Práctica 3:
Creación de una página web para consultar una base de datos

Dpto. Lenguajes y Sistemas Informáticos

Ofimática 2002/2003

Contenidos

1 Introducción a Internet

Internet no es más que la conexión de un conjunto de redes, las cuales no tienen que ser del mismo tipo, muchas veces, lo único que tienen en común es la interconexión que tienen entre ellas.

Internet tiene sus inicios en la guerra fría, en un intento de que la red de comunicaciones norteamericana no estuviera centralizada, evitando el riesgo que suponía perder la red a causa de que se destruyera el nodo principal de la misma. La idea era dividir la información a transmitir en paquetes teniendo cada paquete su identificador, la dirección de destino y su número de paquete. Estos paquetes, una vez llegados a su destino se ordenarían para componer el mensaje inicial.

Con este sistema no existían nodos principales ni subordinados, todos eran iguales y los distintos paquetes podían seguir caminos totalmente distintos, no previsibles a priori. Incluso si se destruía una parte importante de la red, los paquetes podían circular por el resto de la red.

El nacimiento oficial se fija en 1969 con ARPANET, una red experimental patrocinada por la Agencia de Ingestigación de Proyectos Avanzados (ARPA) del Departamento de Defensa (DoD), que tiene el propósito de probar nuevas tecnologías de conexión enlazando centros de investigación.

Los servicios que ofrece intenet son:

- **Correo electrónico:** De los más usados, permite enviar y recibir mensajes entre personas.
- **FTP:** Permite la transferencia de ficheros entre máquinas de la red. Directamente relacionado con el software de dominio público.
- **News:** Grupos de noticias temáticos. Te suscribes a un grupo y cada vez que te conectas puedes leer todo lo que los miembros del grupo han enviado y al mismo tiempo, lo que tu envíes lo podrán leer todos aquellos que estén suscritos. Están organizados en siete categorías principales:
  - *comp:* Temas relacionados con hardware.
  - *news:* Sobre software de administración de news.
  - *rec:* Aficiones de todo tipo.
  - *sci:* Temas científicos.
  - *soc:* Aspectos sociales y culturales.
  - *talk:* Debates sobre política, religión, etc.
  - *misc:* Temas que no se ajusten a ninguno de los anteriores.
  - *alt:* Un grupo que no está en todos los servidores, se conoce como un lugar *sin ley*, sin censuras ni moderadores, donde se discuten temás delicados y a veces ilícitos.
- **Gopher:** A mitad de camino entre ftp y www. Consiste en una estructura jerarquica de menús donde cada entrada del menú puede apuntar a un fichero o a otro menú.
- **Archie:** Permite localizar información a través de la red. Encuentra ficheros a través de su nombre.
- **WAIS:** Complementa a archie, ya que encuentra ficheros a través de su contenido.
- **WWW:** El más popular y el responsable del gran crecimiento de internet. Permite acceder a documentos que a su vez tienen enlaces a otros documentos.

La WWW (World Wide Web) es un sistema de intercambio de información multimedia desarrollado inicialmente en el CERN (Centro Europeo de Investigación Nuclear) de Ginegra en 1989 para el intercambio interno de la documentación de sus proyectos. Pero pronto empezó a difundirse a otras entidades con el apoyo de internet.

Asociados a la WWW se encuentran los siguientes conceptos:

- **URI** (Universal Resource Identifiers) y **URL** (Universal Resource Locators). Son cadenas que identifican los objetos de la red. Son universales, pueden etiquetar objetos exportados con cualquier protocolo.
- **HTTP** (Hypertext Transmission Protocol). Protocolo de comunicaciones, opera sobre TCP/IP y se basa en transferencias de texto. La comunicación solo se mantiene el tiempo necesario para hacer la operación.
- **HTML** (Hypertext Markup Language). Lenguaje que permite integrar en un mismo documento objetos de distintas naturalezas y es lo suficientemente simple como para ser escrito sin la ayuda de software adicional.

La comunicación WWW se basa en la estructura cliente-servidor. En un extremo de la línea se encuentra un servidor HTTP al que accede desde el otro extremo de la línea un cliente o browser que conoce la dirección URL del servidor y le solicita un documento HTML. Cuando lo recibe, lo interpreta y lo muestra al usuario que se encuentra en el cliente.

Las aplicaciones que tiene la WWW son numerosas de las cuales destacaremos: las home-pages (páginas personales de usuario, donde cada cual cuenta un poco de si mismo y sus aficiones), turismo, catálogos de servicios, boletines oficiales, información local, museos, centros de enseñanza, fondos documentales, librerías electrónicas, índices, publicidad y marketing, eventos, venta por catálogo, entretenimiento, información deportiva, anuncios, etc.

2 Introducción a HTML

HTML no es un lenguaje de programación sino un lenguaje de descripción de documentos y sus componentes, que consta de los siguientes elmentos:

Caracteres especiales: Comienzan por `&` seguidos de una orden y finalizan por `;` y nos van a permitir escribir vocales acentuadas (`&aacute;`) o la ñ (`&ntilde;`). Aquí tenemos los más usuales que usamos en nuestro idioma:

&<vocal>acute;	vocal con tilde aguda á
&<vocal>grave;	vocal con tilde grave à
&<vocal>circ;	vocal con tilde circunfleja â
&<vocal>uml;	vocal con diéresis ü
ñ	ñ
ç	ç
<	<
>	>
&	&
"	”
¡	¡
¿	¿

Directivas: Son las órdenes del lenguaje, mediante ellas le indicamos al browser como debe modificar el aspecto del texto. Se incluyen entre los símbolos < y >. Las hay de dos tipos:

- Abiertas: No requieren ir por parejas, como por ejemplo <br> que introduce un salto de línea.
- Delimitadoras: Van por parejas encerrando un texto, por ejemplo hola mundo iría en negrita.

Pueden tener atributos que modifican su significado, en el ejemplo <font size=7>, font es la directiva y size un atributo que indica el tamaño del texto.
Cuando damos una directiva o atributo erróneo, el browser no mostrará ningún mensaje de error, simplemente ignorará la orden.

Comentarios: Como en los lenguajes de programación podemos introducir comentarios encerrándolos entre <! − − y − − >

Hay que tener en cuenta que los espacios, tabuladores y saltos de linea no son interpretados como tales, es decir, que es lo mismo separar dos palabras por un espacio o por diez, el browser lo mostrara igual.

3 Estructura básica

Un documento HTML debe ir encerrado entre las directivas <HTML> y </HTML>.

Dentro nos encontraremos las directivas pareadas <HEAD> para almacenar información sobre el propio documento y <BODY> para almacenar el contenido del documento.

Dentro del bloque HEAD nos podemos encontrar las siguientes directivas:

- <BASE>: Que indica la dirección base a partir del cual se van a construir las direcciones relativas. *Ejemplo:* <BASE href="http://www.ugr.es/"> indica que si después en un enlace hacemos referencia al documento "becas.html" en realidad estamos haciendo referencia al documento "http://www.ugr.es/bec
- <META>: Para darle información no definida en las restantes directivas. Por *ejemplo*, si queremos que un documento se carge automáticamente transcurridos 10 segundos desde que se cargó el actual usaríamos la directiva como sigue:
<META http-equiv=REFRESH content="10; URL=siguiente.html">
- <TITLE>: Directiva pareada para darle título a la ventana del browser.

Ejemplo

```
<html>
<head>
    <title>Pagina de   H o l a   M u n d o </title>
</head>
<body>
    Hola Mundo.
</body>
</html>
```

La directiva <BODY> tiene atributos para controlar los colores y las imágenes de fondo. Dichos atributos son los siguientes:

- `background` : Al que se le asigna el fichero que contiene el fondo deseado.
  `<BODY background="fondos/cielo.gif">`
- `bgcolor` : Para establecer un color de fondo, aunque no tiene efecto si se ha puesto como fondo un fichero usando el atributo anterior.
  `<BODY bgcolor="#RRGGBB">`
- `text` : Color del texto.
- `link` : Color de los enlaces.
- `alink` : Color de los enlaces que aún no se han visitado.
- `vlink` : Color de los enlaces que han sido visitados.

4 Texto

Al introducir texto, este se mostrará todo seguido, sin tener en cuenta los saltos de línea que existen en el código fuente. Por ello cuando queramos incluir un salto de línea que se refleje como tal tendremos que hacerlo con la directiva <BR>.

Aunque no se suele usar para separar párrafos pues para ello existe la directiva pareada <P>. Que además de incluir un salto de línea al final del párrafo deja un espacio mayor con la siguiente línea.

Esta directiva dispone del atributo `align` que puede tomar los valores `center` y `right` para centrar el texto o alinearlo a la derecha respectivamente.

Para evitar que un trozo de texto se divida por un sálto de línea debemos encerrarlo con la directiva pareada <NOBR>. Dentro de un bloque <NOBR> se indican los sitios donde se permiten las divisiones con la directiva <WBR>.

Para incluir texto tal cual aparece en código fuente se usa la directiva pareada <PRE>, muy usado para mostrar ficheros de texto y códigos fuente de los lenguajes de programación.

Con las directivas pareadas <B>, <I>, <U> y <TT> coseguimos texto en negrita, cursiva, subrayado y letra no proporcional respectivamente, aunque se suele aconsejar por compatibilidad usar estilos lógicos que son los siguientes:

- <STRONG>: Para resaltar texto.
- <EM>: Para enfatizarlo.
- <CITE>: Para incluir citas.
- <KBD>: Caracteres de teclado.
- <CODE>: Código.
- <DFN>: Definición.
- <SAMP>: Ejemplo.
- <BLOCKQUOTE>: Bloque de texto indentado.
- <VAR>: Variable.
- <ADDRESS>: Firmas y direcciones.

Los distintos browsers hacen la correspondencia entre estos estilos lógicos y los estilos que pueden mostrar. Normalmente, <STRONG> se hace corresponder con <B> pero si un ordenador no puede mostrar negrita, establecerá la correspondencia de <STRONG> con otro estilo para que el usuario vea el texto resaltado.

Para cambiar el tamaño del texto tenemos dos directivas:

- <BASEFONT>: Que pone el tamaño por defecto para todo el texto.
- <FONT>: Pareada, que pone el tamaño para el texto encerrado por la directiva.

Ambas se usan con el atributo `size` al cual se le pueden dar los valores 1, 2, 3, 4, 5, 6 y 7, estando el valor por defecto en 3. También se le pueden dar incrementos relativos anteponiendo al dígito los signos + y -.

El color del texto se cambia con el atributo `color` al que se le da valores RGB con el formato `#RRGGBB` encerrado entre comillas.

Disponemos de seis directivas pareadas para incluir secciones, desde la mayor <H1> a la más pequeña <H6> que admiten el atributo `align` de alineación.

También podemos incluir líneas de separación con la directiva <HR> que tiene los siguientes atributos:

size= Para indicar el grosor de la línea.

width= Para indicar la anchura de la linea en pixeles o en porcentaje de la ventana.

align= Para alinearla, tiene sentido cuando no ocupa toda la ventana.

noshade= Para eliminar el efecto tridimensional que por defecto tiene.

Ejemplo

```
<html>
<head>
   <title>Pagina de   H o l a   M u n d o </title>
</head>
<body>
  Hola Mundo.
  Hola Mundo. <br>
  Hola Mundo.

  <p align=center> Hola Mundo. </p>
  <p align=right> Hola Mundo. </p>

  ---------- <br>
  Hola Mundo <br>
  ---------- <br>

  <pre>
  ----------
  Hola Mundo
  ----------
```

```
</pre>

<p> <b>Hola Mundo.</b>
     <i>Hola Mundo.</i>
     <u>Hola Mundo.</u>
     <tt>Hola Mundo.</tt></p>
<p> <strong>Hola Mundo strong</strong><br>
     <em>Hola Mundo em</em><br>
     <cite>Hola Mundo cite</cite><br>
     <kbd>Hola Mundo kbd</kbd><br>
     <code>Hola Mundo code</code><br>
    <dfn>Hola Mundo dfn</dfn><br>
    <samp>Hola Mundo samp</samp><br>
    <var>Hola Mundo var</var><br>
    <address>Hola Mundo address</address></p>

  <blockquote>Este texto, est&aacute; sangrado con respecto al
       resto del texto, ello ha sido posible con la directiva
       BLOCKQUOTE, la cual se puede poner cuantas veces se quiera.
  <blockquote>&#191;De acuerdo?</blockquote></blockquote>

  <p>
  <font size=1 color="#0000ff">Hola Mundo.</font>
  <font size=2 color="#00ff00">Hola Mundo.</font>
  <font size=3 color="#00ffff">Hola Mundo.</font>
  <font size=4 color="#ff0000">Hola Mundo.</font>
  <font size=5 color="#ff00ff">Hola Mundo.</font>
  <font size=6 color="#ffff00">Hola Mundo.</font>
  <font size=7 color="#ffffff">Hola Mundo.</font></p>

  <hr size=10 width=75% align=center>

  <h1>Secci&oacute;n 1</h1>
  <h2>Secci&oacute;n 2</h2>
  <h3>Secci&oacute;n 3</h3>
  <h4>Secci&oacute;n 4</h4>
  <h5>Secci&oacute;n 5</h5>
  <h6>Secci&oacute;n 6</h6>
</body>
</html>
```

5 Listas

Podemos hacer dos tipos de listas: sin numerar y numeradas. Las listas sin numerar se realiza con la directiva pareada <UL> donde cada nuevo item se indicará con la directiva pareada <IL>.

Ejemplo

```
<UL>
<LI> Item 1 </LI>
```

```
<LI> Item 2 </LI>
</UL>
```

Ambas directivas cuentan con el atributo `type` para variar el aspecto del caracter de inicio de los items. Este atributo puede tomar los siguientes valores:

Valor	Carácter	
disc	•	(defecto)
circle	◦	
square	□	

Se puede usar el atributo tanto en la directiva <UL> afectando a toda la lista, como en el la directiva <LI> afectando a dicho item y a los siguientes.

En cuanto a las listas numeradas, se realizan con las directivas pareadas <OL> para englobar la lista y <LI> para cada item. Los atributos que poseen son para indicar, los caracteres de numeración `type` (que se puede usar tanto en <OL> como en <LI>) y el valor de inicio del contador `start` (para <OL>) y `value` (para <LI>).

Para el atributo `type` tenemos los siguientes valores:

Valor	Modo de numeración	
1	1, 2, 3, 4, ...	(defecto)
A	A, B, C, D, ...	
a	a, b, c, d, ...	
I	I, II, III, IV, ...	
i	i, ii, iii, iv, ...	

Los atributos `start` y `value` reciben el valor numérico de inicio (siempre en numeración latina).

Los atributos que se pongan en <OL> afectan a toda la lista y los que se pongan en <LI> afectan a ese item y siguientes.

Hay un tercer tipo de listas para definiciones, las directivas son: <DL> para englobar la lista, <DT> para el texto que definimos y <DD> para la definición.

Por último decir que las listas se pueden englobar unas dentro de otras.

Ejemplo:

```
<html>
<head>
  <title>Pagina de   H o l a   M u n d o </title>
</head>
<body>
    <ul>
    <li> Hola mundo </li>
        <ol type=i>
        <li> Hola mundo numerado </li>
        <li> Hola mundo numerado </li>
        </ol>
    <li type=circle> Hola mundo </li>
        <ol type=a start=3>
        <li> Hola mundo que contin&uacute;a la
               numeraci&oacute;n </li>
        <li> Hola mundo </li>
        </ol>
```

```
        <li type=square> Hola mundo </li>
            <dl>
            <dt>Hola mundo</dt>
            <dd>Dicese de la frase usada en programaci&oacute;n
                    para experimentar.</dd>
            <dt>Pepeillo</dt>
            <dd>Otro de los ejemplos m&aacute;s usados.</dd>
            </dl>
        </ul>
    </body>
    </html>
```

6 Enlaces

La gran potencia que proporciona la WWW es la posibilidad de hipertexto, que desde un documento carguemos otro o nos desplacemos a un punto distinto dentro del mismo documento.

Ello es posible gracias a la directiva <A> que con su atributo `href` podemos indicar el documento a cargar y la zona del documento en donde situarnos. Por ejemplo el código

```
<A href="granada.html"> Visita Granada </A>
```

producirá que la frase *Visita Granada* se muestre de otro color y subrayada, y que cuando piquemos con el ratón sobre ella, se cargue el documento `granada.html` que se encuentra en la localización indicada en la directiva <BASE> o en su defecto en la misma localización donde se encuentra el documento actual.

Para movernos dentro de un documento o acceder a una posición del documento distinta del principio, debemos situar marcas de destino. Ello se hace con el atributo `name` de la directiva <A>. Y para acceder a dicha marca se la añadiremos a la localización separada por `#`. En el siguiente ejemplo lo veremos mejor:

```
    <H1>&Iacute;ndice General</H1>
    <ol>
    <li><A href="#intro">Introducci&oacute;n</A></li>
    <li><A href="#tecbas">T&eacute;cnicas b&aacute;sicas</A></li>

    <!-- El resto del indice -->

    <hr> <!-- Linea separadora -->

    <H1><A name="intro">Introducci&oacute;n</A></H1>

    <!-- Texto de la introduccion -->

    <H1><A name="tecbas">T&eacute;cnicas B&aacute;sicas</A></H1>

    <!-- Texto de las tecnicas basicas -->
```

Para poner enlaces a documentos de otros servidores tendremos que darle la dirección completa, es decir:
`http://servidor/directorio/[fichero.html[#marca]]`

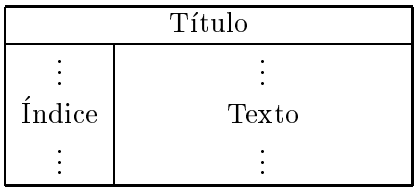
Si no ponemos la `marca` nos situaremos al principio del citado documento. Si no indicamos el fichero, el browser intentará cargar el fichero por defecto que normalmente es `index.html`, si no existe un fichero con dicho nombre, mostrará un listado de los ficheros del directorio para que carguemos el que deseemos haciendo uso del ratón.

Por último indicar que podemos poner enlaces a otros servicios de internet como servidores FTP (ftp://servidor/directorio), ficheros locales (file://fichero), grupos de noticias (news://grupo), terminal (telnet:host) o correo electrónico (mailto:dirección).

7 Marcos

Una práctica muy habitual es dividir la pantalla en zonas para situar en dichas zonas distintas partes de las páginas buscando una mayor claridad de navegación a través de ellas.

Por ejemplo, podemos pensar en una organización como la siguiente:



De modo que ofrecemos al visitante a nuestras páginas una cabecera común a todas ellas y un índice que le permita un acceso rápido a cualquiera de las secciones que componen la información que ofrecemos a los navegantes.

Las directivas para hacer esto son <FRAMESET> (que es pareada) y <FRAME>. Con <FRAMESET> hacemos la división de la pantalla y con <FRAME> le indicamos el contenido de cada división realizada. Las directivas <FRAMESET> pueden estar anidadas, con lo cual podemos realizar nuevas divisiones en los marcos resultantes de una división anterior. Se usan dentro de la directiva pareada <HEAD>.

<FRAMESET> se usa con uno de estos dos atributos: `rows` que realiza divisiones horizontales segun los porcentajes que se le indiquen; y `cols` que realiza las divisiones verticales.

<FRAME> se usa con el atributo `src` al que se le asigna la página a cargar en este marco y opcionalmente con el atributo `name` para darle un nombre al marco al que hacer referencia como destino de otras páginas que deseemos cargar ahí.

Ejemplo

Fichero `principal.html`

```
<html>
<head>
    <frameset rows="10%,90%">
        <frame src="cabecera.html">
        <frameset cols="25%,75%">
            <frame src="Indice.html">
            <frame src="Presentacion.html" name="destino">
        </frameset>
    </frameset>
</head>
</html>
```

Fichero `Indice.html`

```
<html>
<head>
    <base target="destino">
</head>
<body>
    Los contenidos del indice.
</body>
</html>
```

Por defecto, cuando llamamos a una página desde un marco, esta se carga en el mismo marco, pero en el caso del indice interesa que se cargue en otro marco, eso lo hemos conseguido con el atributo `target` de la directiva <BASE> indicándole el nombre del marco en el cual se debe cargar la página. Este atributo también se puede usar con la directiva <A>.

8 Tablas

Para explicar las directivas de tablas fijémosnos en el siguiente ejemplo:

```
<html>
<head>
    <title>Ejemplo de Tabla HTML</title>
</head>
<body>
<center>
<table border=10 cols=4  width=80%  bgcolor=#00ff00>
    <col width=.50*>            <col width=.15*>
    <col width=.15*>            <col width=.20*>
    <caption> Precios de Ordenadores </caption>

    <th rowspan=2> </th>
    <th colspan=2>Valores</th>
    <th rowspan=2>Precio</th>
    <tr>
    <th>CPU</th>    <th>RAM</th>
    <tr>
    <td>Pentium MMX</td>
    <td align=right>200 MHz</td>
    <td align=right>32 MB</td>
    <td align=right>100.000</td>
    <tr>
    <td>Pentium II</td>
    <td align=right>350 MHz</td>
    <td align=right>128 MB</td>
    <td align=right>200.000</td>
    <tr>
</table>
</center>
```

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

Como podemos ver, la tabla se define con la directiva pareada <TABLE> que dispone de los atributos `border` para indicarle que queremos líneas de separación y en su caso le decimos el ancho de dichas líneas; `cols` para indicarle el numero de columnas que va a tener la tabla; `width` para idicarle que porcentaje de la anchura de la ventana queremos que ocupe la tabla o el tamaño en píxeles (si no usamos %); `bgcolor` para indicarle el color de fondo.

Con la directiva <COL> junto con el atributo `width` le indicamos el ancho de cada columna con respecto al ancho total.

La directiva <CAPTION> nos permite darle un título a la tabla.

Con la directiva <TH> definimos celdas que van a actuar como cabeceras de columna, por ello automáticamente el texto saldrá en negrita y centrado.

Con la directiva <TD> definimos celdas normales (no de cabecera). Una fila de celdas se finaliza con la directiva <TR>

Si queremos que una celda ocupe más de una fila haremos uso del atributo `rowspan` y si queremos que ocupe más de una columna haremos uso del atributo `colspan`. La alineación del texto dentro de una celda la conseguimos con el atributo `align`.

9 Gráficos y mapas interactivos

Para incluir gráficos en las páginas web tenemos dos formatos: *GIF* y *JPG*. GIF trabaja con 256 colores, lo cual supone un problema al incluir fotografías, ya que debido al mayor número de colores se debe recurrir al *dithering*[1], con la consiguiente pérdida de calidad en la imagen. Por contra, al usar un algoritmo de compresión simple, la formación de la imagen es rápida, además de permitir fondos transparentes, incluir varias imágenes a modo de animación y añadir sonido.

Por otro lado, el formato JPG trabaja con 24 bits por pixel lo que permite manejar 16.7 millones de colores y obtiene mayores tasas de compresión por lo que tenemos más calidad en la imagen y un menor tamaño en bytes siendo la transmisión más rápida. En contra, al tener un algoritmo de compresión de mayor complejidad, la formación de la imagen es más lenta que en el GIF.

En general, se suele usar con el formato GIF los iconos e imágenes pequeñas, mientras el JPG se usa para las imágenes que desean mostrarse con mayor calidad. Una práctica muy extendida es el uso de *thumbnails* para mostrar fotos con calidad. Consiste en crear una página índice que muestra todas las fotos con un tamaño pequeño y en formato GIF. Y establecer enlaces desde estos *iconos* a las fotos con tamaño grande y formato JPG para mostrarlas con mejor calidad.

Para incluir una imagen en una página usaremos la directiva <IMG> y mediante el atributo `src` le indicamos el fichero que contiene la imagen.

Ejemplo

```
<IMG src="iconos/flecha.gif">
```

Mostramos el icono `flecha.gif` que se encuentra en un directorio de imágenes.

Esta directiva se considera como un carácter más, por lo que se puede poner en una tabla, o dentro de una directiva pareada de enlace.

[1] Se simula un color combinando colores distintos en los píxeles contiguos.

Otra práctica habitual es usar imágenes para mostrar texto, aprovechando así las fuentes de letra más creativos que nos permiten los editores gráficos. Pero no debemos olvidar a aquellas personas que no disponen de un browser gráfico o que por algún error no han podido mostrar las imágenes; en esos casos debemos usar el atributo `alt` para mostrar un texto alternativo.

Ejemplo

```
<IMG src="iconos/indice.gif" alt="Indice">
```

El ejemplo anterior muestra un icono de regreso al *Índice General* pero si alguna causa impide mostrar la imagen, mostrará el texto alternativo *Indice*.

Como las imágenes se pueden poner junto con el texto, existe un atributo que permite alinear el texto con la imagen. El atributo es `align` que acepta los valores `TOP, MIDDLE, BOTTOM, LEFT` y `RIGHT`. Y con los atributos `hspace` y `vspace` indicamos la separación entre el texto y las imágenes.

También disponemos de atributos para modificar el tamaño de la imagen, los atributos son `width` y `height`. Y un atributo para poner un marco a la imagen: `border`.

Otro atributo muy útil es `lowsrc` a la que se le da la misma imagen a baja resolución. De modo que primero se carga esta imagen y a continuación la buena.

Por último, veremos un uso cada vez más extendido de los ficheros gráficos en las páginas web: los mapas interactivos.

Consisten en una imagen que tiene enlaces a distintas páginas web en función de la zona de la imagen en donde se pinche. El ejemplo típico es el que muestra un mapa de un país y podemos viajar hacia las distintas ciudades simplemente pichando sobre ellas.

Para ello habrá que definir para cada área activa las coordenadas del área y la referencia del documento a cargar. Esto se hace con la directiva pareada <MAP> la cual nombramos con el atributo `name`. Entre la pareja de directivas definiremos las áreas con la directiva <AREA> que tiene los atributos `shape` que por ahora solo admite el valor `rect` que indica que el área es rectangular; con el atributo `coords` le indicamos las coordenadas del área en píxeles con el formato `izquierda, arriba, derecha, abajo`; y con el atributo `href` le indicamos la referencia del documento asociado al área.

Una vez definidas las areas activas, debemos indicar a la directiva de imágenes <IMG>, que la imagen va a ser un mapa interactivo y cuales son sus áreas activas. Esto lo hacemos con el atributo `usemap` al que le asignamos el nombre del fichero que contiene la definición de las áreas (que puede ser el fichero actual) y el nombre de la zona de definiciones de áreas (el dado a la directiva <MAP>).

Ejemplo

```
<img src="mapa.gif" usemap="prueba.html#mapa">
<map name="mapa">
<area shape="rect" coords="170,29,329,133" href="html/xv.html">
<area shape="rect" coords="152,140,334,182" href="html/jb.html">
</map>
```

10 Formularios

Hasta ahora hemos aprendido a realizar páginas HTML que funcionan según el modelo siguiente:

1. el autor crea documentos y los pone a disposición en un servidor,

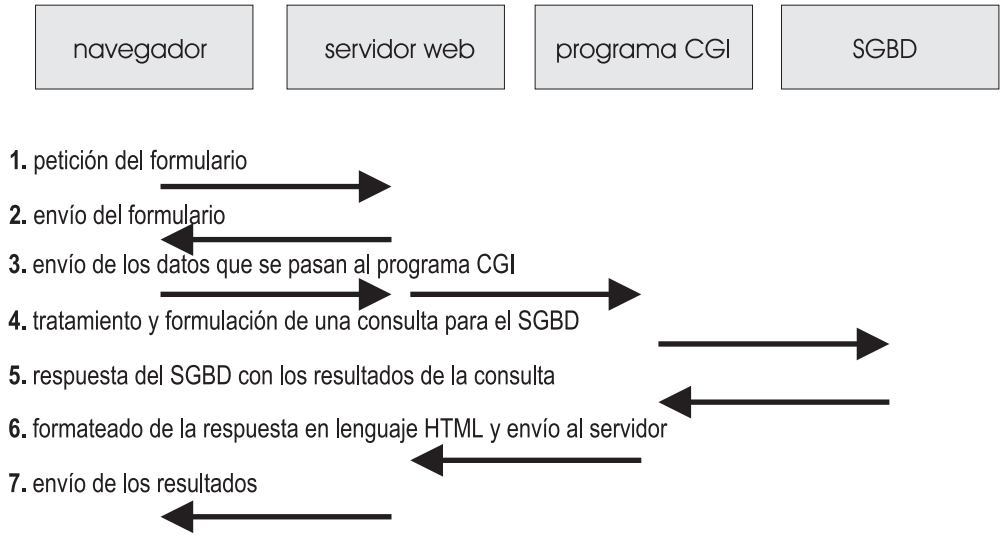
2. el lector utiliza un navegador para consultar estas páginas.

Nos encontramos ante un modo de consulta en el que simplemente se puede elegir el camino por entre las páginas, pero en ningún momento se permite interactividad real entre el usuario y el sistema que proporciona los documentos.

Supongamos, por ejemplo, que quisiésemos realizar una página para ofrecer un servicio de búsqueda en una base de datos. El usuario debería ser capaz de escribir en la página la palabra a buscar, y tras realizar una petición de búsqueda, recibiría en otra página la información buscada.

Los formularios son la respuesta aportada por HTML para generar una pantalla de cuadros de diálogo con el lector. Como en un formulario de papel, se podrá incluir texto, casillas de verificación, listas de selección, etc.

Los formularios no serán otra cosa que ficheros que se rellenan; siempre se necesitará un elemento apropiado para tratar la información que se introduzca. Este elemento será casi siempre un programa que se ejecutará en el servidor: el CGI o *Common Gateway Interface*. El programa CGI permite establecer el diálogo entre el navegador que envía el formulario y el programa ubicado en el servidor (una base de datos, por ejemplo).



Los formularios en HTML se definen con la directiva pareada <FORM>. Esta directiva define una zona del documento HTML en la que van a estar incluidas la zona de entrada de datos y los botones de envíos de datos. Dentro de un mismo documento HTML puede haber varios formularios; en este caso, cada uno de ellos podrá ser enviado de forma totalmente independiente.

Los formularios HTML permiten la introducción de datos de acuerdo con los siguientes tipos de elementos:

- Campos de texto.
- Cajas de validación.
- Listas desplegables.
- Áreas de texto.

Lo más habitual es que el visitante deba introducir algún tipo de texto. Para crear un *campo de texto* utilizaremos la directiva <INPUT>:

```
<INPUT TYPE="TEXT" NAME="nombre" SIZE="anchura" VALUE="valor">
```

Mediante el modificador VALUE es posible indicar un texto que se muestra sin que el usuario lo introduzca. Cuando el visitante envíe el formulario, el servidor recibirá una variable con el nombre indicado en el atributo NAME en la que irá el valor introducido por el usuario.

Un campo de tipo *contraseña* es igual a un campo de texto excepto que el valor introducido no se muestra en pantalla. Podemos crear un *password* o contraseña mediante el código:

```
<INPUT TYPE="PASSWORD" NAME="nombre" SIZE="anchura">
```

Hay dos tipos de *cajas de validación*: aquellas en las que solamente es posible seleccionar una de las opciones (*radio buttons*), y aquellas en las que es posible la selección de múltiples opciones (*check box*). Los dos tipos de cajas de validación se crean en HTML de forma muy similar mediante la directiva <INPUT>:

```
<INPUT TYPE="tipo" NAME="nombre" VALUE="valor">Texto de la opcion
```

La variable `tipo` puede tomar los valores `checkbox` y `radio`, dependiendo del tipo de caja que queramos presentar.

El siguiente ejemplo muestra el código HTML para construir una caja de validación tipo `checkbox`:

```
<html>
 <head>
     <title> Ejemplo de caja de validacion tipo checkbox</title>
 </head>
 <body>
     <center>
     <form>
         <font size=+1 color="red">Elige uno o mas sistemas operativos</font><br>
         <input type="checkbox" name="v1" value="W98"> Microsoft Windows 98 <br>
         <input type="checkbox" name="v2" value="WNT"> Microsoft Windows NT <br>
         <input type="checkbox" name="v3" value="Linux"> Linux <br>
         <input type="checkbox" name="v4" value="Sol"> Sun Solaris <br>
         <input type="checkbox" name="v5" value="Mac"> Apple MacOS <br>
     </form>
     </center>
 </body>
 </html>
```

Las cajas de validación tipo `radio button` se construyen de la misma manera, excepto que ahora todas las opciones tienen el mismo nombre (si no, no sería posible hacer que sólo una de ellas estuviese activa):

```
<html>
 <head>
     <title> Ejemplo de caja de validacion tipo radio button</title>
 </head>
 <body>
```

```
    <center>
    <form>
        <font size=+1 color="red">Elige un sistema operativo</font><br>
        <input type="radio" name="var" value="W98"> Microsoft Windows 98 <br>
        <input type="radio" name="var" value="WNT"> Microsoft Windows NT <br>
        <input type="radio" name="var" value="Linux"> Linux <br>
        <input type="radio" name="var" value="Sol"> Sun Solaris <br>
        <input type="radio" name="var" value="Mac"> Apple MacOS <br>
    </form>
    </center>
</body>
</html>
```

Las *listas desplegables* se implementan con dos elementos: la lista propiamente dicha, y las opciones de la lista:

```
<html>
<head>
    <title> Ejemplo de lista desplegable</title>
</head>
<body>
    <center>
    <form>
        <font size=+1 color="red">Elige tu sistema operativo</font><br>
        <select name = "opcion">
        <option value="W98"> Microsoft Windows 98
        <option value="WNT"> Microsoft Windows NT
        <option value="Linux"> Linux
        <option value="Sol"> Sun Solaris
        <option value="Mac"> Apple MacOS
        </select>
    </form>
    </center>
</body>
</html>
```

Cuando el visitante envíe el formulario, el servidor recibirá una variable `opcion` en la que irá el valor de alguna de las opciones especificadas con la etiqueta <OPTION>

Para especificar un *área de texto* dentro de un formulario utilizaremos la directiva <TEXTAREA>:

```
<TEXTAREA NAME="nombre" COLS="columnas" ROWS="filas">
Texto opcional que se muestra por defecto
</TEXTAREA>
```

Una vez que el visitante ha rellenado el formulario, debe enviarlo. Para ello, HTML permite definir un *botón de envío* del formulario:

```
<INPUT TYPE="SUBMIT" VALUE="nombre">
```

Junto con el botón de envío, existe otro especialmente útil para el visitante. El *botón de reset* limpia todos los campos del formulario y les devuelve su valor original:

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

11 CGI

La idea de la programación CGI es construir el documento HTML en el momento en que se solicita; el documento se envía al cliente a medida que se construye sin necesidad de almacenarse en un archivo.

El cliente indica el nombre de un archivo mediante un URL, no para recibir su contenido sino para solicitar su ejecución en el servidor. Este ltimo ejecuta el programa indicado y devuelve al cliente la salida estándar de dicho programa (es decir, lo que se hubiera obtenido en la pantalla al ejecutar el programa manualmente desde la línea de órdenes).

El ejemplo siguiente muestra un programa CGI que genera una página que contiene la fecha actual:

```
#!/bin/ksh
# Creacion de la cabecera
echo 'Content-type: text/html'
echo ' '
# Creacion del cuerpo del documento
echo '<HTML>'
echo '<HEADER>'
echo '<TITLE>Esta es la fecha de hoy</TITLE>'
echo '</HEADER>'
echo '<BODY>'
echo "La hora del sistema es `date`"
echo '</BODY> '
echo '</HTML>'
```

Si se llama a este programa `fecha` y se ubica en el directorio `/cgi-bin` del servidor se podrá, a partir de un cliente, llamar al enlace `http://url-servidor/cgi-bin/fecha`. El servidor sabe que los ficheros del directorio `cgi-bin` son programas ejecutables, por lo que ejecutará dicho programa y devolverá al cliente la salida generada por éste (es decir, una página HTML que muestra la fecha actual).

Todo lenguaje capaz de producir una salida estándar puede utilizarse para escribir programas CGI, siendo los más utilizados C, Perl, Java, Visual Basic, ...

Para que la salida producida por un programa CGI pueda ser interpretada por el servidor y el cliente, debe tener un formato especial. Éste se compone de dos partes separadas por una línea en blanco.

La primera parte, la cabecera, incluye informaciones que utilizará el servidor para construir la cabecera HTTP de su respuesta al cliente. La segunda, el cuerpo, se compone de los datos que constituyen el documento que se enviará al cliente. En el ejemplo anterior la cabecera contenía en la línea `Content-type` el tipo de documento generado por el programa CGI (en este caso `text/html`). Los siguientes son algunos de valores posibles para `Content-type`:

- `text/html`: el tipo más utilizado, indica que los datos deben interpretarse como código HTML
- `text/plain`: indica que los datos son texto plano que no debe interpretarse
- `image/gif`, `image/jpeg`, ...: los datos deben interpretarse como imágenes gráficas

- `application/postcript`: los datos están en formato postscript

Cuando el cliente rellena y pulsa el botón de envío de un formulario HTML, el navegador enviará al servidor los datos del formulario junto con la petición de ejecutar un programa CGI. La llamada al programa CGI se define mediante el atributo ACTION de la directiva <FORM>. El atributo ACTION especifica la acción a realizar por parte del servidor web cuando reciba los datos del cliente:

```
<FORM ACTION="http://servidor/path/programa-cgi">
```

Cuando el usuario haga clic sobre el botón de envío del formulario, conectará con el `servidor` y ejecutará el `programa-cgi` ubicado en el directorio especificado.

Las *variables de entorno* son variables enviadas por el cliente al servidor y que contienen información acerca del cliente, del servidor, de los datos que se están enviando e incluso del propio programa CGI que va a ejecutarse. Algunas de las variables de entorno más comunes son las siguientes:

- REQUEST_METHOD: método con el que se envía el formulario
- CONTENT_LENGTH: número de bytes de los datos de entrada (método POST)
- QUERY_STRING: datos del formulario (método GET)
- SERVER_PORT: puerto del servidor a través del cual se envían los datos
- SERVER_PROTOCOL: nombre y versión del protocolo utilizado en la comunicación
- GATEWAY_INTERFACE: versión de la especificación CGI utilizada en la comunicación
- HTTP_USER_AGENT: nombre y versión del navegador del cliente

Los datos del formulario son enviados al servidor en forma de una cadena de caracteres que contiene un conjunto de pares *(nombre de campo, valor)*. Cada par va separado por un carácter `&`. La separación entre el nombre del campo y su valor se hace mediante un carácter `=`. Los espacios se reemplazan por `+` y los caracteres especiales por su valor hexadecimal bajo el formato `%xx`:

```
APELLIDO=de+la+Cierva&NOMBRE=Juan
```

Los datos pueden enviarse utilizando dos métodos diferentes:

- GET
- POST

El atributo METHOD de la directiva <FORM> se utiliza para especificar el método que se empleará en el envío del formulario:

```
<FORM METHOD="GET" ACTION="http://servidor/encuesta.exe">
        . . . (elementos del formulario)
</FORM>
```

Cuando se utiliza el *método GET*, la cadena de caracteres que contiene el conjunto de pares se añade al URL que referencia al programa CGI a ejecutar. La separación entre el nombre del script y la cadena se realiza mediante el carácter `?`:

```
http://servidor/cgi-bin/buscar.exe?APELLIDO=de+la+cierva&NOMBRE=Juan
```

En la ejecución del programa CGI, la cadena con los datos del formulario se encontrará en la variable de entorno QUERY_STRING. Para capturarlos, el programa CGI simplemente tendrá que acceder al valor de dicha variable.

Cuando se utiliza el *método POST*, la cadena con los datos del formulario no se añade al URL, sino que se envía al servidor mediante una secuencia HTTP especial. El programa CGI podrá leer la cadena directamente desde la entrada estándar. Para capturar los datos, el programa utiliza el valor de la variable CONTENT_LENGTH para saber la longitud de la cadena recibida por la entrada estándar.

Con cualquiera de los dos métodos, nos enfrentamos con el problema de decodificar las variables y almacenarlas convenientemente. La forma de hacerlo dependerá del lenguaje concreto que utilicemos en el programa CGI.

12 Plantillas

Una alternativa al uso de programas CGI es la utilización de algún tipo de extensión de HTML para producir documentos de HTML dinámico. Las plantillas son una generalización de los documentos HTML. Además de código HTML estático, una plantilla contiene otras instrucciones especiales empotradas. Estas instrucciones son interpretadas cuando el documento es solicitado por el cliente, de manera que éste no ve dichas instrucciones empotradas, sino el resultado de ejecutarlas.

Las plantillas permiten incrustar consultas SQL dentro del código HTML. Cuando el servidor recibe una solicitud de una plantilla HTML por parte del cliente, interpreta las sentencias SQL y las ejecuta contra una base de datos. El resultado de la consulta es capturado por el servidor, y combinado con la plantilla HTML para producir un documento HTML con los resultados de la consulta debidamente formateados.

El servidor que utilizaremos en esta práctica permite utilizar plantillas. Aunque las plantillas no forman parte del lenguaje HTML estándar, los servidores web suelen incorporar alguna facilidad de este tipo. En nuestro caso, el acceso a la base de datos se realiza mediante ODBC (*Open Database Conectivity*).

El siguiente código muestra un formulario HTML para rellenar el campo `apellido1` y solicitar el documento `pagina2.stm`:

```
<HTML>
<BODY>
<FORM method=post action=pagina2.stm>
<INPUT type=text name="apellido1">
<INPUT type=submit>
</FORM>
</BODY>
</HTML>
```

El archivo `pagina2.stm` contiene una directiva especial <RCQ> con una sentencia SQL. El servidor captura la entrada desde el formulario y almacena las variables pasadas como argumentos (`apellido1` en este caso). Dentro de la sentencia SQL, `RC$apellido1` se refiere al argumento `apellido1` enviado desde el formulario. El servidor utiliza la conexión `conexionbd` para realizar la consulta SQL incluida en la plantilla:

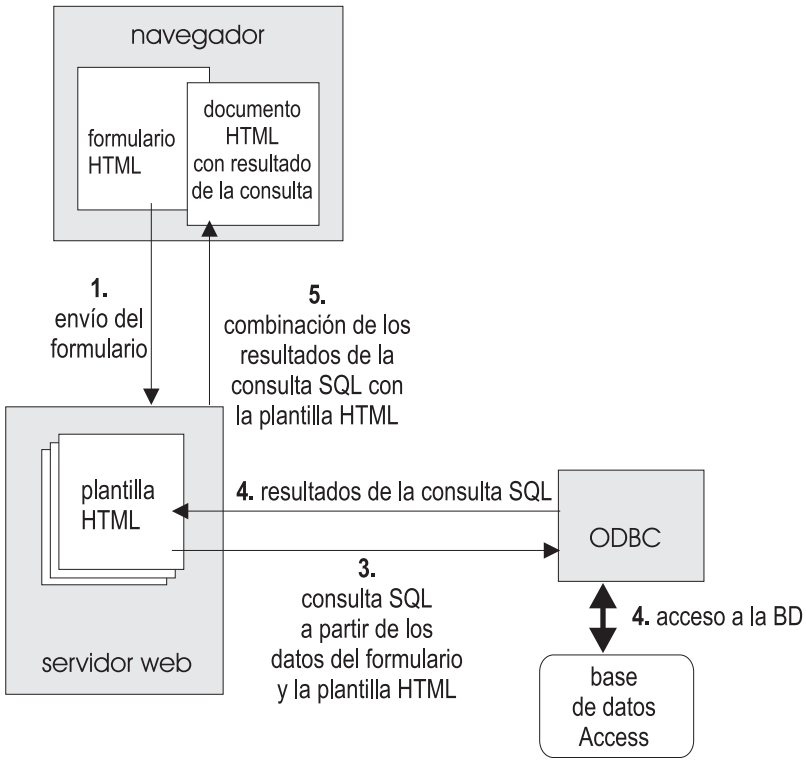
```
<!-- archivo pagina2.stm -->
```

```
<HTML>
<BODY>
<TABLE border=1>
<RCQconexionbd sql="select codigo, nombre, apellido1
                    from afiliados
                    where apellido1 ='RC$apellido1'"
  format="<TR><TD>%s<TD>%s</TD><TD>%s</TD><TR>" >
</TABLE>
</BODY>
</HTML>
```

Los resultados de la consulta son combinados con la plantilla HTML de acuerdo con el formato especificado en el atributo `format` para así producir un documento similar al siguiente, que es enviado al cliente:

```
<!-- archivo pagina2.stm -->
<HTML>
<BODY>
<TABLE border=1>
        <TR><TD>91121213<TD>Juanito</TD><TD>Valderrama</TD><TR>
</TABLE>
</BODY>
</HTML>
```

La figura siguiente muestra el esquema general para la generación de un documento HTML que muestre los resultados de una consulta SQL:



En esta práctica utilizaremos el servidor *Sambar*, disponible en las direcciones *http://tucows.uam.es* y *http://www.sambar.com*. De forma provisional, estará también disponible en el directorio *ofi* de la cuenta del departamento *lsi* de *turing*.

Para poder acceder a una base de datos Access mediante páginas de HTML dinámico, debemos seguir los siguientes pasos:

1. Creamos un origen de datos ODBC para permitir al servidor acceder a nuestra base de datos Access. Para ello, en el panel de control de Windows, haremos doble clic sobre el icono ODBC.
   (a) Agregamos un DSN (*data source name* de sistema, para lo cual será necesario indicar qué controlador utilizaremos: *Microsoft Access Driver*.
   (b) Indicamos el nombre del origen de datos (podemos dar cualquier cualquier nombre, por ejemplo, el de la base de datos).
   (c) Indicamos el nombre y la ubicación de nuestra base de datos, para lo cual pulsaremos el botón *Seleccionar...*)
2. Para probar si la fuente de datos ha sido bien definida utilizamos el programa *iodbc*, en el directorio *bin* del servidor web (para más detalles, ver *http://eiXXXXXX/syshelp/iodbc.htm*)
3. Instalamos el servidor web en nuestro sistema y lo lanzamos. La página principal del servidor es *http://eiXXXXXX*.
4. Habilitamos el servidor web para el acceso a la base de datos (para más detalles sobre este paso, ver la página *http://eiXXXXXX/syshelp/db.htm* en la documentación del servidor):
   (a) Accedemos a la administración del servidor. Para ello abrimos en el navegador la página principal (*http://eiXXXXXX*) y pulsamos en el enlace *System Administrator*. El nombre de usuario es *admin* y la contraseña es la cadena vacía. Tras abrir el enlace *Server Configuration* activamos la casilla *Enable DBMS Usage*.
   (b) Creamos una entrada en el archivo *config/dbconfig.ini* de acuerdo con la fuente de datos ODBC que hemos creado en el primer paso.
   (c) Relanzamos el servidor web para hacer efectivos los cambios.

13 Ejercicio.

Se propone la creación de una página web que permita realizar las siguientes consultas:

- Empresas de una determinada actividad.
- Afiliados de una determinada empresa.
- Recibos pendientes de un afililado.

Para ello, en una página inicial, el usuario seleccionará qué consulta desea realizar. Como resultado de la elección, su navegador mostrará una página con un formulario para especificar la condición de búsqueda para su consulta.

Los formularios debeán tener una lista desplegable que muestre los valores del campo de búsqueda para la consulta. Para ello, será necesario que el formulario sea generado automáticamente mediante una plantilla que contenga una consulta sobre los valores del campo de búsqueda.

Una vez rellenado el formulario, tras pulsar el botón de envío, el servidor generará dinámicamente y devolverá al cliente un documento HTML con los resultados de la consulta.