

Primer contacto con HTML

La Red es el máximo exponente de la sociedad de la información en la que nos ha tocado vivir. Nacida en Estados Unidos hace varias décadas como una forma de conectar entre sí un conjunto distribuido de ordenadores con objetivos científicos y militares, con el paso de los años ha venido evolucionando hasta convertirse en lo que actualmente conocemos, es decir, varios millones de ordenadores interconectados entre sí y facilitando el intercambio de ingentes cantidades de información.

El crecimiento que se ha experimentado a lo largo de la corta historia de Internet, tanto en número de máquinas y usuarios conectados como en cantidad de datos circulando a lo largo y ancho de todo el planeta, se puede calificar de exponencial. En este sentido lejos quedan ya los tiempos en los que conectarse a la Red estaba al alcance tan sólo de unos pocos iniciados.

De este modo, hoy en día la mayoría de los usuarios con ordenador y módem son capaces de conectarse, con mayor o menor provecho, a los múltiples servicios que Internet nos ofrece, como correo electrónico, acceso remoto, transferencia de ficheros entre máquinas remotas, etc., e incluso a servicios de compra de determinados artículos (comercio electrónico).

De todos estos servicios, la Web es sin lugar a dudas el más popular y, ha sido uno de los principales desencadenantes del éxito actual que vive la Red. Precisamente, el sobrenombre popular con el que también se conoce a Internet surge del término Web, como telaraña o red capaz de cubrir el mundo por completo. Esto ha hecho posible que Internet se haya convertido por méritos propios en un mercado sin fronteras, sumamente atractivo y en el que cada día que pasa se nos ofrece un servicio añadido. Ya nadie se extraña si decimos que hemos reservado nuestros billetes o nuestras habitaciones de hotel para las próximas vacaciones, si participamos en subastas, si hacemos la compra del supermercado, si consultamos nuestra cuenta bancaria o si decidimos invertir en Bolsa sentados frente al teclado de nuestro ordenador. En esencia, la Web no es otra cosa que un gigantesco conjunto formado por millones y millones de ficheros que contienen información de todo tipo (textual, gráfica, sonora, etc.), distribuida entre millones de ordenadores y escrita en un determinado formato, que puede ser interpretado de forma adecuada por una categoría especial de programas que denominamos navegadores o *browsers* (como por ejemplo los populares Microsoft Internet Explorer y Netscape Navigator, entre otros).

Por esta razón, el objetivo principal que pretendemos satisfacer con este curso consiste en acercar los mecanismos que hacen posible la Web, y ofrecer los conocimientos necesarios para que seáis capaces de diseñar vuestras propias páginas.

Navegadores y servidores

El servidor web no es otra cosa que un programa que se ejecuta en una máquina física (es decir, un ordenador que, por extensión, también recibe el nombre de servidor Web) a la que se ha asignado una dirección única o **URL** (Uniform Resource Locator, o localizador uniforme de recursos), que la identifica de forma unívoca.

El servidor web se encarga de enviar el contenido de las páginas web que tiene almacenadas cuando el software recibe una solicitud desde otra máquina, por medio de otro programa que se denomina cliente web (aunque a éste se le conoce generalmente como navegador o *browser*), y que se ejecuta en la máquina del usuario que quiere acceder al contenido de dichas páginas.

En dicho envío, el servidor no sólo manda la página HTML, sino también todos aquellos elementos externos a la misma pero que forman parte de ésta (sonidos, imágenes o iconos, por ejemplo), de modo que en la pantalla del usuario que ha realizado la petición se visualice el resultado original, tal y como fue diseñado por su creador.

El lenguaje por medio del cual se efectúa la comunicación entre los clientes y el servidor se denomina **HTTP** (HyperText Transmission Protocol, o protocolo de transmisión de hipertexto). Dicho lenguaje ofrece no sólo los mecanismos necesarios para enviar la información entre máquinas, sino también servicios de encriptación y autenticación del cliente. Sin embargo, dado que ése no es nuestro objetivo, no añadiremos nada más acerca del HTTP.

Un mundo de etiquetas

Las páginas web no son otra cosa que ficheros de texto ASCII (o planos) escritos en formato **HTML** (**HyperText Markup Language**, o lenguaje etiquetado de hipertexto), a las que se puede acceder por medio de un esquema cliente servidor. Este formato especial se basa en intercalar, a lo largo del texto que compone la página, una serie de pares de marcas o etiquetas (**tags**) que indican características especiales de dicho texto. Un tag no es otra cosa que un conjunto de caracteres concreto, que posee un significado específico y que se interpreta de forma correcta por el software cliente o navegador. Este conjunto de

caracteres se encuentra encerrado entre los caracteres < y > si es de apertura, o entre los caracteres </ y > si es de cierre. Los navegadores o *browsers* son capaces de interpretar de forma adecuada estas etiquetas que, salvo excepciones, van en pares apertura/cierre, actuando así sobre el formato de presentación visual de la página.

De hecho, si cargamos una página web con nuestro navegador habitual y pulsamos el botón derecho del ratón sobre la misma, se desplegará un menú en el que podremos seleccionar la opción correspondiente a *Ver fuente*. De este modo, se abrirá una ventana en la que podremos visualizar el texto ASCII de la mencionada página, y si nos fijamos con atención veremos las etiquetas mencionadas.

Aplicaciones para seguir el curso

El elemento básico y fundamental en nuestra particular «caja de herramientas» lo constituyen los denominados editores HTML. Aunque éste es un lenguaje de texto plano (o texto ASCII), es decir, que se puede generar con cualquier editor ASCII (como el clásico **edit** de MSDOS, o el bloc de notas de Windows), lo habitual consiste en trabajar con alguna herramienta que evite la necesidad de tener que introducir manualmente las mencionadas etiquetas, como pueden ser HoTMetaL Pro, Microsoft FrontPage (en versión normal o Express), HotDog Pro, HTML Assistant o el propio Microsoft Word, por citar tan sólo algunos ejemplos.

Como segundo elemento a tener en cuenta, mencionaremos los programas de retoque o diseño gráfico, con los que generar imágenes, fondos, etc., mediante los que podremos dotar nuestras propias páginas de contenidos gráficos. En este caso, la propia Red se encuentra repleta de ellos, con ejemplos tan evidentes como el popular PaintShop Pro, capaz de aceptar incluso plugins de Adobe Photoshop y cuya última versión no tiene nada que envidiar a programas teóricamente más potentes pero mucho más caros; o Lview Pro, una excelente aplicación shareware idónea para la creación de fondos en formato GIF.

Además de estos dos tipos básicos de aplicaciones, existen elementos adicionales (y opcionales), como pueden ser manuales de referencia del lenguaje (en formato electrónico o en papel), iconos, fondos, GIFs animados y fuentes o texturas, así como otros documentos relacionados. Aunque la Red se encuentra repleta de este tipo de elementos, personalmente recomendamos direcciones tan atractivas como:

- The Free Graphics Store (ausmall.com.au/freegraf),
- Iconographics Design Free Graphics (www.iconographics.com/clip_f.htm),
- Texture Land (www.meat.com/textures),
- Fonts & Things (www.fontsnthings.com) o
- Jelane's Free Web Graphics (www.erinet.com/jelane/families).

Finalmente, en nuestro curso tampoco dejaremos de lado otras aplicaciones que han cobrado especial auge en los últimos tiempos, como el popular *Dreamweaver*, explicando así mismo cómo aprovechar algunas de las características de este excelente programa. También mencionaremos, aunque sólo sea brevemente, cómo hacer uso de alguna herramienta de FTP, mediante la cual podremos proceder a la publicación de las páginas que seamos capaces de desarrollar.

Comienza el viaje

La primera decisión que se debe tomar a la hora de diseñar nuestras páginas consiste en ver cómo se va a estructurar la información que se quiere presentar y qué información queremos mostrar. La mejor forma de comenzar nos obliga a coger lápiz y papel y definir a continuación un esquema que nos sirva para plantear un diseño bien estructurado (ojo, que nadie se asuste ante este requisito: no se trata de pensar como analistas o programadores, sino simplemente de actuar con cierto sentido común para no perder el tiempo y obtener un resultado satisfactorio).

La idea primordial debe ser la sencillez y la simplicidad de la información, tratando de organizar contenidos distintos en páginas diferentes: por ejemplo, podemos reservar una página para incluir nuestras fotos familiares, otra para contar nuestras aficiones, otra para recomendar enlaces de interés, etc. Ante todo hay que recordar que cuantos más elementos contenga una página, será tanto más difícil de mantener e incluso de leer por quienes accedan a ella (aquí haremos bueno el dicho de que lo bueno, si breve, dos veces bueno).

Los diez mandamientos del diseño web

- I. Decidir qué se quiere hacer y cómo se quiere mostrar.
- II. Definir una imagen global de las páginas, con algún elemento identificativo y un diseño similar para todas ellas.
- III. Realizar un diseño limpio, claro y sencillo (no perder de vista que lo bueno, si breve, dos veces bueno). Se han realizado estudios que confirman que una página que ocupa más del 50% de la pantalla con texto no suele captar la atención del lector potencial.
- IV. Utilizar distintos niveles de texto, jugando con los elementos de diseño para no ofrecer una imagen monótona.
- V. Tratar de conseguir contenidos visuales, sin olvidar el viejo dicho de que una imagen vale más que mil palabras. Por regla general, se deben utilizar una o dos imágenes por página.
- VI. Plantear una información muy enfocada a los objetivos. Hay que tratar de ser sumamente conciso, ofreciendo enlaces a páginas con información adicional si es necesario.
- VII. Definir en toda página enlaces a la página de presentación e incluso a otras páginas: la movilidad a través de nuestra web es un factor sumamente importante.
- VIII. Mantener un estilo homogéneo y consistente a lo largo del conjunto de páginas.
- IX. Comprobar todos los enlaces y asegurarse de que funcionen y están actualizados. La impresión que causa pulsar sobre un enlace y obtener a cambio un error porque éste no estaba bien definido, es simplemente penosa.
- X. Pedir opinión a otra (u otras) personas, con el fin de recibir comentarios adicionales al respecto, antes de proceder a publicar las páginas.

Una vez diseñado el esquema, conviene plasmarlo en un dibujo que represente el mapa de nuestro web, y en el que podamos identificar de forma clara cómo vamos a permitir que se muevan los visitantes que accedan a nuestras páginas, es decir, la existencia de enlaces dentro del conjunto de páginas. Dicho dibujo debe hacerse teniendo en cuenta que siempre existe una página inicial, que denominaremos página de presentación o **home page**, cuyo nombre físico debería ser *index.html*, *index.htm* o *default.htm*, que constituye la puerta de acceso a nuestra web y a la que resulta conveniente crear un enlace desde cualquier otra de nuestras páginas. Con los dos elementos anteriores completamente diseñados (el esquema y el mapa) es el momento de poner manos a la obra y comenzar a preparar nuestros contenidos, teniendo en cuenta las normas básicas de diseño de páginas web. Llega, por tanto, el momento de comenzar a crear; sin embargo, antes de entrar en materia comenzaremos por ver el esqueleto que da forma a toda página web, es decir, los elementos básicos del lenguaje HTML.

Elementos básicos de toda página

Aun a riesgo de parecer repetitivos, debemos recalcar una vez más que HTML es un lenguaje de etiquetas. El documento web es simplemente un documento de texto, que toma su formato a partir del etiquetado. En general, dicho etiquetado sigue un convenio similar al de los paréntesis que se utilizan al escribir una fórmula matemática: existen tantos de apertura como de cierre, y las aperturas más internas son las que primero se deben cerrar. Las etiquetas, al igual que los paréntesis, afectan única y exclusivamente a los caracteres delimitados por cada pareja principio/ fin. El conjunto formado por las etiquetas de principio y final, más los caracteres delimitados por ambas, recibe el nombre de contenedor. Ahora bien, ni todos los elementos necesitan una etiqueta de final, ni todos los contenedores deben contener texto en su interior, valga la redundancia.

Aunque no existe una norma al respecto, las etiquetas pueden escribirse indistintamente en mayúsculas o minúsculas, pero una buena norma de estilo (y que simplifica notablemente el tiempo dedicado a mantenimiento y revisión del código de las páginas) consiste en escribirlas todas en mayúsculas.

En toda página HTML debe aparecer un mínimo de cuatro etiquetas, con independencia de su contenido. La primera de ellas es la etiqueta o **tag** `<HTML>`, que informa al navegador del tipo de documento que va a tratar. Análogamente, su complementaria `</HTML>` indica al navegador que el documento ha terminado, por lo que debe ser la última del documento. Justo a continuación de la etiqueta `<HTML>` debe colocarse la etiqueta `<HEAD>`, la cual sirve como marca para definir la cabecera del documento, dentro de la cual podremos añadir información complementaria acerca de la página. En el interior de esta cabecera debemos incorporar, obligatoriamente, el tag `<TITLE>`, que sirve para añadir el título de la propia página, el cual no aparecerá como texto del documento, sino en la barra del título de la ventana.

Dicho título es el que usan los motores de búsqueda para incorporar nuestra página a sus índices de resultados. Sin embargo, en este punto conviene hacer una llamada de atención, dado que algunos motores de búsqueda muestran los resultados en orden alfabético: resulta muy conveniente, por tanto, dedicar cierto tiempo a definir el título de nuestra página de presentación, ya que será la llave para acceder a las restantes páginas y conviene que se sitúe en las primeras posiciones de los resultados

ofrecidos por un buscador. Finalmente, una vez concluida la cabecera de nuestro documento, con la etiqueta </HEAD>, llega el momento de incorporar la última etiqueta obligatoria de la página. Esta es <BODY>, y su misión consiste en delimitar el cuerpo o parte central del documento.

El cuerpo de la página

Aunque hemos indicado las etiquetas obligatorias en toda página, lo cierto es que con éstas aún no podemos visualizar nada. En realidad, los contenidos que se visualizarán en nuestra página HTML deben insertarse entre los tags <BODY> y </BODY> indicados en el punto anterior. Para ello, no debemos olvidar que buena parte del contenido de cada página será texto, por lo que para añadirlo necesitamos incorporar un nuevo tag, que denominaremos de párrafo y cuyo aspecto es <P>.

A diferencia de lo que sucede en los procesadores de texto habituales, la forma en que se dividen las líneas de texto en el documento HTML no tiene efecto sobre su presentación real en pantalla. Así, aunque en el código fuente de una página web tecleemos varios retornos de línea y múltiples espacios, todos ellos se ignorarán cuando la página se visualice en un navegador. En este sentido, cada vez que queramos cambiar de párrafo (o, mejor dicho, cada vez que queramos iniciar un párrafo nuevo) deberemos añadir un tag. Sin embargo, esta etiqueta introduce un cierto espacio entre párrafo y párrafo, espacio que puede no ser deseable si estamos tecleando líneas sueltas (por ejemplo, si estamos mostrando una lista de hiperenlaces) o si lo que queremos es partir una línea en dos trozos que se escriban sin separación.

La solución a este aparente problema viene dada por el tag
, que permite cambiar de línea sin dejar espacio con respecto a la anterior. Esta etiqueta es una de las excepciones que mencionábamos con anterioridad, y carece de su equivalente de fin.

Nuestro primer ejemplo

Para concluir esta primera entrega del curso, a continuación mostramos un ejemplo en el que se muestran con mayor claridad las diferencias entre los tags presentados en este apartado, y en el que también se ve un ejemplo de otro elemento íntimamente relacionado con éstos, pero al que apenas hemos prestado atención: se trata de la justificación o alineado, referido única y exclusivamente al párrafo en el que se ha introducido la correspondiente modificación. El código fuente es el siguiente:

```
<HTML>
  <HEAD>
    <TITLE>Ejemplo 1 de pagina Web</TITLE>
  </HEAD>

  <BODY>
    <P>¡Bienvenidos a nuestra primera pagina!</P>

    <P>Esto es un ejemplo de utilizacion</P>
    <P>del tag de parrafo</P>

    <P>Y esto un ejemplo de utilizacion
    <BR>del tag de retorno de linea</P>

    <P ALIGN=right>Para concluir la pagina,
    <BR>se muestra un ejemplo de alineacion derecha</P>

    <P ALIGN=justify>Seguido de otro mas de alineacion
    justificada. En este caso se requiere la presencia
    de mas texto para apreciar el efecto.</P>

    <P ALIGN=center>Y otro de alineacion centrada.</P>

    <P>Notese que si no se indica nada, la alineacion por
    defecto es a la izquierda</P>

  </BODY>
</HTML>
```

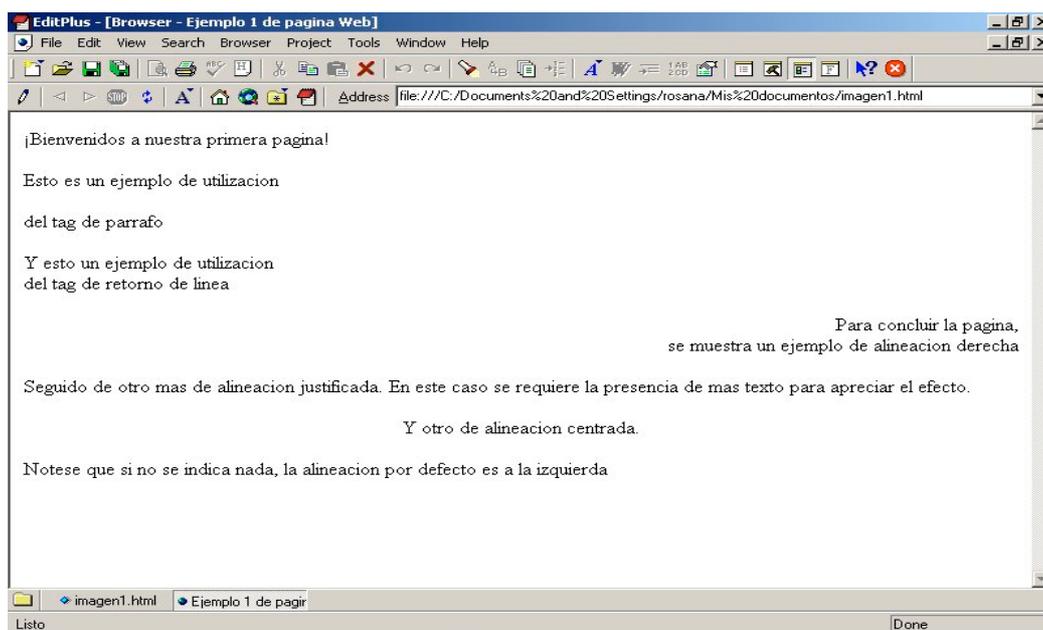


Imagen 1 Resultado de nuestro primer ejemplo.

En resumen

Hasta ahora, hemos sentado las bases iniciales para entender el funcionamiento de la World Wide Web, así como para comenzar a diseñar nuestras propias páginas web, detallando los materiales necesarios y mostrando algunos de los tags propios de HTML.

A partir de ahora seguiremos avanzando en el conocimiento del lenguaje HTML, y veremos cómo hacer uso de estas características desde algunas de las aplicaciones que hemos indicado. En esencia, lo que pretendemos conseguir es que este curso no se convierta en una mera descripción de tags, sino que sea algo realmente práctico y útil.

No nos centraremos por ello en el lenguaje HTML, sino que seguiremos un paso más allá, aprendiendo también los fundamentos básicos de CGI y JavaScript, con el fin de dotar a nuestras páginas de características más avanzadas. Pero esas serán materias que veremos más adelante, a medida que vayamos profundizando en los contenidos del curso. Hasta entonces, el consejo consiste en ir experimentando con lo que hemos aprendido hasta ahora y, sobre todo, en curiosear el contenido de páginas reales.

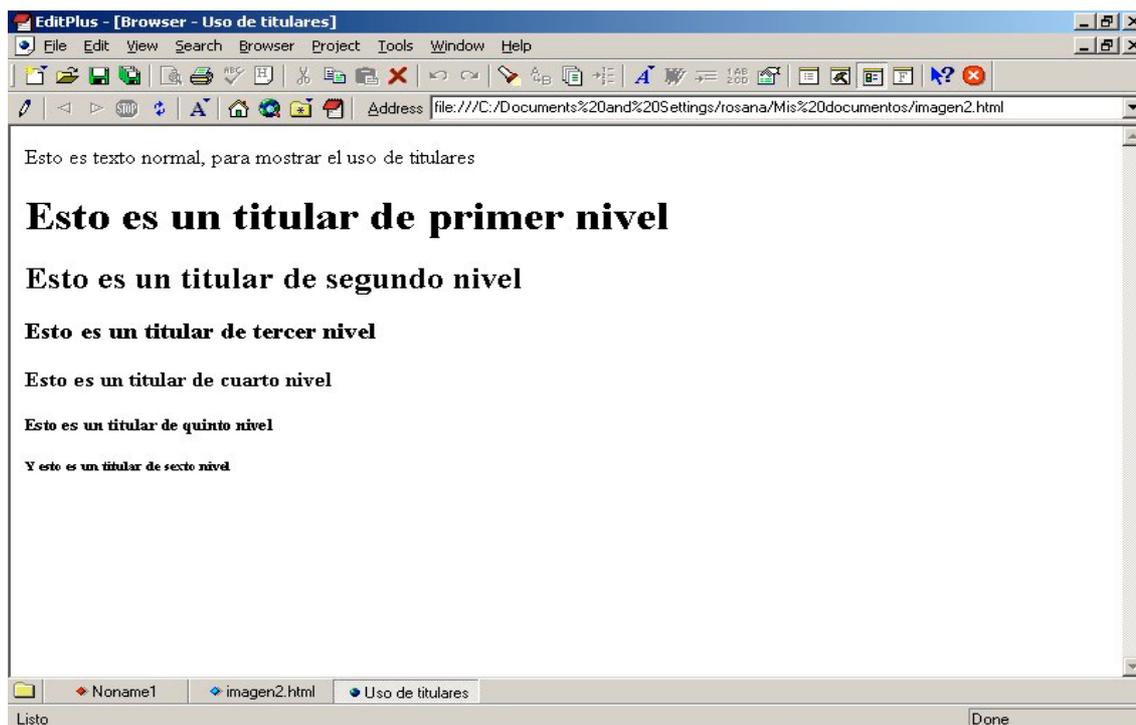
Bloques básicos:

En la primera parte aprendimos la estructura genérica de toda página .HTM, compuesta siempre por una cabecera y un cuerpo, y aprendimos igualmente a incorporar nuestro propio texto a cualquiera de estas páginas. Con estos simples conocimientos ya es posible comenzar a crear contenidos, dado que en toda página web lo realmente importante es la información que contiene, no la forma en que ésta se presenta.

De hecho, aunque tradicionalmente se puede afirmar que una imagen vale más que mil palabras, en la Web sucede justo lo contrario; es decir, las palabras (y, por tanto, la información que encierran) son tanto o más valiosas que las imágenes que puedan adornar nuestras páginas (sin que ello suponga, por supuesto, un menosprecio de éstas y teniendo siempre en cuenta los «mandamientos de diseño» que indicamos en nuestra anterior entrega). Ahora bien, la utilización masiva de texto contradice precisamente una de las mencionadas reglas de diseño, es decir, la de organizar la información y presentarla de forma atractiva y legible. Pues bien, entre otros, el lenguaje HTML nos ofrece el mecanismo de los denominados titulares, que permiten estructurar el texto en distintos niveles de importancia, mediante seis etiquetas distintas, desde <H1> hasta <H6>.

Cada etiqueta de titular funciona como un estilo de encabezado en un procesador de texto o como un nivel dentro de un esquema, permitiendo así la elaboración de una estructura y subdivisión en el documento. Aunque se pueden utilizar hasta seis niveles de titulares, por cuestiones de estilo y legibilidad dentro de cada página es recomendable limitarse a un máximo de cuatro. Nótese que puesto que los titulares no son párrafos, no necesitan hacer uso de las etiquetas <P> y </P>. A modo de ejemplo, en la «Imagen 2» se muestra el resultado del siguiente código:

```
<HTML>
<HEAD>
  <TITLE> Uso de titulares </TITLE>
</HEAD>
<BODY>
  <P> Esto es texto normal, para mostrar el uso de titulares </P>
  <H1> Esto es un titular de primer nivel </H1>
  <H2> Esto es un titular de segundo nivel </H2>
  <H3> Esto es un titular de tercer nivel</H3>
  <H4> Esto es un titular de cuarto nivel</H4>
  <H5> Esto es un titular de quinto nivel</H5>
  <H6> Y esto es un titular de sexto nivel</H6>
</BODY>
</HTML>
```



Sin embargo, como se puede observar en nuestro propio ejemplo, conviene tener en cuenta que el tamaño y el estilo de tales titulares varía dependiendo de cómo esté configurado el navegador con el que se esté presentando el documento. Afortunadamente, para modificar ambas características, el lenguaje HTML nos ofrece una etiqueta adicional, de tipo de fuente, que no es válida sólo para titulares sino para texto en general: .

Textos polifacéticos

El uso de la etiqueta de tipo de fuente combinado con el parámetro FACE delimita el conjunto de caracteres sobre los cuales tiene efecto, con independencia del conjunto de etiquetas que existan en su interior. Esto permite aplicar diferentes estilos a una misma página, utilizando tipos de fuente distintos en aquellos elementos que nos interese destacar, por ejemplo; o bien aplicando un estilo de fuente definido que no dependa de los estilos por defecto propios del navegador con el que se visualice el documento.

Para variar a nuestro antojo las características del texto, la etiqueta de tipo de fuente admite un parámetro adicional, SIZE o tamaño, mediante el cual podremos seleccionar una serie de tamaños predeterminados. A modo de ejemplo, el resultado del siguiente código se puede visualizar en la «Imagen 3»:

```
<HTML>
<HEAD>
<TITLE> Ejemplo: Jugando con los tipos de fuente </TITLE>
</HEAD>
<BODY>
<P> Esta es la fuente configurada por defecto en nuestro navegador
</P>
<FONT FACE= "Verdana">
<P> Ahora seleccionamos una fuente de tipo Verdana </P>
</FONT>
<FONT FACE= "Arial", SIZE=1>
<P>A continuación, cambiamos a una fuente de tipo Arial y tamaño 1
</H1>
</FONT>
</BODY>
</HTML>
```

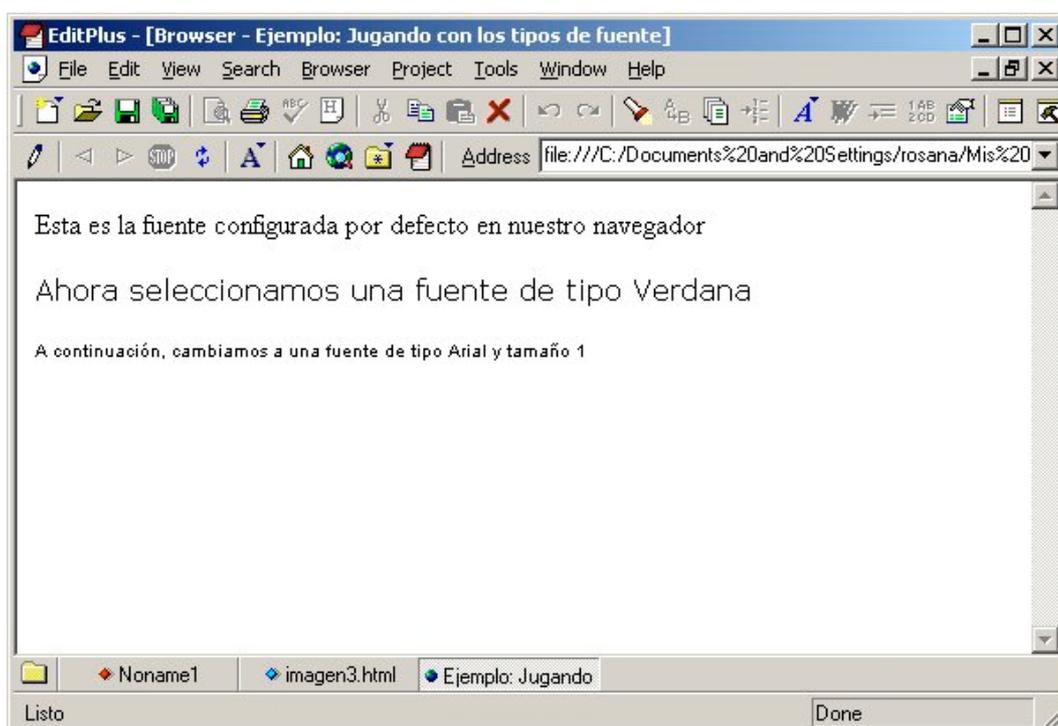


Imagen 3. Mediante la utilización de distintos parámetros podemos jugar con el tipo y tamaño de letra que se aplica en los distintos apartados de nuestras páginas web.

Como sería de suponer por lo que hemos visto hasta el momento, sobre el texto podemos aplicar otras características que nos permitirán mostrar un resultado aún más atractivo, como por ejemplo el centrado (mediante la etiqueta <CENTER>), el enfatizado mediante negritas (con la etiqueta) o el enfatizado mediante cursiva (con la etiqueta). El resultado de todas estas modificaciones se ha aplicado en el siguiente código y su resultado se puede visualizar en la «Imagen 4»:

```
<HTML>
<TITLE> Ejemplo de cómo destacar texto
</TITLE>
</HEAD>
<BODY>
<P> Este es el formato de párrafo que se usa por defecto </P>
<CENTER>
<P> Este es el mismo formato de párrafo, con centrado </P>
</CENTER>
<P> Esto muestra cómo resaltar texto en <STRONG> negrita </STRONG>
</P>
<P> Y esto muestra cómo resaltarlo en <EM> cursiva </EM> </P>
</BODY>
</HTML>
```

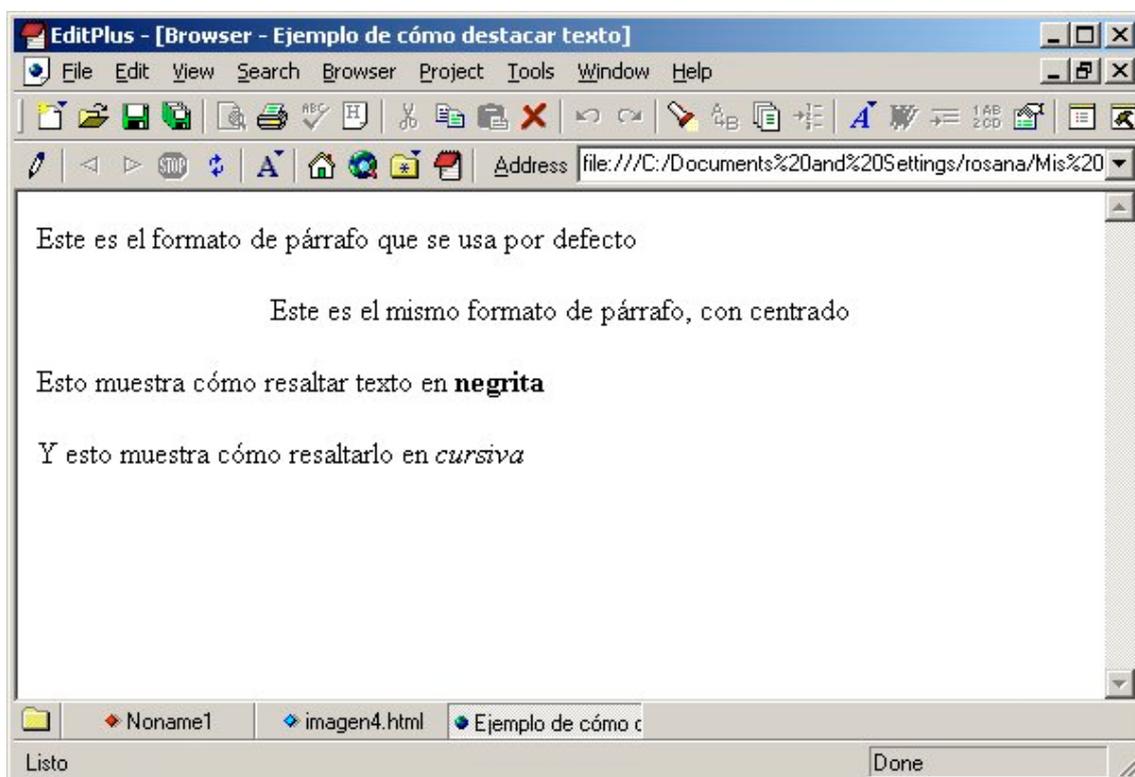


Imagen 4. El lenguaje HTML nos ofrece mecanismos muy simples pero muy efectivos a la hora de destacar parte del texto. No obstante, mas adelante del curso aprenderemos mecanismos más elaborados y bastante más vistosos.

Los atributos de texto resultan sumamente útiles de cara a tener un mayor control sobre la forma de presentar éste, pero en la práctica es interesante conocerlos sólo en el caso de que exista algún problema con alguna palabra o párrafo en concreto. De hecho, la práctica más habitual consiste en utilizar hojas de estilo, tal y como veremos en un capítulo posterior de nuestro curso.

Organización visual

Uno de los elementos visuales más útiles a la hora de dividir una página web en partes claramente diferenciadas son las líneas horizontales. Por defecto se trata de líneas sombreadas que cuando se visualizan sobre un fondo gris parecen barras tridimensionales, trazadas a todo lo ancho de la página.

Para añadir las a nuestras páginas, bastará con hacer uso de la etiqueta <HR>, aunque ésta ofrece cuatro atributos adicionales, que describimos a continuación.

Así, con <HR SIZE=n>, podemos especificar el grosor de la línea, siendo n el número de píxeles que ésta ocupa; mientras que con <HR WIDTH=n>, podemos especificar la anchura de la línea, siendo n el número de píxeles. Por su parte <HR ALIGN=alineación>, indica la alineación de la línea, con los tres valores posibles de izquierda (LEFT), centro (CENTER) y derecha (RIGHT). Finalmente, si utilizamos <HR NOS-HADE> obtendremos una línea sin sombreado horizontal.

En continua renovación

Ahora que hemos explicado los fundamentos más básicos del lenguaje, haremos una parada para explicar una norma no escrita pero de cumplimiento generalizado en la Red, referente al estado de actualización de las páginas. Es decir, lo habitual es que a la hora de diseñar nuestras páginas web reservemos al menos una línea (generalmente en la parte inferior de las mismas), destinada a mostrar tanto la fecha de creación como la fecha de última actualización de los contenidos

La utilidad de dicha información consiste en permitir a nuestros visitantes hacerse una idea bastante aproximada de la calidad de los contenidos (como es natural, no deberíamos esperar unos contenidos «frescos» en unas páginas que no se han actualizado en los últimos tres años, por ejemplo), pero se convierte también en un arma de doble filo, ya que nos exigirá una mayor dedicación al mantenimiento de las mismas. Este último es un elemento del que no habíamos hablado hasta el momento pero, haciendo un breve inciso en el hilo de nuestra narración, debemos dejar claro que el diseño de nuestras páginas web es una actividad sumamente gratificante y bastante absorbente, a poco que deseemos ofrecer un mínimo de calidad. Al igual que sucede con otras actividades que podemos realizar con el ordenador, el diseño de páginas web es una labor de índole creativa y, como tal, motivadora.

Sólo con que nos metamos en este mundillo nuestra curiosidad irá en aumento, y nuestro interés por aprender nuevas técnicas será una constante. En relación con el tema de las fechas tenemos que dejar claro que para éstas no se requieren nuevas etiquetas, sino que es preferible añadir dicho texto con un tamaño muy reducido. Como nota adicional tenemos que decidir si diseñaremos nuestras páginas en castellano o en inglés, dado que en este último caso el convenio de notación es diferente: 10-8-00 no significa 10 de agosto, sino 8 de octubre. Por ello, es aconsejable indicar explícitamente el mes.

El código de ejemplo que se muestra a continuación se puede visualizar en la «Imagen 5»:

```
<HTML>
<HEAD>
<TITLE> Otro ejemplo de página personal </TITLE>
</HEAD>
<BODY>
<CENTER>
  <H1> ¡Bienvenidos a mi nueva página personal! </H1>
</CENTER>
<P> Esta página está en construcción, pero los contenidos se
actualizarán en breve </P>
<HR>
<P>
<CENTER>
  <FONT FACE="Arial", SIZE=1> Página creada el 10 de Octubre de
2001 </FONT>
</CENTER> </P>
</BODY>
</HTML>
```

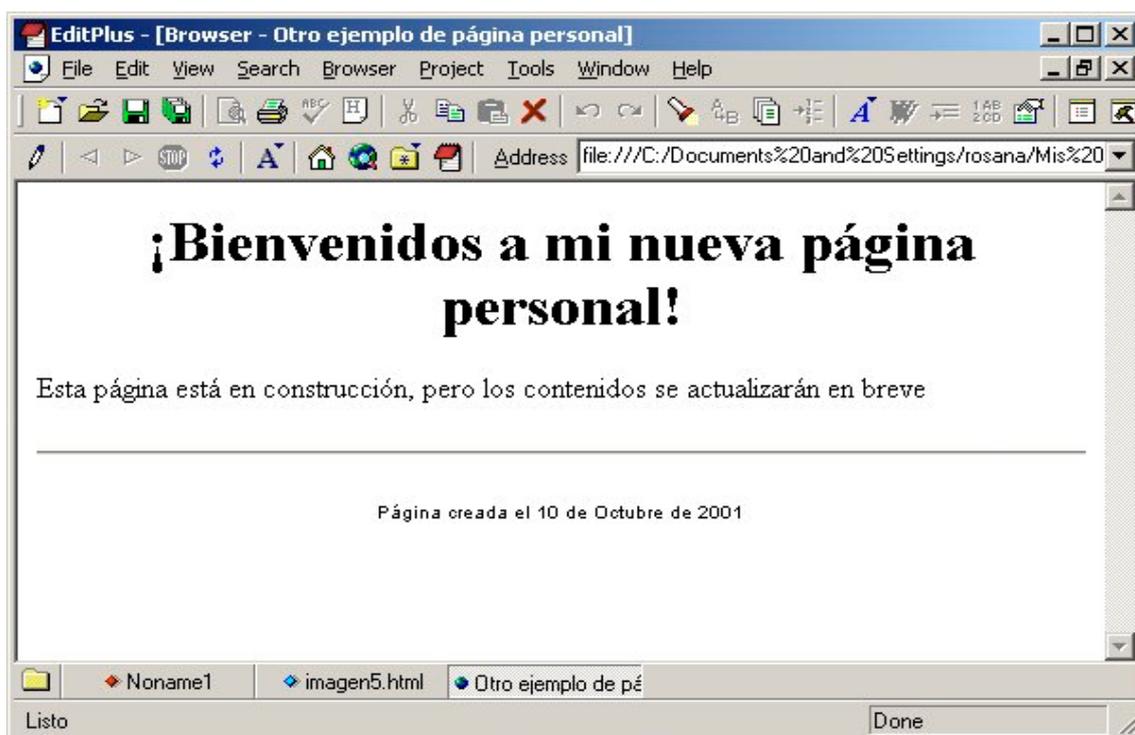


Imagen.5 Una de las características más atractivas que la Red ofrece consiste en relacionar los distintos contenidos presentes en las páginas. Para ello se hace uso de los enlaces.

Enlaces y referencias

Cuando se están diseñando páginas web hay que intentar, de acuerdo con las reglas de diseño que señalamos en la entrega previa, que éstas sean concisas, lo cual nos obligará a dividir nuestros contenidos en un conjunto de páginas y, consecuentemente, a incorporar enlaces entre todas ellas. Es más, en HTML se pueden definir tres tipos distintos de enlaces:

- enlaces dentro de una misma página,
- enlaces a otras páginas situadas dentro del propio sistema y
- enlaces hacia páginas situadas en otros sistemas.

Los enlaces o referencias dentro de una misma página nos conducen hacia otro punto de la misma página web en la que nos encontramos. Los enlaces dentro del propio sistema apuntan a otros elementos (gráficos, imágenes, sonidos) o bien a otras páginas, generalmente creadas por nosotros mismos, que se alojan dentro del mismo servidor. Finalmente, los enlaces hacia otros sistemas apuntan a páginas alojadas en otros servidores. Para establecer un enlace usaremos una etiqueta del tipo `<A HREF>`, con una sintaxis ligeramente diferente en función del tipo de enlace que queramos establecer.

Así, para establecer un enlace dentro de la misma página usaremos una etiqueta auxiliar, del tipo

```
<A NAME="nombre_enlace"> </A>
```

posicionada en el sitio al que queremos establecer el enlace. Al mismo tiempo, en el punto desde el que vamos a efectuar el enlace, utilizaremos la etiqueta:

```
<A HREF="#nombre_enlace">Enlace dentro de la página </A>
```

Para establecer un enlace a otra página o archivo local usaremos la sintaxis

```
<A HREF="nombre_archivo"> Enlace a un archivo local </A>
```

Nótese que, al igual que en el caso del enlace anterior, las comillas son obligatorias y cuando visualicemos el resultado en nuestro navegador aparecerán subrayadas las palabras situadas entre el símbolo `>` de la etiqueta `<A HREF>` y el símbolo `<` de la etiqueta ``.

Finalmente, para establecer un enlace hacia una página situada en un servidor remoto, utilizaremos la misma sintaxis que se emplea en el caso de un enlace local, con la única diferencia de tener que indicar la dirección o URL completo dentro de las comillas.

Además de los indicados en el punto anterior, existe un tipo particular de enlace que permite que los visitantes de nuestras páginas puedan ponerse en contacto con nosotros por medio del correo electrónico para enviarnos sugerencias, comentarios, etc. Su sintaxis es similar a la que hemos visto en los párrafos anteriores, aunque con una ligera diferencia:

```
<A HREF="mailto:dirección">Texto</A>
```

En el siguiente código, se muestran ejemplos de distintos tipos de enlaces:

```

<HTML>
<HEAD>
<TITLE> Mi página personal avanza </TITLE>
</HEAD>
<BODY>
<CENTER>
<H1> ¡Bienvenidos a mi nueva página personal! </H1>
</CENTER>
<P> Esta página está en construcción, pero los contenidos se
actualizarán en breve. Si deseas enviarme cualquier sugerencia,
puedes <A HREF="mailto:rosana@ugr.es"> escribirme.
</A></P>
<P> Desde esta página también puedes acceder a información de <A
HREF="www.intel.com"> Intel </A>
y <A HREF="www.amd.com">AMD</A>. </P>
<HR>
<P> <FONT FACE="Arial" SIZE=1> Página creada el 7 de mayo de 2001.
Fecha de última actualización, 24 de Octubre de 2001 </FONT>
</P>
</BODY>
</HTML>

```

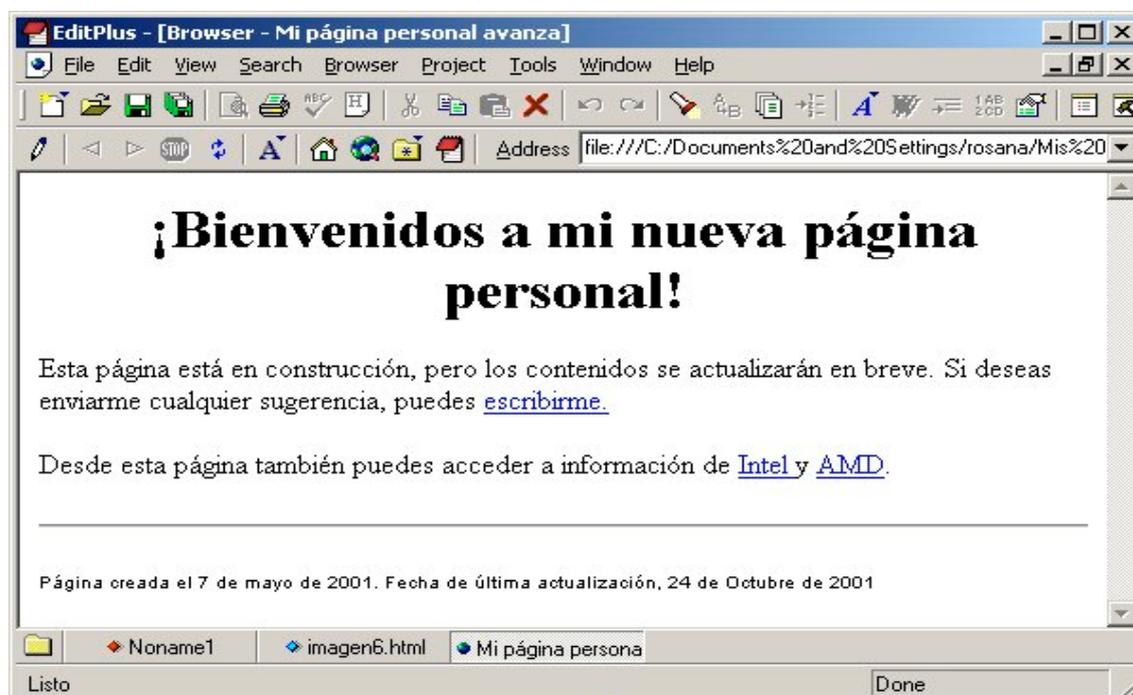


Imagen 6. Diseñar una página implica mucho más, puesto que si queremos que sea útil tenemos que mantenerla actualizada.

Gráficos e imágenes

Aunque antes hemos hablado acerca de la importancia de la información reflejada en nuestras páginas, no debemos perder de vista el poder de la imagen, algo que resulta particularmente cierto en la Web, en la que la mayor parte de las páginas contienen al menos uno o dos elementos gráficos. En general, el tratamiento que daremos a los gráficos e imágenes será el mismo, dado que en ambos casos estamos hablando de elementos visuales contenidos en ficheros externos a nuestras páginas.

Sin embargo, tan sólo mencionaremos un par de ideas acerca del tipo de imágenes a utilizar. En este sentido, los dos formatos más extendidos en la Red son el GIF y el JPG. Ambos tienen sus inconvenientes y sus ventajas propias, pero dadas las características propias de cada uno podríamos establecer una norma de carácter general: utilizar el formato GIF de forma habitual, reservando el JPG para mostrar imágenes ampliadas de gran resolución. De forma sencilla, para incorporar una imagen a un documento HTML, haremos uso de la sintaxis:

```
<IMG SRC="camino/nombre_de_archivo">
```

Nótese que camino suele especificar un path relativo al directorio en que se encuentra almacenada la página HTML desde la que se hace la referencia, teniendo en cuenta que, en este caso, las barras de separación de directorios siguen una sintaxis similar a la de Unix, es decir, mediante la barra /. Sin embargo, la posibilidad de incorporar imágenes cuenta con distintos atributos, como la alineación de las mismas (mediante , tomando posición el valor LEFT para alineación izquierda o RIGHT para alineación derecha, teniendo en cuenta que para centrarlas usaríamos la etiqueta <CENTER> anteriormente descrita) o el tamaño de las mismas (con los atributos WIDTH y HEIGHT para ancho y alto, respectivamente).

Otra posibilidad más elaborada consiste en utilizar una imagen como activador de un enlace, es decir, que al pulsar encima de la imagen fuésemos directos a dicho enlace. Para ello, la sintaxis es relativamente sencilla, y bastará con escribir en el código de nuestra página

```
<A HREF=enlace><IMG SRC=nombre_imagen></A>
```

Incorporación de tablas

Otro elemento sumamente útil para estructurar la información en nuestras páginas son las tablas, las cuales no sirven únicamente para mostrar datos organizados en filas y columnas, sino también para dividir los contenidos en distintas zonas horizontales o verticales.

No obstante, aquí nos centraremos únicamente en la primera posibilidad, en el significado más clásico de tabla, reservando la otra para más adelante. Para ello, dentro de nuestras páginas HTML, las tablas se delimitan con los tags <TABLE> y </TABLE>, intercalando dentro de éstos los contenidos correspondientes a la tabla y tres tipos de etiquetas adicionales, que permiten organizar dichos contenidos: <TH> para información de cabecera (junto con </TH>); <TR>, para información de fila; y <TD>, para los datos propiamente dichos (junto con </TD>).

Para verlo con mayor claridad, la «Imagen 7» muestra el resultado correspondiente al siguiente código:

```
<HTML>
<HEAD>
<TITLE> Ejemplo de uso de tablas </TITLE>
</HEAD>
<BODY>
<CENTER> <H1> Análisis químico </H1> </CENTER>
<P> En la fabricación de nuestro delicioso refresco intervienen
únicamente componentes naturales,
cuyo análisis se muestra a continuación:
</P>
<CENTER>
<TABLE BORDER>
<TR> <TH> Elemento </TH>
<TH>Cantidad (mg/l)</TH>
<TR> <TD> Carbonatos </TD>
<TD> 255,1 </TD>
<TR> <TD> Nitratos </TD> <TD> 23,8
</TD>
<TR> <TD> Sulfuros </TD> <TD> 2,1
</TD>
<TR> <TD> Calcio </TD> <TD> 55,3
</TD>
<TR> <TD> Magnesio </TD> <TD> 47,4
</TD>
</TABLE>
</CENTER>
<HR>
<P> <FONT FACE="Arial", SIZE=1> Página creada el 7 de mayo de
2001. Fecha de última actualización, 9
de mayo de 2001</FONT> </P>
</BODY>
</HTML>
```

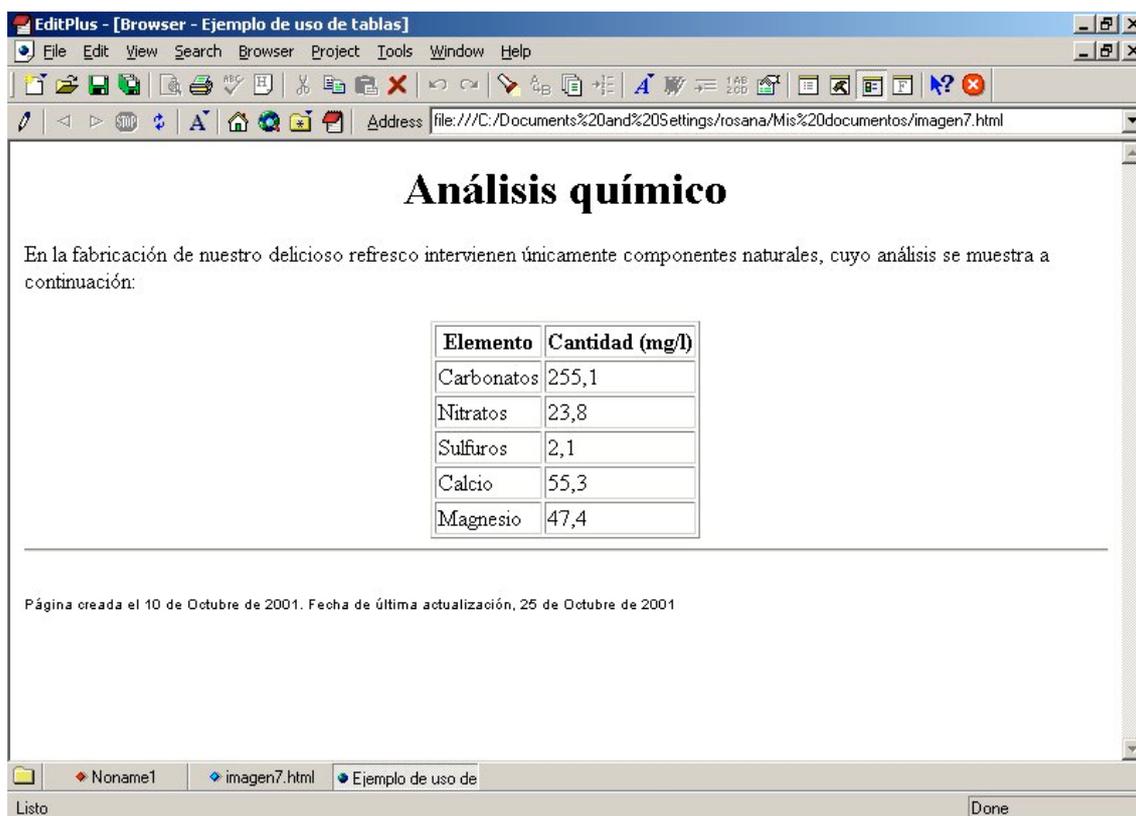


Imagen 7. Con independencia del tipo de información que queramos mostrar, las tablas son un extraordinario mecanismo para presentar ésta de forma ordenada.

Ahora bien, no podemos concluir este apartado sin mencionar tres parámetros que nos permitirán controlar distintos aspectos de nuestras tablas. El primero de ellos es el de justificación, ALIGN, que puede ser izquierda (LEFT), derecha (RIGHT) o centrada (CENTER). Dicho parámetro se puede usar indistintamente con los tags <TH> y <TD>. Por ejemplo, con <TD ALIGN="RIGHT"> conseguiremos que el contenido de la celda en cuestión quede justificado a la derecha. El segundo parámetro de interés es BORDER, que aparece en el anterior ejemplo y que se usa en el tag TABLE para indicar que las celdas de la tabla deben estar delimitadas por líneas.

Finalmente, contamos también con el parámetro WIDTH para especificar el ancho total de la tabla (junto con <TABLE>) o de cada columna (junto con el tag <COL>). Este parámetro se usa en conjunción con otro parámetro adicional, UNITS, que puede tomar los valores PIXEL (es decir, puntos de pantalla) o RELATIVE (es decir, porcentaje sobre el ancho total de la página). Así, por ejemplo, con:

```
<TABLE WIDTH="50" UNITS="RELATIVE">
```

definimos que el ancho de la tabla es la mitad del tamaño de la página, con independencia de la resolución existente en el puesto cliente.

Formularios

Si somos navegantes asiduos de la Red, habremos observado que en numerosas páginas existen formularios mediante los que se puede recabar información adicional de sus visitantes. Este potente mecanismo permite dotar de mayor interactividad a dichas páginas, puesto que establecen un canal de comunicación a través del cual se pueden utilizar las respuestas del usuario para diferentes propósitos, como gestionar pedidos o alimentar una base de datos.

La forma más simple de declarar un formulario dentro de una página HTML consiste en utilizar el tag <FORM ACTION="URL" METHOD="post">, añadiendo el tag </FORM> al final de éste. En este caso, URL es el nombre de un programa CGI que reside en el servidor web, y que básicamente se encarga de recoger información en el formulario y pasarla a dicho servidor.

No obstante, dado que la programación de scripts CGI requiere bastante conocimiento de programación, no continuaremos profundizando en el aspecto de los formularios, con todo el detalle que se merecen y nos quedaremos solo en el apartado de diseño de formularios. A continuación se muestra un ejemplo completo con todos los elementos gráficos de un formulario.

```
<HTML>
<HEAD>
<TITLE>Diseño Formularios</TITLE>
</HEAD>

<BODY>
<H2>Formulario de Respuesta</H2>

<P>Por favor rellene este cuestionario, como forma de mejorar
nuestra web.</P>

<FORM METHOD="POST" ACTION="/cgi-bin/formmail">

<INPUT TYPE="hidden" NAME="receptor" VALUE="rosana@ugr.es">
<INPUT TYPE="hidden" NAME="asunto" VALUE="Feedback Formulario">
<INPUT TYPE="hidden" NAME="redirigir" VALUE="main.html">

<P><STRONG>Nombre: </STRONG>
<INPUT NAME="nombre" TYPE="text" SIZE="25">

<STRONG>Email:</STRONG>
<INPUT NAME="email" TYPE="password" SIZE="25"></P>

<P><STRONG>Comentarios:</STRONG>
<TEXTAREA NAME="comentarios" ROWS="4" COLS="36"></TEXTAREA> </P>

<P><STRONG>Cosas que te gustaron:</STRONG><BR>

Diseño de la web <INPUT NAME="things" TYPE="checkbox"
VALUE="Diseño">
Enlaces <INPUT NAME="things" TYPE="checkbox" VALUE="Enlaces">
Facilidad de uso <INPUT NAME="things" TYPE="checkbox" VALUE="Uso">
Imágenes <INPUT NAME="things" TYPE="checkbox" VALUE="Imágenes">
Codigo Fuente <INPUT NAME="things" TYPE="checkbox" VALUE="Codigo">
</P>

<P><STRONG>¿Como conocistes esta web?:</STRONG><BR>

Motor de búsqueda <INPUT NAME="como conocistes" TYPE="radio"
VALUE="motor búsqueda" CHECKED>
Enlace desde otra web <INPUT NAME="como conocistes" TYPE="radio"
VALUE="enlace">
Web de la Universidad <INPUT NAME="como conocistes" TYPE="radio"
VALUE="ugr.es">
Estaba en un libro <INPUT NAME="como conocistes" TYPE="radio"
VALUE="libro">
Otro <INPUT NAME="como conocistes" TYPE="radio" VALUE="otro">
</P>

<P><STRONG>Puntuanos (1-10):</STRONG>
<SELECT NAME="puntos">
<OPTION SELECTED>Increible :->
<OPTION>10
<OPTION>9
<OPTION>8
<OPTION>7
<OPTION>6
<OPTION>5
<OPTION>4
<OPTION>3
```

```

<OPTION>2
<OPTION>1
<OPTION>Apesta :- (
</SELECT></P>

<INPUT TYPE="submit" VALUE="Enviar datos">
<INPUT TYPE="reset" VALUE="Borrar datos">
</FORM>

</BODY>
</HTML>

```

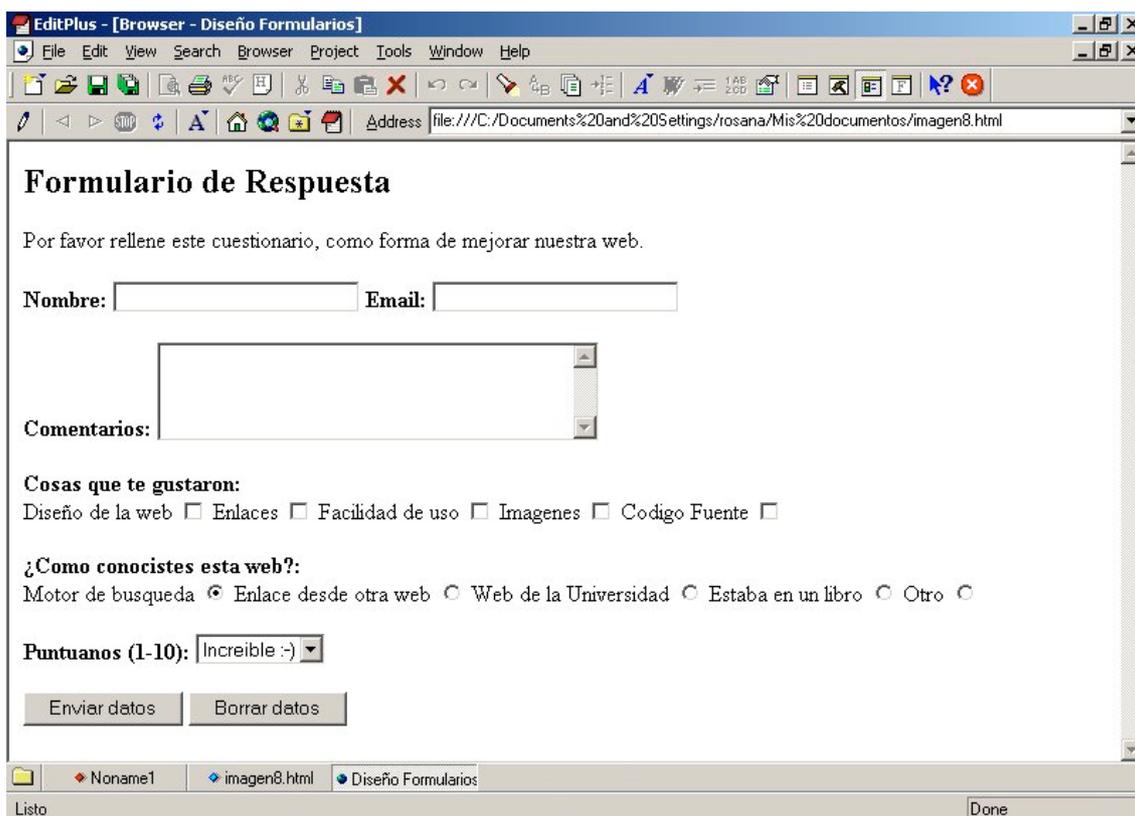


Imagen . Los formularios nos permiten recopilar información por parte del usuario. Sin embargo, dada su complejidad no entraremos con ellos en detalle a estas alturas del curso, posponiendo dicha explicación para más adelante.

Si no vamos a usar un programa CGI, por su complejidad, tal vez podamos usar el campo ACTION del formulario para irnos a otra página en la que agradezcamos al usuario su colaboración, o bien mandarnos los datos tecleados a nosotros mismos por correo. Ambas cosas se hace como si de un enlace se tratase.

Páginas multimedia

Entre las distintas posibilidades que nos ofrece el lenguaje HTML, tenemos que mencionar la posibilidad de incorporar a nuestras páginas tres elementos claramente diferenciadores y que las dotarán de carácter propio: nos referimos a los fondos, sonidos y vídeos. Los fondos (o *backgrounds*) no son otra cosa que imágenes que «tapizan» nuestras páginas, ofreciendo un decorado sobre el que visualizar el resto de los elementos. Para incorporar un fondo a una de nuestras páginas tan sólo necesitamos modificar la definición de la etiqueta <BODY>, para que quede como:

```
<BODY BACKGROUND=nombreficheroFondo>.
```

No obstante, hemos de tener en cuenta que cualquier imagen no sirve como fondo: debe ser agradable a la vista, tener un tamaño adecuado y, sobre todo, no dificultar la legibilidad de los elementos situados encima de éste (por ejemplo, si el texto es negro, no se debe utilizar un fondo de color oscuro tipo azul marino).

En lo que respecta a los sonidos y vídeos, su utilización es idéntica al caso de las imágenes, es decir, estableciendo una referencia contra el fichero que los contiene. En este caso deberemos optar por formatos estándar, dado que tenemos que tener en cuenta que los potenciales visitantes de nuestras páginas deberán tener instalados reproductores de audio y vídeo capaces de interpretar correctamente nuestros enlaces. En caso de que no sea así, no pasará nada, salvo que quienes visiten nuestras páginas no podrán ver o escuchar tales elementos.

Autopistas congestionadas

Actualmente el famoso concepto de «autopistas de la información» no hace referencia a vías de comunicación de alta velocidad, sino a vías de comunicación por las que circula un gran número de datos. Salvando las distancias, el concepto es homólogo al de las autopistas y autovías que encontramos alrededor de las grandes ciudades, dado la habitual congestión que existe en éstas.

En ambos casos, la solución es similar, ya que exige la creación de mejores vías de comunicación, capaces de acelerar el desplazamiento de la información, pero también supone un coste muy elevado, difícil o imposible de asumir para las empresas que se conectan a la infraestructura de Internet. Afortunadamente, la mayor implicación de los gobiernos públicos y las fuertes inversiones que se están efectuando en este terreno tendrán un impacto muy positivo en las prestaciones, posibilitando mayor velocidad y nuevos servicios a lo largo de los próximos años, como podrían ser el vídeo en tiempo real. Hasta entonces tendremos que conformarnos con distintas soluciones de aceleración que comienzan a llegar al mercado, enfocadas fundamentalmente a los proveedores de servicios y contenidos, con el fin de que éstos puedan ofrecer mayor calidad de navegación a sus usuarios. En este sentido, cabe distinguir soluciones basadas en sistemas *cache*, soluciones de *mirroring* (o duplicación de sitios web completos en nodos más cercanos a los usuarios) o soluciones de aceleración de contenidos (basadas en la compresión y descompresión de éstos en las denominadas soluciones con software cliente, o en la optimización de los mismos en las llamadas soluciones *client-free* o sin software cliente).

A modo de resumen

En este segundo tema de programación en la Web hemos finalizado prácticamente nuestro recorrido por los entresijos del lenguaje HTML. Concluiremos la exposición de algunos detalles que se nos han quedado aún en el tintero y mostraremos algunos ejemplos prácticos que sirvan como recopilatorio de todo lo aprendido.

Del mismo modo, estableceremos contacto con algunas de las herramientas de edición HTML más habituales, desarrollando con éstas los ejemplos paso a paso. De esta forma, al mismo tiempo que recopilamos conocimientos veremos cómo implementar éstos de forma aplicada.

La recomendación personal consiste en hacer cuantas prácticas se nos ocurran, jugando con las etiquetas que hemos explicado y explorando las distintas posibilidades que éstas nos ofrecen de cara a la presentación de la información en pantalla.