

## TEMA 3. LA PUBLICACIÓN DE INFORMACIÓN ELECTRÓNICA

### *Introducción*

En el tema anterior estudiamos los mecanismos mediante los cuales un ordenador es capaz de representar información sencilla, así como la posibilidad de, mediante la aplicación de estructuras complejas o de repetición, representar y gestionar información mucho más extensa y estructurada. En temas posteriores se estudiarán dichas estructuras complejas. En este tema, sin embargo, analizaremos desde un punto de vista estrictamente práctico las posibilidades de explotación de la información contenida en dichas estructuras y en combinaciones de las mismas, estudiando la forma de editar contenidos de forma electrónica y hacerlos accesibles por medios informáticos

### *Representación de la información electrónica. Ficheros de datos*

Hasta ahora hemos hablado en todo momento de representación y gestión de la información por parte del ordenador sin preocuparnos en ningún momento de su ubicación ni el medio de almacenamiento presuponiendo, de algún modo, que dicha información estaba contenida en la memoria primaria del ordenador.

En principio, la única memoria directamente accesible por parte de la unidad de control de un ordenador es la *memoria principal*. Todo lo que deseemos manipular con el ordenador deberá residir en este dispositivo. Sin embargo, la memoria principal es por definición cara, pequeña y, lo que es más importante, volátil, por lo que la información contenida se pierde cuando desaparece el suministro de energía. Por tanto, esta memoria, a pesar de ser el único dispositivo accesible directamente por la CPU, no es adecuada para mantener grandes volúmenes de información de forma permanente. Esto justifica la necesidad de memorias periféricas o secundarias, que son más baratas, con mayor capacidad y no volátiles.

La información en la memoria periférica se organiza en **ficheros**. Definiremos fichero como *un conjunto de información almacenada en algún medio accesible por un ordenador, con una organización conocida y con un método de acceso que nos permita leer o modificar dicha información*.

De esta definición se deduce que existirán diferentes tipos de ficheros en función de la información contenida, y del método de organización de la misma. En temas posteriores se estudiarán con mayor profundidad los soportes, organización y métodos de acceso a los ficheros. Por el momento, podemos establecer una primera clasificación de los mismos según la metodología de codificación de la información, dividiéndolos así en dos tipos básicos:

- Ficheros de texto: En los ficheros de tipo texto guardaremos la información en forma de caracteres, tal y como se mostraría en la pantalla. Por ejemplo, el valor entero '25' se transformará en los caracteres '2' y '5'. Por este motivo las operaciones de escritura y lectura de ficheros de este tipo, serán similares a las utilizadas para escribir en pantalla y leer de teclado.
- Ficheros binarios: En los ficheros binarios la información se guarda tal y como está en memoria, es decir, compuesta por unos y ceros. Por ejemplo, el valor entero '25' se guardará como '00011001'. Los métodos de lectura y escritura en ficheros binarios serán, por tanto, diferentes de los de los ficheros de texto.

Sea de un tipo u otro, cualquier fichero contendrá la información organizada mediante un determinado convenio. Denominaremos *formatos de fichero* a los estándares establecidos para estos convenios de organización de la información dentro de los ficheros..

Desde un punto de vista más práctico, podremos clasificar los ficheros de datos según el tipo de información que guardan. A su vez, para cada tipo de fichero existirán multitud de formatos o estándares diferentes de codificación de los contenidos en los mismos. De este modo, podremos hablar de formatos de ficheros de texto, formatos de ficheros de imagen o formatos de ficheros multimedia.

## ***Explotación de la información: La publicación electrónica***

### **Concepto de edición y publicación electrónica**

Definiremos edición electrónica como la utilización de medios y procedimientos informáticos para crear y almacenar información. Definiremos por tanto publicación electrónica como la utilización de medios y procedimientos informáticos para la difusión de dicha información electrónica. En una definición menos formal, podremos hablar globalmente de la edición y publicación electrónica como del hecho de utilizar medios informáticos para la generación y difusión de información en forma de documentos electrónicos susceptibles de ser leídos y gestionados también por medios informáticos. Es importante observar que hablamos de la creación de *documentos electrónicos* y no de ficheros; cada documento electrónico podrá estar compuesto por uno o más ficheros de datos, no necesariamente del mismo tipo ni formato. Por ejemplo, un documento electrónico como puede ser una página web cualquiera estará formada por al menos un fichero de texto HTML y un conjunto de ficheros adicionales de diferentes tipos, entre los que los más habituales serán los de las imágenes que contenga dicha página.

Una vez creados los documentos electrónicos, será necesario hacerlos llegar a los usuarios que han de consultar la información que contienen. El proceso de difundir los documentos electrónicos para hacerlos accesibles a los usuarios se denominará publicación electrónica. Existirán dos métodos básicos para publicar los documentos electrónicos, la publicación fuera de línea y la publicación en línea.

### **Publicación fuera de línea**

El primer método de publicación de documentos electrónicos consistirá en almacenarlos en un soporte de almacenamiento susceptible de ser leído por el ordenador del usuario, como puede ser un CD-ROM, DVD o cinta.

La publicación de documentos electrónicos fuera de línea implica la necesidad de que los destinatarios de la información dispongan de medios informáticos compatibles los medios de almacenamiento y los formatos usados por el editor. Por parte del editor, son necesarios mecanismos de producción, almacenamiento y distribución de esos medios de almacenamiento.

### **Publicación en línea: Redes de ordenadores.**

El segundo método de publicar documentos electrónicos es introducirlos en un ordenador conectado a una red informática y hacerlos disponibles de forma que mediante algún tipo de protocolo los usuarios puedan consultarlos desde otros ordenadores de la red.

## **La red Internet**

### **Breve historia**

Internet es una red de ordenadores de ámbito mundial formada por la interconexión de un gran número de redes diferentes. Internet es por definición heterogénea, ya que tanto los ordenadores conectados a ella, como las tecnologías de comunicación o incluso los servicios ofrecidos a través de ella son muy diversos.

El origen de Internet está en la red ARPAnet, creada por el Departamento de Defensa de Estados Unidos en los años sesenta. El objetivo de este proyecto era crear una red que conectara los centros de seguimiento de defensa y les permitiera coordinar la información que manejasen, implementando los mecanismos necesarios para que se mantuviese la comunicación entre los centros restantes en caso de que un posible ataque destruyese gran parte de ellos. De esta premisa inicial partió un diseño orientado a la escalabilidad de la red y a la utilización de caminos alternativos para enviar la información en caso de que los caminos más cortos hubiesen sido cortados.

En los años setenta se extendió la red al conectarse las principales universidades de Estados Unidos y algunas del resto del mundo. Esta expansión supuso también un cambio de orientación en la función de la red, pasando de ser un proyecto de comunicaciones gubernamentales de defensa a tener un grupo de usuarios principalmente académico y científico.

En los años ochenta, la red continuó expandiéndose y empezó a extenderse sobre organismos no académicos y empresas privadas. Esto último produjo la aparición de nuevos servicios ya orientados a cualquier tipo de usuario.

Desde los años noventa hasta la actualidad, la red se ha popularizado y se ha continuado extendiendo de forma exponencial. El abaratamiento de las tarifas de comunicaciones y los menores costes del equipo informático, así como el despliegue de redes de banda ancha hasta los domicilios de los posibles usuarios han llevado la red al usuario doméstico, con lo que ha aparecido un gran volumen de servicios orientados a este tipo de público.

### **Conceptos básicos**

La red Internet está formada por una gran cantidad de redes diferentes de ordenadores conectadas entre sí. Además, cada una de estas redes está formada por ordenadores con sistemas heterogéneos. El objetivo básico de comunicar todas estas máquinas de forma eficiente se consigue mediante la adopción de una serie de convenios o estándares:

### **Una filosofía básica de diseño, la arquitectura cliente-servidor**

La filosofía básica de diseño de Internet consiste en diferenciar claramente las funciones del ordenador prestador de un servicio, el *servidor* de las del ordenador usuario de este servicio, el *cliente*. Mediante esta estrategia de diseño de servicios, el cliente utiliza una aplicación informática especializada en pedir y utilizar un servicio, la cual es normalmente compacta y ligera comparada con la existente en el extremo opuesto, el servidor. La aplicación que funciona en el servidor se encarga de prestar el servicio en cuestión a tantos clientes como la soliciten.

Esta estrategia de diseño permite un tipo de operación y gestión de los servicios muy flexible y descentralizada. Además, al separar las aplicaciones clientes y servidoras, ordenadores de plataformas completamente diferentes pueden relacionarse a través de estos servicios con sólo disponer de la aplicación adecuada que cumpla el estándar de ese servicio. De este modo, por ejemplo, un ordenador con sistema operativo Microsoft

Windows puede, con la aplicación cliente adecuada (un cliente de correo), utilizar los servicios de correo electrónico de un servidor con sistema operativo UNIX, en el que funciona una aplicación de servidor de correo.

## **Un protocolo de comunicaciones común, el TCP/IP.**

Para que ordenadores de diferente tipo a través de medios también diferentes, es necesario que exista un lenguaje común, un protocolo de comunicaciones. El principio unificador de Internet es un conjunto de protocolos de comunicaciones llamado TCP/IP (*Transmission Control Protocol-Internet Protocol* o Protocolo de Control de transmisión/Protocolo Internet). **Un protocolo de comunicaciones permite conectar computadoras que utilizan sistemas operativos distintos.** IP está disponible para una amplia gama de técnicas de comunicación, principalmente para Ethernet, que es el estándar para las comunicaciones entre diferentes tipos de plataformas. Puede que no sea el estándar de transmisión por red más rápido, pero es el más soportado y probado.

## **Un sistema de denominación común**

Entre estos estándares se incluye un convenio de denominación de los ordenadores conectados a la red. Una dirección IP es un número que identifica la red y un ordenador (cliente) específico en esa red. Así, cada ordenador de Internet tiene asignado de forma única y no ambigua un identificador denominado *número IP*, formado por cuatro números de 0 a 255 separados por puntos. Este número ip denomina sin ambigüedad un ordenador de la red del mismo modo que un número de teléfono denomina un teléfono concreto de la red telefónica. Por ejemplo, el ordenador con número IP147.156.1.46 es el servidor web de la Universidad de Valencia.

Como a las personas les resulta molesto manejarse con números, la dirección IP puede traducirse como un nombre de dominio: **uv.es**, que a su vez podría dividirse en subredes. Asimismo, cada computadora tiene su propio nombre: **www; ftp; gopher; etc.** Nombres que en este caso nos recuerdan los servicios que brindan, pero que no son obligatorios. Bien podrían llamarse informatica, mural, etc. Así el nombre (hipotético) completo del servidor de web de la universidad de valencia se llama **www.uv.es**.

Los dominios se agrupan de acuerdo con el tipo de organización que los sustenta. Por ejemplo, los dominios de empresas comerciales se llaman **.com**, los gubernamentales, **.gov**, los educacionales, **.edu**, etc. Y para distinguir el país de origen se utiliza otra convención similar. Por ejemplo, el código de España es **.es**, el de Argentina es **.ar**, el de Francia, **.fr**, el de Japón, **.jp**, etc.

## **Un conjunto de servicios estandarizados**

La red Internet implementa un grupo de servicios básicos estandarizados. La estandarización de estos servicios supone la definición clara de la función del servicio, de los protocolos de aplicación y de las funcionalidades básicas de la aplicación cliente y la aplicación servidora. A continuación revisaremos algunos de los servicios básicos que ofrece Internet, para centrarnos finalmente en el más utilizado hoy en día, la World Wide Web.

## Servicios básicos de Internet

### Terminal remoto

*Telnet* es un programa que nos permite establecer una sesión de trabajo con una computadora remota. Por ejemplo se puede usar *Telnet* para conectarnos con un ordenador al otro lado del mundo y una vez que se ha establecido la comunicación podremos usar dicha computadora de la misma forma que utilizamos la nuestra propia. (Por supuesto para ello necesitaremos un identificador de usuario y una *password* ).

*Telnet* actúa como un intermediario entre nosotros y la computadora remota de tal forma que cuando pulsamos una tecla en nuestra computadora ésta se la envía a la computadora remota. El resultado final es que nuestro teclado y pantalla parecen estar conectados directamente a la otra computadora.

En la terminología de *Telnet*, nuestra computadora se denomina local. La otra computadora, con la que el programa *Telnet* se comunica se denomina computadora remota. Se utilizan estos términos sin importar la distancia a la que se encuentran las computadoras: tanto si están al otro lado del mundo o en la misma habitación.

### Transferencia de ficheros

FTP ( File Transfer Protocol o Protocolo de Transferencia de Archivos) se refiere a los programas utilizados en Internet que emplean este protocolo. La transferencia de archivos significa simplemente enviar archivos (no mensajes) de una computadora a otra. Por ejemplo, documentos Word, planillas de Excel, sonidos, programas compilados, etc. Cuando se envía un archivo desde nuestra computadora hacia la remota se denomina “hacer un *upload*” o “*subir*” el archivo. Cuando se copia desde la computadora remota hacia la nuestra, por otra parte, se llama “hacer un *download*” o “*baja*” del archivo.

Existen dos modos básicos de uso para el FTP: uno es como usuario *registrado* de la computadora a donde se accede (con login y palabra clave); el otro, como usuario *anonymous* (anónimo). Como usuario anónimo, los privilegios de acceso dentro del host se ven severamente restringidos por razones de seguridad. Por esta razón, los servidores que proveen *anonymous* FTP hacen esto para que estén disponibles sólo los archivos que ellos desean “publicar”. Para poder utilizar FTP en esta modalidad , se debe ingresar como login *anonymous* y, como palabra clave, nuestra *dirección de correo electrónico*. El uso de la dirección de correo electrónico ayuda al proveedor de FTP a registrar qué usuarios están interesados por su material, y también para seguir el rastro de algún problema de seguridad. FTP anónimo es la primera forma que se utilizó en Internet para publicar información y otros tipos de archivos disponibles para el público. Con el advenimiento de la World Wide Web, actualmente es muy fácil acceder a ésta información utilizando la misma interface.

### Mensajería electrónica

El *correo electrónico* (*e- mail*) es una de las primeras aplicaciones desarrolladas para

Internet, y de las que más se utilizan. El correo electrónico es sencillo de enviar, de leer, responder y administrar.

El correo electrónico es rápido y eficiente, lo que justifica su enorme crecimiento. Partiendo de ser otro servicio ofrecido a los investigadores para comunicar ideas y resultados, ha llegado a convertirse en el sistema de mensajería más sofisticado que hoy conocemos.

Comparado con otras tecnologías de comunicación, como el fax o el correo común, el *e-mail* presenta varias ventajas. Es mucho más fácil escribir un *e-mail* que una carta en papel, por ejemplo. Se han realizado estudios que indican que un *e-mail* se responde más rápidamente que una carta común, sobre todo debido a la facilidad con que se realiza esta tarea.

Los *e-mails* se pueden leer en cualquier momento, sin importar horarios, ni husos horarios. Funciona las 24 horas, los 365 días del año. En caso de caída del servidor los *e-mails* no se pierden, sino que el último receptor los retiene hasta que pueda volver a comunicarse con el servidor donde el receptor tiene su cuenta o “casilla de correo”. Todo esto hace posible mantenernos en contacto con nuestros corresponsales en una base diaria, en la medida que se tomen el tiempo para respondernos. El **e-mail es global**, lo que nos permite tener corresponsales en todo el mundo, y responder en minutos sus consultas.

El **correo electrónico es una de las formas más económicas de comunicarse** por Internet. Es mucho más barato que enviar una carta por vía aérea o hacer una llamada telefónica de larga distancia. Generalmente un *e-mail* se puede despachar al destinatario en cuestión de minutos, y éste puede responder con igual presteza. El *e-mail* no requiere papel y es fácil de descartar. Es una forma excelente de comunicarse, que no agrega problemas al solucionar otros.

En Internet se utiliza un protocolo de la familia de TCP/IP, denominado **SMTP** (*Simple Mail Transfer Protocol* o Protocolo de Transferencia de Correo Simple). SMTP también se utiliza para otras redes locales y de área amplia, aunque compite con otros protocolos disponibles para correo electrónico en LANs como el de Microsoft o el de Lotus. Como los otros “**Protocolos de Transferencia**” de Internet (FTP por ejemplo), SMTP tiene capacidades operativas básicas. El protocolo es utilizado por los programas que se encargan de despachar y recibir la correspondencia electrónica. En el entorno Unix, el programa empleado se llama *Sendmail*, y se encarga de despachar la correspondencia a través de la red, sin importar qué programa originó el e-mail. En realidad, para el sistema de despacho no es importante el origen del e-mail, siempre que el software que lo genere se entienda con el mismo en la forma descrita en el protocolo. Las grandes empresas mantienen funcionando constantemente sus servidores de e-mail para poder mantener el servicio 24 horas/365 días. Si el servidor en cuestión no está conectado en línea en Internet, se comunica con el resto de los servidores por módem en forma regular, cada hora aproximadamente, lo que mantiene el servicio “**casi**” en línea.

Por otra parte, los usuarios no interactúan directamente con los programas como el *Sendmail*, sino que lo hacen con un programa cliente o *front-end*, como puede ser Eudora, Microsoft Mail/Exchange, Outlook, etc., donde redactan y reciben la correspondencia electrónica. SMTP administra los mensajes en colas (llamados también

*spools*). Cuando un usuario envía un mensaje desde su cliente de mail al SMTP, se coloca en la cola de salida. SMTP trata de comunicarse con la máquina de destino, tomando el mensaje de dicha cola, cada vez que se comunica con ella. Si el usuario no existe, o si pasa un determinado tiempo (generalmente dos o tres días) y el servidor no se logró comunicar con la máquina de destino, el mismo SMTP devuelve al remitente el e-mail explicando la causa de la falla. Cuando se establece la conexión entre dos servidores, ambos se “presentan” y luego comienzan a intercambiar correspondencia, en forma ordenada. Primero se envía el encabezado y luego el mensaje en sí. SMTP está preparado para optimizar el tráfico de mensajes, incluso si un mensaje tiene varios recipientes, se transmite una vez a través de la red, y el servidor receptor se encarga de distribuir las copias en forma local. La manera en que se expresan las direcciones en e-mail de Internet respeta la convención estándar de Internet: Usuario@Nombre\_del dominio.

### **Otros servicios.**

Existe una gran cantidad de servicios de Internet aparte de los citados en los puntos anteriores. Muchos de ellos están basados en los anteriores, implementando características adicionales aunque basándose en los protocolos conocidos. Entre estos servicios podemos distinguir el de listas de correo y las noticias de USENET.

### **La World Wide Web**

La World Wide Web (Telaraña de ámbito mundial) es uno de los servicios de Internet más usados hoy en día. WWW es un sistema de información global a través de Internet basado en documentos multimedia enlazados mediante enlaces de hipertexto. Cada documento residente en cada servidor contiene datos y enlaces que permiten saltar a otros documentos del mismo u otro servidor, con lo que se establece un entramado en forma de red o *tela de araña*. En teoría, no hay ninguna razón para limitar los datos contenidos en los documentos a información textual, de modo que un documento puede estar conformado por información textual, gráfica o multimedia, y los enlaces hipertextuales pueden partir o llegar a cualquiera de estos elementos.

La web fue creada en 1989 por un grupo de científicos del CERN (Consejo Europeo para la Investigación Nuclear) con un objetivo muy concreto: compartir información heterogénea de forma flexible, sencilla, escalable y distribuida. Uno de los principales responsables del proyecto, Tim Berners-Lee lo definió como *un sistema de información multimedia distribuido, heterogéneo y distribuido*.

El éxito del servicio web para distribuir información en línea a través de Internet se basa en estas características:

- Es multimedia: Maneja textos, imágenes estáticas, imágenes en movimiento, sonido y, en principio, cualquier información codificable por un ordenador.
- Es distribuido: No existe ningún punto central o crítico donde la información esté concentrada, sino que la información está distribuida por un gran número de servidores en todo el mundo y organizada mediante enlaces hipertextuales dependientes del contenido y no de los condicionantes técnicos, en forma de un gran sistema de información o gigantesco hipertexto.
- Es heterogéneo: Tanto la información como los servicios implementados sobre web son de tipos y funcionalidades muy diferentes. Además, sobre web se han implementado la mayoría de los servicios de Internet que le precedieron tales como las noticias de Usenet, el IRC o el FTP, pero además se han desarrollado nuevos servicios derivados como puede ser el comercio electrónico.

Resumiendo, gran parte del éxito del web radica en que se presenta al usuario como un punto de acceso a un gran volumen de información y servicios de ámbito mundial pero con la ventaja de utilizar interfaces de usuario muy sencillas.

## **Características de la WWW**

Algunas de las características fundamentales de la WWW son:

- Se basa en la arquitectura **cliente-servidor**. Existen máquinas cliente que utilizan servicios, es decir solicitan documentos, de otras máquinas servidoras. En esta estrategia de diseño se separan claramente las funciones de usuario de un servicio y servidor del mismo en aplicaciones separadas que funcionan en ordenadores separados. Naturalmente, para que los clientes y los servidores puedan comunicarse deben utilizar un lenguaje común, un **protocolo**.
- La comunicación se basa en el **protocolo HTTP (HyperText Transfer Protocol, protocolo de transferencia de hipertextos)**. El lenguaje común que comparten el cliente a la hora de realizar las peticiones y el servidor a la hora de satisfacerlas es el HTTP.
- La información está organizada en documentos codificados según un sistema que son capaces de interpretar los ordenadores cliente, el **HTML (HyperText Markup Language)**. HTML es un lenguaje estándar de descripción de documentos basado en marcas e independiente de la plataforma. Cualquier cliente que solicite un documento al servidor web tendrá la capacidad de interpretar el lenguaje HTML y mostrar al usuario la información con la organización y aspecto que el editor de la misma deseaba y codificó mediante este lenguaje de descripción. En las secciones siguientes de este documento estudiaremos todos los aspectos relacionados con la codificación de información en lenguaje HTML.

## **ANEXO Lenguajes de marcas: HTML**

### **Apéndice A: El lenguaje HTML.**

#### **A.1 Introducción.**

HTML (Hyper Text Markup Language) es un lenguaje que describe el formato de un documento en base a la inserción de etiquetas, esto es, palabras reservadas del lenguaje comprendidas entre los símbolos < >. Las etiquetas son de tres tipos:

1. **Pareadas:** Cuando siempre aparece una etiqueta de inicio y otra de final. La etiqueta de final es como la de inicio salvo que va precedida del símbolo / y por supuesto encerrada entre los símbolos < > como toda etiqueta. Por ejemplo: `<H1> Título </H1>`.
2. **Sin parear:** Cuando la etiqueta representa la inserción de un elemento, como una imagen, salto de línea, etc. Por ejemplo: `<IMG> <BR>`.
3. **Opcionales:** Son etiquetas pareadas en las cuales no es necesario emplear la etiqueta final, pues el final de un elemento viene dado por el comienzo de otro, tal y como sucede con los párrafos, etc. Por ejemplo: `<P> Aquí comienza un párrafo <P> Y aquí otro </P>`.

Las etiquetas pueden contener en su interior atributos, los cuales representan opciones que modifican el comportamiento por defecto de la etiqueta. Los atributos pueden aparecer en cualquier tipo de etiquetas, excepto que las etiquetas de final no pueden tener atributos. Los atributos pueden tener asignado un valor característico que se incluye con el formato atributo="valor". Este valor puede ser una opción dentro de una lista de posibilidades prefijadas, un nombre de fichero, un número, etc. Si el valor es una opción dentro de una lista de posibilidades se puede prescindir de las comillas. Algunos ejemplos de atributos son:

```
<P ALIGN=left>  
<IMG SRC="foto.gif">  
<HR NOSHADE>
```

HTML no distingue entre mayúsculas y minúsculas, por lo que las etiquetas pueden escribirse con cualquier combinación de caracteres. Sin embargo, determinados valores de atributos en los que sea importante respetar los cambios entre mayúsculas y minúsculas deberán ir entrecomillados.

#### **A.2 Anidamiento de etiquetas.**

HTML permite anidar etiquetas, es decir, agrupar etiquetas de forma que el efecto de presentación conseguido sea la suma de los efectos de cada etiqueta por separado. Por ejemplo: `<B><I>Este texto saldrá en negrita e itálica</I></B>` muestra el texto en negrita e itálica. Las etiquetas no deben solaparse, esto es, si una etiqueta B se declara después de otra etiqueta A, su etiqueta final /B debe aparecer antes que la etiqueta final de A, esto es /A.

### **A.3 Corrección sintáctica de los documentos.**

Como cualquier lenguaje de programación, HTML tiene un conjunto de reglas que fijan la sintaxis correcta de un documento y se centran en tres aspectos:

- Estructura general de un documento.
- Etiquetas y atributos de las mismas.
- Posibles formas de combinar (anidar) etiquetas.

Aunque todo documento escrito en HTML debería ser correcto sintácticamente, es imposible asegurar que todos los documentos que se encuentran disponibles en la red serán correctos, por lo cual todo cliente Web está implementado de forma que sea capaz de procesar documentos incorrectos sintácticamente. Para ello sigue las siguientes reglas:

- Si se encuentra una etiqueta desconocida se ignora.
- Si se encuentra un atributo con un valor erróneo o fuera de rango, el atributo toma el valor por defecto.
- Si se encuentra una combinación de etiquetas anidadas no permitida, se presenta el efecto conjunto si es posible. En caso contrario sólo se presentará la primera etiqueta procesada.

### **A.4 Color en HTML.**

En HTML es posible especificar distintos colores para el texto, fondo, etc. de los elementos. Esto puede hacerse de dos formas. La primera y más sencilla, consiste en conocer en escribir el nombre del color, por ejemplo “black”, “white”, “green”, “yellow”, “red”, “blue”, etc. Sin embargo, esta forma solo es útil para una serie de colores fáciles de recordar. La segunda forma, válida para especificar todos los colores posibles, consiste en escribir el color de la siguiente forma: “#RRGGBB”, donde RR representa el valor de la componente roja del color, GG el valor de la componente verde y BB el valor de la componente azul. Así, el color blanco es “#FFFFFF”, un tono de gris es “#0F0F0F” (esto es rojo=verde=azul), un tono de verde es “#003D00” (rojo=azul=0) y un tono de morado es “#0F000F” (rojo=azul, verde=0).

### **A.5 Estructura general de un documento HTML.**

Un documento HTML es un fichero de texto plano que utiliza una serie de palabras reservadas para indicar los atributos y estructura de la información que contiene. Para que un cliente Web entienda que un fichero contiene información HTML se le debe dar un nombre que termine con la extensión “.html” o “.htm”. También debe introducirse en la primera línea, aunque no es obligatorio, la etiqueta de comienzo de documento HTML, esto es `<DOCTYPE HTML PUBLIC>`. La estructura general de un fichero HTML es:

```
<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
.....
</HEAD>
<BODY>
.....
</BODY>
</HTML>
```

Como puede observarse, toda la información se encuentra rodeada por las etiquetas `<HTML>` y `</HTML>`. Dentro de ellas se pueden identificar dos partes, el

encabezado, comprendido entre las etiquetas `<HEAD>` y `</HEAD>` y el cuerpo, comprendido entre `<BODY>` y `</BODY>`.

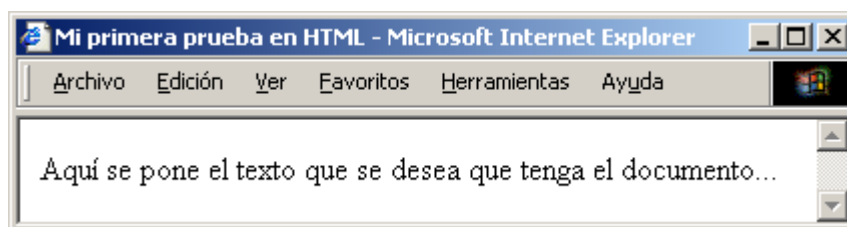
En el encabezado se incluye la información sobre las características del documento. Su contenido no se muestran directamente en el documento, excepto la única etiqueta de inserción obligatoria en el encabezado, denominada `<TITLE>` y `</TITLE>` que muestra la cadena incluida en su interior como título mostrado en la ventana del cliente Web cuando se accede al documento. Esta cadena no debe contener etiquetas de formato.

En el cuerpo se incluye toda la información que forma parte del documento, esto es los datos que se presentarán en pantalla. Los atributos del cuerpo `<BODY>` permiten controlar algunos atributos de la apariencia global del documento, estos atributos son:

Atributo.	Descripción.
BACKGROUND	Especifica la localización de una imagen que será el fondo del cliente.
TEXT	Especifica el color del texto.
LINK	Especifica el color de un link.
VLINK	Especifica el color de un link visitado.
ALINK	Especifica el color de un link marcado.

Por tanto, el formato general de un documento HTML básico es:

```
<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Mi primera prueba en HTML</TITLE>
</HEAD>
<BODY>
Aquí se pone el texto que se desea que tenga el documento...
</BODY>
</HTML>
```



## A.6 Etiquetas para estructurar el texto.

Veremos a continuación las etiquetas permiten dar formato al texto contenido en un documento HTML.

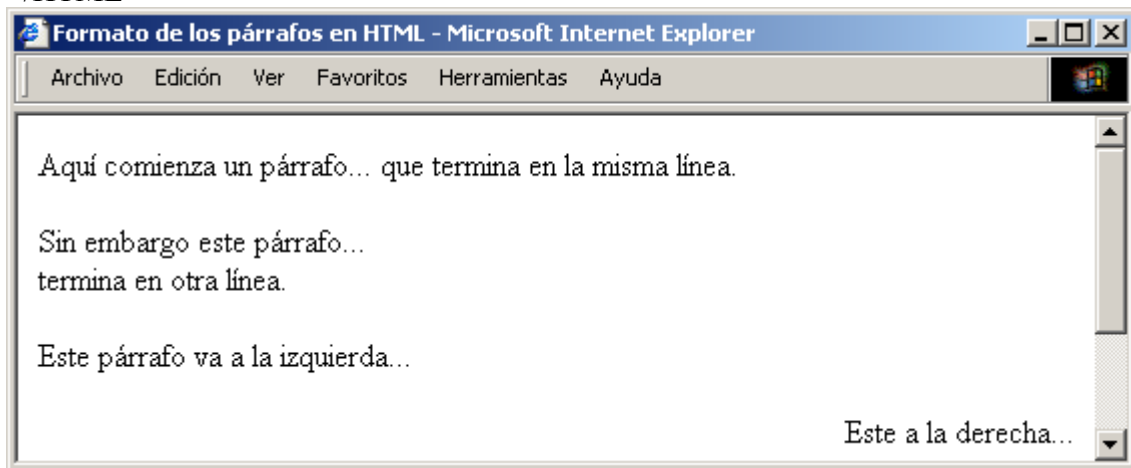
### A.6.1 Párrafos.

El bloque básico de un texto es el párrafo. Dentro de un documento HTML es preciso marcar el comienzo de un párrafo mediante la etiqueta `<P>`. Su terminación sucede al comenzar otro párrafo o bien al aparecer de forma explícita su etiqueta de terminación `</P>`. Como atributo, la etiqueta de inicio de párrafo admite que se le especifique la alineación del párrafo mediante `ALIGN=(left, right, center)`. Si se desea insertar mayor espacio que el que proporciona el fin de un párrafo es necesario introducir saltos de línea, pues varias etiquetas `<P>` seguidas son consideradas como una única etiqueta.

Un salto de línea dentro de un párrafo se especifica mediante la etiqueta `<BR>`. Esta etiqueta admite como atributo `CLEAR=(left, right, center)`, que sirve para

introducir saltos de línea especiales cuando se utilicen “imágenes flotantes” que veremos con posterioridad. Veamos un ejemplo de texto con párrafos y saltos de línea:

```
<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Formato de los párrafos en HTML</TITLE>
</HEAD>
<BODY>
<P>Aquí comienza un párrafo...
que termina en la misma línea.
<P>Sin embargo este párrafo...<BR>termina en otra línea.
<P ALIGN=left>Este párrafo va a la izquierda...
<P ALIGN=right>Este a la derecha...
<P ALIGN=center><BR><BR>Y este centrado y separado...
</BODY>
</HTML>
```

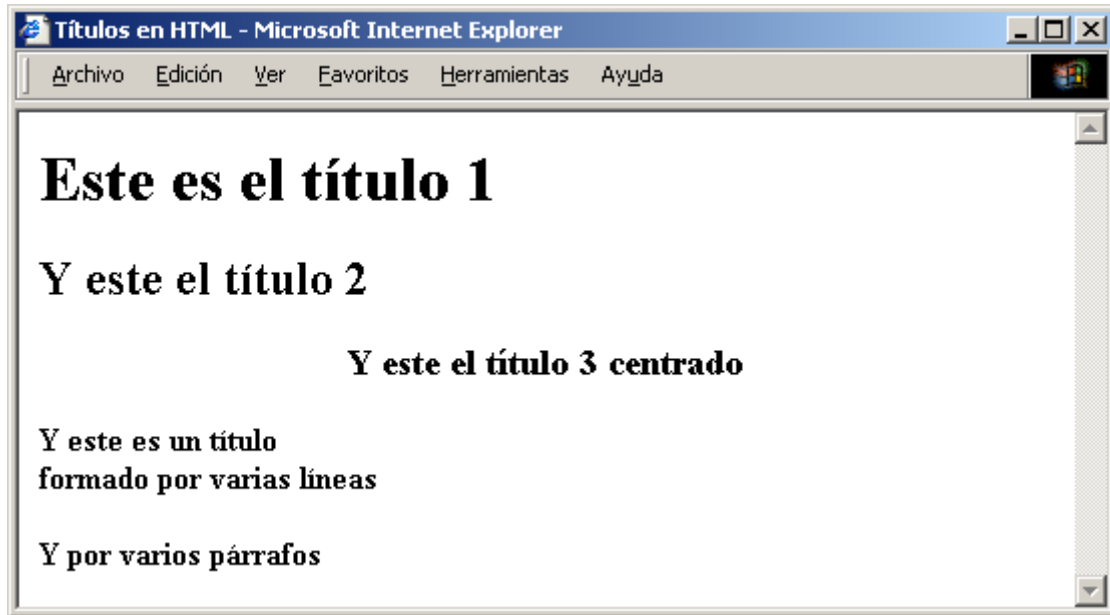


## A.6.2 Títulos.

Los títulos permiten dividir el documento en una jerarquía de secciones, en las que el tamaño de presentación del título se corresponde con la importancia del mismo. Los títulos se indican con las etiquetas `<Hn>` y `</Hn>`, donde *n* es el nivel del título, existiendo seis niveles de títulos, de forma que un número menor se representa más visible que un número mayor. Los títulos admiten como atributo la especificación del alineamiento como `ALIGN=(left, right, center)`. Los títulos insertan un salto de párrafo automático antes y después del mismo. Veamos un ejemplo de títulos:

```
<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Títulos en HTML</TITLE>
</HEAD>
<BODY>
<H1>Este es el título 1</H1>
<H2>Y este el título 2</H2>
<H3 ALIGN=center>Y este el título 3 centrado</H3>
<H4><P>Y este es un título<BR>formado por varias líneas<P>Y por varios
párrafos</H4>
</BODY>
```

</HTML>



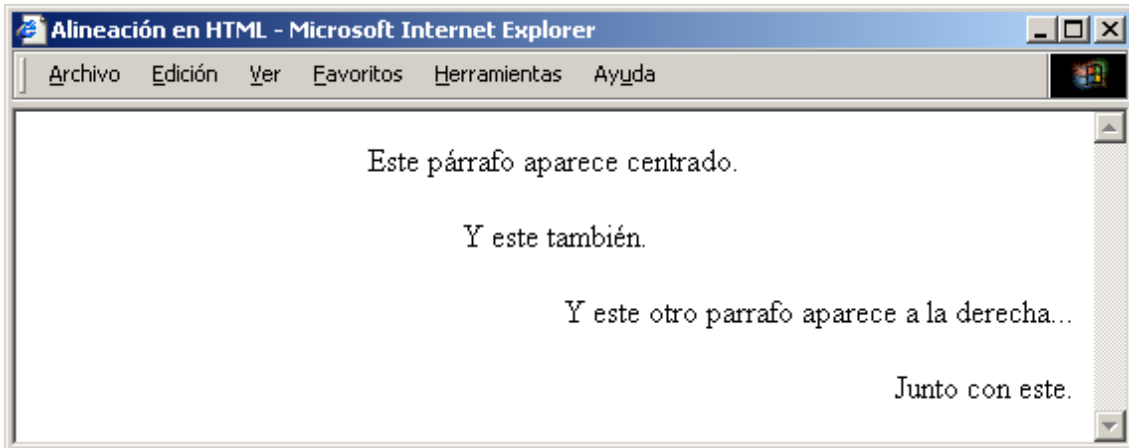
### A.6.3 Alineación de elementos.

Existen algunas etiquetas que permiten especificar la alineación general de los elementos. Estas etiquetas solo funcionan en algunos clientes Web, por lo cual solo comentaremos aquí las dos más extendidas.

La etiqueta `<CENTER>` `</CENTER>` centra todos los párrafos de texto, etc., que se encuentren entre ellas.

La etiqueta `<DIV>` `</DIV>` especifica la alineación de todos los elementos HTML incluidos entre ellas. Su atributo es `ALIGN=(left, right, center)`. Veamos un ejemplo de uso de estas dos etiquetas:

```
<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Alineación en HTML</TITLE>
</HEAD>
<BODY>
<CENTER>
<P>Este párrafo aparece centrado.<P>Y este también.
</CENTER>
<DIV ALIGN=right>
<P>Y este otro parrafo aparece a la derecha...<P>Junto con este.
</DIV>
</BODY>
</HTML>
```



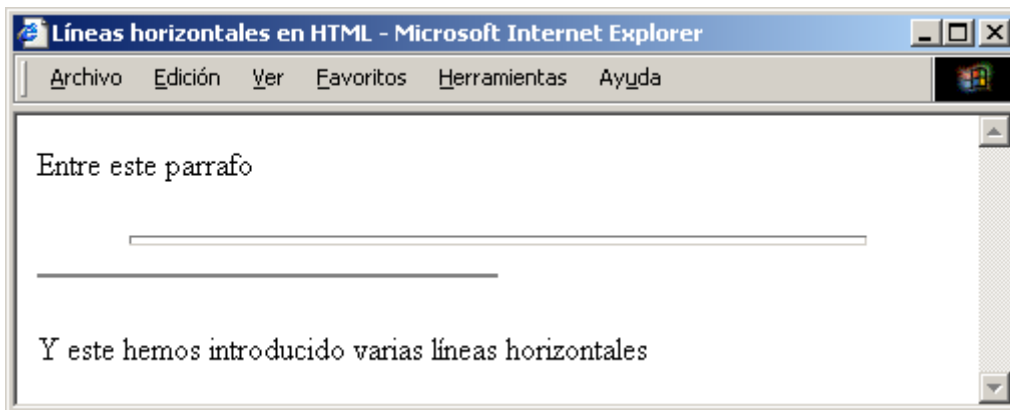
#### A.6.4 Líneas horizontales.

Las líneas horizontales son muy útiles para separar diferentes secciones en un documento HTML. Las líneas horizontales se introducen con la etiqueta `<HR>`, la cual permite especificar diferentes propiedades con los atributos que se muestran en la tabla siguiente:

Atributo	Descripción	Valor por defecto.
SIZE=	Marca el grosor de la línea.	2
WIDTH=	Marca la anchura horizontal de la línea. Se puede expresar en pixels o en porcentaje.	El ancho de la pantalla.
ALIGN=	Especifica la alineación de la línea, puede ser left, right o center.	Si no se especifica, las líneas están centradas.
NOSHADE	Muestra una línea sólida sin apariencia tridimensional.	Línea con apariencia tridimensional.

Veamos un ejemplo de inserción de líneas horizontales:

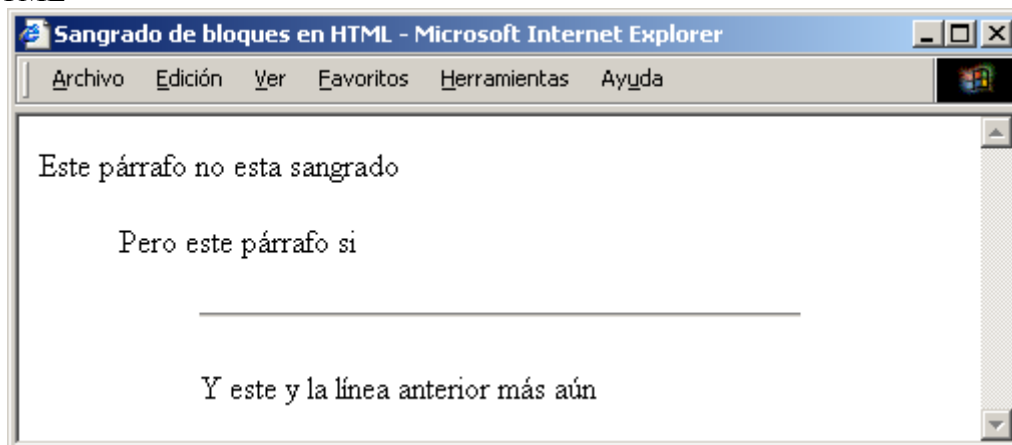
```
<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Líneas horizontales en HTML</TITLE>
</HEAD>
<BODY>
<P>Entre este párrafo</P>
<HR SIZE=5 WIDTH=80%>
<HR WIDTH=50% ALIGN=left NOSHADE>
<P>Y este hemos introducido varias líneas horizontales</P>
</BODY>
</HTML>
```



### A.6.5 Sangrado de bloques.

El sangrado de bloques permite definir una sección del documento a la cual se le aplicara un sangrado. El sangrado se define mediante las etiquetas `<BLOCKQUOTE>` `</BLOCKQUOTE>` y pueden anidarse. El sangrado inserta automáticamente un salto de párrafo antes y después del bloque. Veamos un ejemplo de sangrado de bloques:

```
<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Sangrado de bloques en HTML</TITLE>
</HEAD>
<BODY>
<P>Este párrafo no esta sangrado</P>
<BLOCKQUOTE>
<P>Pero este párrafo si</P>
<BLOCKQUOTE>
<HR>
<P>Y este y la línea anterior más aún</P>
</BLOCKQUOTE>
</BLOCKQUOTE>
</BODY>
</HTML>
```



### A.6.6 Texto preformateado.

En general, no se tiene control sobre la presentación del texto de un documento, siendo el cliente Web el que se encarga de procesar el contenido y mostrarlo en función de las características de la ventana, tipo de letra, etc. Sin embargo, puede ser útil en algunos casos disponer de la capacidad de especificar la posición de un texto en la página. Esto se realiza con las etiquetas `<PRE>` `</PRE>`. Estas etiquetas hacen que se muestre el texto con el mismo formato que contiene en el documento HTML, de forma que se respetan los espacios en blanco, saltos de línea, etc.

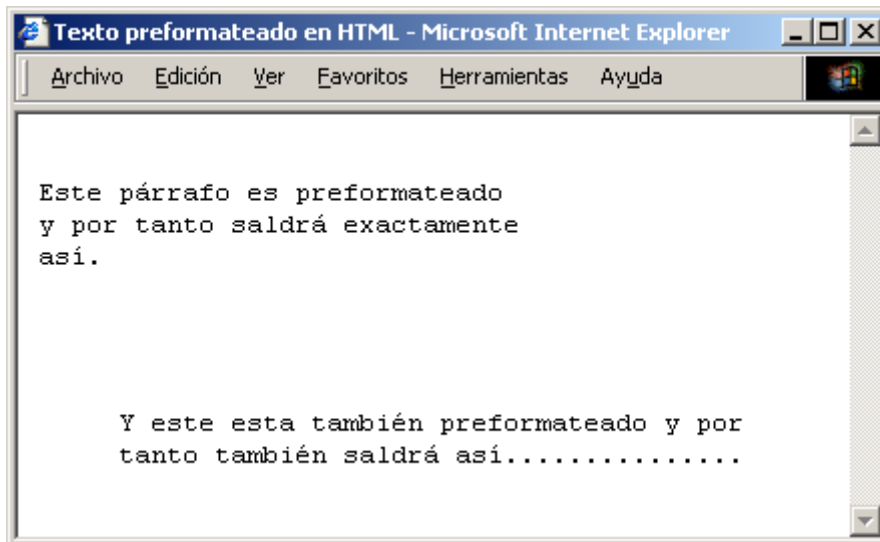
La etiqueta `<PRE>` admite un atributo, `WIDTH=(40,80,132)`, que especifica el ancho de línea a utilizar para mostrar el texto, siendo el cliente Web el encargado de seleccionar la fuente adecuada para cada presentación. Todo bloque `<PRE>` añade un salto de párrafo delante y detrás del mismo. El valor por defecto si no se especifica nada es de 80 caracteres. Veamos un ejemplo:

```
<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
```

```

<TITLE>Texto preformateado en HTML</TITLE>
</HEAD>
<BODY>
<PRE>
<P>
Este párrafo es preformateado
y por tanto saldrá exactamente
así.
</P>
</PRE>
<PRE WIDTH=40>
<BLOCKQUOTE>
<P>
Y este esta también preformateado y por
tanto también saldrá así.....
</BLOCKQUOTE>
</PRE>
</BODY>
</HTML>

```



### A.6.7 Direcciones.

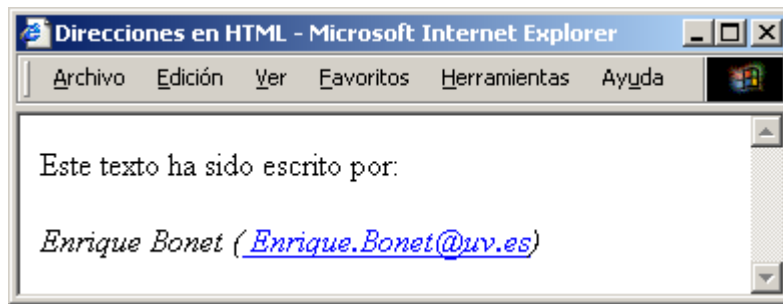
Las direcciones se utilizan para incluir datos sobre el autor del documento, su dirección, etc. Se introducen con la etiqueta `<ADDRESS>` `</ADDRESS>`. Veamos un ejemplo:

```

<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Direcciones en HTML</TITLE>
</HEAD>
<BODY>
<P>Este texto ha sido escrito por:</P>
<ADDRESS>
Enrique Bonet (<A HREF="mailto:Enrique.Bonet@uv.es">
Enrique.Bonet@uv.es</A>)
</ADDRESS>

```

```
</BODY>
</HTML>
```



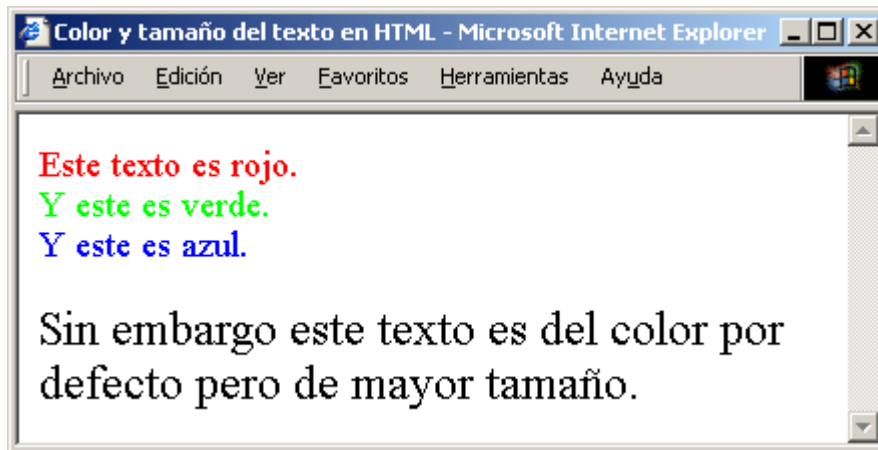
### A.6.8 Color y tamaño del texto.

En HTML es posible definir el color que deseamos que posea el texto. Esto se realiza con la etiqueta `<FONT> </FONT>`. La etiqueta posee dos atributos que pueden verse en la tabla siguiente:

Atributo	Descripción
COLOR	Especifica un color para el texto.
SIZE	Especifica el tamaño de la fuente de texto, puede ir de 1 a 7 y puede especificarse como valor absoluto "1", "5", etc. o como valor relativo "+1", "-3".

La base del tamaño fuente se especifica mediante la etiqueta no pareada `<BASEFONT>`, la cual especifica el tamaño por defecto de la fuente mediante el atributo `SIZE`. Por defecto el tamaño es 3. Un ejemplo de uso puede verse a continuación:

```
<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Color y tamaño del texto en HTML</TITLE>
</HEAD>
<BASEFONT SIZE="4">
<BODY>
<P>
<FONT COLOR="#FF0000">Este texto es rojo.<BR></FONT>
<FONT COLOR="#00FF00"> Y este es verde.<BR></FONT>
<FONT COLOR="#0000FF">Y este es azul.</FONT>
</P>
<P><FONT SIZE="+2">Sin embargo este texto es del color por defecto pero de mayor tamaño.</FONT></P>
</BODY>
</HTML>
```



### A.6.9 Comentarios.

Los comentarios permiten definir un bloque de texto como un comentario. Todo el texto contenido entre la etiqueta de comentario es ignorado por el cliente Web. La etiqueta de introducción de un comentario es: `<!-- -->`. Un ejemplo de comentario, aunque lógicamente no podremos mostrar la salida pues los comentarios son ignorados por el cliente Web es: `<!-- Este es un comentario en un documento HTML -->`.

### A.7 Etiquetas de formato del texto.

Las etiquetas permiten especificar el formato de presentación del texto, pudiendo considerarse el equivalente al “formato de caracteres” de los procesadores de texto. Las etiquetas son de dos tipos:

1. **Etiquetas de formato lógico:** Clasifican los bloques de texto dentro de una serie de clases o tipos predefinidos.
2. **Etiquetas de formato físico:** Fijan el estilo a utilizar en la presentación de un texto, permitiendo mostrarlo en negrita, itálica, etc.

Veamos ambos tipos de etiquetas.

#### A.7.1 Etiquetas de formato lógico.

Los estilos lógicos clasifican el texto en una serie de grupos generales de formato. Cada estilo lógico se muestra con unos determinados atributos. El anidamiento de los estilos lógicos es permitido, aunque carece de sentido por la propia esencia del estilo lógico. Los estilos lógicos existentes y su explicación se encuentran en la siguiente tabla.

Estilo lógico.	Descripción
<code>&lt;CITE&gt; &lt;/CITE&gt;</code>	Cita literal de un texto.
<code>&lt;CODE&gt; &lt;/CODE&gt;</code>	Presentación de código fuente.
<code>&lt;EM&gt; &lt;/EM&gt;</code>	Aplica énfasis al texto.
<code>&lt;KBD&gt; &lt;/KBD&gt;</code>	Secuencias de ordenes o comandos que debe introducir el usuario.
<code>&lt;SAMP&gt; &lt;/SAMP&gt;</code>	Insertar una secuencia de caracteres literales.
<code>&lt;XMP&gt; &lt;/XMP&gt;</code>	Inserta una secuencia de caracteres literales pero ignora todas las etiquetas HTML de su interior.
<code>&lt;STRONG&gt; &lt;/STRONG&gt;</code>	Aplica énfasis al texto.
<code>&lt;VAR&gt; &lt;/VAR&gt;</code>	Define el nombre de una variable.

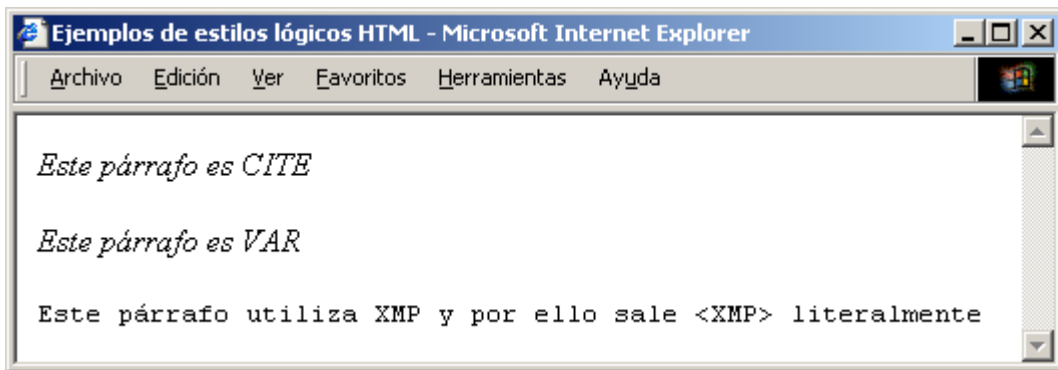
Veamos un ejemplo de uso de los estilos lógicos:

```
<DOCTYPE HTML PUBLIC>
<HTML>
```

```

<HEAD>
<TITLE>Ejemplos de estilos lógicos HTML</TITLE>
</HEAD>
<BODY>
<P><CITE>Este párrafo es CITE</CITE>
<P><VAR>Este párrafo es VAR</VAR>
<P><XMP>Este párrafo utiliza XMP y por ello sale <XMP> literalmente</XMP>
</BODY>
</HTML>

```



## A.7.2 Etiquetas de formato físico.

Las etiquetas de estilos físicos fijan el formato explícito de un conjunto de elementos de texto. El anidamiento de estilos físicos está permitido y tiene sentido, pues es posible querer que un texto tenga un tipo de letra determinado (por ejemplo itálica) y además este por ejemplo en negrita. Los principales estilos físicos y su explicación se encuentran en la tabla siguiente:

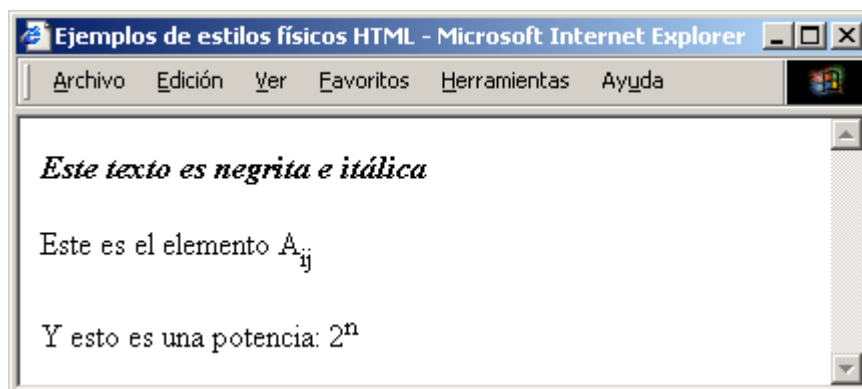
Estilo físico.	Explicación.
<B> </B>	Muestra el texto en negrita.
<I> </I>	Muestra el texto en itálica.
<TT> </TT>	Muestra el texto como una máquina de escribir.
<U> </U>	Subraya el texto marcado.
<SUB> </SUB>	Muestra el texto como un subíndice.
<SUP> </SUP>	Muestra el texto como un superíndice.
<BIG> </BIG>	Muestra el texto con un tamaño superior al por defecto.
<SMALL> </SMALL>	Muestra el texto con un tamaño inferior al por defecto.
<BLINK> </BLINK>	Muestra el texto parpadeante.
<S> </S>	Muestra el texto con una tachadura.

Veamos un ejemplo de uso de los estilos físicos.

```

<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Ejemplos de estilos físicos HTML</TITLE>
</HEAD>
<BODY>
<P><B><I>Este texto es negrita e itálica</I></B></P>
<P>Este es el elemento A<SUB>ij</SUB></P>
<P>Y esto es una potencia: 2<SUP>n</SUP></P>
</BODY>
</HTML>

```



## A.8 Caracteres especiales.

Los documentos HTML están contruidos a partir de caracteres del juego ISO-Latin-1 de 256 caracteres. Sin embargo, no todos los clientes Web contienen elementos comunes para mostrar los 256 caracteres. Generalmente, los juegos de caracteres empleados contienen elementos comunes hasta el 128, donde se incluyen todos los caracteres alfabéticos y numéricos junto con la mayoría de símbolos presentes en el teclado.

Para el resto de caracteres especiales, es preciso disponer de una representación de los mismos que sea independiente del entorno. La solución pasa por asignar a cada carácter especial una nueva etiqueta HTML, denominada referencia a carácter, que se puede incluir en el documento de dos formas:

1. Por su número, con el formato `&#nnn;` (nnn es el código decimal del elemento).
2. Por su nombre mnemotécnico, según se recoge en ISO-LATIN-1.

Por ejemplo, la á se puede representar como `&#225;` o `&aacute;`. El ; final indica el final de la etiqueta. Es de resaltar que algunos caracteres, tales como `<`, `>`, `&`, `#` y `“`, no pueden ser incluidos directamente dentro de un documento ya que serían interpretados como parte del código HTML. Por ello se deben utilizar sus códigos mnemotécnicos. Los principales código mnemotécnicos pueden encontrarse en la siguiente tabla:

Carácter	Código	Carácter	Código	Carácter	Código	Carácter	Código
<	&lt;	>	&gt;	&	&amp;	#	&#35;
“	&quot;	á	&aacute;	é	&eacute;	í	&iacute;
ó	&oacute;	ú	&uacute;	Á	&Aacute;	ñ	&ntilde;
Ñ	&Ntilde;	®	&reg;	@	&copy;	Espacio no divisible.	&nbsp;

El carácter `&nbsp;` puede ser útil para hacer que dos palabras no sean cortadas por un salto de línea automático dentro de un párrafo.

## A.9 Listas.

Las listas son grupos de párrafos a los que se aplica un formato de presentación común. Existen en HTML cuatro tipos de listas:

1. **Listas desordenadas:** Cada elemento de la lista incluye una marca de párrafo (generalmente un punto grande) y un margen interior. Las listas desordenadas se representan mediante la etiqueta `<UL> </UL>`, admitiendo como atributo `TYPE=(circle, disc, square)` que indica el formato de la marca de párrafo.
2. **Listas ordenadas:** Como las anteriores solo que la marca de párrafo se sustituye por un número de orden. Las listas ordenadas se representan mediante la etiqueta `<OL> </OL>`, admitiendo como atributos `START=` que especifica el número de

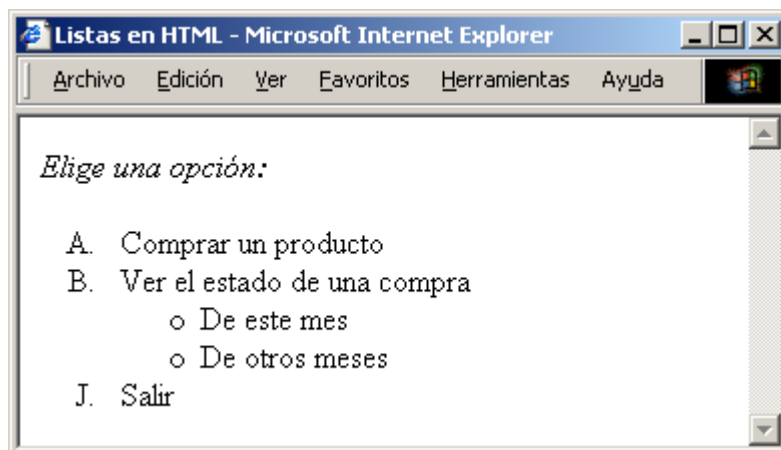
orden del primer elemento de la lista y TYPE="(A, a, I, i, 1)", que especifica el tipo de numeración a emplear.

3. **Menús:** Listas desordenadas que se muestran de forma más compacta que las ordenadas. Se representan mediante la etiqueta `<MENU>` `</MENU>`.
4. **Listas de elementos cortos:** El cliente Web decide como las muestra, si como listas desordenadas o bien en columnas paralelas. Se representan con la etiqueta `<DIR>` `</DIR>`.

Cada uno de los elementos de la lista se representa por la etiqueta de tipo opcional `<LI>` `</LI>` (recordar que en una etiqueta de tipo opcional puede no aparecer su terminador). Esta etiqueta permite los atributos VALUE=, que altera el número de índice del elemento en una lista ordenada y TYPE="(A, a, I, i, 1, circle, disc, square)" que altera la marca de párrafo según la lista sea desordenada u ordenada.

Las listas pueden anidarse unas dentro de otras. Las listas interiores presentarán sangrados adicionales respecto a la anterior. Como siempre es necesario respetar las reglas de anidado de etiquetas. Veamos un ejemplo de uso de listas:

```
<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Listas en HTML</TITLE>
</HEAD>
<BODY>
<P><EM>Elige una opción:</EM></P>
<OL START=1 TYPE="A">
<LI>Comprar un producto</LI><BR>
<LI>Ver el estado de una compra</LI><BR>
<UL TYPE=circle>
<LI>De este mes</LI><BR>
<LI>De otros meses</LI><BR>
</UL>
<LI VALUE=10>Salir</LI>
</OL>
</BODY>
</HTML>
```

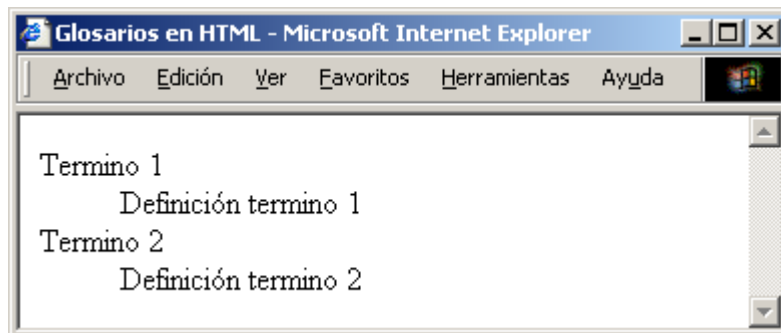


### A.9.1 Glosarios de términos.

Son listas preparadas para realizar un glosario de términos, en el que junto a cada elemento se incluye una descripción del mismo. Sigue las mismas reglas que el resto de las listas. Se marcan con la etiqueta `<DL>` `</DL>`. La etiqueta `<DT>` `</DT>`

indica que un párrafo es un termino del glosario y la etiqueta `<DD>` `</DD>` indica que el párrafo es una definición que puede tener varias líneas. Las dos últimas etiquetas son de tipo opcional con lo cual puede no aparecer la marca del final. Veamos un ejemplo de su uso:

```
<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Glosarios en HTML</TITLE>
</HEAD>
<BODY>
<DL>
<DT>Termino 1</DT>
<DD>Definición termino 1</DD>
<DT>Termino 2</DT>
<DD>Definición termino 2</DD>
</DL>
</BODY>
</HTML>
```



## A.10 Enlaces entre páginas.

La posibilidad de crear enlaces entre diferentes páginas es la principal aportación del Web. Para implementarlo, se necesita la capacidad para localizar, a través de una dirección, cualquier elemento del universo de Internet. Esto se realiza mediante el empleo de URLs (Universal Resource Locator) que asigna una dirección a cualquier recurso disponible en Internet. Los enlaces pueden ser de dos tipos:

1. **Enlaces de destino:** Representan elementos de un documento a los que se puede hacer referencia desde otro punto del mismo o desde otro documento.
2. **Enlaces de origen:** Marcan los elementos que al ser seleccionados, activarán una URL o un enlace de destino.

Los enlaces, tanto de destino como de origen, se seleccionan con la etiqueta `<A>` `</A>`, dentro de la cual es necesario especificar uno y solo uno de los dos atributos de la tabla siguiente:

Atributo.	Descripción.
NAME="..."	Define un enlace de destino. En la zona entrecorillada se debe incluir un nombre, único dentro de este documento, que servirá de referencia del bloque seleccionado.
HREF="..."	Define un enlace de origen. En la zona entrecorillada se debe incluir el URL que el cliente recogerá ante la activación del enlace. La URL podrá incluir un enlace de destino.

Además de estos dos atributos, la etiqueta `<A>` posee otro atributo, el atributo `TARGET="nombre"`, donde nombre será el nombre especificado en el atributo NAME

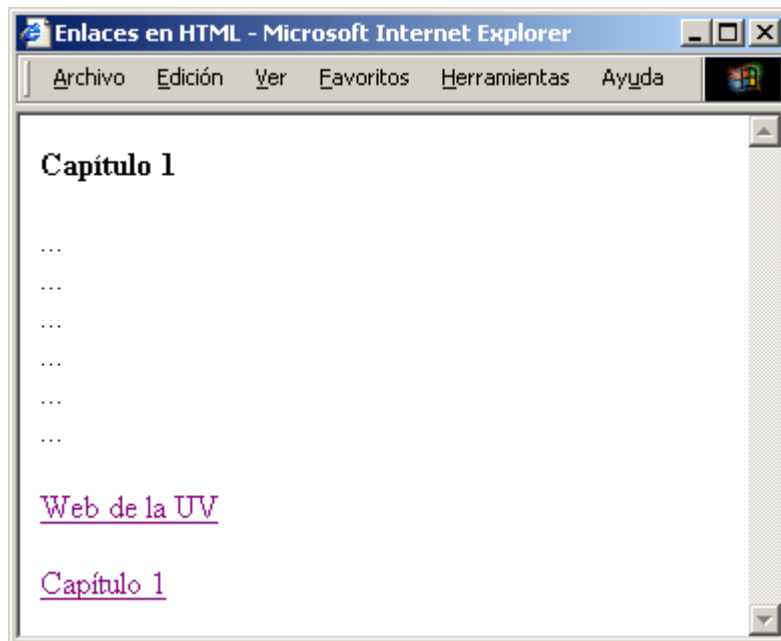
de la etiqueta <FRAME> que veremos con posterioridad al explicar los marcos (frames).

Las URLs solo pueden contener en su interior caracteres ASCII imprimibles (aquellos por debajo del código 128 excepto los caracteres de control). Existen tres caracteres especiales en las URLs:

- El carácter / se utiliza como separador de directorios.
- El carácter # representa una meta local dentro de un documento.
- El carácter % se utiliza para codificar caracteres no permitidos en la URL. Estos caracteres se codifican con el formato %xx, donde xx es el código hexadecimal del carácter. Además, los tres caracteres especiales deben representarse de esta forma, esto es, el / se representa como %2F, el # como %23 y el % como %25.

Veamos un ejemplo de uso de enlace entre páginas Web:

```
<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Enlaces en HTML</TITLE>
</HEAD>
<BODY>
<P><A NAME="Capítulo1"><B>Capítulo 1</B></P>
<P>...<BR>...<BR>...<BR>...<BR>...<BR>...</P>
<A HREF="http://www.uv.es">Web de la UV</A><BR><BR>
<A HREF="#Capítulo1">Capítulo 1</A>
</BODY>
</HTML>
```



### **A.11 Inclusión de imágenes.**

Los clientes Web actuales permiten la inclusión de imágenes de formato GIF (extensión .gif) y de formato JPEG (extensión .jpg o .jpeg). Cuando se recibe una imagen de tipo desconocido (bitmap, tiff, etc.), el cliente Web intenta localizar un aplicación capaz de mostrarla mediante su descripción MIME (Multipurpose Internet Mail Extensions).

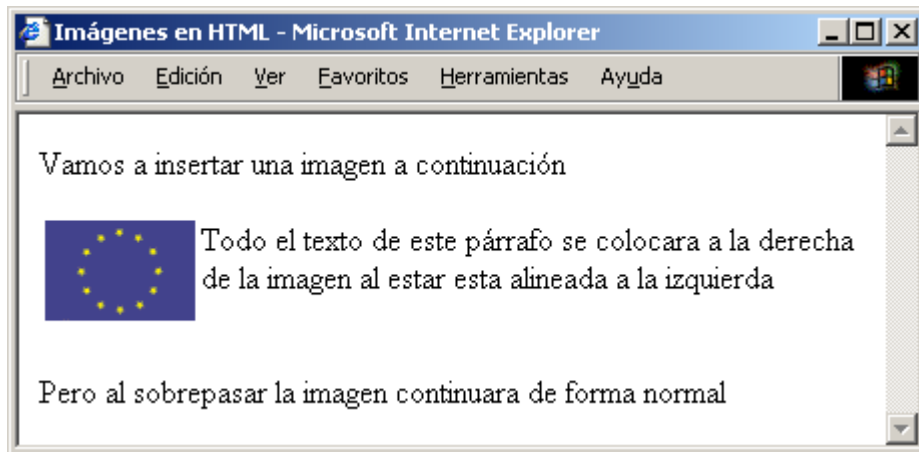
La forma básica de incluir imágenes es hacer un enlace al fichero gráfico, de forma que al activarlo se mostrará la imagen en una pantalla vacía. Por ejemplo: *Pulsa*`<A HREF="/home/quique/quique.jpg">aquí</A>`*para ver mi foto* permite abrir una pantalla en la cual se mostrara una imagen.

Además de mediante enlaces, HTML permite insertar imágenes dentro de un documento mediante la etiqueta sin parrear `<IMG>`. Cuando una imagen es insertada en un documento, va unida al párrafo de texto donde se inserta. La etiqueta `<IMG>` presenta los siguientes atributos:

Atributo	Descripción
SRC="..."	Contiene la URL de la imagen a incluir. Este atributo es obligatorio.
ALT="..."	Asigna un texto descriptivo relacionado con la imagen.
ALIGN=	Controla la alineación de la imagen con respecto al párrafo en que esta insertada. Los valores que puede tomar son: top, middle, bottom, texttop, baseline, absmiddle y absbottom.
ALIGN=	Control de la alineación de la imagen, puede tomar los valores left y right y permite que un párrafo fluya sobre el borde de una imagen, bloqueando ésta a la izquierda o derecha del margen. No puede utilizarse junto con la anterior.
HEIGHT=	Informa al cliente Web de la altura de la imagen a insertar.
WIDTH=	Informa al cliente Web de la anchura de la imagen a insertar.
ISMAP	Informa al cliente Web de que esta imagen es un mapa activo.
USEMAP	Informa al cliente Web del archivo con el mapa activo de la imagen.
BORDER=	Fija la anchura del borde que rodeará a la imagen. Un valor de 0 desactiva el borde.
HSPACE=	Fija la anchura horizontal de los márgenes de la imagen (en pixels).
VSPACE=	Fija la anchura vertical de los márgenes de la imagen (en pixels).

La introducción de la segunda forma del atributo ALIGN= obligo a modificar la etiqueta `<BR>` con el fin de permitir que manejara los márgenes creados al colocar una imagen a la izquierda o derecha. Por ello se añadió a `<BR>` un atributo CLEAR=(left, right, all) que inserta un salto de línea que se desplaza verticalmente hacia abajo hasta encontrar libre el margen izquierdo, derecho o ambos márgenes respectivamente. Un ejemplo de inserción de imágenes en un documento HTML es el siguiente:

```
<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Imágenes en HTML</TITLE>
</HEAD>
<BODY>
<P>Vamos a insertar una imagen a continuación</P>
<P><IMG SRC="imagen.gif" ALT="logo" ALIGN=left>Todo el texto de este párrafo
se colocara a la derecha de la imagen al estar esta alineada a la
izquierda<BR><BR><BR>
Pero al sobrepasar la imagen continuara de forma normal</P>
</BODY>
</HTML>
```



## A.12 Tablas.

El lenguaje HTML permite la creación de tablas que no solo se limitan a la simple inclusión de texto en columnas, sino que permiten realizar combinaciones entre texto e imágenes. Una tabla se define mediante la etiqueta `<TABLE>` `</TABLE>`. En su interior se describe la distribución de la misma a partir de filas y celdas.

Las filas se representan con la etiqueta `<TR>` `</TR>` y son los bloques básicos. Dentro de cada fila se definen las celdas con la etiqueta `<TD>` `</TD>`. Existe un tipo de celda especial, designada por la etiqueta `<TH>` `</TH>`, que representa el encabezado. Además dentro de cada tabla se puede especificar un título de la tabla mediante la etiqueta `<CAPTION>` `</CAPTION>`. Esta etiqueta solo puede aparecer una vez dentro de cada tabla.

Por tanto, el número de filas de una tabla viene especificado por el número de etiquetas `<TR>` `</TR>` que aparezcan en su interior. Sin embargo, el número de columnas viene dado por el máximo de los números de celdas, especificadas por la etiqueta `<TH>` `</TH>`, que posea una fila. Veamos un ejemplo de una sencilla tabla:

```
<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Tablas en HTML</TITLE>
</HEAD>
<BODY>
<TABLE>
<CAPTION>Gastos del mes</CAPTION>
<TR><TH>Concepto</TH><TH>Importe</TH></TR>
<TR><TD>Alimentacion</TD><TD>20.000</TD></TR>
<TR><TD>Vestido</TD><TD>50.000</TD></TR>
</TABLE>
</BODY>
</HTML>
```



## A.12.1 Atributos de modificación de las etiquetas de las tablas.

A continuación veremos los atributos que es posible dar a las etiquetas de una tabla con el fin de modificar sus valores por defecto. Para ello observemos la siguiente tabla:

Etiqueta	Atributos	Descripción
<TABLE>	BORDER=	Fija el ancho del borde exterior de la tabla (en pixels).
	CELLPADDING=	Fija la distancia (en pixels) entre el borde de la celda y su contenido.
	CELLSPACING=	Fija la anchura (en pixels) de las líneas de división de la tabla.
	WIDTH=	Controla la anchura horizontal de la tabla. Se puede especificar en pixels o en porcentaje.
	ALIGN=	Control de la alineación de la tabla, puede tomar los valores left y right y permite que un párrafo fluya sobre el borde de una tabla, bloqueando ésta a la izquierda o derecha del margen
<CAPTION>	ALIGN=	Fija la posición del título con respecto a la tabla. Puede ser top o bottom.
<TR>	ALIGN=	Especifica la alineación horizontal del texto en la fila, puede ser left, right o center.
	VALING=	Especifica la alineación vertical del texto en la fila, puede ser top, bottom, middle o baseline.
<TD>	ALIGN=	Especifica la alineación horizontal del texto en la celda, puede ser left, right o center.
	VALIGN=	Especifica la alineación vertical del texto en la celda, puede ser top, bottom, middle o baseline.
	COLSPAN=	Indica el número de columnas que ocupara esta celda.
	ROWSPAN=	Indica el número de filas que ocupara esta celda.
	NOWRAP=	Obliga al cliente Web a no romper las líneas de texto que contenga la celda.
	WIDTH=	Determina el ancho de la celda. Se puede especificar como un ancho en pixels o en porcentaje.
<TH>		Igual que <TD>. Igual que los atributos de <TD>.

Además de los atributos anteriores, existe un atributo BGCOLOR= que permite cambiar el color del fondo de la tabla, fila, celda, etc., según la etiqueta sobre la que se aplique. El valor del atributo puede especificarse como un código de color hexadecimal o como un mnemotécnico. Veamos un ejemplo:

```
<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Tablas en HTML</TITLE>
</HEAD>
<BODY>
```

```

<TABLE BORDER=2 BGCOLOR="green">
<CAPTION ALIGN=bottom>Gastos del mes</CAPTION>
<TR ALIGN=center BGCOLOR="yellow">
<TH>Concepto</TH><TH>Importe</TH><TH>Fecha</TH>
</TR>
<TR ALIGN=right>
<TD ALIGN=left>Alimentación</TD><TD>20.000</TD><TD>1/1/1990</TD>
</TR>
<TR ALIGN=right>
<TD ALIGN=left>Vestido</TD><TD>50.000</TD><TD>10/1/1990</TD>
</TR>
</TABLE>
</BODY>
</HTML>

```

Concepto	Importe	Fecha
Alimentación	20.000	1/1/1990
Vestido	50.000	10/1/1990

Gastos del mes

### A.12.2 Anidamiento de tablas.

Las tablas pueden anidarse unas dentro de otras. Esto permite especificar dentro de una tabla las tablas que sean necesarias. Veamos un ejemplo de como hacerlo:

```

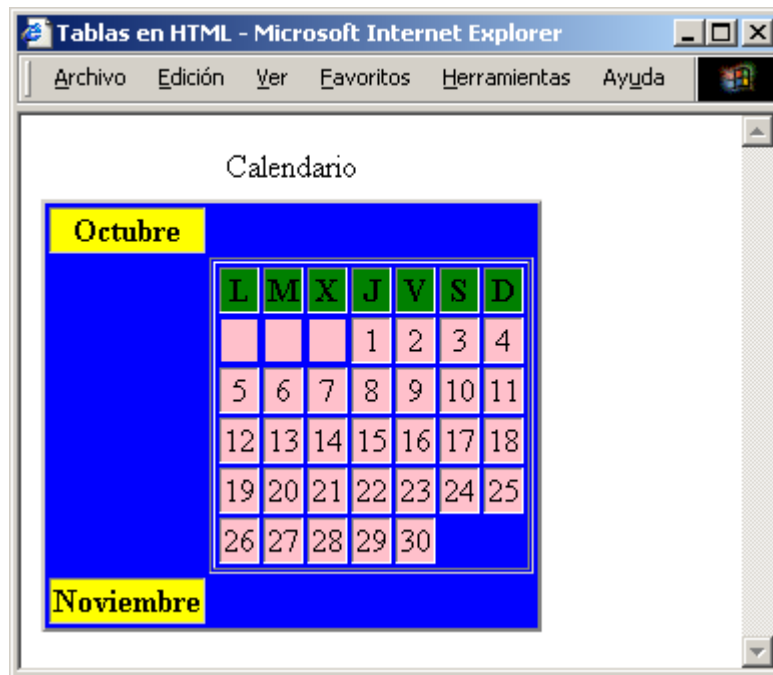
<DOCTYPE HTML PUBLIC>
<HTML>
<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Tablas en HTML</TITLE>
</HEAD>
<BODY>
<TABLE BORDER=2 BGCOLOR="blue">
<CAPTION ALIGN=top>Calendario</CAPTION>
<TR ALIGN=center BGCOLOR="yellow"><TH>Octubre</TH></TR>
<TR ALIGN=right>
<TD></TD>
<TD>
<TABLE BORDER=1>
<TR ALIGN=center BGCOLOR="green">
<TH>L</TH><TH>M</TH><TH>X</TH><TH>J</TH>
<TH>V</TH><TH>S</TH><TH>D</TH></TR>
<TR ALIGN=center BGCOLOR="pink">
<TD></TD><TD></TD><TD></TD><TD>1</TD>
<TD>2</TD><TD>3</TD><TD>4</TD></TR>
<TR ALIGN=center BGCOLOR="pink">

```

```

<TD>5</TD><TD>6</TD><TD>7</TD><TD>8</TD>
<TD>9</TD><TD>10</TD><TD>11</TD></TR>
<TR ALIGN=center BGCOLOR="pink">
<TD>12</TD><TD>13</TD><TD>14</TD><TD>15</TD>
<TD>16</TD><TD>17</TD><TD>18</TD></TR>
<TR ALIGN=center BGCOLOR="pink">
<TD>19</TD><TD>20</TD><TD>21</TD><TD>22</TD>
<TD>23</TD><TD>24</TD><TD>25</TD></TR>
<TR ALIGN=center BGCOLOR="pink">
<TD>26</TD><TD>27</TD><TD>28</TD><TD>29</TD>
<TD>30</TD></TR>
</TD>
</TR>
</TABLE>
<TR ALIGN=center BGCOLOR="yellow"><TH>Noviembre</TH></TR>
</TABLE>
</BODY>
</HTML>

```



### A.13 Marcos (Frames).

Los marcos permiten dividir una página de un cliente Web en varias páginas, cada una de las cuales puede mostrar una página HTML distinta, de forma que cada marco tendrá un URL determinado así como sus propias barras de desplazamiento. Por ejemplo podemos crear un marco para ver el índice de un documento y mediante enlaces, cargar en la parte principal de la ventana los contenidos de esos índices.

La definición de un marco cambia completamente la estructura general de una página HTML, eliminando el cuerpo especificado por la etiqueta `<BODY>` `</BODY>` y sustituyéndolo por la etiqueta `<FRAMESET>` `</FRAMESET>`. Por tanto, el formato general de una página HTML con un marco es el siguiente:

```

<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>

```

```

.....
</HEAD>
<FRAMESET>
.....
</FRAMESET>
</HTML>

```

La etiqueta *<FRAMESET>* posee dos atributos principales. El atributo *ROWS*=”lista de valores” especifica el número de marcos en los que se divide la página y cuantas filas va a ocupar cada marco. El atributo *COLS*=”lista de valores” especifica la longitud en columnas de los marcos. Los valores que puede tomar cada uno de los elementos de lista de valores son números enteros, tantos por ciento y el valor ‘\*’, que especifica que sea tan grande como la ventana. Por ejemplo el siguiente código HTML define dos marcos que ocupan el 30% y el 70% de la ventana y coloca en cada uno de ellos una imagen.

```

<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Marcos en HTML</TITLE>
</HEAD>
<FRAMESET COLS="30%,70%">
<FRAME SRC="imagen.gif">
<FRAME SRC="imagen.gif">
</FRAMESET>
</HTML>

```



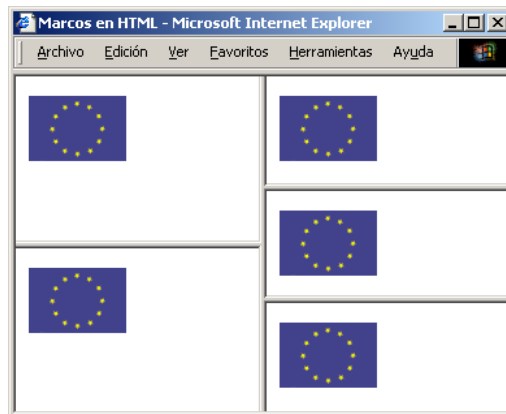
Además de los atributos *ROWS* y *COLS*, la etiqueta *<FRAMESET>* admite un atributo *BORDER* que especifica el tamaño del borde que separa los distintos marcos declarados. El atributo *BORDER* puede tomar el valor cero para indicar que no exista ningún borde definido entre los distintos marcos. Como hemos podido ver, se han incluido dentro de la etiqueta *<FRAMESET>* una nueva etiqueta identificada como *<FRAME>*. La etiqueta *<FRAME>* permite especificar el contenido de cada uno de los marcos. Sus atributos pueden verse en la tabla siguiente:

Atributo	Explicación
SRC=	Nombre del documento que se quiere colocar dentro del marco.
NAME=	Asigna un nombre a un marco de forma que pueda ser el destino de enlaces situados en otros marcos.
MARGINWIDTH=	Especifica el margen lateral (en pixels).
MARGINHEIGHT	Especifica el margen superior e inferior (en pixels).
SCROLLING=(yes, no, auto)	Si un documento es más largo que el marco aparecen unas barras de desplazamiento. Este atributo permite activar, desactivar o automatizar su aparición.
NORESIZE	Evita que el tamaño de los marcos sea alterado.

Para aquellos clientes Web que no permiten la definición de marcos, es posible definir una sección en la cual colocar la información de otra forma para que pueda

funcionar la página Web en estos clientes. Esta sección, cuya etiqueta es `<NOFRAME>` y `</NOFRAME>` es ignorada por los clientes Web que si reconocen los marcos. Los marcos poseen, al igual que sucedía con las tablas, etc., la posibilidad de anidarse. Sin embargo, deberá tenerse mucho cuidado al anidar marcos pues la lectura de la página se puede complicar en exceso. Veamos un ejemplo de una página HTML que define varios marcos:

```
<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Marcos en HTML</TITLE>
</HEAD>
<FRAMESET COLS=50%,50%>
<FRAMESET ROWS=50%,50%>
<FRAME SRC="imagen.gif">
<FRAME SRC="imagen.gif">
</FRAMESET>
<FRAMESET ROWS=33%,33%,33%>
<FRAME SRC="imagen.gif">
<FRAME SRC="imagen.gif">
<FRAME SRC="imagen.gif">
</FRAMESET>
</FRAMESET>
<NOFRAME>Su cliente no permite marcos</NOFRAME>
</HTML>
```



Como indicamos al explicar los enlaces entre páginas, es posible hacer que el destino de un enlace sea un marco distinto. Esto nos permite crear menús que actualizan un marco distinto al que se pulsa. Para ello veamos el siguiente ejemplo de un marco formado por cuatro páginas Web.

```
<!--Fichero que define los marcos -->
<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Marcos en HTML</TITLE>
</HEAD>
<FRAMESET COLS="30%,70%">
<FRAME SRC="menu.html">
<FRAME SRC="portada.html" NAME="principal">
</FRAMESET>
</HTML>
```

```

<!-- Fichero menu.html -->
<DOCTYPE HTML PUBLIC>
<HTML>
<BODY>
<A HREF="imagen.html" TARGET="principal">Cuando pulses aquí cambiara la otra
ventana</A>
</BODY>
</HTML>

```

```

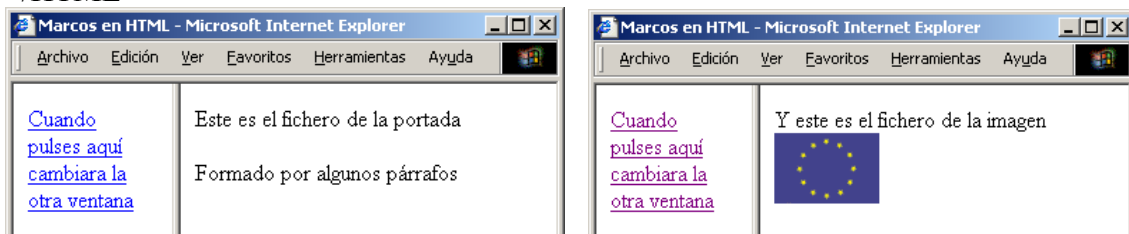
<!--Fichero portada.html -->
<DOCTYPE HTML PUBLIC>
<HTML>
<BODY>
<P>Este es el fichero de la portada</P>
<P>Formado por algunos párrafos</P>
</BODY>
</HTML>

```

```

<!--Fichero imagen.html -->
<DOCTYPE HTML PUBLIC>
<HTML>
<BODY>
<P>Y este es el fichero de la imagen<IMG SRC="imagen.gif"></P>
</BODY>
</HTML>

```



## A.14 Formularios y CGI: Comunicación con el servidor.

En el lenguaje HTML se define un mecanismo que permite un intercambio de información directo entre el cliente y el servidor, son los conocidos formularios o FORMS. Mediante estos formularios, el usuario puede introducir información seleccionando elementos, activando botones, escogiendo elementos de menús e introduciendo texto en cuadros de diálogo.

Los formularios se crean mediante la etiqueta `<FORM>` `</FORM>`, siendo necesario introducir atributos especiales y otros códigos en el cuerpo del formulario. Los atributos de `<FORM>` son:

Atributo	Descripción
ACTION	Especifica el URL del programa CGI que debe procesar los datos que envía el formulario. Si se desea enviar correo electrónico se puede indicar con <i>mailto:dirección de correo</i> .
METHOD	Especifica el método que se usa para enviar la información de los distintos elementos del formulario al programa CGI. Los métodos pueden ser GET (el programa CGI recibe los datos en la variable de entorno QUERY_STRING) y POST (el programa CGI recibe los datos por la entrada standard).

Dentro de los formularios, existen tres etiquetas que especifican campos de entrada, son la etiquetas `<INPUT>`, `<SELECT>` `</SELECT>` y `<TEXTAREA>` `</TEXTAREA>`.

### A.14.1 Campos de entrada `<INPUT>`.

La etiqueta `<INPUT>` puede tomar los siguientes vienen especificados en la siguiente tabla:

Atributos	Descripción
TYPE	Selecciona el tipo de campo de entrada. Es obligatorio
NAME	Define el nombre del identificador cuyo contenido será enviado. Es obligatorio.
VALUE	Inicializa el valor del campo.
SIZE	Especifica el tamaño visible del texto.
MAXLENGTH	Especifica el número máximo de caracteres en un campo de entrada de tipo texto.
CHECKED	Inicializa campos de entrada de tipo checkbox o radio buttons a su estado.
SRC	Especifica el URL para la imagen a usar en el caso de un botón gráfico de envío de la información.
ALIGN	Especifica el alineamiento, puede ser top, middle, bottom (por defecto), left o right.

Como hemos visto en la tabla anterior, el tipo de entrada en el formulario viene especificado por el atributo `TYPE=""` del formulario. Este atributo puede tomar los siguientes valores:

Valor del atributo	Descripción
TEXT	Valor por defecto. Muestra una línea de texto.
PASSWORD	Igual que TEXT excepto que los caracteres introducidos son mostrados como *.
CHECKBOX	Casilla de verificación con dos estados (marcada o no). Es obligatorio el uso de VALUE para especificar el nombre de la variable y el valor que toma ésta.
RADIO	Elementos que operan conjuntamente, solo puede estar activado uno de ellos. Es obligatorio el uso de VALUE. Se puede indicar que un valor es por defecto utilizando el atributo CHECKED de la etiqueta INPUT.
SUBMIT	Define un botón que envía los datos al programa del CGI. Debe existir siempre uno en cualquier formulario.
RESET	Define un botón que inicializa los valores del formulario al valor por defecto.
FILE	Define un método para poder escoger un fichero cuyo contenido será enviado como los datos proporcionados por el formulario.
HIDDEN	No son visibles y proporcionan un método de enviar información sobre el estado, etc. al programa del CGI.
IMAGE	Permite definir un botón gráfico.

Veamos un sencillo ejemplo de formulario con campos de entrada especificados por la etiqueta `<INPUT>`.

```
<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Formularios en HTML</TITLE>
</HEAD>
<BODY>
<FORM METHOD="POST" ACTION="http://computador.uv.es/cgi-bin/prueba.exe">
Nombre:&nbsp;
<INPUT TYPE="text" NAME="nombre" SIZE="10" MAXLENGTH="25"><BR>
Primer apellido:&nbsp;
<INPUT TYPE="text" NAME="apellido1", SIZE="10" MAXLENGTH="25"><BR>
Segundo apellido:&nbsp;
<INPUT TYPE="text" NAME="apellido2", SIZE="10"
MAXLENGTH="25"><BR><BR>
```

```

Curso:&nbsp;<BR>
Primero<INPUT TYPE="radio" NAME="curso" VALUE="Primero" CHECKED>
Segundo<INPUT TYPE="radio" NAME="curso" VALUE="Segundo">
Tercero<INPUT TYPE="radio" NAME="curso" VALUE="Tercero">
Cuarto<INPUT TYPE="radio" NAME="curso" VALUE="Cuarto">
Quinto<INPUT TYPE="radio" NAME="curso" VALUE="Quinto"><BR><BR>
<INPUT TYPE="submit" VALUE="Enviar">
<INPUT TYPE="reset" VALUE="Borrar">
</FORM>
</BODY>
</HTML>

```



### A.14.2 Campos de entrada <SELECT>

La etiqueta <SELECT> </SELECT> permite especificar un menú desplegable. Sus atributos son:

Atributo	Descripción
NAME	Define el nombre del identificador cuyo contenido será enviado. Es obligatorio.
SIZE	Define el número de elementos del menú desplegable que serán visualizados a la vez.
MULTIPLE	Permite que se seleccionen más de un elemento del menú.

Cada uno de los elementos del menú son especificados por la etiqueta <OPTION></OPTION>, la cual puede poseer dos atributos:

Atributo	Descripción
VALUE	Valor de la opción que se enviará si es seleccionada.
SELECTED	Indica que este es el valor seleccionado por defecto. Pueden existir varios si la opción MÚLTIPLE ha sido seleccionada en el menú.

Un sencillo ejemplo de formulario con campo de entrada de tipo <SELECT> es:

```

<DOCTYPE HTML PUBLIC>
<HTML>
<HEAD>
<TITLE>Formularios en HTML</TITLE>
</HEAD>
<BODY>
<FORM METHOD="POST" ACTION="http://computador.uv.es/cgi-bin/prueba.exe">
<P>Procesador:
<SELECT NAME="procesador" SIZE="1">

```





Si se devuelve una referencia a otro documento ya existente en el servidor local o en un servidor remoto, la cabecera será:

Primera línea: *Location: URL\_doc*

Segunda línea: (En blanco).

Veamos un ejemplo típico de un programa CGI escrito en C que recibe los datos de un formulario:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define TAM 1000

void Error(char *msg)
{
    printf("Content-type: text/plain\n\n");
    printf("Error: %s\n",msg);
    exit(EXIT_SUCCESS);
}

int main(void)
{
    char buffer[TAM],*p;
    int longitud;

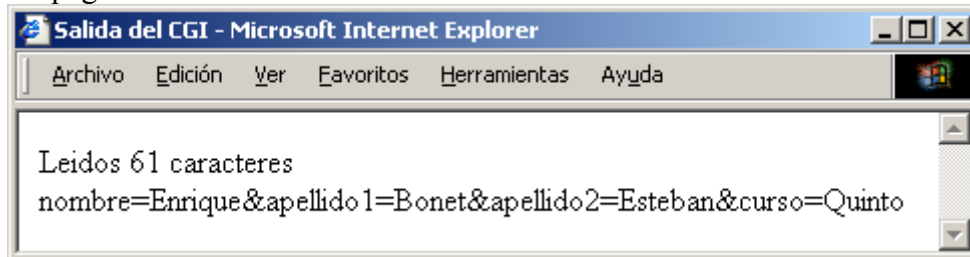
    if ((p=getenv("REQUEST_METHOD"))==NULL)
        Error("REQUEST_METHOD no encontrado");
    if (strcmp(p,"POST",strlen("POST"))==0)
    {
        if ((p=getenv("CONTENT_LENGTH"))==NULL)
            Error("CONTENT_LENGTH no encontrado");
        if ((longitud=atoi(p))>=(TAM-1))
            Error("Longitud excesiva");
        if (fgets(buffer,longitud+1,stdin)==NULL)
            Error("Leyendo los datos");
        buffer[longitud]='\0';
    }
    else
    {
        if ((p=getenv("QUERY_STRING"))==NULL)
            Error("QUERY_STRING no encontrado");
        if ((longitud=strlen(p))>=(TAM-1))
            Error("Longitud excesiva");
        strncpy(buffer,p,TAM);
    }
    /* Creamos ahora el mensaje de salida */
    printf("Content-type: text/html\n\n");
    printf("<HTML>");
    printf("<HEAD><TITLE>Salida del CGI</TITLE></HEAD>");
    printf("<BODY>Leidos %d caracteres<BR>",longitud);
    printf("%s</BODY>",buffer);
}
```

```

printf("</HTML>");
return 0;
}

```

Si ejecutamos un formulario sobre este CGI, obtenemos como respuesta la siguiente página:



Es necesario recordar que para ver el resultado de la ejecución de un CGI es necesario colocarlo en un servidor de Web, no puede ejecutarse directamente desde un cliente Web en el cual se haya cargado la página HTML.

### **A.15 Imágenes sensibles (Mapas de imágenes).**

Las imágenes sensibles son imágenes que contienen zonas activas que actúan como enlaces, en función de la zona de la imagen en la que se “pinche”, se activa un enlace hacia un documento u otro. Existen dos formas de procesar este tipo de imágenes:

1. Imágenes sensibles procesadas en el servidor Web: En las cuales la gestión de la imagen sensible es realizada por el servidor Web, con la sobrecarga de comunicación que esto conlleva. Cuando el usuario pincha sobre la imagen, las coordenadas se transmiten al servidor Web, el cual determina sobre que zona activa se ha pulsado y obtiene el URL asociado al documento especificado en dicha zona, transmitiendo el documento al cliente Web. La inclusión de una imagen sensible procesada en el servidor se hace con el atributo ISMAP de la etiqueta `<IMG>`
2. Imágenes sensibles procesadas en el cliente Web: En la cual el cliente se encarga de la gestión de la imagen sensible. Cuando el usuario pincha sobre la imagen, las coordenadas se procesan en el cliente Web, el cual obtiene la información sobre el URL del documento solicitado, enviando la solicitud de URL al servidor Web. La forma de definir estas imágenes sensibles es usar el atributo USEMAP en la etiqueta `<IMG>`. La sintaxis es la siguiente: `<IMG SRC="imagen" USEMAP="mapa">`.

En ambos casos, las coordenadas sobre las que se ha pulsado el ratón son procesadas para determinar la URL asociada. Esto se consigue mediante la creación de un archivo que contiene el mapa de la imagen. El mapa de una imagen se especifica con la etiqueta `<MAP> </MAP>`, la cual posee como único atributo `NAME="nombre"`, donde “nombre” indica el nombre del mapa que estamos creando. Cada uno de los elementos de la imagen se especifica con la etiqueta sin parrear `<AREA>`, cuyos atributos son:

Atributo	Descripción
HREF	Especifica la URL del documento a cargar si se pulsa sobre esta zona.
ALT	Asigna un texto descriptivo relacionado con la zona.
SHAPE	Especifica el tipo de objeto que define la zona, puede ser “circle”, “rect” o “poly”
COORDS	Especifica las coordenadas de la zona, su formato depende del valor de SHAPE.

Dependiendo del valor de SHAPE, las coordenadas del atributo COORDS se especifican según podemos ver en la tabla siguiente:

SHAPE	Especificación de COORDS
circle	x,y,radio (x,y) son el centro del círculo.
rects	x1,y1,x2,y2 (x1,y1) son la coordenada superior izquierda y (x2,y2) la coordenada inferior derecha del rectángulo.
poly	x1,y1,x2,y2,...,xn,yn, (xi,yi) son las coordenadas de los vértices del polígono, (xn,yn) deben ser igual a (x1,y1).

Un aspecto a considerar es que cada método se evalúa en el orden en que se encuentra dentro del mapa, por lo cual si dos zonas activas se solapan solo se evalúa el primero de los métodos aplicables. Veamos un ejemplo de imagen sensible, para ello usaremos tres archivos Web:

```
<!-- Fichero de la imagen -->
```

```
<DOCTYPE HTML PUBLIC>
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Imágenes sensibles en HTML</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<MAP NAME="mapa_sensible">
```

```
<AREA HREF="cuad.html" ALT="cuadrado" SHAPE="rect" COORDS="10,10,70,70">
```

```
<AREA HREF="circulo.html" ALT="circulo" SHAPE="circle" COORDS="115,105,45">
```

```
</MAP>
```

```
<CENTER>
```

```
<IMG SRC="mapa.bmp" USEMAP="#mapa_sensible">
```

```
</CENTER>
```

```
</BODY>
```

```
</HTML>
```

```
<!-- Fichero del cuadrado -->
```

```
<DOCTYPE HTML PUBLIC>
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Imágenes sensibles en HTML</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<H1 ALIGN="center">!!!Has pulsado sobre el cuadrado!!!</H1>
```

```
</BODY>
```

```
</HTML>
```

```
<!-- Fichero del círculo -->
```

```
<DOCTYPE HTML PUBLIC>
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Imágenes sensibles en HTML</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<H1 ALIGN="center">!!!Has pulsado sobre el círculo!!!</H1>
```

```
</BODY>
```

```
</HTML>
```

