

Nombre: .....

Fecha: /02/2012

Grupo: 1 □

PRÁCTICA 3  
**HTML. GESTIÓN DE UN SERVIDOR WEB POR FTP**

---

En esta práctica introduciremos el lenguaje HTML (Hypertext Markup Language) que nos permitirá crear diversas páginas web. Comenzaremos por subir una página web sencilla que nos permita conocer la estructura básica que cualquier código html debe tener y cómo gestionar un servidor web. Posteriormente, explicaremos el significado de los "tags" más relevantes que pueden ser usados en html. Enfatizaremos la necesidad de usar metainformación que pueda ser usada por los robots y buscadores de Internet. También daremos importancia a la separación entre contenido y presentación. Siguiendo las recomendaciones del DTD "Strict", procuraremos que nuestra web sólo contenga información sobre el contenido de la misma, dejando todo lo relativo a la presentación y aspecto final de la página a los "css" correspondientes que veremos en prácticas futuras. Trabajaremos con elementos que nos permitirán definir listas, enlaces y tablas en nuestros sitios web.

#### Parte 1. Gestión de un servidor web a través de un cliente ftp

El hecho de "publicar" una página web requiere que la misma sea compartida en un servidor web (un servidor que soporte el protocolo http) al cual tengamos acceso.

Para la asignatura de Informática dispondremos de "belenus", un servidor con varias utilidades. Este servidor es para el uso de profesores y alumnos de Grado en Matemáticas y Grado en Ingeniería Informática. Sus funcionalidades son:

- Espacio de 100 MB de disco para cada usuario, accesible vía FTP utilizando la CUASI.
- Espacio para la publicación Web, tanto para profesores como alumnos. Los contenidos deberán ajustarse a trabajos y ejercicios a realizar en las distintas asignaturas.

El uso del servidor debe ser para la funcionalidad que pretende, pudiendo ser restringidos cualquiera de los servicios por un uso indebido. El acceso al servidor se realiza mediante la cuenta CUASI.

1. En primer lugar, vamos a ver qué aspecto tiene nuestro espacio en el servidor. Para ello necesitamos un cliente FTP, es decir, un programa que nos permita conectarnos (y transferir información) a un servidor. En este caso, vamos a usar un programa que se llama Filezilla y que es de libre (y gratuita) distribución (<http://filezilla-project.org/download.php?type=client>). Ejecuta filezilla en tu ordenador.

2. Como puedes observar, la pantalla principal del mismo aparece dividida en dos columnas. La columna de la izquierda representa tu equipo, en particular su sistema de directorios. En la columna derecha aparece la estructura de directorios del servidor al que estés conectado (de momento aparece vacío, no nos hemos conectado a ninguna máquina).

En la barra de herramientas del programa debemos completar los parámetros del servidor al que nos queremos conectar. En este caso, los mismos son:

Servidor:	belenus.unirioja.es
Nombre de usuario:	cuasi
Contraseña:	password
Puerto:	(vacío)

Si has introducido los datos de manera correcta, debería aparecer en la parte derecha de filezilla (bajo la pestaña "Sitio remoto") el sistema de directorios de tu carpeta en el servidor. En el mismo deberían aparecer, al menos, una carpeta de nombre "public\_html" y un fichero de nombre "leeme.txt".

La carpeta "public\_html" es tu carpeta pública, y por tanto todo lo que incluyas en ella será accesible a través de un navegador, por medio de una url o dirección web que veremos más adelante.

3. Vamos a ver ahora cómo podemos transferir un fichero desde el servidor a nuestro equipo. Selecciona el fichero "leeme.txt" de la carpeta en el servidor. Observa en qué carpeta te encuentras en la columna izquierda de filezilla. Seleccionando el fichero "leeme.txt" con el ratón, pulsa el botón derecho del mismo y selecciona la opción "Descargar". El fichero debería aparecer ahora en tu equipo, en la carpeta que apareciese en la columna izquierda del filezilla.

Lee el contenido del fichero.

4. Ahora vamos a ver cómo podemos subir ficheros de nuestro equipo hasta el servidor. Para ello, en primer lugar vamos a crear una primera página web que será el fichero que subamos (por ejemplo, en "Notepad++").

Por el momento sólo pretendemos crear una página funcional, y no entraremos en los detalles sobre la creación de la misma. Utiliza el código siguiente. Observa que su codificación es "utf-8", lo cual quiere decir que deberá ser guardado en formato "utf-8". ¿Qué herramientas conoces que sepan almacenar en ese formato? Utiliza una de ellas. Observa también las rutas "informes/Practica00.txt"; ten cuidado a la hora de subir tus guiones o prácticas a dicha carpeta.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<meta name="keywords" content="Informática, Grado en Ingeniería Agrícola,
    Universidad de La Rioja, tu nombre">
```

```

<meta name="description" content="En esta página se pueden encontrar enlaces a las
soluciones de las prácticas de Informática para del curso 2011/2012.
Todo el contenido ha sido elaborado por nombre.">
<meta name="Author" content="Tu nombre">
<meta name="Date" content="13, Febrero 2012">
<meta name="robots" content="all">
<title>Página web de Tu Nombre</title>
</head>
<body>
<p>
Bienvenido a la página web para la asignatura Informática de Tu Nombre.
</p>
<p>
A continuación puedes ver un listado con enlaces a los informes de las prácticas de la
misma:
</p>
<ul>
<li> <a href="informes/Practica00.txt">Práctica 00</a></li>
<li> <a href="informes/Practica01.txt">Práctica 01</a></li>
<li> <a href="informes/Practica02.txt">Práctica 02</a></li>
<li> Añade tantas entradas como necesites </li>
</ul>
</body>
</html>

```

5. Guarda el código anterior en un fichero "index.htm" (con un editor de textos que te permita guardar en formato "utf-8") y después súbelo por ftp a la carpeta "public\_html" de tu sitio web en belenus.

6. Comprueba con el navegador que la página web que has creado es accesible a través de las url's:

<https://belenus.unirioja.es/~cuasi>  
<https://belenus.unirioja.es/~cuasi/index.htm>

7. Comprueba los enlaces de la página creada. ¿Funcionan? ¿Cómo podemos solucionar esta situación?

8. Abre en otra pestaña la página web "<http://validator.w3.org/>". Comprueba que tu página web es "válida" (con respecto al estándar HTML 4.01). Corta el código que incluye la etiqueta de "página válida" y añádela al final de tu página "index.htm" (cuidado, antes de cerrar el tag "body", si no, no formará parte del cuerpo de la página). Vuelve a subir la página "index.htm" (sobrescribiendo la anterior) por "ftp" a tu sitio web y observa el resultado.

## Parte 2. Creación de páginas web en html

9. Visita la página <http://html.conclase.net/recursos/especificaciones>. Los documentos de especificaciones HTML 4.01, CSS de nivel 1 y CSS de nivel 2 serán nuestra referencia básica. También los puedes encontrar en inglés en la página del w3c (World Wide Web consortium), en castellano en <http://www.w3c.es/>; puedes leer en la misma los fines y aspiraciones de w3c. Los estándares, en inglés, están en <http://www.w3.org/TR/html401/> (HTML 4.01), <http://www.w3.org/TR/CSS1/> (CSS 1.0) y <http://www.w3.org/TR/CSS2/> (CSS2.1).

10. Entramos en <http://html.conclase.net/tutorial/html/>. El tutorial que aquí se presenta no es un tutorial "al uso", pero su lectura debería dejar claros conceptos como:

- Diferencia entre HTTP, HTML e hipertexto.
- La diferenciación clara entre contenido y apariencia en los documentos HTML:

- o Diferencia entre <br> y <p>.
- o Por qué <p> no se usa para introducir líneas en blanco.
- o Las etiquetas <h1>, <h2>... definen estructura, no forma de letra.
- o Las etiquetas <b>, <i>... no deberían usarse porque definen apariencia, las etiquetas <em>, <strong>... sí tienen sentido porque son estructura.
- o Una tabla no debería usarse para maquetar el contenido.
- o La problemática de los caracteres no ASCII:

```
<meta http-equiv="Content-Type" content="text/html; charset=xxxxxxx">
```

11. Existen múltiples editores con los que se pueden crear páginas web. En general los mismos se dividen en dos grandes grupos:

- Visuales: Son entornos generalmente conocidos como "WYSIWYG" ("what you see is what you get"), en los que directamente podemos editar la página web tal y como será vista a través del navegador. Su gran ventaja es que facilitan el desarrollo de código HTML más rápidamente. Su gran inconveniente es que el creador de la web no controla el código HTML al que da lugar el editor. En esta categoría podemos mencionar Dreamweaver, como herramienta de pago, o Kompozer, como herramienta libre.

- De texto: el desarrollador edita directamente el código HTML, introduciendo él mismo las etiquetas que dan lugar a los distintos elementos de la página. Aquí en realidad podemos utilizar cualquier editor de textos, desde el "bloc de notas" hasta otros que incluyen ciertas facilidades, como el coloreado de las palabras reservadas, previsualización de la web o conexión directa con nuestro sitio ftp. Por ejemplo, "Pspad" o "Notepad++" tienen algunas de estas características.

12. Ejecuta la aplicación "Notepad++" o "PsPad" (Notepad++ tiene más funcionalidades, pero puedes elegir la que prefieras). Recupera el "esqueleto" de la página HTML que hemos creado ("index.htm") y cópialo en un nuevo fichero "curri\_vitae.htm" (debes eliminar todo el contenido entre los tags <BODY>...</BODY>, y editar los metas que consideres que han cambiado, como fecha, keywords...).

13. Veamos ahora los distintos componentes que presenta. En primer lugar nos encontramos con la cabecera:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

¿Qué significa? ¿Tiene alguna influencia en el código HTML que se muestra? Lee detenidamente el contenido de <http://www.w3.org/QA/Tips/Doctype> y <http://www.w3.org/QA/2002/04/valid-dtd-list.html>. Recuerda incluir siempre en tus páginas web una declaración de DTD.

14. ¿Por qué hemos elegido el DTD "HTML 4.01 Strict"? Puedes encontrar en <http://www.w3.org/TR/REC-html40/sgml/dtd.html> el DTD correspondiente, así como las características del mismo. Incidimos en la idea de separación entre contenido y presentación, tal y como hace el DTD.

15. Una vez especificado el DTD, comenzamos con el código de nuestro sitio web. El código HTML se especifica por medio de "tags" o etiquetas ([http://es.wikipedia.org/wiki/Etiqueta\\_\(lenguaje\\_de\\_marcado\)](http://es.wikipedia.org/wiki/Etiqueta_(lenguaje_de_marcado))).

Generalmente, los "tags" definen regiones de texto que van a compartir cierta característica. Suele haber un "tag" de apertura y otro de cierre para cada región.

Por ejemplo:

- <HTML>...</HTML> delimitan el contenido de una página web.
- <HEAD>...</HEAD> delimitan la cabecera del sitio web. Como puedes observar, los tags pueden ir situados unos dentro de otros (eso sí, no tendría sentido encontrar una región <HTML>...</HTML> dentro de una región <HEAD>...</HEAD>). Debes tener en cuenta siempre la jerarquía que debe existir entre "tags".
- Los tags <meta ...> no tienen resultado sobre la visibilidad de la página. Como su nombre indica, contienen "metainformación", como cuestiones relevantes a quién ha sido el creador de la página, la fecha de creación de la misma, una descripción del contenido... Toda esa información es utilizada por los "robots" de los buscadores para ayudar a indexar las páginas web, o por el navegador para mostrar la página de forma correcta. Puedes encontrar más ejemplos de metas y una explicación más extendida de los mismos en <http://www.maestrosdelweb.com/editorial/usarmetas/>. Hay un "meta" especialmente relevante que tiene que ver con la codificación de caracteres en la página web y con el tipo MIME de la misma (<http://www.htmlquick.com/es/reference/mime-types.html>):

```
<meta http-equiv="content-type" content="text/html; charset=utf-8">
```

Añade a tu página web los "metas" que consideres oportunos (cuantos más "metas" incluyas, mejor documentada estará).

- <TITLE>...</TITLE> delimita la región que compondrá el título de la página web. ¿Dónde se muestra el mismo en la página?

Añade a tu página web un título descriptivo de la misma.

Como has podido observar, tanto los "meta" como "title" son tags contenidos dentro de la región "head" de la página web. Puedes encontrar una lista exhaustiva de tags que pueden incluirse en la región "head" en [http://www.w3schools.com/html/html\\_head.asp](http://www.w3schools.com/html/html_head.asp) (algunos de ellos se usarán más adelante). Veamos ahora algunos "tags" que van fuera de "head":

- <BODY>...</BODY> delimita el contenido del documento (<http://www.w3.org/TR/html401/struct/global.html#edef-BODY>). Cómo se

muestre ese contenido depende del dispositivo que usemos para cargar la web.

Dentro del tag "body" podemos encontrar casi cualquier otro tag propio de HTML. En las próximas prácticas iremos viendo algunos de ellos. En la práctica de hoy nos centraremos en aquellos que permiten definir cabeceras y crear listas.

- `<H1>...</H1>` a `<H6>...</H6>` delimitan cabeceras de orden 1 (el más grande) a orden 6 (el más pequeño) <http://www.w3.org/TR/html401/struct/global.html#edef-H1>. Utiliza el "tag" H1 para asignar una cabecera a la página. Después crea dos subsecciones en tu página web por medio de dos tags H2. Una será el índice de la página y el otro una sección de datos personales.

Estos tags son elementos de "nivel de bloque". Esto quiere decir, entre otras cosas, que el navegador siempre creará una línea (o bloque) nueva para mostrarlos. Es importante que recuerdes la distinción entre elementos de "nivel de bloque" y los de "nivel de línea" (que veremos más adelante), porque de ello depende, por ejemplo, dónde puede ser usado cada uno de ellos. Dentro de un elemento de nivel de bloque podemos usar otros elementos de nivel de bloque, u otros de línea, o texto. Además, todo el contenido de una página web debe estar incluido en elementos de nivel de bloque (o contenedores). Sin embargo, dentro de un elemento de nivel de línea sólo podemos encontrar texto u otros elementos de nivel de línea.

Nota. Puedes ir comprobando de forma ocasional el aspecto de tu página web en distintos navegadores gracias a la función de previsualización de Pspad o en el menú de herramientas de Notepad++ "Ejecutar - Launch in Firefox (o IE, Chrome, Safari)".

- `<ul>...</ul>` o `<ol>...</ol>` son elementos de nivel de bloque que delimitan una región que será considerada por el navegador como una "unordered list" (lista desordenada) o "ordered list" (lista ordenada). Toda lista está formada por una serie de elementos (o "list items"), que se delimitan por medio del tag `<li>` (en este caso el tag de cierre `</li>` es opcional, aunque recomendable). También se pueden crear sublistas, o listas que sean elementos de otras listas (como elemento de nivel de bloque, dentro de una lista puedes encontrar otros elementos de nivel de bloque, como por ejemplo listas). Crea una lista (ordenada o desordenada), dentro de la sección "Índice", que contenga los elementos etiquetados "Índice" y "Datos personales", y como una sublista de la segunda los campos "Nombre y apellidos", "Dirección", "Dirección electrónica", "Estudios", "Experiencia profesional" e "Intereses". Cada uno de los elementos de esta sublista debe tener su equivalente en la sección "Datos personales" como cabecera de orden 3 (H3).

- `<p>...</p>` y `<br>`. El tag `<p>...</p>` (<http://www.w3.org/TR/html401/struct/text.html#edef-P>) sirve para definir párrafos en nuestro texto. Tal y como está especificado, no puede contener dentro otros "elementos de nivel de bloque" (tales como H1...H6, ul, ol o el propio p, o los dados en <http://htmlhelp.com/reference/html40/block.html>),

aunque sí puede contener elementos de nivel de línea (que veremos más adelante; puedes encontrar algunos de ellos en <http://htmlhelp.com/reference/html40/inline.html>). Otra cuestión importante a la hora de crear tu página web es que la misma debe estar siempre formada por una serie de "contenedores", los cuales son elementos de nivel de bloque (<http://webdesign.about.com/od/cssglossary/g/bldefcontainer.htm>). Como conclusión, es importante que tengas cuidado de que cualquier fragmento de tu código esté siempre incluido dentro de un bloque p, H1...H6, etc. Por su parte, <br> es un elemento de nivel de línea (podrá ser utilizado dentro de bloques o de otros elementos de línea). Su utilidad es introducir un salto de línea (<http://www.w3.org/TR/html401/struct/text.html#edef-BR>) y tiene prohibida la introducción de tag de cierre.

Con los anteriores tags p y br o haciendo uso de algunos de los anteriores, completa ahora todos los campos que has incluido en tus datos personales ("Nombre y apellidos", "Dirección"... "Intereses").

Por último, vamos a ver cómo podemos incluir enlaces dentro de una página web. Para poder introducir un enlace a tu propio código hacen falta dos pasos:

1. Definir el lugar al que va a apuntar nuestro enlace. Para ello debemos definir un "ancla" (del inglés "anchor") en el fragmento de código al que queremos apuntar. La sintaxis usada es <a name="datos\_personales"> Fragmento de código al que pretendo apuntar</a>. En este caso, el identificador "datos\_personales" será válido dentro de toda la página web.

2. Ahora debemos seleccionar el fragmento de nuestro código HTML que apuntará hacia el "ancla" anterior. En nuestro caso, será el selector correspondiente de la lista de nuestro índice. La sintaxis para enlazar el fragmento de código (en este caso, el elemento de la lista) con el ancla es <a href="#datos\_personales">Datos personales</a>. Observa la almohadilla, que indica que el ancla forma parte de nuestra propia página. Si quieres enlazar con otra página distinta, la sintaxis sería:

```
<a href="http://www.unirioja.es">Universidad de La Rioja</a>
```

Si quieres enlazar con un punto concreto de otra página, finalmente, combinando las dos posibilidades anteriores, puedes hacerlo como:

```
<a href="http://mi_url#datos_personales">Datos personales</a>
```

O, por ejemplo, si quieres incluir un enlace a tu email:

```
<a href="mailto:mi_direccion@xxxx.xxx">Escríbeme aquí</a>
```

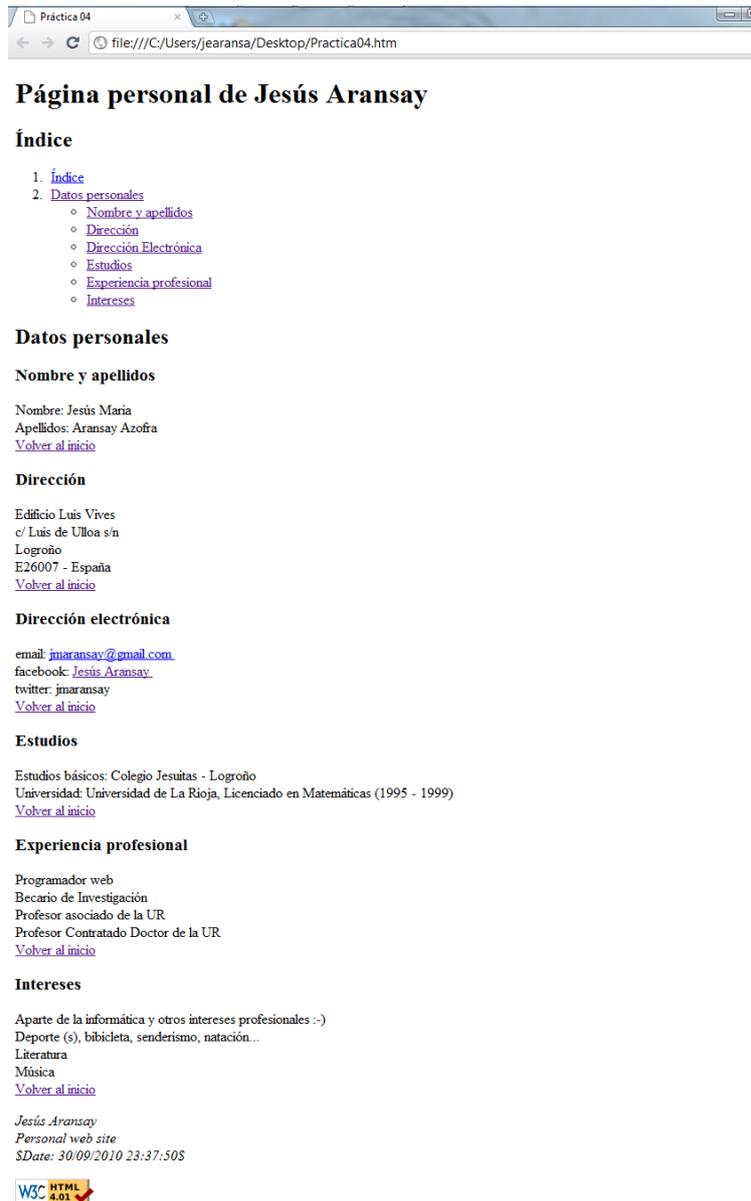
Ten en cuenta que el tag "a" es un elemento de nivel de línea y por tanto puedes utilizarlo dentro de otros tags de bloque o línea, pero, a priori, no se puede meter un elemento de nivel de bloque en un tag a.

Completa tu página web con enlaces a todas las secciones y subsecciones que contiene desde el índice. También debes crear un ancla en el texto superior de la página ("Página personal de..."), a la que se apunte desde cada sección de la misma.

- `<ADDRESS>...</ADDRESS>` suele incluirse al final de nuestra página web, y delimita una región que generalmente incluye información de contacto del autor de la misma (<http://www.w3.org/TR/html401/struct/global.html#edef-ADDRESS>). Es un tag de nivel de bloque. Crea tu propio tag ADDRESS e incluye información relativa al autor de la página y la forma de contactarlo.

Una vez completada la página, y comprobado que el aspecto de la misma es satisfactorio, debes validar la misma con respecto al DTD HTML 4.01 Strict. Esto lo puedes hacer en la página web <http://validator.w3.org/> o también directamente desde los menús de Notepad++ (TextFX – TextFX Convert – Submit to W3C HTML Validator). Si la página pasa la validación, incluye el icono correspondiente.

Para terminar, debes subir la página a tu directorio web en belenus (por FTP) y enlazarla desde tu página "index.htm" para que se pueda acceder a la misma directamente.



Aspecto aproximado que debería tener la página “curri\_vitae.htm”

### Parte 3. Tablas en html

Antes de entrar a explicar en más detalle los tags que permiten definir tablas y formularios vamos a explicar alguna propiedad más de los tags HTML que nos permitan entender mejor lo que haremos más adelante. En general, cualquier tag HTML (p. ej. <a>...</a>) tiene asociados dos tipos de componentes (diferentes para cada tipo de tag, y que son opcionales):

1. Atributos: los atributos nos permiten definir características adicionales de los tags. Su sintaxis de uso es:

```
<tag atributo1="valor1" atributo2="valor2">...</tag>
```

Por ejemplo, si recuperamos el tag <a>...</a> que nos permitía definir orígenes y destinos de vínculos (anclas o enlaces), podemos

ver que en su lista de atributos (<http://html.conclase.net/w3c/html401-es/struct/links.html#edef-A>) contiene algunos de los siguientes:

- `name`: asigna un nombre, de modo que el mismo pueda actuar como destino de un vínculo (tal y como hicimos en la práctica anterior).
- `href`: especifica la localización de un sitio en la web (o en nuestra propia página), enlazando así el elemento actual con dicho sitio.

Puedes encontrar la lista exhaustiva y sus funciones en el enlace anterior.

2. Elementos: los elementos de un tag son aquel conjunto de tags que podemos encontrar dentro del mismo. Por ejemplo, dentro de un tag `<li>...</li>` no podemos encontrar un tag `<body>`, porque el navegador no sabría interpretarlo de forma correcta. Para conocer los tags que pueden ser elementos de otro, puedes consultar la especificación de HTML 4.01. Por ejemplo, dentro de una lista (`<ol>...</ol>` o `<ul>...</ul>`), consultando <http://html.conclase.net/w3c/html401-es/struct/lists.html#h-10.2> podrás comprobar que los elementos disponibles son:

- `<li>`: Una o más copias del atributo `<li>`.

Por lo tanto, dentro de una lista, el único elemento que podemos encontrar son `<li>`. Dentro de cada `<li>` podremos encontrar los elementos que permita el mismo. ¿Cuáles son? (Verás que los mismos están definidos como "flow", y pulsando el enlace puedes comprobar que esto incluye cualquier elemento de bloque, como H1...H6, p, ol, ul, table... o de línea como sub, sup, span, br...)

Con las ideas anteriores intentaremos presentar ahora cómo construir tablas y formularios en HTML, basándonos tanto en los atributos como los elementos que pueden contener ambas.

Las tablas en HTML son una forma de organizar la información para que la misma se muestre por medio de filas y columnas. Son extremadamente útiles y permiten conseguir efectos que con otros elementos de nivel de bloque (H1...H6, p, ol, ul...) no se podrían conseguir. Eso ha llevado a que incluso muchos desarrolladores web las utilicen para dar formato a sus sitios web, aunque esta práctica está desaconsejada incluso por la especificación estándar de HTML 4.01 (Capítulo 7, Tablas): "No deberían usarse tablas con la única finalidad de organizar la presentación de los contenidos de un documento (es decir, de crear el "layout"), ya que esto puede ocasionar problemas cuando se represente en un medio no visual. Además, al incluir gráficos, estas tablas pueden forzar a los usuarios a hacer desplazar horizontalmente la pantalla para ver una tabla diseñada en un sistema con una pantalla más grande. Para minimizar estos problemas, los autores deberían usar hojas de estilo en lugar de tablas para organizar la presentación."

Vamos a ver cuáles son los principales "tags" que pueden ser utilizados en el desarrollo de una tabla en HTML:

- `<TABLE>...</TABLE>`: Puedes encontrar su especificación en <http://html.conclase.net/w3c/html401-es/struct/tables.html#edef-TABLE>. El tag table marca el principio y el final del contenido de la tabla (dentro del mismo se incluirá el título de la misma, las cabeceras, el pie o cualquier información que contenga la misma).

Atributos más relevantes del tag table (por tanto, irán situados como `<table atributo1="value" atributo2 ="value" ... >`):

- `summary`: resumen del contenido de la tabla, pensado para dispositivos (navegadores) que se presentan en medios no visuales (para invidentes...). Muchos de los navegadores convencionales no lo muestran de manera visual, pero es conveniente incluirlo.
- `width`: anchura de la tabla. La anchura se puede especificar por medio de un valor absoluto en píxeles (`width="600"`), o un valor porcentual con respecto a la anchura de la ventana de visualización (`width="75%"`). De las dos opciones, la segunda es más recomendable porque permite al navegador redimensionar las tablas dinámicamente. Si no ponemos un valor, el navegador tomará uno por defecto.
- `border`: permite definir la anchura del marco que rodea a una tabla (y las celdas de la misma). Se mide en píxeles, y se puede especificar como `border="5"` (borde de 5 píxeles).
- `cellspacing`: permite definir la distancia que debe haber entre los bordes de la tabla y las celdas de la misma, así como entre las celdas de la tabla. Por ejemplo, `cellspacing="2"`, definiría 2 píxeles de distancia.
- `cellpadding`: permite definir la distancia entre el borde de las celdas de la tabla y el contenido de las mismas. También se define, como `cellspacing`, en píxeles (o en porcentaje, aunque su significado es menos claro). Puedes encontrar una representación gráfica del significado de `cellpadding` y `cellspacing` en <http://html.conclase.net/w3c/html401-es/struct/tables.html#h-11.3.3>.

Hay algunos otros atributos que puedes encontrar en <http://html.conclase.net/w3c/html401-es/struct/tables.html#edef-TABLE> que aquí no detallaremos.

Elementos propios del tag table (irán todos ellos situados dentro del bloque `<TABLE>...</TABLE>`):

- `<caption>...</caption>`: el elemento `caption` debe definir el título, contenido o descripción de la tabla. Los agentes visuales (navegadores) lo mostrarán por defecto encima de la tabla.
- `<colgroup>...</colgroup>`: permite definir grupos de columnas que vayan a tener unas mismas propiedades (especificadas en los atributos de `colgroup`). Los más representativos son:
  - `span`: define el número de columnas que comprenden el grupo de columnas afectado por `colgroup`.

- o `width`: permite definir la anchura por defecto para las columnas afectadas por el `colgroup`.

Por ejemplo, el tag `<COLGROUP span = "6" width = "16%"></COLGROUP>` define un grupo de 6 columnas, la anchura de las cuales será cada una del 16% del total (serán todas de la misma anchura).

- `<col>`: el elemento `col` tiene atributos similares a los de `colgroup`. Además, lo podemos usar dentro de `colgroup` para definir subgrupos de columnas que compartan características.

Pasamos ahora a ver los tags que nos van a permitir definir las distintas filas de la tabla (no es obligatorio usar los tres tipos en cada tabla):

- `<THEAD>...</THEAD>`: Define las cabeceras de la tabla. No es obligatorio incluirla. La misma debe estar compuesta por, al menos, una fila, que será un elemento de la forma `<TR>...</TR>` (veremos más adelante cómo definir las filas).
- `<TBODY>...</TBODY>`: Define el cuerpo de la tabla. Al igual que `thead`, debe estar compuesta por, al menos, una fila, de la forma `<TR>...</TR>`.
- `<TFOOT>...</TFOOT>`: Define el pie de la tabla. Al igual que los anteriores estará compuesto por filas de la forma `<TR>...</TR>`.

El único elemento que podemos encontrar en los anteriores tags `thead`, `tbody` y `tfoot` es:

- `<TR>...</TR>`: permite definir las distintas filas (`tr = table row`) de la tabla. Cada fila, a su vez, sólo puede estar compuesta por los siguientes elementos:
  - o `<TH>...</TH>`: celda de encabezado. Generalmente mostrada en negrita. El tag de cierre es opcional.
  - o `<TD>...</TD>`: celda de datos. El tag de cierre es opcional.

Dentro de los tags `td` y `th` podemos encontrar cualquier elemento de nivel de línea (`a`, `strong`, `em`...) o de bloque (`p`, `ol`, `ul`, `table`...).

Los tags `th` y `td` tienen varios atributos. De ellos, los que más nos interesan son los siguientes:

- `rowspan`: permite hacer que una celda ocupe varias filas (`<TD rowspan="3">` haría que la celda definida ocupara 3 filas).
- `colspan`: permite hacer que una celda ocupe varias columnas (`<TD colspan="2">` haría que la celda definida ocupara 2 columnas).

Ambos tags se pueden usar de forma conjunta para definir una celda que ocupe varias celdas y columnas.

- `align`: define la alineación del contenido de la celda con respecto a la misma. Puede tomar, entre otros, los valores "left", "center" y "right".

- `valign`: define la alineación vertical del contenido de la celda con respecto a la misma. Sus valores pueden ser "top", "middle", "bottom" y "baseline".

Los dos atributos anteriores también se pueden aplicar a `tr`, afectando a todas las celdas de una fila, a `thead`, `tbody` y `tfoot`, afectando a todas las celdas de esas áreas, y a `colgroup` y `col`, afectando a todas las celdas de esas columnas.

En la página web <http://www.htmlquick.com/es/tutorials/tables.html> puedes encontrar varios ejemplos de tablas que hacen uso de los anteriores elementos.

16. Crea un fichero "horario.htm" (incluye en el mismo cabeceras y metas como hemos hecho en los anteriores ficheros). Haz uso de los anteriores ejemplos para realizar la tabla HTML correspondiente a tu horario del segundo curso, segundo cuatrimestre de GIA. Puedes encontrar el horario en [http://www.unirioja.es/facultades\\_escuelas/fceai/horarios/horarios\\_11\\_12/hor2\\_2grado\\_iagr\\_11.pdf](http://www.unirioja.es/facultades_escuelas/fceai/horarios/horarios_11_12/hor2_2grado_iagr_11.pdf). Selecciona el grupo de prácticas, el grupo reducido, laboratorios y asignaturas (de primer o segundo curso) que te corresponda. Recuerda validarla al terminar, e incluir el icono correspondiente. Por ejemplo, la tabla HTML, para una persona de GIA que estuviera en todos los grupos y laboratorios debería quedar como:

Periodo Lectivo: Curso 2011-12 Semestre 2  
 Titulación: Grado en Ingeniería Agrícola

HORARIO DEL CURSO I - GRUPO I.1

Horario	Lunes	Martes	Miércoles	Jueves	Viernes
9:00 10:00	Botánica GRUPO GRANDE J. MARTINEZ L. Microscopia L - 233	Cultivos GRUPO GRANDE M. MARTINEZ CT - Aula 103	Botánica GRUPO GRANDE J. MARTINEZ L. Microscopia L - 233 (S. Impar)	Botánica GRUPO GRANDE J. MARTINEZ L. Microscopia L - 233	
10:00 11:00	Cultivos GRUPO GRANDE F. PAEZ DE CT - Aula 103	Electrotecnia, máquinas y motores GRUPO GRANDE I. BARRIO CT - Aula 103	Informática GRUPO GRANDE J. ARANSAY CT - Aula 103 / Aula 201 (S. Impar)	Informática GRUPO GRANDE J. ARANSAY CT - Aula 103 / Aula 201	
11:00 12:00	Informática GRUPO GRANDE J. ARANSAY CT - Aula 103	Topografía, Cartografía y SIG GRUPO GRANDE J. SANTAMARIA CT - Aula 103	Topografía, Cartografía y SIG GRUPO LABO J. SANTAMARIA Departamental - L203 Departamental - Aula Info 1 (S. Par)	Electrotecnia, máquinas y motores GRUPO GRANDE F. AZOFRA CT - Aula 103	
12:00 13:00	Informática GRUPO GRANDE J. ARANSAY CT - A - 201	Topografía, Cartografía y SIG GRUPO GRANDE J. SANTAMARIA CT - Aula 103	Cultivos GRUPO GRANDE F. PAEZ CT - Aula 103 (S. Impar)	Topografía, Cartografía y SIG GRUPO GRANDE J. SANTAMARIA CT - Aula 103 (S. Par)	
13:00 14:00	Informática GRUPO GRANDE J. ARANSAY CT - A - 201	Producción Vegetal GRUPO LABO J C. MENENDEZ CT - L - Hortofru	Topografía, Cartografía y SIG GRUPO GRANDE J. SANTAMARIA Departamental - Aula Info 1 (S. Par)	Producción Vegetal GRUPO LABO J C. MENENDEZ CT - L - Hortofru	
14:00 15:00		Producción Vegetal GRUPO LABO J E. PRADO CT - L - Hortofru	Cultivos GRUPO REDU J M. MARTINEZ CT - Aula 103 (S. Par)	Producción Vegetal GRUPO LABO J C. MENENDEZ CT - L - Hortofru	
15:00 16:00	Producción Vegetal GRUPO LABO J E. PRADO CT - L - Hortofru				
16:00 17:00	Producción Vegetal GRUPO LABO J E. PRADO CT - L - Hortofru		Botánica GRUPO GRANDE J. MARTINEZ L. Microscopia L - 233	Electrotecnia, máquinas y motores GRUPO LABO J. BARRIO CT - Lab Maquin	
17:00 18:00			Cultivos GRUPO LABO J F. PAEZ CT - L Hortofru		
18:00 19:00			GRUPO LABO J M. MARTINEZ CT - L Hortofru	Electrotecnia, máquinas y motores GRUPO LABO J. BARRIO CT - Lab Maquin	
19:00 20:00			Botánica GRUPO GRANDE J. MARTINEZ L. Microscopia L - 233		
			Cultivos GRUPO LABO J F. PAEZ CT - L Hortofru		
			GRUPO LABO J M. MARTINEZ CT - L Hortofru		
			Cultivos GRUPO LABO J F. PAEZ CT - L Hortofru		
			GRUPO LABO J M. MARTINEZ CT - L Hortofru		