

Curso de HTML

1. ESTRUCTURA BÁSICA DE UN DOCUMENTO HTML
2. DANDO FORMA AL TEXTO DEL DOCUMENTO HTML
3. CARACTERES ESPECIALES
4. ENLACES
5. IMÁGENES
6. ALINEACION Y DIMENSIONADO DE IMÁGENES
7. FONDOS Y COLORES
8. TABLAS
9. FORMULARIOS
10. MAPAS
11. FRAMES
12. SONIDOS

1. ESTRUCTURA BÁSICA DE UN DOCUMENTO HTML

El principio esencial del lenguaje HTML es el uso de las etiquetas (tags). Funcionan de la siguiente manera:

`<XXX>` Este es el inicio de una etiqueta.
`</XXX>` Este es el cierre de una etiqueta.

Las letras de la etiqueta pueden estar en mayúsculas o minúsculas, indiferentemente.

Lo que haya entre ambas etiquetas estará afectado por ellas. Por ejemplo, todo el documento HTML debe estar entre las etiquetas `<HTML>` y `</HTML>`:

`<HTML>` [Todo el documento] `</HTML>`

Un documento HTML en sí está dividido en dos zonas principales:

- * La cabecera, comprendida entre las etiquetas `<HEAD>` y `</HEAD>`
- * El cuerpo, comprendido entre las etiquetas `<BODY>` y `</BODY>`

Dentro de la cabecera hay información del documento, que no se ve en la pantalla principal del navegador que es utilizado para visualizar el documento HTML, principalmente la información encontrada en la cabecera es el título del documento, comprendido entre las etiquetas `<TITLE>` y `</TITLE>`. El título debe ser breve y descriptivo de su contenido, pues será lo que vean los demás cuando añadan nuestra página a su lista de favoritos (o bookmark).

Dentro del cuerpo está todo lo que queremos que aparezca en la pantalla principal (texto, imágenes, etc.)

Por tanto, la estructura de un documento HTML queda de esta manera:

```
<HTML>
  <HEAD>
    <TITLE> Título de la página </TITLE>
  </HEAD>
  <BODY>
    [Aquí van las etiquetas que visualizan la página]
  </BODY>
</HTML>
```

2. DANDO FORMA AL TEXTO DEL DOCUMENTO HTML

Cuando escribimos en el documento el texto que queremos que aparezca en la pantalla, veremos que éste se acomoda a ella, sin que tengamos que pulsar el retorno del carro. Si queremos separar el texto en distintos párrafos debemos usar la etiqueta `<P>`, (que tiene su correspondiente etiqueta de cierre `</P>` pero no suele ponerse)

El texto puede tener unos encabezados, comprendidos entre las etiquetas `<H1>` y `</H1>`, `<H2>` y `</H2>`, etc. (hasta el número 6), 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>` (no la soportan todos los navegadores, aunque sí la mayoría de ellos). Nos centra todo lo que esté dentro de ella, ya sea texto, imágenes, etc.

También tenemos los separadores (horizontal rules), que se consiguen con la etiqueta `<HR>` (no existe la correspondiente de cierre). Con ella se obtiene una raya horizontal tan ancha como la pantalla, y con la apariencia de estar embutida sobre el fondo, como se puede observar a continuación:

Ejemplo

```
<HTML>
  <HEAD>
    <TITLE> Mi pagina en el Web - 1 </TITLE>
  </HEAD>
  <BODY>
    <H1> <CENTER> Primera Pagina </CENTER> </H1>
    <HR>
    Esta es mi primera pagina, aunque todavía es muy sencilla. Como el lenguaje HTML
    no es difícil, pronto estaré en condiciones de hacer cosas más interesantes.
    <P> Aquí va un segundo párrafo, ¿ qué les parece ?
  </BODY>
</HTML>
```

Una etiqueta puede estar anidada dentro de otra. En el ejemplo anterior lo está la etiqueta `<CENTER>` dentro de la etiqueta `<H1>`.

Cuando queremos poner un texto sin ninguna característica especial, lo ponemos directamente. La separación entre párrafos (dejando una línea en blanco) la conseguimos con la etiqueta `<P>`.

Si queremos separar los párrafos, o cualquier otra cosa, pero sin dejar una línea en blanco, usamos una etiqueta pensada para ello: `
` (break, o romper). No tiene etiqueta de cierre.

Al escribir el texto, si ponemos más de un espacio en blanco entre dos palabras observamos que el navegador sólo reconoce uno de ellos. Si queremos forzarle a que lo haga, debemos poner el código "` `"; (non-breaking space).

Para destacar alguna parte del texto se pueden usar:

<code></code> y <code></code>	para poner algo en negrita (bold).
<code><I></code> y <code></I></code>	para poner algo en cursiva (italic).

Otra etiqueta interesante es `<PRE>` y `</PRE>`. El texto que se encuentre entre ella estará preformateado, es decir que aparecerá como haya sido escrito y con una fuente de espaciado fijo (tipo Courier). Se respetarán los espacios en blanco y retornos del carro, tal como estaban en nuestro documento HTML. Es muy apropiada para confeccionar tablas poco elaboradas y otros documentos similares.

Con la etiqueta `<TT>` y `</TT>` conseguimos que el texto tenga un tamaño menor y la apariencia de los caracteres de una máquina de escribir (typewriter). La diferencia con la anterior es que no preformatea el texto, sino que únicamente cambia su apariencia.

La etiqueta `<BLOCKQUOTE>` y `</BLOCKQUOTE>` se utiliza para destacar una cita textual dentro del texto general. Este párrafo está escrito entre ambas etiquetas. Deja márgenes a ambos lados, por eso se usa para poner sangrías.

En las fórmulas matemáticas puede interesar poder escribir índices y subíndices, que se consiguen con las etiquetas `^{` `}` y `_{` `}` respectivamente.

Listas

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

- * Listas desordenadas (no numeradas)
- * Listas ordenadas (numeradas)
- * 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> Un elemento
  <LI> Otro elemento
  <LI> Otro más
  <LI> etc.
</UL>
```

Es decir, toda la lista está dentro de la etiqueta `` y ``, y luego cada elemento 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 elementos en un orden determinado. Su estructura es similar a la anterior. La diferencia estriba en que aparecerá automáticamente un número correlativo para cada elemento.

```
<OL>
  <LI> Primer Elemento
  <LI> Segundo Elemento
  <LI> Tercer Elemento
  <LI> etc.
</OL>
```

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

El tercer tipo lo forman las listas de definiciones. 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:

- * El nombre de la cosa a definir, que se consigue con la etiqueta <DT> (definition term).
- * La definición de dicha cosa, que se consigue con la etiqueta <DD> (definition definition).

```
<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>
```

Comentarios no visibles en la pantalla

A veces es muy útil escribir comentarios en el documento HTML sobre el código que escribimos, que nos pueden servir para recordar posteriormente sobre lo que hicimos, y que no queremos que se vean en pantalla.

Esto se consigue encerrando dichos comentarios entre estos dos símbolos: <!-- y -->

Ejemplo

```
<HTML>
  <HEAD>
    <TITLE> Mi pagina del Web - 2 </TITLE>
  </HEAD>
  <BODY>
    <CENTER><H1> Mis aficiones </H1></CENTER>
    <HR>
    Sin un orden particular, mis <B> aficiones </B> son las siguientes:
    <!-- Una lista sin orden alguno -->
    <UL>
      <LI> El cine
      <LI> El deporte
    </UL>
      <LI> Natacion
      <LI> Baloncesto
    </UL>
    <LI> La musica
  </UL>
  La musica que más me gusta es <I> (en orden de preferencia): </I>
  <!-- Una lista con un orden -->
  <OL>
    <LI> El rock
    <LI> El jazz
    <LI> La musica clasica
  </OL>
</BODY>
</HTML>
```

3. CARACTERES ESPECIALES

Existen algunas limitaciones para escribir el texto. Una de ellas se debe a que las etiquetas se forman como un comando escrito entre los símbolos `<` y `>`. Por tanto, si se quisiera escribir estos caracteres como parte normal del texto, daría lugar a una ambigüedad, ya que el programa navegador podría interpretarlos como el comienzo o final de una etiqueta, en vez de un carácter más del texto.

Para resolver este problema, existen unos códigos especiales, pensados para poder escribir estos caracteres y otros relacionados con las etiquetas.

<code>&lt;</code>	para <code><</code> (less than, menor que)	<code>&amp;</code>	para <code>&</code> (ampersand)
<code>&gt;</code>	para <code>></code> (greater than, mayor que)	<code>&quot;</code>	para <code>"</code> (double quotation)

Como se ve, estos códigos empiezan siempre con el signo `&` y acaban siempre con `;`

Antes ya citamos el espacio no rompible, cuando aparece, el navegador sabe que no debe partir una línea por ese lugar. Se usa también para añadir espacios:

` ` espacio no rompible (non breaking space)

De una manera similar, existen códigos para escribir letras específicas de distintos idiomas. Hay muchos de ellos, pero, lógicamente, los que más nos interesan son los propios del castellano (las vocales acentuadas y con diéresis, la ñ y Ñ, los signos ¿ ;)

Los códigos de las vocales acentuadas se forman comenzando con `&`, seguido de la vocal en cuestión, seguido de la palabra **acute** (aguda) y terminando con el signo `;`

<code>&aacute;</code>	Para la á	<code>&Aacute;</code>	Para la Á
<code>&eacute;</code>	Para la é	<code>&Eacute;</code>	Para la É
<code>&iacute;</code>	Para la í	<code>&Iacute;</code>	Para la Í
<code>&oacute;</code>	Para la ó	<code>&Oacute;</code>	Para la Ó
<code>&uacute;</code>	Para la ú	<code>&Uacute;</code>	Para la Ú

El resto de los códigos son:

<code>&ntilde;</code>	Para la ñ	<code>&Ntilde;</code>	Para la Ñ
<code>&uuml;</code>	Para la ü	<code>&Uuml;</code>	Para la Ü
<code>&#191;</code>	Para ¿	<code>&#161;</code>	Para ;

Todo esto, que como se ve es muy laborioso, puede parecer inútil ya que si escribimos nuestro texto sin hacer ningún caso de estas convenciones, escribiendo las letras acentuadas y demás signos directamente, es muy posible que el resultado lo veamos correctamente en nuestro navegador, pero no podemos estar seguros de que les ocurra lo mismo a todos los que accedan a nuestras páginas con otros navegadores distintos.

4. ENLACES

La característica que más ha influido en el espectacular éxito de la Web ha sido, aparte de su carácter multimedia, la posibilidad de unir los distintos documentos repartidos por todo el mundo por medio de enlaces hipertexto.

En general, los enlaces tienen la siguiente estructura:

```
<A HREF="XXX"> YYY </A>
```

Donde **XXX** es el destino del enlace (Observe las comillas). **YYY** es el texto indicativo en la pantalla del enlace (con un color especial y generalmente subrayado)

Tipos de enlaces

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.

Supongamos que queremos enlazar con la página creada en el ejemplo 2, que hemos llamado **mipag2.html**. En este caso, simplemente sustituimos lo que hemos llamado **XXX** (el destino del enlace) por el nombre del archivo:

```
<A HREF="mipag2.html"> Ejemplo de mi segunda pagina </A>
```

Si queremos que vaya a un sitio concreto de otra página nuestra en vez de ir al principio de la página, adonde va por defecto, en ese sitio tenemos que colocar una marca (ver la Enlaces dentro de la misma página), y completar el enlace con la referencia a esa marca.

Lo veremos con el siguiente ejemplo: `` es la marca que colocaremos en nuestra página, a la que deseamos acceder desde otra nuestra. Entonces la etiqueta con la que efectuaremos la llamada tiene que ser de esta manera: ` En mi otra página `.

Una observación importante: Pudiera ocurrir que nuestro sitio Web estuviera organizado con un directorio principal, y otros subdirectorios auxiliares. Si la página a la que deseamos acceder está, por ejemplo en el subdirectorio llamado **misubdir**, entonces en la etiqueta tendría que colocarse **misubdir/mipag2.html**.

Y a la inversa, si quiero saltar desde una página a otra que está en un directorio anterior, en la etiqueta tendría que haber puesto **../mipag2.html**. Esos dos puntos colocados al principio hacen que se dirija al directorio anterior. Obsérvese que se debe utilizar el símbolo **/** para indicar los subdirectorios, y no **** que es propio únicamente de Windows.

Si nos queremos evitar todas estas complicaciones, podemos tener todo junto en un único directorio, pero esto tiene el inconveniente de que esté todo más desordenado, y sea más difícil hacer futuras modificaciones.

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). El URL podría ser, además de la dirección de una página Web, una dirección de FTP, Gopher, etc.

Una vez conocida la dirección (o URL), la colocamos en vez de lo que hemos llamado XXX (el destino del enlace). Si queremos enlazar con la página de Netscape (<http://home.netscape.com>), la etiqueta sería:

```
<A HREF="http://home.netscape.com"> Página inicial de Netscape </A>
```

Es muy importante copiar estas direcciones correctamente (respetando las mayúsculas y minúsculas, pues los servidores UNIX sí las distinguen)

4. Enlaces con una dirección de e-mail

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

```
<A HREF="mailto:dirección de e-mail"> Texto del enlace </A>
```

Ejemplo

```
<A HREF="mailto:president@whitehouse.gov">Escribe al presidente de USA</A>
```

Hay algunos navegadores que no subrayan el comentario de este tipo de enlace. Una manera recomendable y más segura para conocer la dirección e-mail sería poner algo así como:

```
<A HREF="mailto:president@whitehouse.gov">president@whitehouse.gov</A>
```

Es decir, es conveniente, por la razón dicha anteriormente, poner también en el texto del enlace la dirección de correo electrónico. Así no puede haber duda de a qué se refiere.

Ejemplo

```
<HTML>
  <HEAD>
    <TITLE> Mi pagina del Web - 3 </TITLE>
  </HEAD>
  <BODY>
    <CENTER><H1> Mis paginas favoritas </H1></CENTER>
    <HR>
    Estas son mis paginas favoritas:
    <P>
      <A HREF="http://home.netscape.com"> Netscape </A>
      <BR> <A HREF="http://www.microsoft.com"> Microsoft </A>
      <BR> <A HREF="http://www.yahoo.com"> Yahoo! </A>
    </P>
  </BODY>
</HTML>
```


5. IMÁGENES

La etiqueta que nos sirve para incluir imágenes en nuestras páginas Web es muy similar a la de enlaces a otras páginas, que hemos visto en el capítulo anterior. La única diferencia es que, en lugar de indicar al programa navegador el nombre y la localización de un documento de texto HTML para que lo cargue, se le indica el nombre y la localización de un archivo que contiene una imagen.

La estructura de la etiqueta es:

```
<IMG SRC="imagen.gif">
```

Con el comando **IMG SRC** (image source, fuente de la imagen) se indica que se quiere cargar una imagen llamada, como está indicado, **imagen.gif** (o el nombre que tenga).

Dentro de la etiqueta se pueden añadir otros comandos, como **ALT**

```
<IMG SRC="imagen.gif" ALT="descripción">
```

Con el comando **ALT** se introduce una descripción (una palabra o una frase breve) indicativa de la imagen. Este comando, que en principio se puede omitir, es en beneficio de los que accedan a nuestra página con un programa navegador en forma de texto como el Lynx. Ya que no son capaces de ver la imagen, por lo menos pueden hacerse una idea sobre ella. Pero no es sólo por esto. Hay casos, como veremos más adelante, en los que se utiliza una imagen como enlace a otra página. Si se omitiera este comando, los que utilizan dichos navegadores no podrían de ninguna manera acceder a esas páginas.

Con respecto a la localización del archivo de esa imagen, se puede decir aquí lo mismo que en el capítulo anterior referente a los enlaces. Si no se indica nada especial, como en el caso expuesto, quiere decir que el archivo **imagen.gif** está en el mismo directorio que el documento HTML que estamos escribiendo. Si no es así, se siguen los mismos criterios indicados para los enlaces.

Las imágenes deben estar guardadas en formato **GIF** (máximo 256 colores, para dibujos) o en formato **JPG** (para imagen fotográfica).

Un aspecto muy importante a tener en cuenta es el tamaño de las imágenes, pues una imagen grande supone un archivo grande, y esto puede resultar en un tiempo excesivo de carga, con el consiguiente riesgo de que quien esté intentando cargar nuestra página se canse de esperar, y desista.

Para elegir la posición de la imagen con respecto al texto hay distintas posibilidades. La más sencilla es colocarla entre dos párrafos, con un titular a un lado. Los navegadores actuales (como el Netscape Navigator y el Microsoft Internet Explorer) permiten que el texto pueda rodear a la imagen.

De momento nos vamos a limitar a escoger la posición del titular con respecto a la imagen. Se puede poner arriba, en medio o abajo del lado de la imagen. Para ello se añade el comando **ALIGN** a la etiqueta, de la siguiente manera:

<code></code>	Titular alineado arriba
<code></code>	Titular alineado en medio
<code></code>	Titular alineado abajo

Otra posibilidad muy interesante es la de utilizar una imagen como enlace a otra página. Para estos casos se utilizan generalmente imágenes pequeñas (iconos), aunque se puede usar cualquier tipo de imagen.

Según vimos en el capítulo anterior, la estructura general de un enlace es:

```
<A HREF="XXX"> YYY </A>
```

En este caso sustituimos XXX por el nombre del archivo de la página a la que queremos acceder. Y en lugar de YYY ponemos la etiqueta completa de la imagen (que queda así englobada dentro de la etiqueta del enlace).

Como por ejemplo (`hombre.gif`) para acceder al ejemplo práctico del capítulo 2 (`mipag2.html`):

```
<A HREF="mipag2.html"><IMG SRC="hombre.gif"></A>
```

Pulsando la imagen comprobamos que efectivamente enlaza con la página deseada. Obsérvese además que la imagen está rodeada de un rectángulo del color normal en los enlaces. Si no se desea que aparezca ese rectángulo, hay que incluir dentro de la etiqueta de la imagen el atributo `BORDER="0"`, es decir:

```
<A HREF="mipag2.html"><IMG SRC="hombre.gif" BORDER="0"></A>
```

Posicionando el cursor sobre esta última imagen, comprobamos que actúa también como enlace aunque carezca del rectángulo de color. Esto puede resultar más estético, pero se corre el riesgo de que el usuario no se dé cuenta de que la imagen sirve de enlace.

También podemos utilizar una imagen para enlazar con otra imagen. Supongamos que queremos enlazar con la imagen `estaimagen.gif` por medio de esta otra imagen `desdeesta.gif`:

```
<A HREF="estaimagen.gif"><IMG SRC="desdeesta.gif"></A>
```

Por último, otra posibilidad es la de utilizar un texto para enlazar con una imagen. En este caso sustituimos XXX (el destino del enlace) con el nombre del archivo de la imagen a la que queremos acceder e YYY (lo que aparece en pantalla como el enlace) por el texto.

Ejemplo

```
<A HREF="isla.gif"> un paraíso tropical </A>
```

Un tipo de imágenes del que se hace abundante uso y que sirven para mejorar la presentación de la página son los iconos, botones, barras separadoras, etc. A pesar de su tamaño o forma, son imágenes como cualquier otra.

6. ALINEACION Y DIMENSIONADO DE IMÁGENES

Alineación de las imágenes

Si se quiere lograr diseños fantásticos y rodear con textos los gráficos se puede **ALIGN** junto con la etiqueta ya conocida ``. Donde quiera que se desee que aparezca una imagen basta con insertar:

<code></code>	Alinea a la izquierda de la página
<code></code>	Alinea al centro de la página
<code></code>	Alinea a la derecha de la página

Si se quiere interrumpir el proceso de rellenado del texto a los lados de la imagen, para que salte hasta debajo de ella, es decir, dejar un espacio en blanco parcialmente, se pueden emplear atributos de la etiqueta `
`:

<code><BR CLEAR="left"></code>	Busca el primer margen libre (clear) a la izquierda.
<code><BR CLEAR="right"></code>	Busca el primer margen libre a la derecha.
<code><BR CLEAR="all"></code>	Busca el primer margen libre a ambos lados.

Ejemplo

```
<IMG SRC="imagen.gif" ALIGN="left"> Este texto esta a un lado de la imagen.  
<BR> Este tambien esta a un lado de la imagen, en la linea siguiente.  
<BR CLEAR="left"> Este otro texto, en cambio, ha buscado el primer margen libre a la izquierda.
```

Dimensionando la imagen

Los programas navegadores cuando cargan un documento HTML y encuentran una etiqueta de una imagen, interrumpen el proceso de carga y solicitan al servidor que le envíe dicha imagen, quedando a la espera hasta que se complete el envío, repitiéndose este proceso con cada una de las imágenes.

Esto es especialmente molesto cuando, como ocurre frecuentemente, en la cabecera de la página se encuentra una imagen grande, ya que durante un tiempo relativamente largo no se verá nada en la pantalla.

Para evitar este inconveniente existen unas extensiones de la etiqueta de imagen `` que sirven para indicar al navegador cuáles son sus dimensiones en pixels. (Obtenidas de algún programa gráfico).

En este caso, el navegador actúa de una forma más favorable, ya que entonces, como conoce las dimensiones de las imágenes les reserva un espacio en la pantalla y va colocando el texto de forma apropiada, sin ninguna interrupción, a la vez que va rellenando esos espacios reservados a las imágenes. Estos comandos o atributos son **WIDTH** (ancho) y **HEIGHT** (alto).

Por ejemplo, para la imagen `isla.gif` situada más arriba:

```
<IMG SRC="imagen.gif" WIDTH="120" HEIGHT="94">
```

Es conveniente hacer esto con todas las imágenes, incluso con las más pequeñas (iconos, botones, etc), para que no haya ninguna interrupción en el proceso de carga del documento.

Se puede también, si se quiere, dimensionar las imágenes con unos valores distintos a los que realmente tienen, variando el tamaño, la anchura o la altura. Esto es muy conveniente, por ejemplo para poner en la página un thumbnail (reproducción en pequeño de una imagen), que hace de enlace a la imagen en su verdadero tamaño. De esta manera no recargamos demasiado una página, y el usuario será quien decida qué imágenes desea cargar.

Para hacer que una imagen reducida sea el enlace con la imagen en su tamaño original, lo conseguimos con:

```
<A HREF="imagen.gif"><IMG SRC="imagen.gif" WIDTH="150" HEIGHT="75"></A>
```

También se puede conseguir esto de otra manera, más correcta aunque más laboriosa. Es la de reducir en un programa gráfico esta imagen, por ejemplo a 150x75, guardarla con otro nombre, y luego hacer que la pequeña sea el enlace de la grande. Es más correcta esta otra solución porque no todos los navegadores reconocen los comandos **WIDTH** y **HEIGHT**, como algunas de las versiones más antiguas de Netscape.

7. FONDOS Y COLORES

Se puede cambiar el fondo de dos maneras distintas:

1. Con un color uniforme
2. Con una imagen

1. Fondos con un color uniforme

Se consigue añadiendo el comando BGCOLOR a la etiqueta <BODY> (situada al principio del documento), de la siguiente manera:

```
<BODY BGCOLOR="#XXYYZZ">
    XX    Es un número indicativo de la cantidad de color rojo
    YY    Es un número indicativo de la cantidad de color verde
    ZZ    Es un número indicativo de la cantidad de color azul
```

Estos números están en numeración hexadecimal. Esta numeración se caracteriza por tener 16 dígitos (en lugar de los diez de la numeración decimal habitual). Estos dígitos son:

0 1 2 3 4 5 6 7 8 9 A B C D E F (A=10, B=11, C=12, D=13, E=14, F=15)

Es decir, que en nuestro caso, el número menor es el 00 y el mayor el FF (=255). Así, por ejemplo, el color rojo es el #FF0000, porque tiene el máximo de rojo y cero de los otros dos colores. Los colores más simples son:

#FF0000	Rojo	#FFFFFF	Blanco	#FF00FF	Magenta
#00FF00	Verde	#000000	Negro	#00FFFF	Celeste
#0000FF	Azul	#FFFF00	Amarillo	#CCCCCC	Gris

Para hacer un color más oscuro, hay que reducir el número de su componente, dejando los otros dos invariables. Así, el rojo #FF0000 se puede hacer más oscuro con #AA0000, o aún más oscuro con #550000.

Para hacer que un color tenga un tono más suave (más pastel), se deben variar los otros dos colores haciéndolos más claros (número más alto), en una cantidad igual. Así, podemos convertir el rojo en rosa con #FF7070.

Colores del texto y de los enlaces

Si no variasen los colores habituales del texto y de los enlaces (negro y azul, respectivamente), podría ocurrir que su lectura contra un fondo oscuro fuese dificultosa o imposible, si el fondo fuese precisamente negro o azul.

Para evitar esto, podemos escoger los colores del texto y de los enlaces, añadiendo a la etiqueta (si se desea) los siguientes comandos:

TEXT	color del texto	LINK	color de los enlaces
VLINK	color de los enlaces visitados	ALINK	color de los enlaces activos (al ser pulsados)

Los códigos de los colores son los mismos que los que se han visto anteriormente.

La etiqueta, con todas sus posibilidades, sería:

```
<BODY BGCOLOR="#XXYYZZ" TEXT="#XXYYZZ" LINK="#XXYYZZ"
    VLINK="#XXYYZZ" ALINK="#XXYYZZ">
```

El comando TEXT explicado anteriormente (que va englobado dentro de la etiqueta <BODY>) cambia el color de la totalidad del texto de la página.

Tanto el Netscape Navigator 2, como el Microsoft Explorer soportan una etiqueta de color de la fuente con la que se puede cambiar sólo una parte del texto:

```
<FONT COLOR="#XXYYZZ"> Este texto es de color XXYYZZ </FONT>
```

2. Fondos con una imagen

El fondo de una página puede ser también una imagen, ya sea en formato GIF o JPG. Esta imagen se repite por toda la página, como un mosaico. La estructura de la etiqueta puede ser:

```
<BODY BACKGROUND="imagen.gif">  
<BODY BACKGROUND="imagen.jpg">
```

No todos los navegadores soportan este formato.

Se puede añadir también a esta etiqueta todos los comandos para cambiar los colores del texto y de los enlaces, vistos anteriormente. Esto es imprescindible a veces para conseguir que el texto sea legible, en contraste con el color o imagen de fondo.

Hay que prever la posibilidad de que quien acceda a nuestra página haya deshabilitado la carga automática de imágenes, en cuyo caso tampoco cargaría la imagen que sirve como fondo y sólo vería el fondo estándar. Esto podría ser muy perjudicial si hemos escogido unos colores para el texto y los enlaces que no contrastan bien con ese fondo. La solución a este problema es poner dentro de la etiqueta <BODY> los dos comandos BACKGROUND y BGCOLOR (en este orden), teniendo cuidado en escoger un color uniforme de fondo parecido al de la imagen.

Por ejemplo, supongamos que queremos poner como fondo la imagen `nubes.jpg`. Escogemos entonces un color de fondo azul claro, `#CCFFFF`. La etiqueta quedaría así:

```
<BODY BACKGROUND="nubes.jpg" BGCOLOR="#CCFFFF">
```

Esto tiene la ventaja adicional de que, incluso aunque no se deshabilite la carga automática de imágenes, al cargar la página, lo primero que se ve es ese fondo de color uniforme, que luego es reemplazado por el de la imagen.

8. TABLAS

Las tablas son un modo sencillo de disponer el texto en columnas o de añadir un titular a una ilustración, pero hay más aplicaciones. La etiqueta `<TABLE>` puede ser una poderosa herramienta de formato. Se puede hacer uso de la etiqueta `<TABLE>` para ubicar texto e imágenes con precisión, en prácticamente casi cualquier lugar de una página.

Estructura de una tabla

Vamos a ver ordenadamente (de fuera hacia dentro) las etiquetas necesarias para confeccionar las tablas.

```
<TABLE>
    [resto de las etiquetas]
</TABLE>
```

- Es la etiqueta general, que engloba a todas las demás.

```
<TABLE BORDER="n">
    [resto de las etiquetas]
</TABLE>
```

- Presenta los datos tabulados con un borde, haciendo las tablas más atractivas, y el grosor es de n pixeles.

```
<TR>
    [etiquetas de las distintas celdas de la primera fila]
</TR>
```

- Permite formar cada fila de la tabla. Hay que repetirla tantas veces como filas queremos que tenga la tabla.

```
<TD>
    [contenido de cada celda (imágenes, texto, etc.)) ]
</TD>
```

- Permite formar las celdas que contendrá cada fila. Hay que repetirlas tantas veces como celdas deba tener la fila.

```
<TH>
    [encabezamiento de tabla]
</TH>
```

- Es utilizada para colocar encabezamientos en negrita sobre las columnas

Ejemplo

```
<HTML>
  <HEAD>
    <TITLE> Página de prueba para tablas </TITLE>
  </HEAD>
  <BODY>
    <TABLE BORDER>
      <TR>
        <TH>Columna 1</TH><TH>Columna 2</TH><TH>Columna 3</TH>
        <TD>fila1-celda1</TD><TD>fila1-celda2</TD><TD>fila1-celda3</TD>
      </TR>
      <TR>
        <TD>fila2-celda1</TD><TD>fila2-celda2</TD><TD>fila2-celda3</TD>
      </TR>
    </TABLE>
    <BR> [Aquí van las etiquetas que visualizan la página]
  </BODY>
</HTML>
```

Si el número de celdas de una fila es distinto al de otra el navegador forma el número de filas y columnas que haga falta, dejando espacios en blanco en las filas que tengan menos celdas.

Titular de tabla

Se puede añadir un título a la tabla, es decir un texto situado encima de la tabla que indica cuál es su contenido. Se consigue con la etiqueta `<CAPTION>` y `</CAPTION>`. Por ejemplo:

```
<CAPTION> Esta es mi primera tabla </CAPTION>
<CAPTION ALIGN="top">Título</CAPTION>      Coloca el título sobre la tabla.
<CAPTION ALIGN="bottom">Título</CAPTION>    Coloca el título bajo la tabla.
```

Contenido de las celdas

Solo se ha puesto hasta ahora en el ejemplo, texto normal dentro de las distintas celdas. Pero se puede poner en ellas cualquier otro elemento de los que van en un documento HTML, como imágenes, enlaces, etc. No hay más que poner dentro de la etiqueta de la celda la etiqueta correspondiente a una imagen, un enlace, etc.

Ejemplos

a) Con una imagen

```
<TABLE BORDER>
  <TR>
    <TD> <IMG SRC="imagen.gif"> </TD>
  </TR>
</TABLE>
```

b) Con un enlace

```
<TABLE BORDER>
  <TR>
    <TD><A HREF="index.html"> Mi página principal </A></TD>
  </TR>
</TABLE>
```

Posicionamiento del contenido dentro de la celda

Normalmente, el contenido de una celda está alineado a la izquierda. Pero se puede cambiar esto añadiendo dentro de la etiqueta de la celda los siguientes atributos:

```
<TD ALIGN="center"> Al centro </TD>
<TD ALIGN="right"> A la derecha </TD>
<TH ALIGN="left"> A la izquierda </TH>
```

El alineamiento por defecto en el sentido vertical es en el medio. También se puede cambiar, añadiendo dentro de la etiqueta de la celda los siguientes atributos:

```
<TD VALIGN="top"> Arriba </TD>
<TD VALIGN="bottom"> Abajo </TD>
```

Variando las dimensiones de la tabla

El navegador se encarga normalmente de dimensionar el tamaño total de la tabla de acuerdo con el número de filas, de columnas, por el contenido de las celdas, espesor de los bordes, etc.

A veces nos puede convenir forzarle para que la tabla tenga unas dimensiones totales mayores que las que le corresponden, tanto en anchura como en longitud. Esto se consigue añadiendo dentro de la etiqueta de la tabla los atributos **WIDTH** y **HEIGHT** igual a un porcentaje de la dimensión de la pantalla, o a una cifra que equivale al número de pixels.

Por ejemplo, si colocamos:

```
<TABLE WIDTH="60%">
```

O, por ejemplo ponemos:

```
<TABLE HEIGHT="200">
```

Celdas que abarcan a otras varias

A veces puede interesarnos que una celda se extienda sobre otras varias. Esto se consigue añadiendo dentro de la etiqueta de la celda los atributos **COLSPAN="número"** para extenderse sobre un número determinado de columnas, o **ROWSPAN="número"** para extenderse verticalmente sobre un número determinado de filas.

Por ejemplo, en la primera tabla del ejemplo anterior vamos a añadir una fila con una sola celda, que abarca a dos columnas:

```
<TR> <TD COLSPAN="2"> Celda sobre 2 columnas </TD> <TR>
```

O, en la misma tabla, vamos a añadir una celda en la primera fila, pero que abarque también a la siguiente:

```
<TD ROWSPAN="2"> Celda junto a 2 filas </TD>
```

Color de fondo en las tablas

Podemos conseguir que las tablas tengan un color de fondo, siguiendo un procedimiento totalmente análogo al empleado para que una página tenga un color de fondo uniforme (según vimos en el capítulo 7). Para ello debemos utilizar el atributo **BGCOLOR="#XXYYZZ"**, visto en dicho capítulo.

Se puede conseguir así:

```
<TABLE BORDER BGCOLOR="#XXYYZZ">  
[resto de las etiquetas]  
</TABLE>
```

La totalidad de la tabla tendrá un color de fondo.

```
<TD BGCOLOR="#XXYYZZ">  
[contenido de la celda]  
</TD>
```

Solo una celda tendrá color de fondo, distinto al resto.

Imágenes de fondo en las tablas

El Explorer de Microsoft soporta la colocación de imágenes de fondo en el interior de las tablas, de una manera análoga a como se hace en una página (según vimos en el capítulo 7).

Para ello debemos, utilizar el atributo **BACKGROUND="imagen.gif"** o **BACKGROUND="imagen.jpg"**, visto en dicho capítulo.

Si se utiliza dentro de la etiqueta **<TABLE>** la imagen en cuestión se multiplicará detrás de todas las celdas.

(Con el Netscape no se verá ninguna imagen de fondo, sólo con el Explorer)

Si, por el contrario, sólo se pone este atributo dentro de la etiqueta de una celda concreta (**<TD>** o **<TH>**), entonces la imagen de fondo se verá sólo en esa celda.

Separación entre las celdas de una tabla

Por defecto, la separación entre las distintas celdas de una tabla es de dos pixels. Pero se puede variar esto con el atributo **CELLSPACING**, que se pone dentro de la etiqueta **TABLE**.

Por ejemplo, para obtener una separación de 20 pixels entre celdas ponemos:

```
<TABLE BORDER CELLSPACING="20">
```

A primera vista parece como si esto fuera lo mismo que si hubiéramos aumentado el espesor de los bordes. Pero para comprobar que no es así, hagamos que en el caso anterior, tenga además unos bordes de 5 de espesor:

```
<TABLE BORDER="5" CELLSPACING="20">
```

Separación entre el borde y el contenido dentro de las celdas

Por defecto, la separación entre el borde y el contenido dentro de las celdas es de un pixel. Se puede cambiar esto con el atributo **CELLPADDING**, que se pone dentro de la etiqueta **TABLE**.

Por ejemplo, para obtener una separación de 20 pixels entre el contenido y los bordes, dentro de cada celda:

```
<TABLE BORDER CELLPADDING="20">
```

Se puede combinar este atributo con **CELLSPACING** (visto en el apartado anterior).

Por ejemplo, una tabla con bordes de 5 de espesor, separación entre celdas de 15 y separación del contenido con respecto a los bordes de las celdas de 20, lo obtendríamos con:

```
<TABLE BORDER="5" CELLSPACING="15" CELLPADDING="20">
```

9. FORMULARIOS

La manera general para que los lectores de nuestra página se puedan comunicar con nosotros es por medio de un enlace a nuestra dirección de e-mail, con lo que recibiríamos un e-mail convencional.

Pero puede ser que lo que necesitemos sea solamente una respuesta concreta a unas opciones que presentaremos nosotros mismos, o un comentario del usuario, para ello le suministraremos un espacio en donde introducirlo.

Se puede, hacer todo esto, además de otras cosas, utilizando los formularios, con los que se puede confeccionar páginas que contengan los elementos necesarios para ello, tal como botones de radio, listas de selección, cajetines de introducción de texto y de control, etc., como vamos a ver.

Los formularios permiten que los demás nos envíen la información directamente a nosotros o bien a nuestro servidor, en donde hemos instalado un programa que procese esta información. Por ejemplo, vamos a suponer que queremos crear una lista de correo. Los usuarios pueden introducir sus nombres y direcciones de e-mail y pulsar un botón de envío.

Esos datos los podemos recibir "en bruto" en nuestro correo, con los que confeccionaríamos manualmente dicha lista de correo, sin necesitar ningún programa para ello. Este proceso es el que vamos a comentar en este capítulo.

La otra posibilidad, de la que únicamente se va a hacer esta mención, es que hubiéramos instalado en nuestro servidor un programa especial para procesar esos datos y añadirlos a la lista de correo, y que incluso pudiera devolver automáticamente al usuario algún tipo de información. Para conseguir esto, los formularios necesitan ejecutar programas o scripts por medio del CGI (Common Gateway Interface). El CGI permite a los formularios ser procesados por programas escritos en cualquier lenguaje, aunque los más usados en Internet son el Perl y el C.

El tipo de formulario que se describe a continuación no funcionará en todos los navegadores, por lo que es conveniente suministrar al final del formulario, como una alternativa, un enlace de e-mail ordinario (ver el Capítulo 4)

La estructura general de un formulario es:

1. Etiqueta de inicio:

```
<FORM ACTION="mailto:email" METHOD="POST" ENCTYPE="TEXT/PLAIN">
```

2. Cuerpo del formulario

(Elementos para introducir los datos).

3. Botones de envío y de borrado.

4. Etiqueta de cierre </FORM>

1. Etiqueta de inicio

El atributo **ACTION** indica la acción que se debe efectuar y que es que los datos sean enviados por e-mail a la dirección indicada. (Si hiciéramos uso del CGI, sería precisamente aquí donde indicaríamos su localización en el servidor, que habitualmente es el directorio cgi-bin, para que procese los datos).

El atributo **METHOD="POST"** indica que los datos sean inmediatamente enviados por correo a la dirección de correo electrónico, nada más pulsar el usuario el botón de envío.

Con el atributo **ENCTYPE="TEXT/PLAIN"** se consigue que las respuestas las recibamos como un archivo de texto, perfectamente legible y sin codificar.

2. Cuerpo del Formulario (Elementos para introducir los datos)

La introducción de los datos se consigue por medio de la etiqueta:

```
<INPUT TYPE="XXX" NAME="YYY" VALUE="ZZZ">
```

XXX Es la palabra que indica el tipo de datos a introducir.
YYY Es el nombre que le asignamos nosotros a la variable de introducción del dato.
ZZZ Es la palabra asociada a un elemento.

Los vamos a dividir en tres clases:

- Introducción por medio de texto
- Introducción por medio de menús
- Introducción por medio de botones

a. Introducción por medio de texto

Introducción por medio de texto (una línea)

En este caso es XXX=TEXT, es decir, INPUT TYPE="TEXT". El atributo VALUE no procede en este caso.

Ejemplo. Vamos a poner un ejemplo solicitando el apellido del usuario:

```
<FORM ACTION="mailto:e-mail" METHOD="POST" ENCTYPE="TEXT/PLAIN">
  Escribe tu apellido:
  <BR><INPUT TYPE="TEXT" NAME="Apellido">
</FORM>
```

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

Apellido=Ruiz

La longitud de este formulario es por defecto de 20 caracteres. 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, incluyendo el atributo MAXLENGTH="número".

En el caso que hemos visto, si hubiéramos cambiado la etiqueta correspondiente por:

```
<INPUT TYPE="text" NAME="Apellido" SIZE="10" MAXLENGTH="12">
```

(Se puede comprobar cómo no se pueden introducir más de 12 caracteres).

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". En el último ejemplo, podríamos cambiar la etiqueta correspondiente por:

```
<INPUT TYPE="password" NAME="Apellido" SIZE="10" MAXLENGTH="12">
```

(Se puede comprobar cómo los caracteres introducidos se representan por asteriscos)

Introducción por medio de texto (múltiples líneas)

Cuando el texto a introducir puede alcanzar una gran longitud, por ejemplo un comentario, es conveniente utilizar un formulario de texto de múltiples líneas.

Esto se consigue con la etiqueta de inicio:

```
<TEXTAREA NAME="YYY" ROWS="número" COLS="número">
```

(No se utiliza INPUT TYPE y donde ROWS representa el número de filas, y COLS el de columnas).

```
</TEXTAREA>
```

Ejemplo: un formulario solicitando los comentarios del usuario:

```
<FORM ACTION="mailto:e-mail" METHOD="POST" ENCTYPE="TEXT/PLAIN">
  Introduce tus comentarios:
  <BR><TEXTAREA NAME="Comentarios" ROWS="6" COLS="40">
  </TEXTAREA>
</FORM>
```

(El salto de línea del texto introducido no se efectúa automáticamente).

Una vez que el usuario haya escrito sus comentarios dentro del formulario, y haya pulsado el botón de envío, recibiremos un e-mail suyo con el siguiente texto:

Comentarios=mensaje escrito por el usuario

b. Introducción por medio de menús

Si queremos que el usuario, en vez de introducir un texto, como hemos visto en los casos anteriores, escoja entre varias opciones que le presentamos nosotros, haremos uso de un formulario en forma de menú.

Se consigue con la etiqueta de inicio `<SELECT NAME="YYY">` y la de cierre `</SELECT>`.

Las distintas opciones a escoger se consiguen con la etiqueta `<OPTION>`.

Ejemplo: Pedimos al usuario que elija su color preferido:

```
<FORM ACTION="mailto:e-mail" METHOD="POST" ENCTYPE="TEXT/PLAIN">
  <BR>Cuál es tu color preferido?<BR>
  <SELECT NAME="ColorPreferido">
    <OPTION SELECTED>
    <OPTION>Rojo
    <OPTION>Verde
    <OPTION>Azul
    <OPTION>Amarillo
  </SELECT>
</FORM>
```

Si el usuario ha escogido, p. ej. Azul y ha pulsado el botón de envío, recibiremos un e-mail suyo con el texto:

ColorPreferido=Azul.

En el ejemplo anterior, sólo es visible en el formulario una opción. Si queremos que sean visibles múltiples opciones a la vez, añadimos en la etiqueta los atributos `MULTIPLE SIZE="número"`, donde especificamos el número de opciones visibles.

Si cambiamos en el ejemplo anterior la etiqueta correspondiente por:

```
<SELECT NAME="ColorPreferido" MULTIPLE SIZE="2">
```

c. Introducción por medio de botones

Caja de confirmación (checkbox)

Si queremos que el usuario confirme una opción determinada, podemos hacer uso de un formulario de confirmación, o checkbox, que se consigue con la etiqueta:

```
<INPUT TYPE="CHECKBOX" NAME="YYY">
```

Ejemplo: Solicitamos al usuario que confirme su inclusión en una lista de correo:

```
<FORM ACTION="mailto:email" METHOD="POST" ENCTYPE="TEXT/PLAIN">
  <INPUT TYPE="checkbox" NAME="Lista">
  Sí, deseo ser incluido en la lista de correo.
</FORM>
```

Si el usuario marca este formulario y pulsa el botón de envío, recibiremos un e-mail suyo con el texto:

Lista=On

Si queremos que el formulario aparezca inicialmente como marcado (el usuario no necesitará hacerlo), basta con añadir el atributo `CHECKED` dentro de la etiqueta. En el ejemplo anterior sustituimos la etiqueta equivalente por:

```
<INPUT TYPE="CHECKBOX" NAME="Lista" CHECKED>
```

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">
```

Donde YYY es el nombre que le ponemos a la variable que se trata de elegir, y ZZZ es el nombre de cada una de las opciones en concreto.

Ejemplo: solicitamos al usuario que defina cuál es su sistema operativo preferido:

```
<FORM ACTION="mailto:email" METHOD="POST" ENCTYPE="TEXT/PLAIN">
 Cuál es tu sistema operativo preferido?<BR>
  <INPUT TYPE="radio" NAME="Sistema" VALUE="Linux">Linux
  <INPUT TYPE="radio" NAME="Sistema" VALUE="MacOS" CHECKED>MacOS
  <INPUT TYPE="radio" NAME="Sistema" VALUE="OS/2">OS/2
  <INPUT TYPE="radio" NAME="Sistema" VALUE="Unix">Unix
  <INPUT TYPE="radio" NAME="Sistema" VALUE="Windows">Windows
</FORM>
```

Obsérvese el atributo opcional CHECKED que se ha añadido en la segunda etiqueta. Esa será la opción que aparece marcada por defecto.

Obsérvese también que no es posible escoger más de una opción.

Si el usuario ha escogido la opción MacOS y pulsa el botón de envío, recibiremos un e-mail suyo con el texto:

Sistema=MacOS

3. Botones de envío y de borrado

Hasta ahora, en todos los ejemplos que hemos visto, faltaba un elemento esencial en cualquier formulario, y es el botón de envío de los datos, que se consigue con la etiqueta:

```
<INPUT TYPE="submit" VALUE="ZZZ">
```

En donde ZZZ es el texto que queremos que aparezca en el botón.

Vamos a añadirlo al primer ejemplo, en el que se solicitaba el apellido del usuario:

```
<FORM ACTION="mailto:email" METHOD="POST" ENCTYPE="TEXT/PLAIN">
  Escribe tu apellido:
  <BR><INPUT TYPE="text" NAME="Apellido">
  <P><INPUT TYPE="submit" VALUE="Enviar datos">
</FORM>
```

Otro botón interesante es el de borrado de los datos introducidos, muy conveniente en un formulario con muchos elementos. Es muy similar al de envío, pues se consigue con la etiqueta:

```
<INPUT TYPE="RESET" VALUE="ZZZ">
```

En donde ZZZ es el texto que queremos que aparezca en el botón.

Podríamos añadirlo al ejemplo anterior:

```
<P><INPUT TYPE="reset" VALUE="Borrar datos">
```

Se puede comprobar su funcionamiento, escribiendo algo en el formulario y pulsando luego el botón de borrado.

Consideraciones finales

Hasta ahora hemos visto uno a uno los diferentes elementos que se pueden utilizar. Pero no hay ningún inconveniente en usar, dentro del mismo formulario, distintos tipos de introducción de datos. Al pulsar el usuario el botón de envío recibiríamos en e-mail suyo con las distintas parejas **NAME=VALUE** de cada elemento, encadenadas con el símbolo **&**.

Ejemplo

Haremos como ejemplo práctico un libro de visitas a nuestra página.

```
<HTML>
  <HEAD>
    <TITLE>Libro de Visitas</TITLE>
  </HEAD>
  <BODY>
    <H2>Libro de visitas</H2><P>
    <FORM ACTION="mailto:email" METHOD="POST" ENCTYPE="TEXT/PLAIN">
      Tu nombre:
      <BR><INPUT TYPE="text" NAME="Nombre">
      <P>Escribe tus comentarios:<BR>
      <TEXTAREA NAME="Comentarios" ROWS="6" COLS="40">
      </TEXTAREA><P>
      <INPUT TYPE="submit" VALUE="Enviar datos">
      <INPUT TYPE="reset" VALUE="Borrar datos">
    </FORM>
    <P><HR>
  </BODY>
</HTML>
```

10. MAPAS

Hemos visto anteriormente dos maneras distintas de enlazar una página con otra: por medio de un enlace de texto o por medio de una imagen. Pero en este último caso se hacía uso de una imagen para enlazar a una única página.

Se puede utilizar una única imagen para enlazar con varias páginas, yendo a una u otra según la zona en donde se pulse el ratón. Este tipo de imágenes se llaman mapas.

Veremos cómo se crea un mapa partiendo de una imagen, y cómo se implementa dicho mapa con el lenguaje HTML en nuestra página.

Hay dos tipos de mapas:

1. Mapas gestionados por el servidor

En este tipo de mapas al desplazar el cursor sobre la imagen que forma el mapa se van obteniendo las distintas coordenadas. Cuando se pulsa el ratón en un punto determinado, el navegador envía esas coordenadas al servidor, y éste comprueba en un archivo MAP (situado en su directorio cgi-bin) cuál es la página que se corresponde con estas coordenadas, envía esta información al navegador, y éste solicita a su vez al servidor que le enlace con dicha página.

Estos mapas, tienen la ventaja de poder ser utilizados por todos los navegadores, incluso las versiones más antiguas.

Pero tienen varios inconvenientes: No todos los servidores tienen implementado el programa CGI necesario para que pueda funcionar un mapa de este tipo. Además hay que conocer el tipo de servidor (NCSA, el más común, o CERN), pues el código a aplicar es distinto en un caso u otro. Por otra parte, al requerir un trasiego de información entre el navegador y el servidor, el tiempo de respuesta es mayor que en el otro tipo, como veremos a continuación.

2. Mapas gestionados por el cliente

Estos mapas son gestionados por el cliente (nuestro navegador) y no por el servidor, como en el caso anterior.

En este tipo de mapas, al ir desplazando el cursor se observa que en ciertas zonas se convierte en el símbolo normal para enlazar con otras páginas (una mano, habitualmente).

Si se pulsa en estas zonas activas (HOTSPOTS), el navegador consulta con el documento HTML que ha recibido, y decide a qué página o dirección corresponde esa zona activa.

Es decir, no necesita enviar la información al servidor y esperar la respuesta de éste, con lo que el proceso es más rápido que en el caso anterior, reduciéndose además el tráfico de datos y la sobrecarga al servidor. Otras ventajas son que al pasar por una zona activa se muestra la dirección a la que se corresponde (al contrario que en el caso anterior, en que sólo se ven unas coordenadas). Además, estos mapas se pueden utilizar off-line, al contrario también que en el otro tipo, ya que no requieren la ayuda del servidor.

Pero tiene el inconveniente de que estos mapas no son implementados por todos los navegadores, como en el caso anterior, aunque sí por la mayoría de los utilizados actualmente, desde las versiones siguientes: Netscape 2.0, Microsoft Explorer 2.0, Mosaic 2.1, etc. Pero se puede obviar este inconveniente suministrando enlaces alternativos para los navegadores que no implementen este tipo de mapas.

En lo que sigue, vamos a ver exclusivamente cómo implementar este tipo de mapas, es decir, mapas gestionados por el cliente (client-side image maps).

Confección de la imagen del mapa

Antes que nada, es necesario confeccionar la imagen que va a servir como mapa. Se hará con un programa gráfico, y pueden utilizarse los formatos GIF o JPG.

Para crear una mapa para utilizarlo en nuestra página HTML, antes se confecciona con un programa gráfico la imagen que nos va a servir como mapa.

Para definir un área activa rectangular, necesitamos conocer las coordenadas de su ángulo superior izquierdo (X1, Y1) y las de su ángulo inferior derecho (X2, Y2).

Estas coordenadas las obtenemos con el programa gráfico con el que hemos confeccionado la imagen. Al obtener los datos necesarios (las coordenadas) podemos escribir el código HTML que convierta esta imagen en un mapa.

Código HTML para definir un mapa gestionado por el cliente

En este caso, el código que debemos escribir es el siguiente:

```
<MAP NAME="nombre">
  <AREA SHAPE="RECT" COORDS="X1,Y1,X2,Y2" HREF="paginaXX.html">
  <AREA SHAPE="RECT" COORDS="W1,W1,Z2,Z2" HREF="paginaYY.html">
  <AREA SHAPE="DEFAULT" NOHREF>
</MAP>
<IMG SRC="imagen.gif" USEMAP="#nombre">
```

<MAP NAME="nombre">

- Esta es la etiqueta de apertura del mapa, y en donde se define su nombre.

<AREA SHAPE="RECT" COORDS="X1,Y1,X2,Y2" HREF="paginaXX.html">

- Esta etiqueta define la primera zona activa. Se indica que su forma (SHAPE, en inglés) es rectangular, y que las coordenadas de su ángulo superior izquierdo son (X1,Y1), y que las del ángulo inferior derecho son (X2,Y2). A continuación se indica cuál es el enlace que se desea obtener.

<AREA SHAPE="DEFAULT" NOHREF>

- Con esta etiqueta se define el área completa del mapa, indicando que su forma es la que tiene por defecto (default), o sea rectangular. Si se quisiera que el mapa no abarcara la totalidad de la imagen, o incluso que tuviera una forma distinta, habría que indicarlo aquí, en lugar de la instrucción DEFAULT, haciendo uso del mismo tipo de instrucciones que las de las zonas activas.

</MAP>

- Etiqueta de cierre del mapa.

- Con esta etiqueta se pide al servidor que envíe una imagen llamada imagen.gif. Con USEMAP="#nombre" se indica que esta imagen es el mapa definido anteriormente con ese nombre, y que debe actuar en consecuencia a las pulsaciones del ratón, solicitando un enlace determinado en las zonas activas, o no haciendo nada en el resto.

Con el atributo NOHREF, se define que el área completa de la imagen es una zona no activa, excepto en las áreas definidas anteriormente como activas.

Enlaces alternativos de texto

Como se ha dicho anteriormente, no todos los navegadores implementan este tipo de mapas gestionados por el cliente. Por ello, es necesario suministrar unos enlaces alternativos convencionales de texto.

Por ejemplo, a continuación del mapa, se pueden poner enlaces de texto, correspondientes a las dos zonas activas (reduciendo la fuente, para hacerlo más estético):

```
<FONT SIZE=-1>
  <A HREF="paginaXX.html">Mi página XX</A>
  <A HREF="paginaYY.html">Mi página YY</A>
</FONT>
```

Zonas activas en forma de círculos o polígonos

Hasta ahora hemos definido las zonas activas en forma de rectángulos. Pero también se puede definir estas zonas activas como círculos o como polígonos.

Para definir una zona activa circular es necesario conocer las coordenadas de su centro (X,Y) y la longitud de su radio (R). Una vez sabidos estos datos, la etiqueta es como sigue:

```
<AREA SHAPE="CIRCLE" COORDS="X,Y,R" HREF="dirección_de_la_página">
```

Para definir una zona activa poligonal hay que conocer las coordenadas de los distintos puntos (X1,Y1), (X2,Y2), (X3,Y3), etc. Con estos datos, la etiqueta se escribe como sigue:

```
<AREA SHAPE="POLYGON" COORDS="X1,Y1,X2,Y2,X3,Y3,... " HREF="enlace">
```


11. FRAMES

Frames (en inglés, marcos o cuadros) es un procedimiento del lenguaje HTML para dividir la pantalla en diferentes zonas, o ventanas, que pueden actuar independientemente unas de otras, como si se tratara de páginas diferentes, pues incluso cada una de ellas puede tener sus propias barras deslizadoras. Los navegadores que lo implementan son el Netscape 2.0, y el Explorer 2.0 en adelante.

Una de sus características más importantes es que pulsando un enlace situado en un frame, se puede cargar en otro frame una página determinada. Esto se utiliza frecuentemente para tener un frame estrecho en la parte lateral (o superior) con un índice del contenido en forma de diferentes enlaces, que, al ser pulsados cargan en la ventana principal las distintas páginas. De esta manera se facilita la navegación entre las páginas, pues aunque se vaya pasando de unas a otras, siempre estará a la vista el índice del conjunto.

Para comprender los distintos conceptos vamos a desarrollar un ejemplo, creando una página con dos Frames. El de la izquierda va a servir de índice de lo que veamos en el de la derecha, y en éste veremos inicialmente una página de presentación.

Documento de definición de los frames

Lo primero que tenemos que hacer es crear un documento HTML en el que definiremos cuántas zonas va a haber, qué distribución y tamaño van a tener, y cuál va a ser el contenido de cada una de ellas.

En el ejemplo que vamos a desarrollar, la página va a tener dos frames distribuidos en columnas (es decir, uno al lado del otro, en vez de uno encima del otro, lo que sería una distribución en filas).

Con respecto al tamaño, haremos que el primero (el de la izquierda) ocupe el 20% del ancho de la pantalla, y el otro, el 80% restante.

Y con respecto al contenido, el frame de la izquierda va a contener un documento HTML que va a servir de índice de lo que veamos en el otro (y que vamos a llamar `mi_indice.html`), y el de la derecha otro documento HTML que va a servir de página de presentación (al que llamaremos `mi_presentacion.html`).

Todo lo anterior se refleja en el siguiente documento HTML:

```
<HTML>
  <HEAD>
    <TITLE>Mi primera pagina con frames</TITLE>
  </HEAD>
  <FRAMESET COLS="20%,80%">
    <FRAME SRC="mi_indice.html">
    <FRAME SRC="mi_presentacion.html" NAME="principal">
  </FRAMESET>
</HTML>
```

Obsérvese lo siguiente:

Es un documento parecido a los que conocíamos hasta ahora. La diferencia está en que en vez de utilizar la etiqueta **BODY**, que sirve normalmente para delimitar lo que se va a ver en la pantalla, se hace uso de la etiqueta **FRAMESET** (definir los frames).

En este caso, con la etiqueta `<FRAMESET COLS="20%,80%">` se define que va a haber dos frames y que van a ir en columnas. Si hubiéramos querido que fueran en filas, habríamos puesto **ROWS** (filas, en inglés). También se define el espacio en anchura que va a ocupar cada uno de ellos en la pantalla. Se ha puesto como porcentajes del total, pero se podría también haber puesto una cifra absoluta, que representaría el número de pixels a ocupar.

Ya se ha definido el número de frames, su distribución y su tamaño, pero falta por definir el contenido de cada frame. Esto se hace con las etiquetas:

```
<FRAME SRC="mi_indice.html">
<FRAME SRC="mi_presentacion.html" NAME="principal">
```

Con esto se define que el contenido del primer frame (izquierda) sea el documento HTML que hemos llamado `mi_indice.html` y el del segundo (derecha) sea el documento HTML `mi_presentacion.html`.

Obsérvese que en la etiqueta del segundo se ha incluido el atributo **NAME="principal"**, pero no así en el primero. El motivo es que se necesita dar un nombre al segundo frame, pues, como veremos a continuación, en el documento del primer frame va a haber unos enlaces que van a ir dirigidos hacia él. En este caso sólo tenemos dos frames, pero podría haber más, y es necesario distinguirlos unos de otros. Y el primero no necesita nombre, pues no va a haber enlaces en el segundo dirigidos hacia él.

A este documento le vamos a llamar **mi_pagina.html**, pero todavía no lo vamos a guardar, pues falta por añadir algo que veremos más adelante.

Documentos HTML de cada frame

Necesitamos ahora confeccionar el documento HTML de cada uno de los frames. Recuérdese que son como páginas independientes, que pueden tener cada una su propio fondo, etc., y todo lo que queramos añadir en ellos.

Documento del frame de la izquierda

Va a tener un fondo amarillo, y va a contener dos enlaces dirigidos al frame de la derecha.

```
<HTML>
  <HEAD>
    <TITLE> Indice </TITLE>
  </HEAD>
  <BODY BGCOLOR="#FFBB00">
    <P><A HREF="mi_presentacion.html" TARGET="principal">Presentación</A>
    <P><A HREF="otra_pagina.html" TARGET="principal">Esta es otra página</A>
    <P><IMG SRC="imagen.gif">
  </BODY>
</HTML>
```

Dentro de las etiquetas de los enlaces podemos observar algo nuevo, y es el atributo **TARGET** (en inglés: objetivo, blanco), que sirve para hacer que al ser activado el enlace no se cargue en el propio frame, sino en otro, precisamente en el que hemos llamado con ese nombre en el documento de definición de los frames.

En nuestro caso, le hemos dado el nombre de **"principal"** al frame de la derecha, y es por tanto ahí donde se van a cargar los documentos HTML.

Guardamos este documento con el nombre de **mi_indice.html**

Documento del frame de la derecha

Va a tener un fondo negro, y va a contener sólo un texto.

```
<HTML>
  <HEAD>
    <TITLE> Presentación </TITLE>
  </HEAD>
  <BODY BGCOLOR="#000000" TEXT="#0000FF">
    <CENTER><FONT SIZE=+3><STRONG>
      <P>Esta es la versión
      <BR><FONT COLOR="#FF0000">con frames</FONT>
      <BR>de mi página
    </STRONG></FONT></CENTER>
  </BODY>
</HTML>
```

Guardamos este documento con el nombre de **mi_presentacion.html**

Con lo visto anteriormente, ya tenemos las tres piezas que necesitamos:

- Documento de definición de los frames
- Documento del contenido del primer frame
- Documento del contenido del segundo frame

Alternativa para los navegadores que no soportan frames

Hay que prever el caso en que los navegadores no soportan o desconocen las etiquetas **FRAMESET** y **FRAME**.

Para estos casos está prevista la etiqueta **<NOFRAMES>** y **</NOFRAMES>**. Se añaden al final del documento de definición de los frames, y se pone entre ambas lo que verán los que acceden con un navegador que no soporta frames. Puede incluso ser el código HTML de una página completa.

En nuestro caso, pudiéramos poner un mensaje advirtiendo de esta circunstancia, y dirigiendo al usuario, con un enlace normal, hacia una página que hayamos definido como principal o de presentación para estos casos. (También podríamos no decir nada sobre frames, y colocar aquí el código de la página de presentación)

```
<HTML>
  <HEAD>
    <TITLE>Mi pagina con frames</TITLE>
  </HEAD>
  <FRAMESET COLS="20%,80%">
    <FRAME SRC="mi_indice.html">
    <FRAME SRC="mi_presentacion.html" NAME="principal">
  </FRAMESET>
  <NOFRAMES>
    Estas utilizando un navegador que no soporta frames.
    <P>Pulsa para visitar mi <A HREF="presentacion.html"> página</A>.
  </NOFRAMES>
</HTML>
```

Ahora sí podemos guardar este documento con el nombre de `mi_pagina.html`.

Atributos de la etiqueta <FRAMESET>

La etiqueta **FRAMESET**, como hemos visto en el ejemplo, es la que define la distribución, el número y tamaño de los frames. Tiene dos atributos: **COLS** (columnas) y **ROWS** (filas):

```
<FRAMESET COLS="XX,YY,ZZ,...">
<FRAMESET ROWS="XX,YY,ZZ,...">
```

Define la distribución de los frames en columnas o en filas, según se use uno u otro atributo. Por ejemplo:

```
<FRAMESET COLS="XX,YY">           (habrá dos frames en columnas)
<FRAMESET COLS="XX,YY,ZZ">        (habrá tres frames en columnas)
```

Define el tamaño de los frames, según el valor que demos a **XX**, **YY**, **ZZ**.... Este valor se puede expresar en un porcentaje del ancho del pantalla (para las columnas), o del alto de la pantalla (para las filas). Así, por ejemplo:

```
<FRAMESET COLS="%20,%80">
(la columna de la izquierda ocupará el 20% del ancho de la pantalla, y la de la derecha el 80% restante)

<FRAMESET ROWS="%10,%70,%20">
(la fila superior ocupará el 10% del alto de la pantalla, la del medio el 70%, y la inferior el 20%)
```

Un número absoluto que representa el número de pixels que ocupará cada frame a lo ancho o a lo alto (según sean filas o columnas). Así, por ejemplo:

```
<FRAMESET COLS="40,600">
(la columna de la izquierda tendrá 40 pixels de ancho y la de la derecha 600).
```

Pero es peligroso utilizar sólo valores absolutos, pues el tamaño de la pantalla varía de un usuario a otro. Si se va a usar algún valor absoluto para un frame, es mejor mezclarlo con alguno relativo, como los que vamos a ver a continuación, para que se ajuste el total a la pantalla del usuario

Si se pone un valor relativo poniendo un asterisco (*), en vez de un número, se interpreta como que ese frame debe tener el espacio restante. Por ejemplo:

```
<FRAMESET ROWS="100,*,100">
(Habrá tres filas, la superior y la inferior de una altura de 100 pixels, y la del medio el espacio restante).
```

Si hay más de un frame con *, ese espacio restante se dividirá por igual entre ellos. Si hay un número antes del *, ese frame obtiene esa cantidad más de espacio relativo. Así "2*,*" daría 2/3 para el primer frame y 1/3 para el otro.

Frames sin bordes

Si se desea que no haya un borde de separación entre los frames, se deben incluir el atributo `FRAMEBORDER="0"` dentro de la etiqueta `FRAMESET`. No todas las versiones de los navegadores lo implementan.

Para que también desaparezcan los huecos de separación entre frames hay que añadir otros dos atributos (el primero es para el Explorer y el segundo para el Netscape): `FRAMESPACING="0"` y `BORDER="0"`, con lo que la etiqueta completa quedaría:

```
<FRAMESET FRAMEBORDER="0" FRAMESPACING="0" BORDER="0" COLS="xx,yy">
```

Atributos de la etiqueta <FRAME>

Esta etiqueta define las características de un frame concreto, no del conjunto de los frames, como era el caso con la etiqueta `<FRAMESET>`. Puede tener los siguientes posibles atributos, que van dentro de la etiqueta `<FRAME>`:

`SRC="dirección"`

- Esta dirección puede ser la de un documento HTML (tal como hemos utilizado en el ejemplo), o cualquier otro recurso Web (o URL). Con este atributo se indica lo que se cargará inicialmente en el frame. Si no se le pone este atributo a la etiqueta `<FRAME>`, entonces dicho frame aparecerá inicialmente vacío, aunque tendrá las dimensiones asignadas.

`NAME="nombre_de_la_ventana"`

- Este atributo se usa para asignar un nombre a un frame. De esta manera se podrá "dar en el blanco" (en inglés, target) en esta página, desde un enlace situado en otra página. Es decir, que pulsando en otra página un enlace, se cargará precisamente en ésta, tal como hemos visto en el ejemplo. El atributo `NAME` es opcional. Por defecto, todas las ventanas carecen de nombre. Los nombres que se escojan deben comenzar por un carácter alfanumérico.

`MARGINWIDTH="número"`

- Se utiliza este atributo cuando se quiere controlar el ancho de los márgenes dentro de un frame. El número que se ponga representa los pixels de los márgenes. Este atributo es opcional.

`MARGINHEIGHT="número"`

- Igual que el anterior, pero referido a los márgenes en altura.

`SCROLLING="yes | no | auto"`

- Este atributo se utiliza para decidir si el frame tendrá o no una barra deslizadora. Si se elige "yes" tendrá siempre una barra deslizadora. Si se elige "no" no la tendrá nunca, y si se elige "auto", será el navegador quien decida si la tendrá o no. Este atributo es opcional. Su valor por defecto es "auto".

`NORESIZE`

- A este atributo no se le asigna un valor numérico, como a los demás. Es un indicador para que la ventana no se pueda redimensionar (en inglés, resize) por parte del usuario. Es un atributo opcional. Por defecto, todos los frames son redimensionables.

`FRAMEBORDER="no"`

- Este atributo elimina el borde en un frame, pero si se quiere que se elimine completamente, también hay que ponérselo al frame contiguo. Si se quiere eliminar los bordes de todos los frames, se debe colocar en la etiqueta `FRAMESET`, como hemos visto anteriormente.

El atributo TARGET

En el ejemplo hemos visto que, como queríamos que los enlaces situados en el frame de la izquierda surtieran efecto no en él mismo, sino en otro frame (el situado a la derecha), teníamos que poner dentro de cada enlace el siguiente atributo `TARGET="principal"`, siendo "principal" el nombre que habíamos dado al segundo frame, en el documento de definición de frames. Es decir, hemos utilizado el atributo de esta manera:

`TARGET="nombre_dado_a_otro_frame"`

Estos nombres, que los escogemos nosotros, pueden ser cualquiera, pero con la condición que el primer carácter sea alfanumérico (letra o número).

Pero hay unos nombres reservados (es decir, que no se pueden usar para denominar a un frame), que hacen que este atributo efectúe unas funciones especiales. Para que cumplan su cometido, es imprescindible escribir estas palabras reservadas (blank, self y top) en minúsculas.

TARGET="_blank". Hace que se abra una nueva copia del navegador, y el enlace activado se carga en ella, a pantalla completa. Es decir, tendríamos dos copias funcionando a la vez.

TARGET="_self". Hace que el enlace se cargue en el propio frame.

TARGET="_top". Hace que el enlace se cargue a pantalla completa, suprimiendo todos los frames, pero sin que se cargue una nueva copia del navegador. Un error común es olvidar poner este atributo en los enlaces que están en un frame, con lo que al ser activados, la página llamada se carga en el propio frame, lo cual es muy molesto si esa página pertenece a otro sitio web. Esto se evita poniendo este atributo en las etiquetas de los enlaces.

Frames anidados dentro de otros frames

Hasta ahora hemos contemplado sólo la posibilidad de tener una distribución de los frames en filas o en columnas, (dependiendo que se utilice el atributo ROWS o COLS en la etiqueta FRAMESET), pero no ambos a la vez. Se puede obtener distribuciones complejas anidando filas en una columna, o a la inversa, columnas dentro de una fila.

Supongamos que queremos la siguiente distribución:

- Un frame estrecho en horizontal en la parte superior, de lado a lado de la pantalla (altura, el 15%)
- Otro frame estrecho en vertical en la parte izquierda, debajo del anterior (anchura, el 20%)
- Un tercero ocupando el resto de la pantalla.

Vemos que, en realidad, esto equivale a la siguiente distribución:

- Dos filas. La superior ocupa el 15% y la inferior el resto.
- La fila inferior dividida en dos columnas. La primera (la de la izquierda) ocupa el 20% y la otra, el resto.

El documento de definición de las dos filas (olvidémonos de momento que la de abajo está subdividida), sería:

```
<HTML>
  <HEAD>
    <TITLE>Pagina con dos filas</TITLE>
  </HEAD>
  <FRAMESET ROWS="15%,*">
    <FRAME SRC="documento_fila_superior">
    <FRAME SRC="documento_fila_inferior">
  </FRAMESET>
</HTML>
```

Como la fila inferior, en realidad, son dos columnas (con una distribución del 20% y resto), sustituimos (anidando) la etiqueta <FRAME SRC="documento_fila_inferior"> por:

```
<FRAMESET COLS="20%,*">
  <FRAME SRC="documento_columna_izqda">
  <FRAME SRC="documento_columna_dcha">
</FRAMESET>
```

Con lo que queda el documento definitivo así:

```
<HTML>
  <HEAD>
    <TITLE>Pagina con fila superior y dos columnas inferiores</TITLE>
  </HEAD>
  <FRAMESET ROWS="15%,*">
    <FRAME SRC="documento_fila_superior">
    <FRAMESET COLS="20%,*">
      <FRAME SRC="documento_columna_izqda">
      <FRAME SRC="documento_columna_dcha">
    </FRAMESET>
  </FRAMESET>
</HTML>
```

12. SONIDOS

Una página Web puede tener sonidos incorporados, bien sea como un fondo sonoro que se ejecuta automáticamente al cargar la página, o como una opción para que la active el propio usuario.

Capacidades sonoras de los navegadores

Para poder escuchar los sonidos es necesario disponer, como es lógico, de una tarjeta de sonido con sus correspondientes altavoces. Pero esto no es suficiente, pues no todos los programas navegadores funcionan igual.

Explorer de Microsoft

A partir de la versión 2.0 es capaz de reproducir fondos sonoros sin necesidad de añadir nada. Además, a partir de la versión 3.0 del Explorer, es incluso compatible con los plug-ins del Netscape.

Netscape

Las versiones anteriores a la 2.0 requieren que se activen programas auxiliares. La versión 2.0 sí es capaz de reproducir un fondo sonoro, pero es necesario que tenga instalado un plug-in llamado Crescendo. La versión 3.0 lleva el plug-in Live Audio (sólo la versión completa, no la reducida). que le hace capaz de reproducir un fondo sonoro.

Pero a todas estas complicaciones de las distintas versiones de los navegadores, hay otra más y es que el servidor donde esté alojada la página Web debe tener configurados como MIME los formatos, .MID y .WAV. Si esto no es así, aunque depositemos en el servidor nuestro documento HTML acompañado por el archivo de sonido, éste no se ejecutará.

Este problema no existe, sin embargo para el Explorer, con el que no nos tenemos que preocupar por este tema.

Fondo sonoro para Microsoft Internet Explorer

Para las versiones 2.0 en adelante, se utiliza la etiqueta:

```
<BGSOUND SRC="archivo_de_sonido" LOOP="n">
```

El archivo de sonido puede estar en formato .MID o .WAV

El atributo LOOP (en inglés, lazo) sirve para especificar el número (n) de veces que se debe ejecutar el archivo de sonido. Si se escoge el número n="-1" o se pone LOOP="infinite", el sonido se ejecutará indefinidamente. Se puede omitir este atributo, y entonces el archivo se ejecutará una sola vez.

La etiqueta para que se ejecute el archivo sonido.mid dos veces en el Explorer es:

```
<BGSOUND SRC="sonido.mid" LOOP="2">
```

Fondo sonoro para Netscape

La etiqueta básica para Netscape es:

```
<EMBED SRC="archivo_de_sonido" WIDTH=XXX HEIGHT=YY>
```

donde WIDTH es la anchura y HEIGHT la altura de una consola que aparece, y que tiene diferentes teclas (Play, Stop, Pausa, etc.). Respecto a las dimensiones XXX e YY estas dependerán del plug-ins que existe para Netscape.

El archivo de sonido puede estar en formato, .MID o .WAV, pero recuérdese la advertencia hecha anteriormente, de que estos formatos deben estar configurados como MIME por el servidor donde esté alojada la página.

Dentro de la etiqueta podemos añadir los siguientes atributos opcionales:

```
AUTOSTART="true"   Arranca automáticamente.  
LOOP="true"        Se ejecuta ininterrumpidamente
```

Para conseguir que la consola sea invisible hay que añadirle el atributo HIDDEN="true".

```
<EMBED SRC="sonido.mid" HIDDEN="TRUE">
```

Fondo sonoro combinado para el Explorer y el Netscape

Se puede combinar los dos tipos distintos de etiquetas para conseguir que un fondo sonoro sea escuchado por usuarios que utilicen tanto el Explorer como el Netscape (siempre que éste último esté preparado para ello). En este caso, el Explorer ignorará la etiqueta del Netscape, y viceversa.

Las dos etiquetas necesarias para que se reproduzca un archivo de sonido ya sea en formato, .WAV o .MID, como sonido de fondo, tanto por el Explorer como por el Netscape (lo hará una sola vez):

```
<BGSOUND SRC="sonido.mid">           Para el Explorer.  
<EMBED SRC="sonido.mid" HIDDEN="true"> Para el Netscape.
```

Como vimos anteriormente para que un archivo de sonido, .MID o .WAV se ejecute como sonido de fondo, tanto por el Explorer como por el Netscape y que se ejecute indefinidamente, a la etiqueta del Explorer basta con añadirle el atributo `LOOP="infinite"`, como se ha visto anteriormente. Y con respecto a la etiqueta del Netscape, teóricamente debería también bastar añadirle el atributo `LOOP="true"`, pero curiosamente esto no es así, porque hay que poner las dimensiones de la consola (que no se va a ver).

Las dos etiquetas quedarían de esta manera:

```
<BGSOUND SRC="sonido.mid" LOOP="infinite">  
<EMBED SRC="sonido.mid" WIDTH="200" HEIGHT="55" AUTOSTART="true"  
      LOOP="true" HIDDEN="true">
```

Activación del sonido por el propio usuario

Hemos visto cómo poner un sonido de fondo en una página. Hay otra opción, mucho más sencilla, y es la de poner un enlace a un archivo de sonido, de tal manera, que al pulsarlo se ejecute el archivo. (Ver la Capítulo 4, en el que se explica cómo crear enlaces).

Por ejemplo, poner un enlace a un archivo sonido.mid:

```
Escucha esta <A HREF="sonido.mid"> musica </A>
```

Al pulsar el enlace se activa, en una ventana aparte, el programa que ejecuta el sonido. Esto es válido para todos los navegadores, incluso las versiones más antiguas, con la única condición de que se haya configurado un programa auxiliar capaz de ejecutar archivos .MID o .WAV

Se puede hacer el enlace con un icono (Ver el Capítulo 5). Por ejemplo al hacer un enlace con un icono que se llamara `snd.gif` al archivo de sonido `sonido.wav` el enlace quedaría así:

```
<A HREF="sonido.wav"> <IMG SRC="snd.gif"> </A>
```

En el capítulo anterior creamos una versión con frames, en la que aparece inicialmente en el frame de la derecha una página de presentación.

Se puede colocar a esta página una música de fondo, para que sea ejecutada por el Explorer y el Netscape. Para ello debemos hacer lo siguiente:

En el documento de la derecha que llamamos `presentacion.html` debemos añadir entre las etiquetas situadas al final `` y `` lo siguiente:

```
<BR><FONT COLOR="#FF0000"> ¡Escucha la música de fondo! </FONT>  
<BGSOUND SRC="sonido.mid">  
<EMBED SRC="sonido.mid" WIDTH="2" HEIGHT="0" AUTOSTART="true">
```