélite de 56 Kbps a través del Instituto de Astronomía en la Ciudad de México con el Centro Nacional de Investigación Atmosférica (NCAR) de Boulder, Colorado, en los Estados Unidos.

1992 Enero - surge MEXnet, formada por:

ITESM, Universidad de Guadalajara, Universidad de las Américas,

ITESO, Colegio de Postgraduados, LANIA, CIQA, Universidad de Guanajuato, Universidad Veracruzana, Instituto de Ecología, Universidad Iberoamericana, IT de Mexicali

Junio - MEXnet establece una salida digital de 56kbps.

1993 El CONACYT y el ITAM se conectan a Internet; existían ya una serie de Redes en el País: MEXnet, Red UNAM, Red ITESM, BAJAnet, Red Total CONACYT, RUTyC (desapareció ese mismo año), SIRACyT

1994 Se forma la Red Tecnológica Nacional (RTN), Backbone nacional integrado por MEXnet y CONACYT y agrupando a un gran número de instituciones educativas y comerciales en toda la República, desde Baja California hasta Quintana Roo.

Internet se abre a nivel comercial en nuestro país, ya que hasta entonces, solamente instituciones educativas y de investigación podían realizar su enlace a Internet.

1995 Diciembre - Se crea el Centro de Información de Redes de México (NIC-México) el cual se encarga de la coordinación y administración de los recursos de Internet asignados a México, tales como la administración y delegación de los nombres de dominio ubicados bajo MX.

1996 El INFOTEC crea el Centro de Tecnologías Avanzadas cuyo objetivo es desarrollar servicios de contenido de valor agregado en Internet y todos aquellos que se desarrollan con nuevas tecnologías del Intranet y de Multimedia

Nace la Sociedad Internet, Capítulo México, una asociación internacional no gubernamental no lucrativa para la coordinación global y cooperación en Internet.

1997 Existen más de 150 Proveedores de Acceso a Internet (ISP'S) que brindan sus servicios en el territorio mexicano, ubicados en los principales centros urbanos: Cd. de México, Guadalajara, Monterrey, Chihuahua, Tijuana, Puebla, Mérida, Nuevo Laredo, Saltillo y Oaxaca, por mencionar sólo algunos.

Los detalles de la información anterior se pueden encontrar en la referencia [8].

La Word Wide Web es una colección de cientos de millones de documentos en Internet y escritos en HTML.

Es importante recordar que la Web es sólo una parte de todo Internet. Mucha gente piensa que la web e Internet es la misma cosa, pero no están en lo cierto. Todos los protocolos de Internet permiten la creación de una amplia gama de aplicaciones basadas en Internet, incluyendo el ya familiar correo electrónico y los populares grupos de noticias. La World Wide Web es uno de estos protocolos, y el Hypertext Markup Language (HTML) o Lenguaje de marcas de hipertexto, es sólo uno de los muchos lenguajes usados para ofrecer información vía Internet.

En su expresión más temprana, el HTML era un lenguaje de formateado. Toda su labor era simplemente administrar el formateado básico de los documentos de hipertexto. Los controles de formato se limitaban a elementos como cabeceras y párrafos, y el texto se limitaba a poder formatearse en negrita o cursiva.

El HTML en su versión actual, 4.0 añade una completa gama de potentes opciones al lenguaje original, llevándolo desde un limitado formateo a una herramienta de diseño avanzado. Por ejemplo: **Background** (pone una imagen como fondo).

La web empezó como un entorno basado en hipertexto que se podía visualizar sólo por navegadores de texto. Pero en 1993, Marc Andreessen y Eric Bina desarrollaron un interfaz gráfico de usuario (*GUI, Graphic User Interface*) en el Centro Nacional de Aplicaciones de Supercomputación (*NCSA, National Center for Supercomputing Applications*). Uno de los factores más importantes en el lanzamiento de Internet al gran uso comercial ha sido el navegador Mosaic (creado por Marc Andreessen en 1993). Su interfaz era fácil de usar, y el hecho de que pudiera mostrar gráficos era una característica muy atractiva [8].

Un navegador es un programa que permite interactuar con recursos de Internet (www, ftp etc).

La llegada de Mosaic literalmente creó la oportunidad para el nacimiento de la Web como un medio gráfico. Desde esta base creció la enorme y popular industria que tenemos hoy día.

Después de que Mosaic entrara en escena, era inminente la llegada de un nuevo navegador: Cello, también conocido como *Netscape Navigator*, algo muy similar a Mosaic. *Netscape* lanzó esta primera versión de *Navigator* en 1994, tan sólo un año después que apareciera Mosaic. *Netscape* rápidamente se colocó con la delantera en la carrera de los navegadores debido a sus características nuevas en constante evolución y a la posibilidad de funcionar entre muchas plataformas (Unix y Windows). Una plataforma se refiere al tipo de interfaz de usuario que utiliza la computadora [4].

En esencia, *Netscape* había heredado la visión de Mosaic, y, más aún, una invitación a ser comercializado. El potencial de la navegación por la web llegó y continúa avanzado.

Sin esperar, *Microsoft* siempre interesada en Internet como un medio, y pensando cómo crear aplicaciones integradas de software, con un apogeo económico, una enorme infraestructura de soporte técnico, y recursos de desarrollo sin igual, hizo su entrada en el mercado de los navegadores.

Tras obtener el código primitivo de Spry Mosaic, evaluado en las convenciones de *Netscape*, y añadiendo características tecnológicas para hacer del interfaz algo propio, el navegador *Microsoft Internet Explorer* había nacido. *Microsoft* luego le añadió un fuerte y funcional soporte que incluía soporte de vídeo, de sonido de fondo, marquesinas horizontales, y un gran número de nuevas etiquetas de HTML.

Tanto *Netscape* como *Microsoft* siguieron compitiendo de forma agresiva, cada uno intentaba superar al otro en términos de funcionalidad, interacción entre plataformas, y nuevas tecnologías. Y, por supuesto, la meta última era dominar el mercado.

Actualmente el navegador Web más utilizado, es el *Microsoft Internet Explorer* de *Microsoft*, casi un 80% de los navegantes lo utilizan, frente a tan sólo un 20% que usa el *Navigator* de *Netscape*. Esto ha sido influenciado en gran medida por el hecho que *Microsoft* distribuye su navegador junto con el sistema operativo Windows, mientras que si se quiere utilizar el otro navegador, tenemos que molestarnos en ir a la página Web de *Netscape* y descargarlo[4].

1.2 ¿QUÉ ES HTML?

El HTML es un lenguaje de *scripting* usado en principio para crear páginas web. Es inseparable del concepto y la identidad referencial al World Wide Web. El marco de trabajo de su invención y creación han sido atribuidos a *Tim Berners-Lee* [31], un programador del Centro Europeo para Física de Partículas (CERN). *Berners-Lee* desarrolló el HTML para:

- Ofrecer un medio que permitiera a los científicos publicar, buscar y recibir información 24 horas al día.
- Crear un lenguaje internacional de codificación en ordenadores que facilitara el acceso universal independientemente de la plataforma, red, o terminal.

Desde el punto de partida de ofrecer una solución específica para la comunidad científica, el HTML originalmente ofreció acceso a través de un entorno basado en texto y facilitó el cambio de información.

HTML es el lenguaje internacional y de acceso universal. Estos aspectos, originalmente utilizados para enlazar papeles a referencias, han lanzado la Web al estado de "meta-medio". Esto es la Web: un medio que abarca todo y que soporta

múltiples formas de medio como texto, gráficos, y una variedad de objetos que pueden ser enlazados a otras páginas u objetos.

Los orígenes del HTML son como un lenguaje de codificación específico para aplicaciones basado en el modelo *Standard Generalized Markup Language* (SGML). Como un lenguaje de formateado, el HTML utiliza las declaraciones SGML y la definición de tipos de documento (DTD, *definition type document*).

El HTML es el lenguaje que permite este proceso desde el lado del cliente (o usuario final). El modo de ofrecer el HTML es mediante HTTP, (Protocolo de Transferencia de Archivos de Hipertexto). Protocolo que enlaza, a través de Hipervínculos, las páginas de Hipertexto (HTML) que forman la World Wide Web. El Protocolo HTTP consiste en un conjunto de reglas que se aplican a las peticiones que hace un cliente o navegador y a las respuestas que entrega un servidor.

1.3 HARDWARE

El hardware es la parte física de la computadora (teclado, monitor, ratón, disco duro, tarjeta madre etc.).

Nuestra computadora para funcionar óptimamente no sólo debe estar provista con los mejores y más nuevos programas del mercado, sus componentes físicos también deben ser atendidos con cierta regularidad, y actualizados con una cierta frecuencia para mantenerla al día con las nuevas normas tecnológicas.

El hardware afectará a la manera en que se carguen tus páginas.

- **Monitores, tarjetas de vídeo, y RAM de vídeo.** Estos tres componentes de hardware funcionan conjuntamente para crear la señal visual. Si un visitante de la web tiene un hardware de imagen muy limitado, no será capaz de ver de manera correcta su sitio Web.
- **Soporte audio, vídeo y multimedia.** Con el creciente interés de la multimedia basada en Web, el deber de los diseñadores HTML es tener en mente que el usuario final posee o no el hardware capaz de manejar audio, vídeo, y un sinnfín de aplicaciones multimedia.
- **Memoria RAM y espacio en disco duro.** La RAM es esencial para dar velocidad a los navegadores y aumentar así su aprovechamiento. Los navegadores también dependen de la *cache*, el cual es un método de almacenamiento de los gráficos y otros datos en el disco duro del visitante. Este proceso ayuda así a acelerar la muestra en la pantalla de la computadora la información basada en HTML.

- **Acceso y ancho de banda.** Mientras que un creciente número de fanáticos de la Web están conectados a Internet por líneas tremendamente rápidas, a través de tecnologías avanzadas de ancho de banda, la mayoría de la gente en sus hogares utiliza módems analógicos que marcan a velocidades desde 14.4 Kbps (kilo bites por segundo) a 56 Kbps. El ancho de banda es un tema de gran importancia cuando se diseñan los sitios y prácticamente dicta todas las decisiones de usar uno u otro navegador y una u otra plataforma.

Los detalles de la información anterior pueden ser consultados en la referencia [27].

1.4 ACCESIBILIDAD

Lo primero para empezar es decidir qué información debería ser hecha accesible y qué información puede quedar relegada al entorno más avanzado. Una vez que la decisión esté realizada, se puede diseñar un código HTML apropiadamente, o elegir descargar archivos de texto desde el sitio Web como una alternativa.

Las siguientes guías harán tus páginas Web más accesibles.

- **Facilidad de lectura.** Esto es asegurar que una página Web es legible en un entorno de sólo texto.
- **Claridad del contexto.** Un importante objetivo tanto en el acceso HTML y en el diseño, es la idea de que todos y cada uno de los elementos de una página tienen que tener alguna justificación del lugar donde estén. La razón no siempre tiene que ser funcional, también puede ser estética. Pero no añadas una tecnología específica sólo porque tú sabes escribir el código. Elige y escoge únicamente las que son apropiadas para el contexto de tu trabajo.
- **Funcionalidad.** Los formularios de pedidos, los carritos de la compra, y las búsquedas deben ser accesibles, o si no puede ser, deben ser ofrecidas algunas alternativas más sencillas. Si estoy visitando tu sitio Web con un módem de 14.4 Kbps desde un país muy lejano con tecnología limitada y un navegador de sólo texto, puedes apostar a que voy a verme frustrado si intento rellenar un formulario de un producto que requiero para hacer mi trabajo. Asegurar que existen alternativas a los formularios y otros dispositivos alternativos en un sitio es bastante simple de hacer y ayudará a este tipo de usuarios en gran medida.

1.5 CÓMO PUBLICAR UNA PÁGINA

Una vez terminada una página se tiene que transferir a un servidor web de modo que la gente pueda acceder a ella.

Un servidor web o **host** no es otra cosa que una computadora conectada las 24 horas a Internet con software y hardware especial.

Cada vez que alguien visita una página web, el navegador debe conectarse a una computadora a la que le debe pedir la página solicitada.

La computadora que atiende estos pedidos y distribuye las páginas es el servidor web.

El término servidor también puede referirse al programa que corre en ese servidor para atender dichas peticiones [34].

1.5.1 ¿Cómo encontrar un host?

A no ser que tenga su propio servidor, se tendrá que pagar a una de miles de empresas que la hacen de host (Por ejemplo Terra), la mayoría cobra una cantidad mensual que depende de los servicios que ofrecen.

Algunos ofrecen sus servicios gratuitos a cambio de anunciarse en su página. (Por ejemplo: <http://angelfire.lycos.com/>)

Antes de elegir un host se deben considerar los siguientes detalles:

- ¿Cuánto espacio en disco necesitará la página? No pague más de lo que necesita.
- ¿Ofrecen soporte técnico? ¿Por teléfono o por correo electrónico?
- ¿Es rápida la conexión a Internet? Esto determinará la rapidez con que tus páginas se ofrecerán a los visitantes. ¿Cuenta con conexiones múltiples para el caso en que una de ellas no funcione?
- ¿Ofrecen contador de visitantes? [34]

1.5.2 ¿Cómo conseguir su propio nombre de dominio?

Generalmente, una dirección de página web está constituida por el nombre de un servidor junto con la ruta de acceso al archivo ubicado en ese servidor.

Ejemplo: si el nombre del servidor web es www.servidor.com la dirección de las páginas debe ser algo así como www.servidor.com/~mipagina.

Si no quiere que el nombre de su host web aparezca en la dirección, puede registrar su propio nombre de dominio (por una tarifa) y pedirle a su host que cree un dominio virtual en su servidor con su nombre de dominio, por ejemplo si registra el nombre armapc, la dirección para acceder a sus páginas será www.armapc.com.

Para ver si un dominio esta disponible visite la página www.nic.mx visite <http://www.virtualis.com.mx/dominios.html> [34].

1.5.3 ¿Qué hacer para que visiten su página?

Los usuarios utilizan los buscadores para encontrar una página en relación con un tema particular.

Para presentar su sitio web a un buscador hay que conectarse al motor de búsqueda o buscador de su elección y encontrar la liga que permite agregar su URL.

La de altavista se encuentra en altavista.digital.com/av/content/addurl.htm

La de yahoo se encuentra en www.yahoo.com/docs/info/include.html

La de lycos en www.lycos.com/addsite.html

Para que el motor de búsqueda sepa el contenido de nuestra página y así informar al usuario que nuestra página tiene la información que busca, se usan los metatag:

En la sección **head** de su página teclee:

```
<meta name="keywords" content="palabras describan el tema tratado separadas por coma y espacio">
```

```
<meta name="description" content="una o dos frases breves que describan su página, algo que convenga a la gente de entrar a su página">
```

```
<meta name="autor" content="nombre de la persona que escribió la página">
```

En dos semanas trate de buscar su página en el motor de búsqueda, si no la encuentra vuelva a presentarla de nuevo [34].

En este capítulo se ha visto que es Internet, cómo publicar una página y ahora es necesario que se presente un manual de HTML para realizar páginas web.

CAPITULO DOS

HTML

2.1 HTML

HTML es el formato con el cual se le dice a un *Web Browser* (*Explorador de Internet*, como por ejemplo el *MS Internet Explorer* o el *Netscape Navigator*) cómo mostrar una página Web: formato de los textos, colores, orden de los párrafos, etc. Los documentos HTML son simplemente archivos de texto con "*tags*", etiquetas, especiales o códigos mediante los cuales el *Web Browser* sabe cómo interpretar y mostrar en pantalla la información.

Las etiquetas o tags le indicarán al navegador la posición relativa de los elementos de la página, su tamaño, las tipografías y colores a utilizar, etc. Existen dos clases de etiquetas:

I. Etiquetas vacías

Tiene el siguiente formato:

<etiqueta>

Se utilizan para introducir saltos de línea, líneas horizontales, y otros elementos no asociados al formato de textos o imágenes.

II: Etiquetas contenedoras

Tienen el siguiente formato:

<etiqueta> ... </etiqueta>

Estas etiquetas dan un formato al texto o porción del documento que engloban. Este tipo de etiquetas tiene una marca inicial, y una marca final igual que la inicial, pero con el carácter "/". Todo lo que esté contenido entre ambas marcas quedará sujeto al formato indicado por la etiqueta.

Las letras dentro de la etiqueta pueden estar en mayúsculas o minúsculas indiferentemente [38].

Toda página HTML debe comenzar por la etiqueta <html> y terminar con </html>.

El documento HTML se divide siempre en dos partes: cabecera y cuerpo.

La **cabecera** figura entre las etiquetas `<head>` y `</head>`. En ella se especifica el título, el autor, y otras características del documento que no se muestran físicamente en la página. En la cabecera también deben incluirse los scripts que necesite el documento (JavaScript, JScript, etc.) [37].

Un **script** no es más que un conjunto de archivos que contienen, unos comandos y/o eventos programados por alguien. A los programas escritos por un lenguaje interpretado se les conoce como scripts.

Dentro de la cabecera debe ir el título comprendido entre las etiquetas `<title>` y `</title>`, este título debe ser corto y descriptivo del contenido de la página ya que será lo que verán las personas que guarden la página en alguna agenda de direcciones.

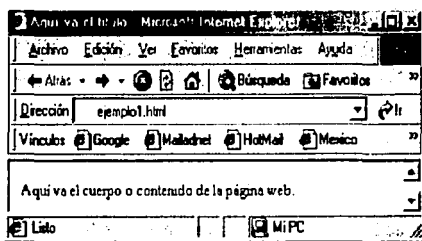
El **cuerpo** que es en donde propiamente esta la página (contenido), debe de ir entre las etiquetas `<body>` y `</body>`.

Ejemplo 1.

Escribimos en el Bloc de Notas de Windows (o un editor de texto que no formatee el texto ya que con estos editores como Word o Word Pad el navegador presenta problemas, es decir, no puede traducir el código, en caso de usarlos guardamos en modo de *solo texto*) lo siguiente y lo guardamos con extensión html (ejemplo1.html).

```
<html><head>
<title> Aquí va el titulo </title></head>
<body>Aquí va el cuerpo o contenido de la página web.</body>
</html>
```

Para poder ver el resultado de este ejemplo solo es necesario abrir el navegador y en el menú archivo seleccionamos abrir, esto nos abre una ventana de dialogo con un botón llamado examinar, buscamos el archivo que en el ejemplo se llama ejemplo1.html y damos aceptar, el resultado es el siguiente:



2.2 CREACIÓN DE PÁRRAFOS.

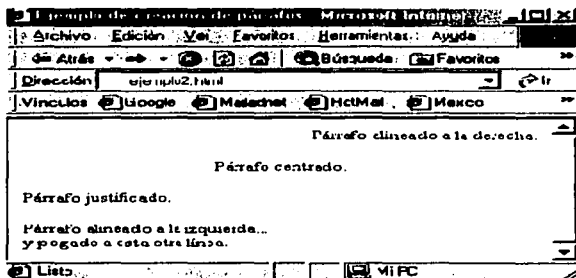
Toda página HTML está compuesta, principalmente, por párrafos de texto combinados con otros elementos, como pueden ser imágenes, tablas o formularios. La inserción de párrafos se realiza mediante la utilización de la etiqueta `<p>` y su correspondiente cierre, `</p>` [32].

Con esta etiqueta se puede modificar la justificación del párrafo con el parámetro `align`, es decir, podemos alinear a la izquierda `left`, a la derecha `right`, justificar `justify` o centrar `center`.

Si se quiere separar un párrafo o cualquier otra cosa sin dejar una línea en blanco se puede usar la etiqueta `
` que no tiene cierre.

Ejemplo 2.

```
<html>
  <head>
    <title>Ejemplo de creación de párrafos</title>
  </head>
  <body>
    <p align="right">Párrafo alineado a la derecha.</p>
    <p align="center">Párrafo centrado.</p>
    <p align="justify">Párrafo justificado.</p>
    <p align="left">Párrafo alineado a la izquierda...<br>y pegado
    a esta otra línea.</p>
  </body>
</html>
```



2.3 FORMATO DEL TEXTO.

En una página se pueden modificar el color, tamaño y tipo de letra de cada párrafo con la etiqueta `... `.

Para modificar el tipo de letra de un párrafo basta ponerlo entre:

` ... ` por ejemplo si queremos escribir Hola! con letra Garamond el código sería: `<p> Hola! </p>`.

También se pueden poner cabeceras, títulos (en negritas) con las etiquetas `<h1>..</h1>`, `<h2>..</h2>` ... `<h6>..</h6>` siendo el número indicativo del tamaño, el tamaño mayor es el correspondiente al número 1.

Una etiqueta muy interesante es la de centrado `<center>` y `</center>`. Nos centra todo lo que esté dentro de ella, ya sea texto, imágenes, etc. Un separador horizontal se logra con la etiqueta `<hr>`.

Para darle color al texto se utiliza la etiqueta ` ... `, donde XXYYZZ es un número que indica el color y a su vez XX es un número indicativo de la cantidad de color rojo, YY de color verde y ZZ de color azul.

Estos números están en numeración hexadecimal. Esta numeración tiene 16 dígitos que son:

0 1 2 3 4 5 6 7 8 9 A B C D E F

El número menor es 00 y el mayor FF. Por ejemplo el color rojo es "#FF0000" por que tiene máximo de rojo y cero de los otros colores.

"#000000" = negro
 "#FFFFFF" = blanco
 "#FFFF00" = amarillo
 "#00FF00" = verde
 "#0000FF" = azul

Por ejemplo si queremos un párrafo en color azul el código sería:
`<p> "párrafo" </p>`.

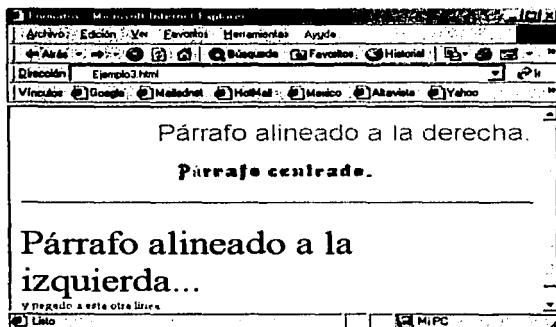
Si lo que queremos es modificar el tamaño, le etiqueta a usar es ` ... `, donde tamaño es un número que puede indicar el tamaño absoluto (con rango de 1 a 7) o el relativo. Por ejemplo, si se utiliza `size="2"` el texto se mostrará en tamaño 2. Si se utiliza `size="+2"`, se le indica al navegador que muestre el texto 2 veces más grande que el tamaño definido predeterminadamente.

Se pueden combinar todos los atributos de las etiquetas por ejemplo:

```
<p><font face="Courier" color="red" size="4">¡Hola!</font></p>
```

Ejemplo 3.

```
<html>
  <head>
    <title> Formatos </title>
  </head>
  <body>
    <p align="right"><font color="#cc8080" size="+3" face="Arial">
    Párrafo alineado a la derecha.</font></p>
    <h2><p align="center"><font color="#8899dd" face="Advert">
    Párrafo centrado.</font></p></h2><hr>
    <p align="left"><font size=7>Párrafo alineado a la izquierda...
    <br></font><font color="#FF33EE">y pegado
    a esta otra línea.</font></p>
  </body>
</html>
```



2.4 FONDOS.

Hay dos maneras de cambiar el fondo:

1. Con un color uniforme.
2. Con una imagen.

Los fondos con un color uniforme se consiguen añadiendo el atributo **bgcolor** a la etiqueta `<body>` de la siguiente manera:

`<body bgcolor="#XXYYZZ">` donde XXYYZZ es un número que indica el color tal como se vio al cambiar el color al texto de un párrafo [38].

El fondo de una página puede ser también una **imagen**, ya sea en formato **GIF** o **JPEG**. Esta imagen se repite por toda la página, de una manera análoga al tapiz de Windows. Esto se consigue añadiendo el atributo **background** a la etiqueta `<body>` de la siguiente manera:

`<body background="imagen.gif">` ó `<body background="imagen.jpg">`.

Esta imagen debe existir en la misma carpeta que contenga la página HTML de lo contrario hay que poner la dirección de la imagen.

2.5 COLOR DEL TEXTO.

El color predefinido del texto es negro pero hay la posibilidad de cambiarlo añadiendo el atributo **text** a la etiqueta `<body>` de la siguiente manera:

`<body text="#XXYYZZ">` donde XXYYZZ es el número que indica el color que tendrá el texto a lo largo de la página.

2.6 CARACTERES ESPECIALES.

En caso de que el navegador no acepte ciertos caracteres (c) aquí esta su código.

código	c	código	c	código	c
©	á	©	©	¶	¶
«	é	«	«	·	·
¬	í	¬	¬	¸	,
­	ó	­	™	¹	¡
®	ú	®	®	º	º
¯		¯	¯	»	»
°	¢	°	º	¼	¼
±	£	±	±	½	½
²	¥	²	²	¾	¾
³	¡	³	³	¿	¿
´	§	´	´	×	×
µ	¨	µ	µ	÷	÷

código <i>e</i>	
&	&
<	<
>	>
"	"

En las fórmulas matemáticas puede interesar poder escribir **índices** y **subíndices**, que se consiguen con las etiquetas `^{` `}` y `_{` `}` respectivamente [34]. Así, por ejemplo:

m^2 se consigue de la siguiente manera: `m²`

v_x se consigue con: `v_x`

2.7 IMÁGENES.

No solo se pueden insertar imágenes como fondo sino como ligas a otra página o solo para adornar la página.

Es recomendable que la imagen este en el mismo directorio que la página, ya que de lo contrario en vez de poner solo el nombre de la imagen tendríamos que poner la ruta completa de su localización.

La etiqueta que nos sirve para incluir imágenes en nuestras páginas del Web es:

```
 ó 
```

En donde `imagen.gif` ó `imagen.jpg` es el nombre de la imagen.

Dentro de la etiqueta se pueden añadir atributos como `alt` con el que introducimos una pequeña descripción de la imagen y el código quedaría así:

```

```

Esto lo podemos ver pasando el mouse por encima de la imagen.

Otro atributo es `align` que permite especificar la alineación de la imagen y del texto alrededor de la imagen. Puede tomar los valores `left`, `right`, `top`, `texttop`, `middle`, `absmiddle`, `baseline`, `bottom`, `absbottom` por ejemplo:

```
 Texto alineado en el centro...
```

```
 Texto alineado abajo...
```

```
 Texto alineado arriba...
```

Un atributo más es **width / height**: permiten especificar el tamaño de la imagen (ancho y alto). Si no se incluyen, la imagen se muestra a su tamaño real.

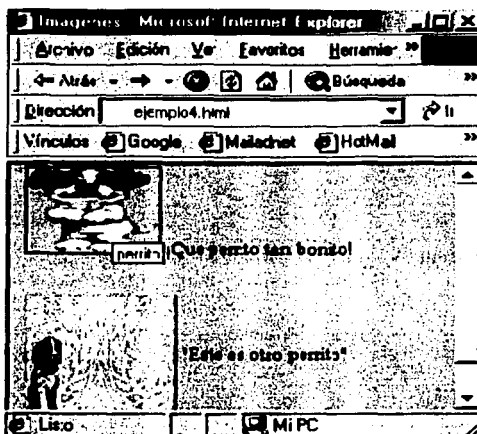
Ejemplo:

`` donde 150 y 100 son píxeles de la página.

Otro atributo es **border**: que es un borde para la imagen, este borde puede tomar incluso un valor cero.

Ejemplo 4.

```
<html>
  <head>
    <title>Imágenes</title>
  </head>
  <body text="#0000a0" bgcolor="#9dc6f7" >
    
    &icxcl;Que perrito tan bonito!<br><br><br>
    
    &quot;Este es otro perrito&quot;
  </body>
</html>
```



2.8 COMENTARIOS.

Los comentarios dentro del código son útiles ya que nos pueden servir para recordar posteriormente lo que hicimos. Los comentarios se ponen entre `<!-- y -->` [32].

2.9 CREACIÓN DE ENLACES.

Una de las mayores virtudes del lenguaje HTML, y la que le da su nombre, es la posibilidad de crear enlaces o hipervínculos que relacionan diferentes páginas entre sí. HTML son las siglas de *HiperText Mark-up Language*, que podría traducirse como lenguaje de etiquetas hipertexto.

Los enlaces, vínculos o hipervínculos (también conocidos como *links*, su denominación inglesa) son simplemente caminos hacia otras páginas de la *world wide web*.

La etiqueta para crear enlaces es la siguiente:

```
<a href="xxx"> yyy </a>
```

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

Hay varios tipos de enlaces:

1. Enlaces dentro de la misma página.
2. Enlaces con otra página nuestra.
3. Enlaces con una página fuera de nuestro sistema.
4. Enlaces con una dirección de email.

TESIS CON
FALLA DE ORIGEN

1. Enlaces dentro de la misma página

A veces, en el caso de documentos (o páginas) muy extensos, nos puede interesar dar un salto desde una posición a otra determinada.

En este caso, lo que antes hemos llamado XXX, es decir, el **destino** del enlace, en este caso el sitio dentro de la página a donde queremos saltar, se sustituye por #marca (la palabra marca puede ser cualquier palabra que queramos). Lo que hemos llamado antes YYY es la palabra (o palabras) que aparecerán en la pantalla en color (en forma de hipertexto). Su estructura es, entonces:

```
<a href="#marca"> yyy </a>
```

Y en el sitio exacto a donde queremos saltar, debemos poner la siguiente etiqueta:

```
<a name="marca"> </a>
```

2. Enlaces con otra página nuestra

Puede ser que tengamos una sola página. Pero lo más frecuente es que tengamos varias páginas, una inicial (o principal) y otras conectadas a ella, e incluso entre ellas mismas.

NOTA: Se va a suponer que las páginas están en el mismo directorio.

Supongamos que queremos enlazar con la página que contiene el ejemplo 3, llamada ejemplo3.html. En este caso sustituimos XXX por el nombre de la página, en este caso ejemplo3.html y el código se vería así:

```
< a href="ejemplo3.html"> Enlace al ejemplo 3 </a>
```

Lo anterior nos llevaría al inicio de la página del ejemplo 3, si quisiéramos ir a un sitio concreto tendríamos que poner una marca y concretar el enlace con la referencia a esa marca y quedaría así:

```
< a href="ejemplo3.html#marca"> Enlace al ejemplo 3 en la marca </a>
```

3. Enlaces con una página fuera de nuestro sistema

Si queremos enlazar con una página que esté fuera de nuestro sistema (es decir, que esté en un servidor distinto al que soporta nuestra página), es necesario conocer su dirección completa, o URL (*Uniform Resource Locator*).

Una vez conocida la dirección (o URL), lo colocamos en vez de XXX, por ejemplo si queremos enlazar a el sitio de la Facultad de Ciencias sería así:

```
<a href="http://www.ciencias.unam.mx"> Página de la Facultad de Ciencias
</a>
```

4. Enlaces con una dirección de email

En este caso, sustituimos lo que se ha llamado antes xxx (el destino del enlace) por **mailto**: seguido de la dirección de email. La estructura de la etiqueta es:

```
<a href="mailto: dirección de email"> Texto del enlace </a>
```

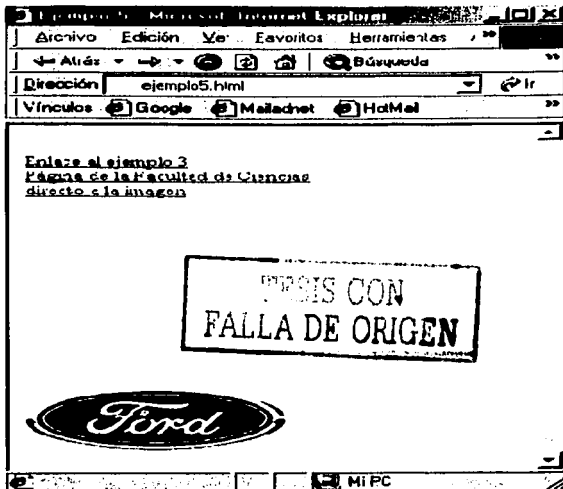
NOTA: Podemos poner en vez de algún texto de enlace en cualquier enlace una imagen [13].

Ejemplo 5.

```
<html>
  <head>
    <title>Ejemplo 5 </title>
  </head>
  <body>

    <br><a href="Ejemplo3.html">Enlace al ejemplo 3</a>
    <br><a href="http://www.ciencias.unam.mx"> Página de la Facultad de Ciencias
    </a>
    <br><a href="ejemplo5.html#aqui"> directo a la imagen </a><br><br><br>
    <br><br><br><br><br><br><br><br><br>
    <br><a name="aqui"><a href="http://www.ford.com.mx">
    </a>

  </body>
</html>
```

2.10 LISTAS.

A menudo nos interesará presentar las cosas en forma de **listas**. Podemos escoger entre tres tipos distintos:

1. Listas desordenadas (no numeradas)
2. Listas ordenadas (numeradas)
3. Listas de definición.

Las **listas desordenadas** (*unordered lists*) sirven para presentar cosas que, por no tener un orden determinado, no necesitan ir precedidas por un número. Su estructura es la siguiente:

```
<UL>
<LI> Una cosa
<LI> Otra cosa
<LI> Otra más
<LI> Etc.
</UL>
```

Es decir, toda la lista está dentro de la etiqueta `` y ``, y luego cada cosa va precedida de la etiqueta `` (*list item*).

Se puede anidar una lista dentro de otra. Por ejemplo:

```
<UL>
<LI> Mamíferos
<LI> Peces
  <UL>
    <LI> Sardina
    <LI> Bacalao
  </UL>
<LI> Aves
</UL>
```

Las **listas ordenadas** (*ordered lists*) sirven para presentar cosas en un orden determinado. Su estructura es muy similar a la anterior. La diferencia estriba en que en el resultado aparecerá automáticamente un número correlativo para cada cosa.

```
<OL>
<LI> Primera cosa
<LI> Segunda cosa
<LI> Tercera cosa
<LI> Etc.
</OL>
```

Al igual que las listas desordenadas, también se pueden anidar las listas ordenadas.

El tercer tipo lo forman las **listas de definición**. Como su nombre indica, son apropiadas para glosarios (o definiciones de términos). Toda la lista debe ir englobada entre las etiquetas <DL> y </DL>. Y a diferencia de las dos que hemos visto, cada renglón de la lista tiene dos partes: 1) el nombre de la cosa a definir, que se consigue con la etiqueta <DT> (*definition term*) y 2) la definición de dicha cosa, que se consigue con la etiqueta <DD> (*definition definition*) [34].

```
<DL>
<DT> Una cosa a definir
<DD> La definición de esta cosa
<DT> Otra cosa a definir
<DD> La definición de esta otra cosa
</DL>
```

Ejemplo 6.

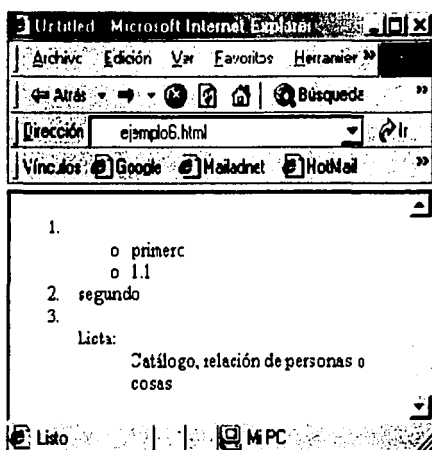
```
<html>
  <head>
    <title>Untitled</title>
  </head>
  <body>
    <ol>
```

```

<li><ul><li>primero
<li>1.1</ul>
<li>segundo
<li>
<dl>
<dt>Lista:<dd>Catálogo, relación de personas o cosas
</dl>
</ol>
</body>
</html>

```

TEXTO CON
FALLA DE ORIGEN



2.11 TABLAS.

Las tablas son una forma de presentar los datos, nos ayudan a acomodar la información en filas y columnas, mejorando el aspecto visual de la información.

Antes de crear una tabla, es importante tener una idea clara de los que se va a construir, es decir, saber cuantas filas y columnas necesita y donde situar cada elemento, es aconsejable diseñar la tabla en una hoja de papel.

La utilidad de las tablas va más allá de la presentación ordenada de los datos. El lenguaje HTML tiene carencias en posicionamiento de imágenes, márgenes, alineación del texto. Los diseñadores han superado estas deficiencias utilizando tablas.

En HTML, las tablas se construyen especificando filas y celdas. El conjunto de celdas de una fila irá encerrado en las etiquetas de fila, y el conjunto de filas de la tabla, en las etiquetas de tabla.

El comienzo y fin de una tabla se define mediante la etiqueta `<table> ... </table>`.

Para especificar cada fila de la tabla se utilizan las etiquetas `<tr> ... </tr>`. Finalmente, para especificar cada celda de una fila habrá que usar las etiquetas `<td> ... </td>` [37].

Las etiquetas que definen la tabla admiten diferentes parámetros para lograr el correcto dimensionamiento de las celdas y de los bordes que forman la tabla. También es posible especificar la separación que debe existir entre el texto y el borde de la tabla.

Los parámetros que rigen estas características son los de la siguiente tabla:

Parámetro	Etiqueta	Significado
width	<code><table></code> <code><td></code>	Anchura total de la tabla o anchura de la celda. Puede especificarse en % o en pixels.
height	<code><td></code>	Altura de la celda. Puede especificarse en % o en pixels.
cellspacing	<code><table></code>	Separación entre celdas.
cellpadding	<code><table></code>	Separación entre el texto y el borde de la celda.

Tabla 1: parámetros con sus etiquetas y su significado.

Parámetro	Significado
align	Alineación horizontal del texto. Puede tomar los valores left (izquierda), center (centro) y right (derecha).
valign	Alineación vertical del texto. Puede tomar los valores top (superior), middle (central), bottom (inferior) y baseline (línea de base).
border	Le pone un borde en niveles a la tabla

TESIS CON
FALLA DE ORIGEN

	si el valor es cero, el borde no se visualiza.
bordercolor	Para la etiqueta <td>, determina el color del borde (valores Hexadecimales).
bgcolor	Determina el color de la celda en la etiqueta <td> o de la tabla en la etiqueta <table> de igual manera toma valores hexadecimales.
background	Para poner una imagen de fondo en la tabla o en alguna celda
Colspan	Para extenderse un número determinado de columnas
Rowspan	Para extenderse un número de filas

Tabla 2: significado de algunos parámetros.

Se puede añadir un título a la tabla con la etiqueta <caption> ... </caption>.

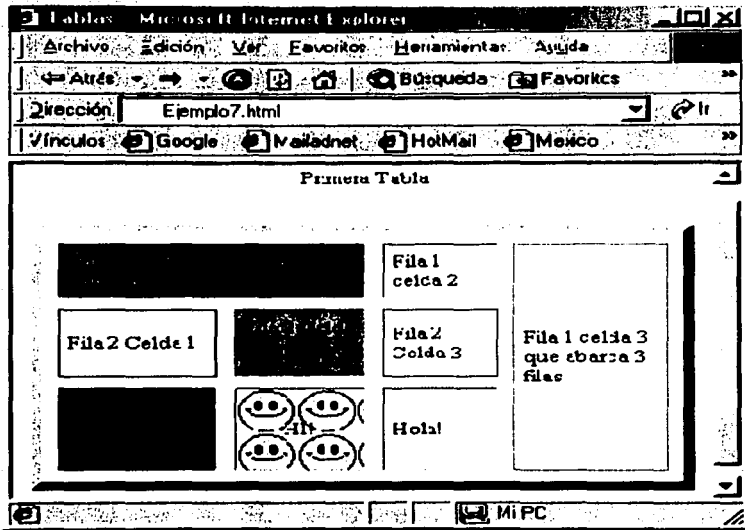
Ejemplo 7.

```
<html>
  <head>
    <title>Tablas</title>
  </head>
  <body>
    <table align="center" cellspacing="9" cellpadding="5" border="6">
      <caption>Primera Tabla</caption>
    <tr>
      <td bgcolor=#87aacd colspan="2"> Fila 1 celda 1 que abarca 2 columnas</td>
      <td>Fila 1 celda 2</td>
      <td bordercolor=#fb9b62" rowspan="3">Fila 1 celda 3<br>que abarca 3
    filas</td>
    </tr>
    <tr>
      <td bordercolor=#ff0000">Fila 2 Celda 1</td>
      <td align="justify" valign="top" bgcolor=#7be278">Fila 2 Celda 2 </td>
      <td width="52" height="52" align="left" bordercolor=#8c9f2f">Fila 2 Celda
    3</td>
    </tr>
  </tr>
</body>
```

```

<td width="86" height="64" align="center" bgcolor="#ff0080">Fila 3 Celda
1</td>
<td background="gobly.gif" align="center">HII</td>
<td bgcolor="#99ffff">Hola!</td>
</tr>
</table>
</body>
</html>

```



2.12 FORMULARIOS.

Los formularios dan a las páginas web un grado de interactividad que es bastante práctico para muchas cosas, ya que podemos pedir la información que necesitamos.

Los formularios estarán compuestos por tantos campos como informaciones queramos obtener.

La estructura general de un formulario es:

1. Etiqueta de inicio:

```
<form action=" " method=" " enctype="text/plain ">
```

2. Cuerpo del formulario, con los distintos elementos para poder introducir los datos.

3. Botones de envío y de borrado.

4. Etiqueta de cierre `</form>`

El atributo **action** indica la acción a realizar por el formulario ejemplo mailto (envío de correo) o URL (abre una página de internet).

El atributo **method** indica el método de transferencia de las variables. Post, si se envía a través del STDIO (entrada estándar). Get, si se envía a través de la URL, la diferencia radica en que con el método get podemos ver los parámetros pasados en la URL, pero el resultado es el mismo.

El atributo **enctype** es usado cuando el *action* es mailto y nos sirve para recibir las respuesta o información en forma legible.

Elementos para introducir los datos:

Los datos se pueden introducir por medio de tres elementos: Texto, menús y botones.

Para introducir datos en forma de texto y botones utilizamos la etiqueta:

```
<input type="xxx" name="yyy" value="zzz">
```

En donde:

xxx es la palabra que indica el tipo de introducción.

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

zzz es la palabra asociada a un elemento, es como inicializar la variable.

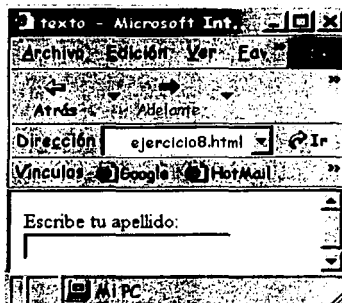
Introducción por medio de texto (una línea):

En este caso es **xxx=text**, es decir, `input type="text" [13]`.

Ejemplo 8.

```
<HTML>
<head>
  <title> texto </title>
</head><body>
```

```
<FORM ACTION="mailto:dirección_de_email" METHOD="POST"
  ENCTYPE="TEXT/PLAIN">
  Escribe tu apellido:
  <BR><INPUT TYPE="text" NAME="Apellido">
</FORM> </body>
</html>
```



Si el usuario introduce su apellido, p. ej. Ruiz, y pulsa el botón de envío (que veremos más adelante), recibiremos un email suyo con el siguiente texto:
Apellido = Ruiz

La **longitud** de este formulario es de 20 caracteres predefinidamente. Se puede variar incluyendo en la etiqueta el atributo `size="número"`. Por otra parte, sea cual sea la longitud del formulario, si no se indica nada, el usuario puede introducir el **número de caracteres** que quiera. Se puede limitar esto, incluyendo en la etiqueta el atributo `maxlength="número"`.

También se puede hacer que el texto introducido **no sea reconocible**, es decir que todos los caracteres se representen por asteriscos.

Basta con cambiar en la etiqueta `input type="text"` por `input type="password"`. Introducción por medio de texto (múltiples líneas):

Hay ocasiones en las que los datos que necesitamos constan de múltiples líneas como en los comentarios, por lo que necesitamos de un espacio con gran longitud, esto lo logramos con la etiqueta:

```
<textarea name="yyy" rows="número de filas" cols="número de columnas">
Aquí podemos indicar que va un comentario </textarea>
```

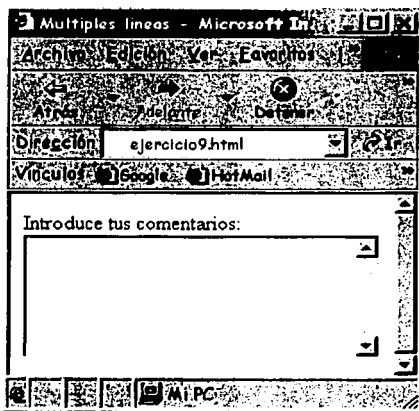
TESIS CON
FALLA DE ORIGEN

Ejemplo 9.

```

<html>
<head>
  <title>Multiples lineas </title>
</head>
<body>
  <FORM ACTION="mailto:dirección_de_email" METHOD="POST"
  ENCTYPE="TEXT/PLAIN">
    Introduce tus comentarios:
    <BR><TEXTAREA NAME="Comentarios" ROWS="6" COLS="30">
    </TEXTAREA>
  </FORM>
</body>
</html>

```

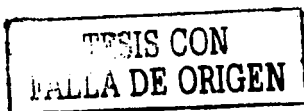


Una vez que el usuario haya escrito sus comentarios dentro del formulario, y haya pulsado el botón de envío, recibiremos un email suyo con el siguiente texto:
Comentarios = Bla bla bla...

Introducción por medio de menús:

Se puede hacer que el usuario en vez de introducir texto, escoja entre varias opciones dadas por nosotros, esto lo logramos con la etiqueta:

```
<select name="yyy" [multiple size="número"]><option >opción 1 ...
```



`<option [value="valor" selected]> opción n </select>`

Lo que tenemos entre corchetes [] es opcional y significa lo siguiente.

Multiple: Indica que podemos escoger varias opciones.

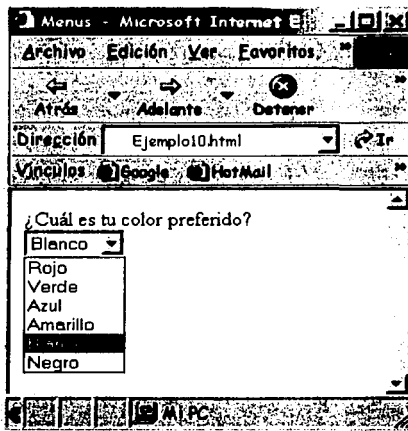
Size: Nos sirve para limitar el número de opciones a escoger.

Value: Es para darle un valor a la opción elegida. Si no se usa retorna el contenido del elemento, es útil por ejemplo en exámenes, en este atributo pondríamos el número de respuesta (ej. a,b,c) y recibiríamos la variable (en este caso el número de la pregunta) igualada a la respuesta que eligió el usuario (a,b,c), si no se usa value regresa el contenido.

Selected: Indica que esta opción se seleccione inicialmente, si ninguna opción tiene este atributo la opción seleccionada será la primera.

Ejemplo 10.

```
<html>
<head>
  <title>Menus</title>
</head><body>
<FORM ACTION="mailto:dirección_de_email" METHOD="POST"
ENCTYPE="TEXT/PLAIN">
  ¿Cuál es tu color preferido?
  <BR><SELECT NAME="ColorPreferido">
    <OPTION value="rojo">Rojo
    <OPTION>Verde
    <OPTION value="el_mio">Azul
    <OPTION>Amarillo
    <OPTION selected>Blanco
    <OPTION>Negro
  </SELECT >
</FORM>
</body>
</html>
```



Si el usuario a elegido el color azul recibiremos un email con el texto: `ColorPreferido = el_mio`.

Formulario de confirmación (checkbox):

Si queremos que el usuario confirme una opción determinada, lo podemos lograr con los checkbox o formularios de confirmación, con la siguiente etiqueta:

```
<input type="checkbox" name="yyy" [checked value="valor"]>
```

En donde:

Checked. (que por estar entre corchetes []) es opcional y lo que hace es marcar la opcion al abrir la hoja.

Value. Es el valor que regresa en caso de ser seleccionado. De no llevar este atributo regresa On.

Ejemplo 11.

```
<html>
<head>
  <title>Checkbox</title>
</head><body>
<form action="mailto:dirección_de_email" method="post" enctype="text/plain">
  <INPUT TYPE="checkbox" NAME="info">
```

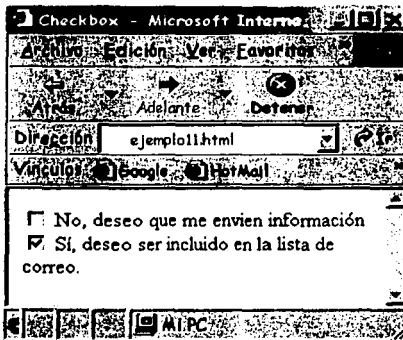
TESIS CON
 FALLA DE ORIGEN

No, deseo que me envíen información

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

Sí, deseo ser incluido en la lista de correo.

</FORM> </body>
 </html>



Si el usuario envía así este formulario recibiremos un email con el texto:
 Lista=On

Lo cual significa que escogió ser incluido en la lista de correo, como no nos llegó la variable *info* significa que no eligió esa opción.

Botones de radio:

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

```
<input type="radio" name="yyy" value="zzz" checked>
```

En donde:

yyy es el nombre que le ponemos a la variable que se trata de elegir.

zzz es el nombre de cada una de las opciones en concreto.

Checked es opcional y sirve para marcar una opción al abrir la hoja.

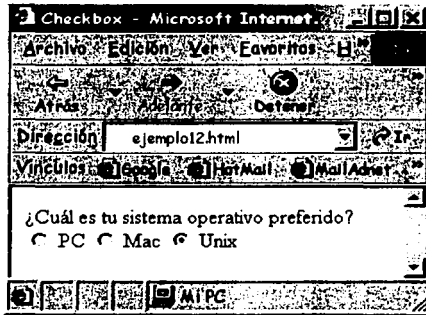
Ejemplo 12.

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

```

<title>Checkbox</title>
</head><body>
<form action="mailto:dirección_de_email" method="post" enctype="text/plain">
  ¿Cuál es tu sistema operativo preferido?
  <BR>
  <input type="radio" name="sistemaoperativo" value="pc" > PC
  <input type="radio" name="sistemaoperativo" value="mac"> Mac
  <input type="radio" name="sistemaoperativo" value="unix" checked > Unix
</form> </body>
</html>

```



Si el usuario envía así el formulario recibiremos un email con el texto:

Sistemaoperativo = unix

Botones de envió y de borrado:

Los botones de envió son esenciales en cualquier formulario y se consiguen con la etiqueta:

```
<input type="submit" value="zzz">
```

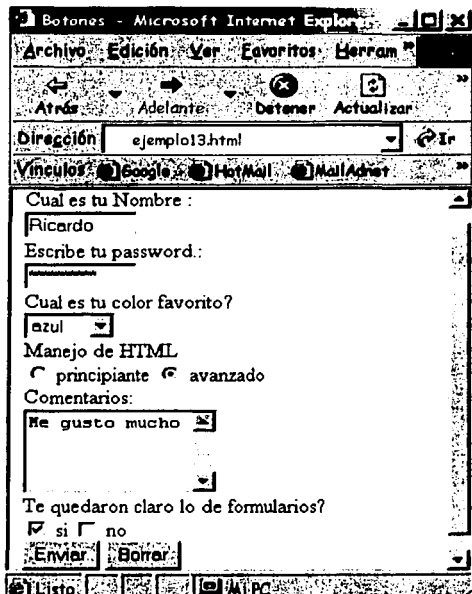
El botón de borrado o limpiado (del formulario) se consigue con la etiqueta:

```
<input type="reset" value="zzz">
```

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

Ejemplo 13.

```
<html>
  <head>
    <title>Botones</title>
  </head><body>
<form action="mailto:tesis@fciencias.unam" method="post" enctype="text/plain">
  Cual es tu Nombre :<br>
  <input type="text" name="nombre" size="10" maxlength="10"><br>
  Escribe tu password.:<br>
  <input type="password" name="password" size="10" maxlength="10"><br>
  Cual es tu color favorito?<br>
<select name="color" size="1">
  <option value="rojo"> rojo
  <option value="blanco"> blanco
  <option value="verde"> verde
  <option value="rosa"> rosa
  <option value="negro"> negro
  <option value="azul"> azul
  <option value="cafe"> cafe
</select><br>
  Manejo de HTML<br>
  <input type="radio" name="nivel" value="principiante"> principiante
  <input type="radio" name="nivel" value="avanzado" selected> avanzado<br>
  Comentarios:<br>
<textarea name="comentarios" cols="15" rows="4">
  </textarea><br>
  Te quedaron claro lo de formularios?<br>
<input type="checkbox" name="gusto" value="si"> si
<input type="checkbox" name="gusto" value="no"> no<br>
<input type="submit" value="Enviar">
<input type="reset" value="Borrar">
</form>
</body>
</html>
```



1.13 TEXTO EN MOVIMIENTO.

Con HTML se puede lograr texto en movimiento (marquesinas), una marquesina es una ventana en la que se desplaza texto y solo son validas para el Explores de Microsoft, la etiqueta básica es:

`<marquee> texto que se desplaza </marquee>`

Como no hemos añadido ningún atributo dentro de la etiqueta, el comportamiento de la marquesina es el que tiene predeterminado: ocupa todo el ancho de la pantalla, tiene la altura de una línea y el texto se desplaza lentamente de derecha a izquierda.

Vamos a ver los distintos **atributos** que modifican su apariencia y comportamiento:

WIDTH, HEIGHT

Ajustan la **anchura** y **altura**, respectivamente, de la marquesina. Pueden ser igual a un número de pixels, o a un porcentaje de la pantalla. Ejemplo:
<marquee width=50% height=60> Esta marquesina ocupa el 50% del ancho de la pantalla y tiene una altura de 60 pixels </marquee>

ALIGN

Modifica el alineamiento del texto que rodea a la marquesina, que puede ser *TOP* (arriba), *MIDDLE* (en medio) o *BOTTOM* (abajo). Ejemplo:
<marquee width=50% height=60 align=bottom>La palabra "¡Hola!" estará alineada con la parte inferior de la marquesina</marquee> ¡Hola!

BEHAVIOR

Este atributo (que quiere decir en inglés *comportamiento*) sirve para definir de qué manera se va a efectuar el desplazamiento del texto. Si es igual a *SCROLL* (el valor predeterminado), el texto aparece por un lado, se desplaza hasta el otro hasta desaparecer por él, y vuelve a empezar (como los casos que hemos visto anteriormente). Si es igual a *SLIDE*, aparece por un lado y se desplaza hasta llegar al otro extremo, y se para ahí. Si es igual a *ALTERNATE* se desplaza alternativamente hacia un lado y otro, siempre dentro de los límites de la marquesina. Ejemplo:
<marquee behavior=alternate>Este texto se mueve a un lado y otro, sin desaparecer</marquee>

BGCOLOR

Con este atributo se modifica el **color de fondo** de la marquesina. Ejemplo:
<marquee bgcolor="#ff7070"> esta marquesina tiene un fondo de color rosa </marquee>

DIRECTION

Este atributo sirve para modificar la **dirección hacia** la que se dirige el texto. El valor predeterminado es *LEFT* (izquierda). Se puede hacer que el texto se dirija hacia la derecha igualando este atributo a *RIGHT*. Ejemplo:
<marquee direction=right> Este texto se dirige hacia la derecha </marquee>

SCROLLAMOUNT

Define la **cantidad** de desplazamiento del texto en cada movimiento de avance, expresado en pixels. Cuanto mayor es el número, más rápido avanza. Ejemplo:
`<marquee scrollamount=50> Doy saltos grandes </marquee>`

SCROLLDELAY

Define el **tiempo** entre cada movimiento de avance, expresado en milisegundos. Cuanto mayor es el número más lento avanza. Ejemplo:
`<marquee scrolldelay =200> Espero mucho entre cada salto </marquee>`

LOOP

Especifica el **número de veces** que aparecerá el texto. Si no se utiliza, es indefinido.

HSPACE, VSPACE

Definen, respectivamente, la separación en sentido horizontal o vertical del texto que está **fuera** de la marquesina.

Ejemplo 14.

```
<html>
  <head>
    <title>marquesinas</title>
  </head>
  <body>
    <marquee> texto que se desplaza </marquee><br>
```

```
<marquee width=50% height=60> Esta marquesina ocupa el 50% del ancho de la
pantalla
y tiene una altura de 60 pixels </marquee><br>
```

```
<marquee width=50% height=60 align=bottom>La palabra "¡Hola!" estará alineada
con
la parte inferior de la marquesina</marquee> ¡Hola!<br>
```

```
<marquee behavior=alternate>Este texto se mueve a un lado y otro, sin
desaparecer</marquee><br>
```

```
<marquee bgcolor="#ff7070"> esta marquesina tiene un fondo de color rosa
</marquee><br>
```

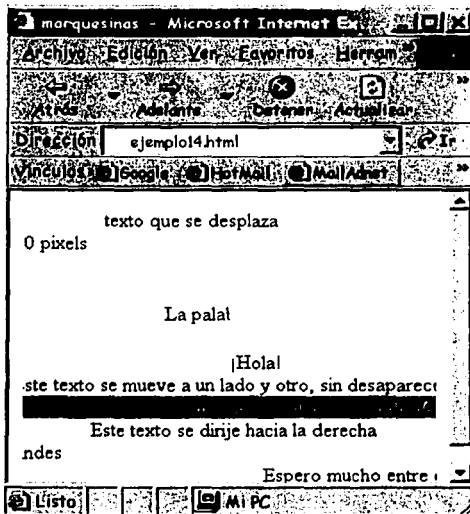
```
<marquee direction=right> Este texto se dirije hacia la derecha </marquee><br>
```

```
<marquee scrollamount=50> Doy saltos grandes </marquee><br>
```

```
<marquee scrolldelay =200> Espero mucho entre cada salto </marquee><br>
```

```
</body>
```

```
</html>
```



Existen ciertos programas que nos ayudan a automatizar este proceso, pero es muy conveniente comenzar a hacerlo de una manera manual, para comprender bien la estructura del lenguaje HTML.

Editores de HTML

Para Windows

- En castellano:
Web Scope, por Jaime Saz Rodríguez-Vigil
CastWeb, por Alonso Rodríguez Zamora
- En inglés:
HotDog
HomeSite
HoTMetal
HTML Writer
HTMLed Professional
NaviPress

Para Macintosh

- En inglés :
PageSpinner
HoTMetal
BBEdit HTML extensions
BBEditTools
html-helper-mode for EMACS
HTML Editor

En este capítulo se aprendió a cambiar el color de fondo de una página, el tamaño y tipo de letra, hacer tablas, formularios, enlazar páginas, insertar imágenes, es decir, a crear páginas web con HTML, por si sólo HTML no puede validar formularios, hacer cálculos, mandar mensajes al usuario, incorporar fecha y hora, cambiar una imagen cuando el usuario la señala, controlar el tamaño de una ventana, es decir, hacer la página web más interactiva, para lograr todo esto se necesita de JavaScript que permitirá ampliar las capacidades de HTML.

CAPITULO TRES

JAVASCRIPT

En este capítulo se verá la relación que existe entre HTML y JavaScript y cómo se utiliza JavaScript.

3.1 ¿QUE ES JAVASCRIPT?

JavaScript es un lenguaje de scripts compacto basado en objetos y en eventos, cuando un usuario da click sobre un enlace o mueve el puntero sobre una imagen se produce un evento. JavaScript no es un lenguaje de programación propiamente dicho.

JavaScript es un lenguaje interpretado, es decir, no requiere compilación. El navegador del usuario se encarga de interpretar las sentencias JavaScript contenidas en una página HTML y ejecutarlas adecuadamente.

Originariamente era denominado *LiveScript*, y fue desarrollado por Netscape [35] para su navegador *Netscape Navigator 2.0*. Fue éste el primer cliente en incorporarlo. Se ejecuta sobre 16 plataformas diferentes, incluyendo los entornos de *Microsoft* e incluso el *MS Explorer* lo incorpora en su versión 3.0.

JavaScript permite la realización de aplicaciones de propósito general a través de la WWW.

3.2 DIFERENCIAS ENTRE JAVA Y JAVASCRIPT.

Se podría decir que la única similitud entre Java y JavaScript es la sintaxis. En la siguiente tabla se muestran las diferencias.

JavaScript	Java
Interpretado (no compilado) en cliente.	Compilado en servidor antes de la ejecución del cliente.
Basado en objetos. Usan objetos, pero no clases ni herencia.	Programación orientado a objetos. Los applets constan de clases objeto con herencia.
Código integrado en el código HTML.	Applets diferenciados del código HTML (accesibles desde las páginas HTML).
No es necesario declarar el tipo de las variables.	Necesario declarar los tipos.
Enlazado dinámico. Los objetos referenciados deben existir en tiempo de ejecución (lenguaje interpretado).	Enlazados estáticos. Los objetos referenciados deben existir en tiempo de compilación (lenguaje compilado).

Tabla 3: diferencias entre JavaScript y java.

3.3 JAVASCRIPT Y HTML.

El script de JavaScript lo tenemos que colocar dentro de la cabecera, después del título, es decir, entre las etiquetas `</title>` y `</head>`, contenido dentro de la etiqueta:

```
<script language="javascript"> </script >
```

Ejemplo:

```
<html>
<head>
<title>
</title>
<script language="javascript">
    [aquí debe ir colocado el script]
</script >
</head>
<body>
</body>
</html>
```

Además de esto, se debe añadir algo dentro del cuerpo, esto lo veremos mas adelante.

Es conveniente poner el código de javascript como si fueran comentarios de html (`<!--` y `-->`) para los navegadores que no soporten javascript.

3.4 ELEMENTOS BÁSICOS.

Comentarios.

En JavaScript existen dos tipos de comentarios y son `//` y `/*...*/` por ejemplo:

`//` este es un comentario que consta de una línea.

`/*` este es el otro comentario en javascript que puede constar de varias líneas y termina en cuanto se encuentre con el símbolo `*/`

Literales.

Se llama así a los distintos valores que puede tomar una variable.

Las cadenas van entre comillas. Por ejemplo:

Nombre = "JASMIN";

Se pueden especificar vectores. Por ejemplo:

Frutas = ["mango", "fresa", "piña"];

Para los valores numéricos y booleanos es así:

Total = 2548765;

Valor = false;

Dentro de las cadenas podemos indicar varios caracteres especiales, con significados especiales. Por ejemplo:

Carácter	Significado
\n	Nueva línea
\t	Tabulador
\'	Comilla simple
\"	Comilla doble
\\	Barra invertida
\999	El número ASCII (según la codificación Latin-1) del carácter en hexadecimal

Tabla 4: significado de caracteres.

Sentencias y bloques.

Las sentencias y bloques se separan por un punto y coma y se agrupan con llaves { }.

3.5 TIPOS DE DATOS.

Un tipo de datos es la clase de valores que puede tomar un identificador (es decir, una variable o una constante). Si el tipo de datos es fecha, el identificador que tenga ese tipo sólo podrá almacenar fechas. En Javascript los tipos de datos se asignan dinámicamente según asignamos valores a las distintas variables y son los clásicos: cadenas, varios tipos de enteros y reales, valores booleanos, vectores, matrices, referencias y objetos.

Variables.

Las variables son nombres que ponemos a los lugares donde almacenamos la información. En Javascript, deben comenzar por una letra o un subrayado (_), pudiendo haber además dígitos entre los demás caracteres. **No es necesario declarar una variable**, pero cuando se hace es por medio de la palabra reservada `var`. Una variable, cuando no es declarada, tiene siempre ámbito global, mientras que en caso contrario será de ámbito global si está declarada fuera de una función, y local si está definida dentro.

Ejemplo:

```
var x; // Accesible fuera y dentro de la función pruebas
y = 2; // Accesible fuera y dentro de la función pruebas
function pruebas() {
  var z; // Accesible sólo dentro de pruebas
  w = 1; // Accesible fuera y dentro de pruebas
}
```

Se pueden declarar varias variables en una misma sentencia separándolos por comas:

```
var x, y, z;
```

El tipo de datos de la variable será aquel que tenga el valor que asignemos a la misma.

Vectores y matrices.

Estos tipos de datos complejos son un conjunto ordenado de elementos, cada uno de los cuales es en sí mismo una variable distinta. **En Javascript, los vectores y las matrices son objetos.** Como veremos que hacen todos los objetos, se declaran utilizando el operador new:

```
miVector = new Array(20)
```

El vector tendrá inicialmente 20 elementos (desde el 0 hasta el 19). Si queremos ampliarlo no tenemos más que asignar un valor a un elemento que esté fuera de los límites del vector:

```
miVector[25] = "Algo"
```

De hecho, podemos utilizar de índices cualquier expresión que deseemos utilizar. Ni siquiera necesitamos especificar la longitud inicial del vector si no queremos:

```
vectorRaro = new Array(); vectorRaro["A colocar en los bookmark"] = "HTML";
```

Hacer una matriz bidimensional es más complicado, ya que tenemos que hacer un bucle que cree un vector nuevo en cada elemento del vector original.

3.6 OPERADORES.

Los operadores nos permiten unir identificadores y literales para formar expresiones.

Los operadores aritméticos se describen en la siguiente tabla.

Descripción	Símbolo	Expresión de ejemplo	Resultado del ejemplo
Multiplicación	*	2*4	8
División	/	5/2	2.5
Resto de una división entera	%	5 % 2	1
Suma	+	2+2	4
Resta	-	7-2	5
Incremento	++	++2	3
Decremento	--	--2	1
Menos unario	-	-(2+4)	-6

Tabla 5: operadores aritméticos.

Operadores de comparación.

Descripción	Símbolo	Expresión de ejemplo	Resultado del ejemplo
Igualdad	==	2 == '2'	Verdadero
Desigualdad	!=	2 != 2	Falso
Igualdad estricta	===	2 === '2'	Falso
Desigualdad estricta	!==	2 !== 2	Falso
Menor que	<	2 < 2	Falso
Mayor que	>	3 > 2	Verdadero
Menor o igual que	<=	2 <= 2	Verdadero
Mayor o igual que	>=	1 >= 2	Falso

Tabla 6: operadores de comparación.

Operadores lógicos.

Descripción	Símbolo	Expresión de ejemplo	Resultado del ejemplo
Negación	!	!(2 = 2)	Falso
Y	&&	(2 = 2) && (2 >= 0)	Verdadero
Ó		(2 = 2) (2 <> 2)	Verdadero

Tabla 7:operadores lógicos.

Operadores de asignación.

Operador	Significado	Operador	Significado
x += y	x = x + y	x -= y	x = x - y
x /= y	x = x / y	x *= y	x = x * y
x % y	x = x % y		

Tabla 8:operadores de asignación.

Operadores especiales.

Descripción	Símbolo	Expresión de ejemplo	Resultado del ejemplo
Crear un objeto	new	a = new Array()	a es ahora un vector
Borrar un objeto	delete	delete a	Elimina el vector anteriormente creado
Referencia al objeto actual	this		

Tabla 9:operadores especiales.

3.7 ESTRUCTURAS DE CONTROL.

Condicionales.

IF...ELSE

Sintaxis: [37]

```
if(condición){
    sentencias //acciones a ejecutar si la condición es cierta
}
else { //el else es opcional
    sentencias //acciones a ejecutar si la condición es falsa
}
```

Ejemplo:

```
if(n>34){
    alert("Eres mayor");
}
else{
```

```
    alert("Eres joven");  
}
```

SWITCH

Sintaxis: [36]

```
switch(variable){  
    case "valor1":  
        acciones; //a realizar en caso de que la variable tome el valor 1  
        break;  
    case "valor2":  
        acciones; //a realizar en caso de que la variable tome el valor 2  
        break;  
    .....  
    case "valorN":  
        acciones; //a realizar en caso de que la variable tome el valor N  
        break;  
    default:  
        acciones; //a realizar si la variable no toma ningún valor anterior  
}
```

Ejemplo:

```
switch(grado){  
    case "1":  
        alert("vas en el 1");  
        break;  
    case "2":  
        alert("vas en el 2");  
        break;  
    ....  
    default:  
        alert("ya terminaste");  
}
```

Bucles.

FOR

Sintaxis: [29]

```
for ( inicio; condición; incremento ) {  
    acciones  
}
```

```
Ejemplo:  
for ( i=0; i<52; i++ ) {  
    alert(i);  
}
```

DO...WHILE

```
Sintaxis: [36]  
do {  
    acciones //se realizan por lo menos una vez, el numero depende de la condición  
} while(condición)
```

```
Ejemplo:  
do {  
    n=n+1;  
} while(n<12)
```

WHILE

```
Sintaxis: [36]  
while(condición){  
    acciones //a realizar si la condición es cierta  
}
```

```
Ejemplo:  
while(s==f){  
    alert("eres mujer");  
}
```

Con la sentencia "break", se puede salir de una sentencia de bucle sin limitaciones.

Con la sentencia "continue", se termina el bucle actual y se comienza con el siguiente.

Existen dos sentencias(**for** y **with**) que permiten acceder a las propiedades de un objeto de forma rápida.

La sentencia **for** recorre todas y cada una de las propiedades de un objeto con un índice.

```
for(variable in objeto){  
    acciones  
}
```

Ejemplo:

```
for (i in navigator) {
  document.write("Propiedad :"+i);
  document.writeln (" valor: " +navigator[i]);
}
```

"WITH" establece un objeto al que aplica un conjunto de acciones:

```
with(objeto){
  acciones
}
```

Ejemplo:

```
with (alguien) {
  nombre= nombre + ' Pommeret';
  edad= edad + 1;
  nacion= "Rusia";
}
```

3.8 FUNCIONES.

Una función es un elemento del programa creado con la finalidad de realizar una determinada acción. Una función puede ser llamada desde otra.

En JavaScript, las funciones se definen en la cabecera del documento HTML. Su sintaxis es:

```
function nombreFunción([parámetros]) {
  acciones
}
```

Para devolver un valor de retorno desde la función se utiliza la palabra reservada return.

3. 9 FUNCIONES PREDEFINIDAS.

Fecha.

getDate() - Día del mes
getDay() - Día de la semana
getMonth() - Número del mes
getFullYear() - Año con dos dígitos
getTime() - Milisegundo transcurridos desde 1/1/1970
getHours() - Hora entre 0 y 23

getMinutes() - Minutos entre 0 y 59
getSeconds() - Segundos entre 0 y 59

Existen las mismas funciones pero con "set" en lugar de "get" de modo que nos permiten cambiar el valor de esas variables.

Ventanas.

Tenemos 3 formas de mostrar un mensaje al usuario en una ventana de aviso:

Alert - Muestra el contenido de lo que le pasamos en una ventana con un botón de aceptar.

```
alert("Esto es una prueba")
```

Confirm - Muestra un mensaje como alert, pero muestra dos botones: **aceptar** y **cancelar**. Si se evalúa, al pulsar aceptar devuelve true y con cancelar, false.

Prompt - Muestra una ventana donde podemos escribir un valor, de modo que después pueda ser asignado a una variable por ejemplo para pedir el nombre del usuario y saludarlo.

```
prompt("mensaje",[valor])
```

Matemáticas.

abs() - Valor absoluto
max(v1,..,vn) - Valor máximo
min(v1,..,vn) - Valor mínimo
round() - Redondear
exp() - Exponencial
log() - Logaritmo
pow(base,exponente) - Potencia
sqrt() - Raíz cuadrada
sin() - Seno
cos() - Coseno
tan() - Tangente
asin() - Arcoseno
acos() - Arcocoseno
atan() - Arcotangente

Cadenas.

```
eval(cadena)
```

Evalúa la expresión contenida en la cadena que recibe como parámetro y la transforma en número.

Ej:

```
x=10; y=20; z=30; eval("x+y+z+900")=960
```

```
parseInt(cadena [, base])
```

Convierte en un número entero la cadena que recibe, asumiendo que está en la base indicada. Si este parámetro falta, se asume que está en base 10.

parseFloat(cadena)

Convierte en un número real la cadena que recibe.

Otras.

isNaN(valor)

Devuelve true sólo si el argumento es NaN, es decir, si no es un número.

isFinite(numero)

Devuelve true si el número es un número válido y no es infinito.

Number(referencia)

String(referencia)

Convierten a número o cadena (respectivamente) el objeto que se les pase como argumento [30].

3.10 OBJETOS.

Un objeto es una estructura que contiene tanto variables (llamadas propiedades) como las funciones que manipulan dichas variables (llamadas métodos).

Propiedades y métodos.

La forma de acceder a las propiedades y métodos de un objeto es:

nombredelobjeto.propiedad

nombredelobjeto.metodo(parámetros)

También se puede referir a una propiedad de un objeto por su índice en la creación. Los índices comienzan por 0:

```
casa[0]=casa.localidad
```

```
casa[1]=casa.superficie
```

```
casa[2]=casa.precio
```

La forma de crear un objeto es la siguiente:

1.- Crear una función constructora.

Ejemplo:

```
function casa(localidad,superficie,precio){
    this.localidad=localidad
    this.superficie=superficie
    this.precio=precio
}
```

2.- Crear objetos con "new"

```
casa1=new casa("Xola",45,565251)
casa2=new casa("Aldama",50,584712)
```

Dos observaciones importantes:

Gracias a new creamos nuevos objetos con las propiedades de los ya creados. this hace referencia al propio objeto.

A un objeto se le pueden seguir añadiendo propiedades tras ser definido, aunque es una práctica que no se aconseja, pues todos los objetos ya creados hasta entonces añaden también esa propiedad con valor nulo. Para añadir propiedades se utiliza la palabra prototype:

```
casa.prototype.año=null
casa.año="1990"
```

Otra forma de crear objetos es de forma literal:

```
nombreobjeto={propiedad1:valor;propiedad2:valor;.....propiedadN:valor}
```

Finalmente, solo añadir que una propiedad puede estar formada por varias subpropiedades de modo que para referenciarlas deberemos seguir la notación similar a la anterior:

```
nombreobjeto.nombrepropiedad.nombreSubpropiedad
```


Ejemplo:

```
casa1 = {localidad:"Xola",precio:565251,superficie: {interior:45,terrazza:10}}
```

En este ejemplo, `casa1` es un objeto en el cual, la propiedad `superficie`, a su vez está formada por dos subpropiedades: `interior` y `terrazza`.

Veamos ahora como se crean los métodos.

Un método no es mas que una función asociada a un tipo de objeto:

```
objeto.nombremétodo=nombrefunción
```

Veamos un ejemplo. Este método sirve para mostrar las propiedades de un objeto `casa`:

```
function VerCasa() {  
    var mostrar="La casa está en " + this.localidad + ", tiene " + this.superficie  
    +" m2 y cuesta " +this.precio +" pesos."  
    return(mostrar)  
}
```

Si en la función constructora de `casa` añadimos:

```
this.VerCasa=VerCasa
```

Ya tenemos un método llamado `VerCasa` que nos permite ver las propiedades de cualquier objeto de tipo `casa`.

Una vez creado un objeto, si lo que queremos es eliminarlo, sólo tenemos que llamar al operador "delete".

```
delete casa
```

Este operador es interesante, pues, si consigue eliminar el objeto, devuelve `true` si se evalúa, por lo que se puede saber cuando un objeto ha sido borrado satisfactoriamente.

3.11 EVENTOS.

Un evento, como su mismo nombre indica, es algo que ocurre. Para que una rutina nuestra se ejecute deberemos llamarla desde un controlador de eventos. Estos controladores se asocian a un elemento HTML.

Ejemplo:

```
<A HREF="http://www.mexico.com" onMouseOver="MiFuncion()">
```

Lo que sucede en este ejemplo es que cuando pasamos el ratón por encima de la liga a mexico.com se llama a la función MiFuncion().

Eventos más usados:

Evento	Descripción	Elementos (HTML) que lo admiten
onLoad	Terminar de cargarse una página	<BODY...> <FRAMESET...>
onUnload	Salir de una página (descargarla)	<BODY...><FRAMESET...>
onMouseOver	Pasar el ratón por encima	<A HREF...> <AREA...>
onMouseOut	Que el ratón deje de estar encima	<A HREF...> <AREA...>
onSubmit	Enviar un formulario	<FORM...>
onClick	Pulsar un elemento	<INPUT TYPE="button, checkbox, link, radio"...>
onBlur	Perder el cursor	<INPUT TYPE="text"...> <TEXTAREA...>
onChange	Cambiar de contenido o perder el cursor	<INPUT TYPE="text"...> <TEXTAREA...>
onFocus	Conseguir el cursor	<INPUT TYPE="text"...> <TEXTAREA...>
onSelect	Seleccionar texto	<INPUT TYPE="text"...> <TEXTAREA...>

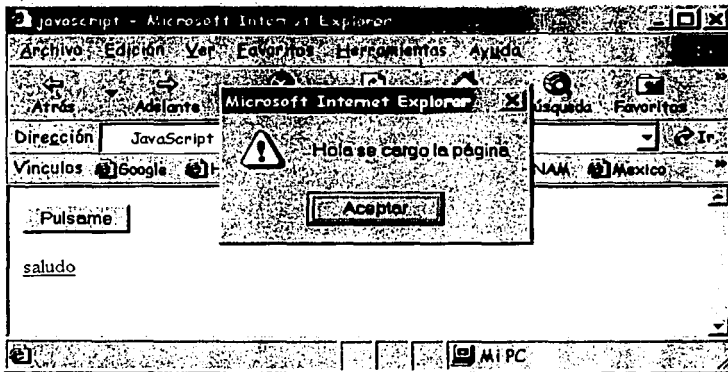
Tabla 10: descripción de eventos.

Ejemplo.

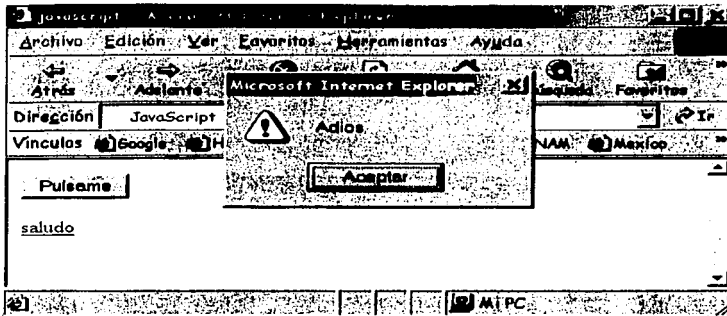
```

<html>
<head>
<title>javascript</title>
<script language="JavaScript">
function saludo() {
alert("¡Hola se cargo la página");
}
function despedida() {
alert("¡Adios ");
}
</script>
</head>
<body onLoad="saludo(); onUnload="despedida()";>
<form>
<input type="button" value="Pulsame" onClick="despedida()" ; >
</form>
<a href="mailto=yo@yo.com" onMouseOver="saludo()" ;>saludo</a>
</body>
</html>

```



Este cuadro de diálogo sale al cargar la página y al pasar el mouse por el link saludo.

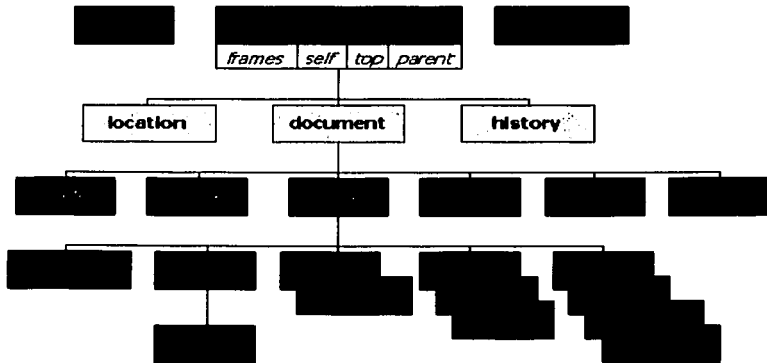


Este otro cuadro de dialogo sale al descargar la página y al hacer clic en el botón Pulsame.

3.12 JERARQUIA.

Cuando funciona en un navegador, el lenguaje JavaScript dispone de una serie de objetos que se refieren a cosas como la ventana actual, el documento sobre el que trabajamos, o el navegador que estamos utilizando.

La jerarquía de estos objetos es la siguiente:



TESIS CON
FALLA DE ORIGEN

Por ejemplo si nos queremos referir al campo de un formulario sería:

window.document.nombreformulario.nombrecampo.value

3.13 OBJETOS ESPECIALES.

JavaScript dispone de varios objetos predefinidos.

Objeto Array:

vector = new array (longitud);

Este objeto permite crear vectores.

Entre sus propiedades y métodos tenemos:

length

Propiedad que contiene el número de elementos del vector.

concat(vector2)

Añade los elementos de *vector2* al final de los del vector que invoca el método, devolviendo el resultado. No funciona en Explorer 3.

sort(funcionComparacion)

Ordena los elementos del vector alfabéticamente. Si se añade una función de comparación como parámetro los ordenará utilizando ésta. Dicha función debe aceptar dos parámetros y devolver 0 si son iguales, menor que cero si el primer parámetro es menor que el segundo y mayor que cero si es al revés.

Objeto Date:

fecha = new Date();

fecha = new Date(año, mes, día);

fecha = new Date(año, mes, día, hora, minuto, segundo);

Este objeto nos permite manejar horas y fechas.

Entre sus propiedades y métodos tenemos:

getTime()

setTime(milisegundos)

Obtienen y ponen, respectivamente, la fecha y la hora tomados como milisegundos transcurridos desde el 1 de enero de 1970.

getFullYear()**setYear(año)**

Obtienen y ponen, respectivamente, el año de la fecha. Éste se devuelven como números de 4 dígitos excepto en el caso en que estén entre 1900 y 1999, en cuyo caso se devolverán las dos últimas cifras. Hay que tener cuidado, ya que la implementación de éstos métodos puede variar en las últimas versiones de Netscape.

getFullYear()**setFullYear(año)**

Realizan la misma función que los anteriores, pero sin tantos líos, ya que siempre devuelven números con todos sus dígitos. Funciona en Explorer 4 y Netscape 4.06 y superiores.

getMonth()**setMonth(mes)****getDate()****setDate(día)****getHours()****setHours(horas)****getMinutes()****setMinutes(minutos)****getSeconds()****setSeconds(segundos)**

Obtienen y ponen, respectivamente, el mes, día, hora, minuto y segundo de la fecha, también respectivamente.

getDay()

Devuelve el día de la semana de la fecha en forma de número que va del 0 (domingo) al 6 (sábado).

Objeto String:

Este es un objeto que se puede confundir con los datos normales de tipo cadena. Conviene utilizar estos últimos, ya que los objetos String tienen un comportamiento extraño cuando se utilizan como cadenas normales. Además, al crear una cadena estamos creando a la vez un objeto String asociado.

Entre sus métodos están:

charAt(pos)**charCodeAt(pos)**

Devuelven el carácter o el código numérico del carácter que está en la posición indicada de la cadena. El último no funciona en Explorer 3.

indexOf(subcadena)

Devuelven la posición de la subcadena dentro de la cadena, o -1 en caso de no estar.

split(separador)

Devuelven un vector con subcadenas obtenidas separando la cadena por el carácter separador. No funciona en Explorer 3.

```
cadena = "Navidad,Semana Santa,Verano";  
vector = cadena.split(",");
```

En el ejemplo, el vector tendrá tres elementos con cada una de las vacaciones de un escolar.

concat(cadena2)

Devuelve el resultado de concatenar cadena2 al final de la cadena. No funciona en Explorer 3.

substr(indice, longitud)

substring(indice1, indice2)

Devuelven una subcadena de la cadena, ya sea cogiendo un número de caracteres a partir de un índice o todos los caracteres entre dos índices.

toLowerCase()

toUpperCase()

Transforman la cadena a minúsculas y mayúsculas, respectivamente.

Objeto Document:

Este objeto representa al documento html en el que estamos. Se accede a el por medio de la referencia *document*.

Algunos de sus métodos son:

lastModified

Contiene la fecha y hora en que se modificó el documento por última vez y se suele usar en conjunción con *write* para añadir al final del documento estas características.

bgColor

Modifica el color de fondo del documento. El color deberá estar en el formato usado en HTML. Es decir, puede ser red o FF0000.

forms[]

Vector que contiene los formularios del documento. El primero será el número 0, el segundo el 1, etc..

images[]

Vector que contiene las imágenes del documento. Se ordenan igual que en el anterior caso, aunque también permiten ser accedidas con el nombre como índice. Es decir, a una imagen definida como `` se puede acceder con `document.images["miImagen"]`.

links[]

Vector que contiene los enlaces del documento. Se ordenan igual que en los dos anteriores, aunque no se suele utilizar en el código Javascript. Su razón de ser es que, al ser los enlaces objetos, permiten incluir código Javascript en ellos por medio de la pseudo-URL `javascript:codigo`.

write(cadena)**writeln(cadena)**

Escribe el código HTML indicado en cadena en nuestro documento HTML. `writeln` hace lo mismo, pero incluyendo al final un retorno de carro.

open()

Abre un nuevo documento para escribir. Un documento debe estar abierto para poder escribir en él. Sin embargo, no se utiliza mucho ya que los dos métodos anteriores abren automáticamente el documento si no lo está ya.

close()

Cierra el documento, impidiendo escribir de nuevo en él.

cookie()

Crea una cookie, que es un archivo que se guarda en la maquina del usuario con la información que se le indique.

Ejemplo final:

```
<html>
<head><title>Ejemplo</title>
<script language="JavaScript">
function loca() {
a=document.formita.mayusculas.value;
```



```

alert(a.toUpperCase());
}
function color(frm, clr, val) {
v = eval("0x" + frm[clr].value) + val;
if (v < 0 || v > 255) v -= val;
v = v.toString(16).toUpperCase();
while (v.length < 2) v = "0" + v;
frm[clr].value = v; nc = "";
for(i = 1; i < 8; i += 3) nc += frm.elements[i].value;
document.bgColor = nc;
}
function setval(item) {
v = prompt("New value for " + item.name + " (00 - FF)", item.value);
if (v) {
v = eval("0x" + v);
if ((v & 255) == v) {
item.value=v.toString(16).toUpperCase();
while (item.value.length < 2) item.value = "0" + item.value;
color(document.f, item.name, 0);
}
}
}
Stamp = new Date();
document.write("<font size='2' face='Arial'><B>' + (Stamp.getMonth() + 1)
+'/' + Stamp.getDate() + '/' + Stamp.getYear() + '</B></font><BR>");
var Hours;
var Mins;
var Time;
Hours = Stamp.getHours();
if (Hours >= 12) {
Time = "P.M.";
}
else {
Time = "A.M.";
}
if (Hours > 12) {
Hours -= 12;
}
if (Hours == 0) {
Hours = 12;
}
Mins = Stamp.getMinutes();
if (Mins < 10) {
Mins = "0" + Mins;

```

```

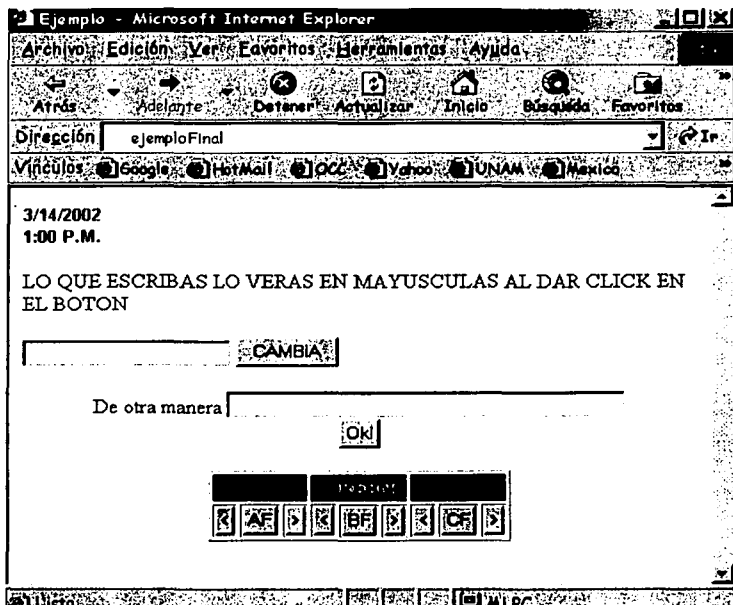
}
document.write('<font size="2" face="Arial"><B>' + Hours + ':' + Mins + Time +
'</B></font> ');
</script>
</head>
<body><br><br>
LO QUE ESCRIBAS LO VERAS EN MAYUSCULAS AL DAR CLICK EN EL
BOTON<BR>
<form name="formita">
<input type="text" name="mayusculas" size="20" maxlength="40">
<input type="button" value="CAMBIA"OnClick="loca0">
</form>
<center>
<form name="capsform">De otra manera
<input type="text" name="caps" size=40
onChange="javascript:this.value=this.value.toUpperCase();"
<br>
<input type="button" value="Ok!">
</form>
</center>
<center>
<form name=f>
<table border=1>
<tr>
<td colspan=3 align=center bgcolor="#ff0000">RED</td>
<td colspan=3 align=center bgcolor="#00ff00">GREEN</td>
<td colspan=3 align=center bgcolor="#000ff">BLUE</td>
</tr>
<tr>
<td><input type=button name=rm value="<" onclick = "color(this.form, 'Red' , -1);">
</td>
<td><input type=button name=Red value="AF" onclick = "setval(this);"></td>
<td><input type=button name=rp value=">" onclick = "color(this.form, 'Red', 1);"> </td>
<td><input type=button name=gm value="<" onclick = "color(this.form, 'Green' , -1);">
</td>
<td><input type=button name=Green value="BF" onclick = "setval(this);"></td>
<td><input type=button name=gp value=">" onclick = "color(this.form, 'Green', 1);">
</td>
<td><input type=button name=bm value="<" onclick = "color(this.form, 'Blue' , -1);">
</td>
<td><input type=button name=Blue value="CF" onclick = "setval(this);"></td>
<td><input type=button name=bp value=">" onclick = "color(this.form, 'Blue', 1);"> </td>
</tr>
</table>
</form>

```

```

</center>
</body>
</html>

```



Hasta este momento se ha mostrado el uso teórico de HTML y JavaScript, para una mejor comprensión de estas herramientas se creó una aplicación practica que se puede visualizar en <http://www.angelfire.com/amiga/acruaria/armapc2.html>. El siguiente capítulo muestra el manual de uso de esta aplicación y en el apéndice de este trabajo se muestra el código de la página creada.

CAPITULO CUATRO

CREACIÓN DE LA TIENDA VIRTUAL

4.1 CREACIÓN DE LA TIENDA VIRTUAL ARMA TU PC

En los capítulos anteriores se ha mostrado la manera de utilizar sentencias o instrucciones de HTML y JavaScript, en este capítulo se construirá una tienda virtual llamada "ARMA TU PC" que es una aplicación práctica del uso de estas herramientas.

Con HTML lo que se logra hacer es el diseño de la página, como lo es insertar títulos, imágenes, tablas, texto, formularios y ligas. También sirve para darle formato a todo lo anterior, en cambio, con JavaScript podemos poner la hora, validar formularios, hacer que las imágenes cambien, incluso que el texto cambie de minúsculas a mayúsculas y viceversa, hacer operaciones con variables, abrir y escribir en una nueva ventana etc. En resumen crear nuestras páginas más interactivas.

Para empezar a crear la tienda virtual necesitamos definir las imágenes que se usarán para los productos que se venderán, el fondo de la página y cualquier otra imagen que se crea necesaria para la tienda, las cuales se almacenarán juntas en una carpeta llamada imágenes ya que será más fácil si las tenemos organizadas como se menciono anteriormente en el capítulo 2.

Se crearon 2 archivos html, un archivo llama al otro a través de la acción del botón factura, a su vez esos dos archivos o páginas con la ayuda de JavaScript abren ventanas con diferentes funciones que se verán más adelante.

En ambos archivos lo primero que se hizo fue poner la etiqueta de inicio, la cabecera en donde se define el título, después el cuerpo en donde se pone lo que serían las imágenes, texto, tablas y formularios, al igual que los botones de acción y todo lo que le daría formato a la página, así como los eventos necesarios para llamar a las funciones y eventos creados en JavaScript para darle mayor interactividad y funcionalidad a la página web.

En este caso, el código de JavaScript se colocó después de la etiqueta `</body>`, ya que si se colocaba dentro de la cabecera, provocaba problemas ya que el servidor que aloja la página pone publicidad en la misma y lo hace con JavaScript, pone el código en la cabecera y se encima con el código que se tiene originalmente, por lo que la solución es poner el código al terminar el cuerpo de la página.

En el primer archivo o página se creo un formulario con los productos, el cual se envía por correo electrónico. Los elementos del formulario como imágenes y botones de radio se introducen en una tabla para ordenarlos dentro de la página. Para esta página se crearon las siguientes funciones con JavaScript que en conjunto se usan para la acción de los botones:

Función Guardar: con la cual se crea una cookie o galleta (archivo que se guarda en la maquina del usuario) en la cual se guardan los elementos seleccionados del formulario.

Función Consultar: la cual verifica si hay o no una galleta.

Función VerCarrito: esta función llama a la función Consultar si esta función regresa una respuesta positiva, es decir, si existe una galleta, entonces crea dos arreglos, en el primero inserta el contenido de la galleta con nombre del producto y precio, y en el segundo separa el nombre del precio para después abrir una ventana en la cual se mostraran los productos que ha elegido el cliente o usuario, además de insertar una imagen simulando un carrito de compras.

Función AveriguarPrecio: verifica que elementos están seleccionados, busca los precios y los suma, abre una ventana y muestra el resultado de la suma.

En el segundo archivo o página, se crea un formulario para pedir los datos del cliente, al inicio de la página se inserta la hora y la fecha con ayuda de JavaScript, y se crean las siguientes funciones:

Función FloatFix: que lo que hace es redondear los resultados a dos decimales.

Función ConsultarGalleta: verifica la existencia de una galleta.

Función CompruebaNumero: verifica si una cadena es de números.

Función CompruebaLetra: verifica si una cadena es de letras del abecedario.

Función Valida1: comprueba que en el campo teléfono se encuentren 8 números y si no es así manda un mensaje de error.

Función Valida2: Comprueba que el en campo tarjeta de crédito, se encuentres 16 números y si no es así manda otro mensaje de error.

Función Valida3: Comprueba que en el campo nombre se encuentren letras y si no es así manda un mensaje de error.

Función Valida4: Comprueba que en campo correo electrónico tenga una @ y un punto, si no es así manda un mensaje de error.

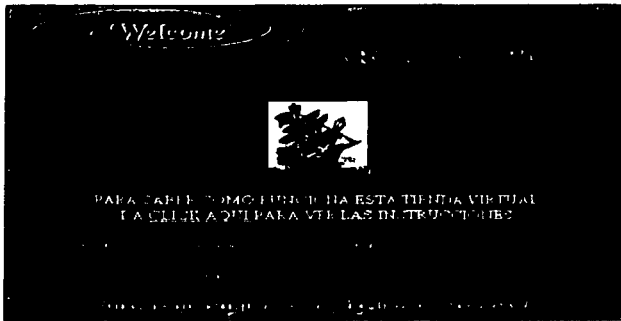
Función Valida: llama de nuevo a todas las funciones de validación al enviar el formulario.

Función Cliente: crea una galleta con los datos del cliente.

Función Fac: esta función revisa las galletas que contienen los datos del cliente y de los productos que selecciono abre una ventana y muestra estos datos en forma de factura, llama a la función floatfix y calcula es descuento si es necesario.

A continuación se mostraran algunas pantallas de la tienda virtual.

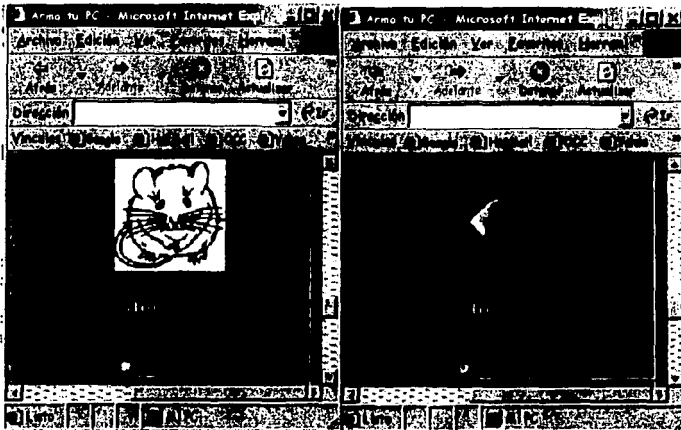
En la siguiente pantalla solamente se uso el lenguaje HTML, con el cual se inserto el título "ARMA TU PC", las imágenes "Welcome" y "Pirata", texto en color blanco y negro, así como un enlace hacia el manual del uso de la tienda que es un documento de word.



Si se quiere proteger el código en la etiqueta body se inserta `'oncontextmenu=""return false''` que es un evento de JavaScript para que no este disponible el código fuente al dar clic con el botón derecho.

Las siguientes imágenes cambian de una a otra al dar un clic en el botón de radio, este evento se logra a través de JavaScript, creando un arreglo de imágenes y llamando con el evento onclick de JavaScript a la imagen al dar clic en el botón de radio.

TESIS CON
FALLA DE ORIGEN



En la siguiente imagen se muestran los botones que tiene la pantalla principal, para la acción de los botones se utilizó las funciones antes descritas.



Para el botón cotizar se utiliza el evento onclick para llamar a la función AveriguarPrecio que lo que hace es sumar los precios de los productos seleccionados, dichos precios se ponen en el valor de cada botón de radio. La pantalla siguiente aparece al oprimir el botón cotizar, esto se logra con la función open de JavaScript y se escribe en ella con la función write el resultado de la suma que devuelve la función.

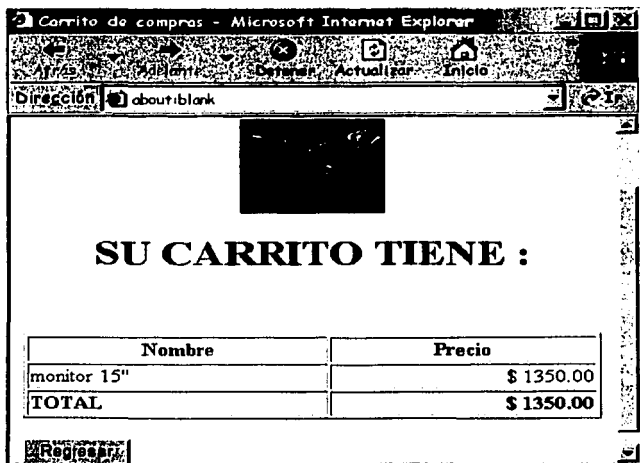
TESIS CON
FALLA DE ORIGEN



El botón poner en el carrito, lo que hace es guardar en una galleta los artículos seleccionados, esto gracias a JavaScript.

EL botón ver carrito, llama a la función `Vercarrito()` creada en JavaScript, que lo que hace es leer la galleta en la cual se encuentran los productos seleccionados y desplegarlos en una ventana en la cual escribimos como en la anterior con la función `write` de JavaScript, cabe mencionar que lo que se escribe en estas ventanas es código HTML. Por ejemplo: `carrito.document.write("<html><head><title>" + "Carrito de compras" + "</title></head><body></body></html>")` lo que hace es escribir en una ventana abierta llamada carrito, el código necesario para ponerle título a la página. Esta ventana también tiene un botón regresar que lo que hace es cerrar la ventana.

La siguiente pantalla aparece al dar clic en el botón ver carrito.

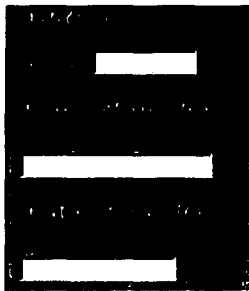


TESIS CON FALLA DE ORIGEN

El botón borrar borra la selección de productos, con el atributo `type="reset"` aquí solo se utilizó HTML, más no borra la galleta la cual se actualiza al dar clic en el botón poner en el carrito.

El botón factura tiene 2 funciones, la primera es mandar por correo electrónico los productos seleccionados, esto se hace con HTML y la otra función es abrir la página en la cual se pedirán los datos para la facturación, estos datos que se piden se validan con JavaScript, cuando el usuario comete un error al escribir los datos este error lo encuentra la función creada con JavaScript para tal cuadro de texto, para decirle al usuario que se ha equivocado utilizamos el evento alert, con el cual se le manda un mensaje al usuario diciéndole por ejemplo que para el teléfono solo se le piden 8 dígitos; el evento que se utilizó en el caso de esta tienda virtual para llamar a las funciones de validación es el evento onchange, es decir, cada que cambiamos de cuadro de texto se llama a la función de validación del cuadro anterior.

La siguiente imagen es una parte de la pantalla que aparece al dar clic en el botón factura, en donde se piden los datos del cliente.




La siguiente pantalla aparece al enviar los datos del cliente, es decir, al dar clic en el botón enviar que se encuentra en la ventana anterior, el cual llama a la función cliente(), que se programa con JavaScript, la cual guarda los datos del cliente en una galleta o cookie para después abrir una ventana en la cual se acomodan en una tabla, los datos del cliente y de los productos seleccionados que se buscan en las galletas previamente generadas para mostrarlos en forma de factura. En esta ventana es donde se muestra el descuento si es que lo hay, se desglosa el IVA y se muestra el total de la compra.

Gracias por comprar con nosotros - Microsoft Internet Explorer

La AT&T... Adelante... Opciones Actualizar Inicio Búsqueda

Dirección



Nombre: Jasmin Gamas Velasco
Sexo: Femenino
Teléfono: 56572556
E-mail: jasmin@mexico.com
Tarjeta de crédito: *****
País: MEXICO
Dirección: azafran 489

Nombre	Precio
monitor 15"	\$ 1350
SUBTOTAL	\$ 1350
DESCUENTO	\$ 0
IVA	\$ 202.5
TOTAL	\$ 1552.50

Control

**TESIS CON
FALLA DE ORIGEN**

CONCLUSIONES

Manejar HTML y JavaScript es de mucha utilidad para cualquier persona que desee hacer su propia página web, para personas que tengan un negocio y requieran tener una página donde ofrezca el bien o servicio de acuerdo a su negocio para ampliar su clientela, o para aquellas personas que se quieran dedicar a realizar páginas web a manera de negocio o a petición de una empresa.

JavaScript es un complemento importante de HTML con el cual interactuamos con el usuario de nuestra página, también proporciona mayor control y versatilidad al diseñar y es apropiado para implantar programas sencillos y pequeños, esta planeado para tareas repetitivas o invocadas por eventos.

Después de realizar esta tesis queda muy claro que tanto HTML como JavaScript son herramientas muy importantes no solo para el actuario enfocado al área de sistemas, sino para cualquier persona que quiera comprender cómo funciona la Internet, que será a futuro el medio de comunicación más utilizado y esto conlleva a que se desarrollen nuevas tecnologías para su optimización.

Cómo aplicación práctica se realizó una tienda virtual llamada "Arma tu PC" para que el lector de esta tesis comprenda de una manera clara el manejo de estas herramientas, la tienda virtual hace cotizaciones, muestra los artículos seleccionados, en caso de compra realiza una factura con los datos del cliente, los artículos vendidos, desglosa el IVA, los descuentos si es que los hubo y el importe total de la compra.

Hay que decir que existen lenguajes de programación para Internet como PHP y ASP que manejan bases de datos y que hacen posible otro tipo de aplicaciones más complicadas, estos lenguajes están fuera del contexto de la tesis.

En cuanto a HTML y JavaScript, se puede decir que una bondad de HTML es que es nemotécnico y muy fácil de aprender pero no permite interacción con el usuario, cómo ventajas de JavaScript se tiene la creación dinámica e interactiva de documentos web, es mucho más fácil empezar a trabajar con JavaScript que con Java, es un lenguaje más simple; JavaScript puede validar formularios, cómo desventajas de JavaScript se tiene que algunos navegadores no lo soportan, lo cual reduce el número de lectores, no se pueden manejar bases de datos, no sirve para aplicaciones muy complicadas.

Uno de los problemas de la web es que contiene mucha información y en algunas ocasiones es difícil encontrar lo que se está buscando. Incluso el mejor motor de búsqueda habrá catalogado menos del 15% de la web. XML (Extensible Markup Language) es parte de la respuesta a este problema, es una especie de primo de HTML, permite que los creadores de páginas web creen sus propias etiquetas de acuerdo a sus necesidades y todos los datos están identificados por las etiquetas XML entre las que

están encerrados. XML será de alguna manera el futuro de la web pero por lo pronto tiene problemas ya que los navegadores anteriores al *Internet Explorer 5* no pueden leer archivos XML, necesitan convertir XML en HTML, además no es tan tolerante como HTML de hecho es quisquilloso, entre otras cosas, le afectan las letras mayúsculas y minúsculas, las comillas, y las etiquetas de cierre. Para un trabajo futuro XML puede ser un tema muy interesante y sería la continuación de esta tesis.

Se sugiere necesario un curso de HTML y JavaScript ya que en un futuro cualquier persona que tenga un servicio que ofrecer y lo quiera ofrecer por éste medio necesitara de una página web y es necesario que las personas sepan crear sus propias páginas evitando así contratar personal especializado para ello y ésta tesis puede servir de guía para dicho curso. La ventaja de HTML es que no se necesitan conocimientos previos de programación para poder hacer una página web, y lo que hace JavaScript es hacerla más interactiva para el usuario.

APÉNDICE

MANUAL DE LA TIENDA VIRTUAL

La función principal de las herramientas de HTML y JavaScript es diseñar y crear páginas web, que son las más utilizadas para mostrar información a través de Internet, por esta razón es importante conocer como elaborarlas, dado que es nuestra tarjeta de presentación ante los usuarios o cibernautas.

La presentación de la página fundamentalmente se realiza a través de HTML, en cambio JavaScript se utiliza principalmente para que los botones contenidos dentro de la página ejecuten acciones, otra función de JavaScript es hacer la página HTML más amigable para poder interactuar con los usuarios de la página más fácilmente.

Los mensaje de error, los cálculos matemáticos del descuento si es que lo hay y la suma del precio de los productos se realizan con JavaScript.

Las páginas que se diseñen pueden ser utilizadas para fines educativos, personales, para mostrar u obtener información o para cuestiones de negocios, se presenta una tienda virtual como una aplicación de lo estudiado hasta ahora y que se desarrollará en este capítulo. Desde el punto de vista actuarial podemos realizar una página que de información acerca de las coberturas que ofrece un seguro en particular o crear una página que nos muestre el estado que guarda un siniestro en cierto tiempo para poder efectuar un seguimiento del mismo.

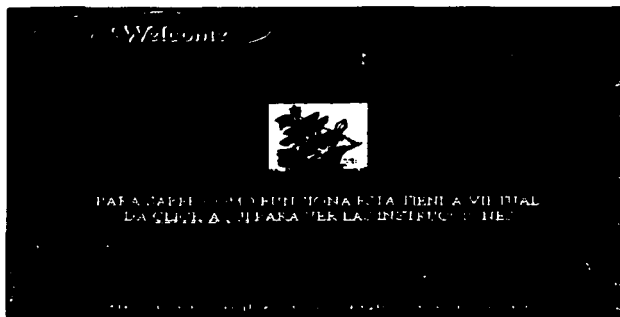
Para este caso práctico se ha creado una tienda virtual llamada ARMA TU PC, cuyo objetivo es vender los aditamentos necesarios para armar una computadora, esta página ha sido puesta en Internet en la siguiente dirección:

<http://www.angelfire.com/amiga/actuaria/armapc2.html>

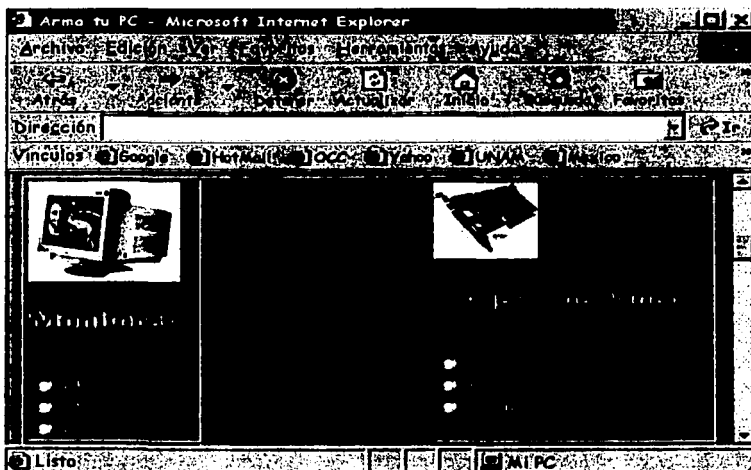
En Internet la veremos con publicidad ya que todas las páginas que son hospedadas de forma gratuita contienen anuncios.

A continuación se presenta de manera detallada las instrucciones para el manejo adecuado de la aplicación referente a la tienda virtual.

Al conectarse al sitio o a la página se muestra una pantalla que contiene todo los productos que se pueden adquirir, en su parte superior aparece una breve explicación de la forma que se puede ser acreedor de algún descuento, así como una liga o enlace que nos envía a un archivo que nos servirá de ayuda para entender el funcionamiento de la misma, este archivo esta en formato zip por lo que es necesario contar con un programa que pueda descomprimirlo como el Winzip.



Más abajo dentro de la página principal se muestran todos los artículos que están disponibles para su venta, como se observa en la siguiente figura:

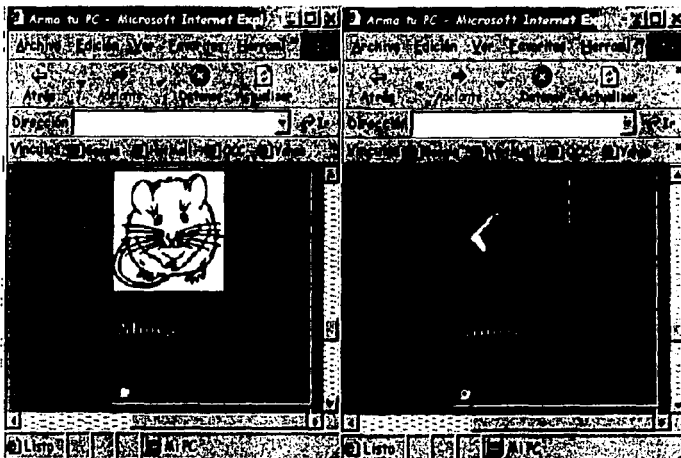


La manera de seleccionar cualquiera de los productos mostrados en la pantalla es dando un clic en el botón de radio (son los círculos que anteceden el nombre del artículo) del producto que deseamos.

TESIS CON
FALLA DE ORIGEN

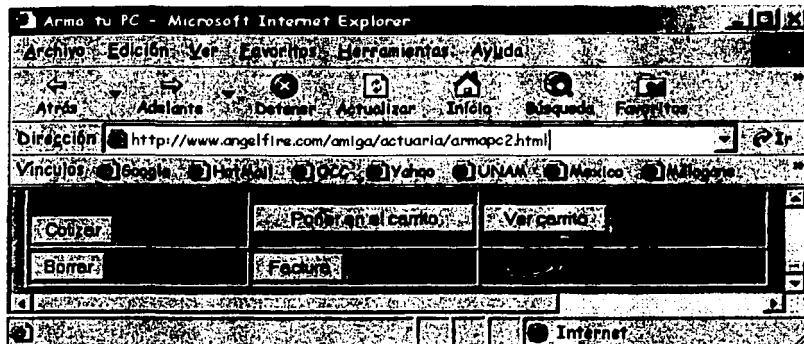
El objetivo de venta de la tienda está orientado para que el usuario pueda armar por sí mismo una PC, por lo cual solamente se puede escoger una cosa de cada grupo, es decir, solamente podemos escoger un monitor por cada compra, si se requieren más monitores se necesitaría efectuar más compras.

Cada vez que seleccionamos un producto la imagen de este cambiará, por ejemplo al seleccionar el mouse se puede observar como cambia la imagen predeterminada del artículo. Esto se ve claramente en las siguientes figuras.



Después que el usuario haya seleccionados los artículos que desee adquirir al final de la página de la tienda virtual se muestra un menú que contiene algunos botones que ejecutan diversas acciones como cotizar, borrar, etc., los cuales se explicarán a detalle más adelante. Enseguida se muestra la pantalla con los botones antes mencionados.

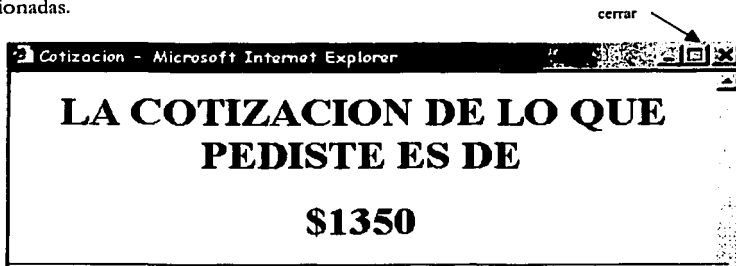
TESIS CON
FALLA DE ORIGEN



A continuación se explica el funcionamiento de cada uno de los botones.

Cotizar: La función de este botón es mostrar al cliente cuanto le costará adquirir los artículos seleccionados. El precio que se muestra en pantalla es el total de todos los artículos que se seleccionaron, para conocer el valor de cada producto en particular se necesita dar un clic al botón de **Poner en el carrito** y después a través del otro botón llamado **Ver carrito** se mostrará el precio de cada artículo, otra opción podría ser seleccionar únicamente un solo artículo y dar clic en **Cotizar**, se puede realizar tanta cotizaciones como el usuario así lo requiera.

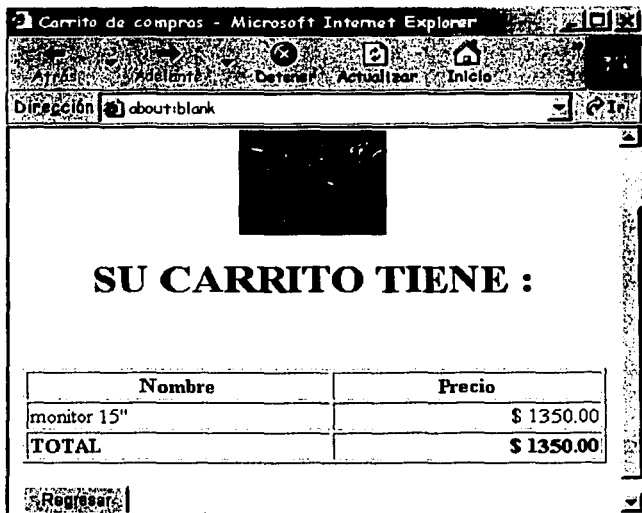
Al dar clic en el botón de **Cotizar** aparecerá la siguiente pantalla. (En este caso se supone que solamente se va adquirir un monitor de 15"). Para salir de esta pantalla es necesario darle un clic al icono de cerrar (✕), inmediatamente después de efectuar esta operación se estará en la pantalla principal, con las opciones previamente seleccionadas.



Borrar: Limpia la pantalla principal, mostrando la página como se vio al inicio al conectarse, es decir, no existe algún artículo seleccionado, esto con el fin de eliminar los artículos que por error se hayan seleccionado para guardarlos en el carrito o para conocer su costo total a través del botón de **Cotización**. También se utiliza cuando se requiere efectuar una nueva cotización.

Poner en el carrito: Este botón lo que hace es almacenar en el carrito los artículos seleccionados por los compradores. Se pueden efectuar tantas selecciones como quiera el usuario, oprimiendo cada vez el botón **Poner en el carrito**, el cual actualizará la información del carrito con lo que se encuentre en ese momento seleccionado, perdiendo los datos anteriormente guardados, por eso mismo no es posible adquirir dos artículos iguales en una misma compra.

Ver carrito: Nos muestra los artículos que hemos almacenado en el carrito mediante el botón antes descrito, así como el precio de cada uno de ellos. Cada vez que seleccionamos el botón de **Poner en el carrito**, se limpia la información que contiene el carrito y se alimenta con la nueva selección. También nos informa del valor total de las compras sin incluir los descuentos ni el IVA, como se observa en la siguiente pantalla.



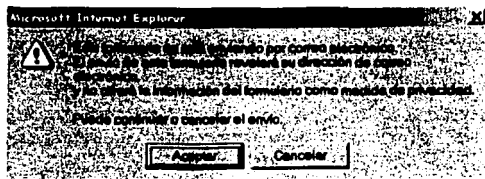
Nombre	Precio
monitor 15"	\$ 1350.00
TOTAL	\$ 1350.00

Regresar

Al dar clic al botón de **Regresar**, se vuelve a mostrar la pantalla de inicio.

TESIS CON
FALLA DE ORIGEN

Factura: Este se utiliza para realizar la compra y emitir la factura correspondiente siempre y cuando se hayan guardado antes los artículos que queremos a través del botón de **Poner en el carrito**. Al dar clic en el botón factura aparece el siguiente mensaje:



En esta pantalla se tiene que dar clic en aceptar para enviar por mail los productos que seleccionamos y así el proveedor se entere de lo que queremos comprar, si se da clic en cancelar no se envía el correo y el proveedor no podrá conocer que artículos se quieren adquirir. Al dar clic en Aceptar nos envía directamente a la aplicación del correo electrónico, al terminar de enviar el correo electrónico se deben llenar los datos que están dentro de la página llamada **Factura**, para acceder a esta hoja es necesario seleccionarla a través de la barra de tareas. Si por error se da clic en cancelar simplemente hay que volver a dar clic al botón **factura**, donde nuevamente se abre otra ventana con el nombre de factura. Es necesario que cierres alguna de las dos ventanas de factura.

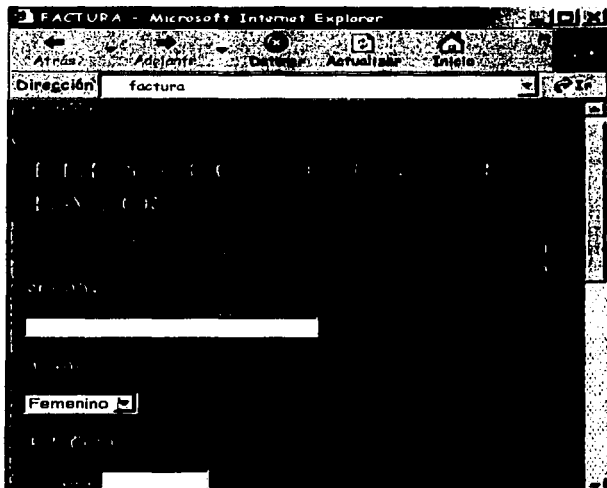


Si la pantalla anterior de envío de correo no aparece, puede ser que no se tenga activada una cuenta de correo y por lo tanto no se puede realizar la compra a menos que envíes un correo a jasmin@mexico.com con los artículos que deseas y los datos que se piden en la pantalla de factura. Si deseas tener una cuenta de correo puedes obtenerla en cualquier servidor de correo, por ejemplo en www.mexico.com.

En ocasiones al abrirse el correo no aparece la dirección del destinatario, en este caso la dirección a la que debes mandar el correo es jasmin@mexico.com.

En esta pantalla se solicitarán algunos datos personales, como los que se muestran a continuación

TESIS CON
FALLA DE ORIGEN



Es necesario llenar los datos correctamente, dado que se utilizarán para conocer al comprador y poder enviarle lo más rápido posible y sin equivocaciones los artículos solicitados, para este fin existen algunas validaciones como es el campo de teléfono donde si escribimos letras o un número de caracteres distinto a ocho, la aplicación nos envía un mensaje de error, así como en el caso de escribir números en la parte correspondiente al nombre. Este mensaje aparece en el momento de seleccionar otro campo de llenado.

Los campos que se deben llenar con mayor cuidado son el referente al correo electrónico y teléfono, dado que por medio de estos contactaremos al cliente en caso de existir alguna duda o aclaración, por esta razón si se tecldea una dirección invalida nos manda un mensaje de error. Otro campo importante es el referente a la tarjeta de crédito, pues es necesario para efectuar la compra.

Por ejemplo si escribimos mal la dirección del correo se mostrará el siguiente mensaje:

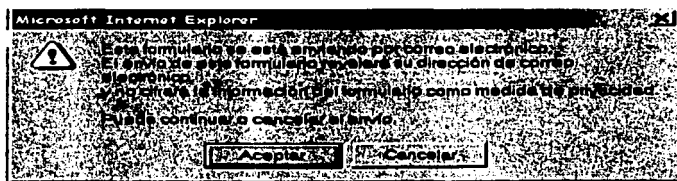


79
TESIS CON ESTA TESIS NO SALE
FALLA DE ORIGEN DE LA BIBLIOTECA

Abajo de la pantalla factura existen dos botones que son: **enviar y regresar**. En el caso de dar un clic en **regresar** se mostrará la página principal, perdiéndose los datos que se capturaron en la página de la Factura. La finalidad de dar clic en **regresar** es el cancelar la compra actual para realizar otra o simplemente salirse de la aplicación sin adquirir ningún producto.

Si los campos de los datos personales no son validos aparecerán nuevamente los mensajes de error al dar clic en **enviar** y la pantalla de envío de correo, a la cual hay que darle cancelar, dado que los datos no son validos y por lo tanto no tiene caso enviarlos.

Cuando llenamos la factura y los campos son validos hay que dar clic en el botón **enviar** y aparecerá inmediatamente un mensaje de envío de mail con la siguiente leyenda.




Al dar clic en el botón de Aceptar, se está enviando por mail los datos personales del comprador. Esta es la manera que tiene el proveedor de los equipos de cómputo de enterarse de las compras realizadas y saber los datos del cliente, por otro lado si da un clic en cancelar simplemente no se podrá efectuar la compra.

También en este caso se abrirá la aplicación del correo electrónico, el cual habrá de cerrarse y después se buscará en la barra de tareas lo que sería nuestra factura final con el nombre "Gracias por comprar", donde aparecerán los datos enviados por correo.

TESIS CON
FALLA DE ORIGEN


Gracias por comprar con nosotros - Microsoft Internet Explorer

Dirección: about:blank



Nombre: Jasmin Gamas Velasco
Sexo: Femenino
Teléfono: 56572556
E-mail: jasmin@mexico.com
Tarjeta de crédito: *****
País: MEXICO
Dirección: azafraan 489

Nombre	Precio
monitor 15"	\$ 1350
SUBTOTAL:	\$ 1350
DESCUENTO	\$ 0
IVA	\$ 202.5
TOTAL	\$ 1552.50



Esta sería la factura final de la compra. El cliente puede realizar tantas compras deseé, mientras se encuentre dentro de la aplicación. En cada compra se deberá de llenar nuevamente los datos personales del cliente.

Es necesario aclarar que la compra será exitosa siempre y cuando se hayan llenado y enviado a través del correo todos los datos, en caso de que el formulario se encuentre incompleto la compra será cancelada.

Al dar clic en el botón cerrar nos regresa a la página de captura de datos personales, la cual hay que cerrar dando clic en el botón regresar y así regresar a la página principal y si así lo queremos realizar otra compra.

**TESIS CON
FALLA DE ORIGEN**

REFERENCIAS

- [1] <http://www.poliedric.com/docs/buscadores1.htm>
- [2] www.brujula.net/cursos/lista.html
- [3] <http://www.0bytes.cl/>
- [4] www.comsto.org/tapas/05/netscape.htm
- [5] <http://microsist.com.mx/noticias/internet/gspin0304.shtml>
- [6] http://www.incrplus.com.ar/sitios/Preguntas_WWW.asp
- [7] <http://www.learnthenet.com/spanish/glossary/glossary.htm>
- [8] <http://www.soho.com.mx/knowledgebase/history/h0005.htm>
- [9] http://www.internautas.org/curso_servidores/
- [10] www.svan.ac.uk/hispanic/coursephss_texto3.htm
- [11] http://www.htmlweb.net/redes/tcp_ip/capa_3/red_2.html
- [12] <http://www.elcodigo.net/index2.html>
- [13] <http://wmaestro.com/webmaestro>
- [14] <http://www.ivia.es/htmlref/htmlref.html>
- [15] http://www.internautas.org/curso_html/
- [16] <http://www.webestilo.com/html/>
- [17] <http://html.programacion.net/js/>
- [18] <http://148.216.5.25/JavaScriptTut/indice.htm>
- [19] <http://www.unav.es/cti/manuales/TutorialJavaScript/indices/>
- [20] <http://wmaestro.com/webmaestro/docs/cap16.html#javascript>
- [21] www.htmlpoint.com/javascript/corso/js_26.htm
- [22] <http://www.elcodigo.net/tutoriales/javascript/javascript.html>
- [23] <http://www.redestb.es/javaaula/cursjava.htm>
- [24] <http://usuarios.tripod.es/Mjavas/NMjava/Servlet/manu1.htm>
- [25] <http://www.ciudadfutura.com/javascriptdesdeccro>
- [26] http://lovecraft.dic.udcc.cl/prog_lang/JavaScript/cap1.htm
- [27] <http://www.internet.com.uy/alvacano/clatajo/estandar/hardware/hardware.htm>

BIBLIOGRAFÍA

- [28] Armand Danesh (1996), "Aprendiendo JavaScript en una semana", Editorial: Samsnet, 535 páginas.
- [29] Armand Danesh y Wes Tatters (1996), "JavaScript 1.1 Developer's guide", Editorial: Samsnet, 591 páginas.
- [30] Bill Anderson (1998), "Netscape Developer's Guide to JavaScript 1.2", Editorial: Prentice Hall, 745 páginas.
- [31] Brent Heslop y Larry Budnick (1995), "HTML Publishing on the internet", Editorial: Ventana Press, 480 páginas.
- [32] Daniel Leduc y Armand ST-Pierre (1999), "HTML creación y difusión de páginas web", Editorial: Trillas, 264 páginas.
- [33] Dick Oliver (1997), "Páginas web mágicas", Editorial: Prentice Hall, 456 páginas.
- [34] Elizabeth Castro (2000), "HTML 4", Editorial: Prentice Hall, 384 páginas.
- [35] Ian S. Graham (1997), "HTML SOURCEBOOK", Editorial: John Wiley & Sons, 620 páginas.
- [36] Isaac Cohen (1996), "CGI/PERL y JavaScript creación de páginas HTML interactivas", Editorial: Ediciones Gestión 2000, 275 páginas.
- [37] Juan Carlos Orós Cabello (2000), "Diseño de páginas web interactivas con JavaScript", Editorial: Alfaomega, 313 páginas.
- [38] Linda Weinman y William Weinman (2002), "Diseño Creativo HTML.2", Editorial: Prentice Hall, 544 páginas.
- [39] Marc Johnson (1996), "JavaScript manual of style", Editorial: Ziff-Davis Press, 277 páginas.
- [40] Vivian Neon y Mimi Recker (1994), "HTML CD an Internet Publishing Toolkit", Editorial: Prentice Hall, 311 páginas.

**TESIS CON
FALLA DE ORIGEN**