

TEMA 2 – HTML.

ÍNDICE.....	1
1. Esquema de funcionamiento de un servicio web.....	2
2. Introducción a HTML.....	3
3. Estructura de un documento HTML.....	8
3.1. Identificación SGML	9
3.2. Etiqueta <html>.....	10
3.3. Cabecera <head>	10
3.4. Cuerpo del documento <body>.....	12
3.4.1 Secciones. Etiquetas semánticas.	13
3.4.2 Títulos.....	14
3.4.3 Etiquetas de contenido	15
3.4.4 Elementos generales	16
3.4.5 Etiquetas de textos	16
3.4.6 Etiquetas de control de las fuentes de texto.....	18
3.4.7 Listas	15
3.4.8 Hiperenlaces o hipervínculos	19
3.4.9 Imágenes	21
3.4.10 Mapas sensibles.....	22
3.4.11 Tablas	24
3.4.12 Formularios.....	30
4. Páginas estáticas vs páginas dinámicas	46
5. Lenguajes de scripts	47
6. Bibliografía	49
El Gran Libro De HTML5, CSS3 Y Javascript. Juan Diego Gauchat	49

1. Esquema de funcionamiento de un servicio web

La Web funciona siguiendo el denominado **modelo cliente-servidor**, habitual en las aplicaciones que funcionan en una red: existe un servidor, que es quien presta el servicio, y un cliente, que es quien lo recibe.

Cliente web

El cliente web es un programa con el que el usuario interacciona para solicitar a un servidor web el envío de páginas de información. Estas páginas se transfieren mediante el protocolo HTTP.

Las páginas que se reciben son documentos de texto codificados en lenguaje HTML. El cliente web debe interpretar estos documentos para mostrárselos al usuario en el formato adecuado.

Además, cuando lo que se recibe no es un documento de texto, sino un objeto multimedia (vídeo, sonido, etc.) no reconocido por el cliente web, éste debe activar una aplicación externa capaz de gestionarlo.

Entre los clientes web (también conocidos como visualizadores o navegadores) más usuales están: *Mozilla Firefox*, *Microsoft Internet Explorer*, *Opera*, *Safari* y *Google Chrome*. La mayoría de ellos soportan también otros protocolos, como por ejemplo el FTP (*File Transfer Protocol*), para la transferencia de ficheros.

Servidor web

El servidor web es un programa que está permanentemente escuchando las peticiones de conexión de los clientes.

El servidor funciona de la siguiente manera: si encuentra en su sistema de ficheros el documento HTML solicitado por el cliente, lo envía y cierra la conexión; en caso contrario, envía un código de error que cierra la conexión. El servidor web también se ocupa de controlar los aspectos de seguridad, comprobando si el usuario tiene acceso a los documentos.

El proceso completo, desde que el usuario solicita una página hasta que el cliente web se la muestra con el formato adecuado, es el siguiente:

1. El usuario especifica en el cliente web la dirección (URL) de la página que desea consultar.

2. El cliente establece la conexión con el servidor web (por ejemplo, Apache).
3. El cliente solicita la página deseada.
4. El servidor busca la página que ha sido solicitada en su sistema de ficheros, Si la encuentra, la envía al cliente; en caso contrario, devuelve un código de error. Si la petición necesita la generación dinámica de una página HTML, por ejemplo, por necesitar hacer una consulta a una base de datos, debe haber un intérprete del lenguaje utilizado (p.e. PHP o ASP) en el servidor, que interactúa con un programa que maneja la base de datos y va generando paginas en HTML que se envían al cliente (lo veremos con detenimiento más adelante).
5. El cliente interpreta los códigos HTML y muestra la página al usuario.
6. Se cierra la conexión.

2. Introducción a HTML

Definición:

HTML (*HyperText Markup Language*) es una implementación del estándar SGML (*Standard Generalized Markup Language*). SGML es un estándar internacional para la definición de texto electrónico independiente de dispositivos, sistemas y aplicaciones.

Los autores utilizan un código de formato (en inglés *markup*) en sus documentos para representar información.

Cada lenguaje de formato de documentos definido con SGML se llama aplicación SGML. Una aplicación SGML se caracteriza generalmente por:

1. Una declaración SGML. La declaración SGML especifica qué caracteres y delimitadores pueden aparecer en la aplicación.
2. Una definición del tipo de documento (*document type definition*, DTD) que define la sintaxis de las estructuras de formato. El DTD puede incluir definiciones adicionales, tales como referencias a entidades de caracteres.
3. Una especificación que describe la semántica que se debe conferir al código de formato. Esta especificación también impone restricciones de sintaxis que no pueden expresarse dentro del DTD.

4. Documentos que contienen datos (contenido) y código (*markup*). Cada documento contiene una referencia al DTD que debe usarse para interpretarlo.

HTML es, por tanto, un caso particular de lenguaje de formato de documentos. Veamos un ejemplo de un documento HTML:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>mi primer documento html</title>
</head>
<body>
  <p>¡hola mundo!</p>
</body>
</html>
```

HTML, por lo tanto, es un lenguaje basado en marcas; una marca afecta a un trozo de texto dándole un formato, estructura o significado determinado. Un documento HTML contiene tanto la información que se desea presentar como instrucciones (marcas) para describir su presentación. También podemos decir que es un lenguaje de hipertexto, ya que dentro del documento existen áreas sensibles (hipervínculos) que al activarlas permiten acceder a otros elementos.

Un documento HTML se divide en una **sección de cabecera** (aquí, entre <HEAD> y </HEAD>) y un **cuerpo** (aquí, entre <BODY> y </BODY>). El título del documento aparece en la cabecera (junto con otras informaciones sobre el documento), y el contenido del documento aparece en el cuerpo. El cuerpo de este ejemplo contiene únicamente un párrafo, codificado o marcado como <P>.

Versiones:

- [HTML 3.2](#), W3C Recommendation, 14 January 1997, Dave Raggett, Author
- [HTML 4.01](#), W3C Recommendation, 24 December 1999, Dave Raggett, Arnaud Le Hors, Ian Jacobs, Editors
- [HTML5](#), W3C Recommendation, 28 October 2014.

Para más información:

- Especificaciones html:
<http://www.w3.org/community/webed/wiki/HTML/Specifications>
- HTML5:

- Recomendación [HTML5](#) publicado en Octubre del 2014.
- http://www.w3.org/html/wiki/FAQs#When_can_I_use_HTML5.3F
- <http://es.wikipedia.org/wiki/HTML5>

Editores:

¿Dónde hay que editar el código fuente? Cualquier editor de texto sin formato vale, eso sí, los ficheros deben tener extensión html (aunque se pueden encontrar documentos html muy antiguos con extensión htm). En Windows, por ejemplo, bastaría con el Bloc de notas.

Existen editores de html más avanzados que, por ejemplo, codifican las etiquetas con algún sistema de colores. Dos de los más utilizados son el Sublime Text, el NOTEPAD++PORTABLE y visual Studio Code.

Navegadores:

El navegador es el software que interpreta el lenguaje HTML. Existen muchos navegadores. Una página web que usa HTML estándar es accesible desde cualquier navegador. Pero existen diferencias entre unos navegadores y otros a la hora de interpretar determinado código no estándar HTML, por eso lo mejor es ajustarse al estándar y no usar etiquetas que hacen cosas muy chulas pero sólo valen para un determinado navegador. Nuestro objetivo debe ser que cualquier usuario pueda ver bien (tal y como nosotros queremos) nuestra página web. Es absurdo ver mensajes que digan "Página optimizada para..." y ahí un navegador o resolución de pantalla (¿vas a tener que cambiar la resolución o el navegador sólo para ver una página?). Tus webs tienen que ser *optimizadas* para verse con cualquier navegador.

URL (Uniform Resource Locator):

La URL define objetos en una red Internet. En ella se contienen datos sobre:

- El tipo de objeto (objetos asociados con alguno de los protocolos o servicios disponibles en Internet: ftp, http, mailto, telnet, etc.).
- El nodo de la red en que se encuentra dicho objeto.
- El fichero físico que contiene el objeto.

Estructura: ***Servicio://[host]:[puerto]/[path del fichero]***

Ej: <http://microsoft.com/download/aspdock.zip>

Si se omite el puerto se tomaría el válido por defecto para el protocolo o servicio utilizado (ej: puerto 80 para servicios web).

Tipos: absolutas y relativas:

- Absolutas: especifican un path completo.
<http://www.sobreas.com/download/aspddoc.zip>
- Relativas: especifican un path relativo a la url del documento.
imagenes/dibujo1.gif

Sintaxis:

Generalidades:

- Son válidos todos los caracteres incluidos en Unicode / ISO 10646.
- El formato es libre. El formato introducido en el fichero fuente (saltos de línea, líneas en blanco, etc.) es irrelevante para el formato final del documento.
- Determinados caracteres tienen un significado especial:
 - < Marca el comienzo de una etiqueta.
 - > Marca el final de una etiqueta.
 - & Marca el comienzo de una referencia a entidad.

Estos caracteres, en caso de que sea necesario utilizarlos, se sustituyen por el nombre de la entidad que los representa en el repertorio ISO Latín 1 (ISO 8859-1):

& se escribe &

< se escribe <

> se escribe >

espacio en blanco se escribe

Los nombres de las entidades están compuestos por el signo &, luego el nombre de la entidad y al final ;. Si se trata de un número de entidad en lugar de un nombre, añadiremos # después del ampersand (&) y a continuación el número de la entidad – si se expresa en hexadecimal irá precedido de una x - y al final ";".

Por ejemplo, para mostrar el signo "<" tenemos tres posibilidades:

Por su nombre de entidad **<**

Indicando su correspondencia decimal en la tabla ASCII **<**

Indicando su correspondencia hexadecimal en la tabla ASCII **<**

El uso más común de los caracteres especiales es el espacio en blanco.

Si en un texto figuran cinco espacios en blanco seguidos, HTML automáticamente borra cuatro.

Para incluir espacios en blanco usamos " " y HTML los dejará en su lugar.

Otro uso muy frecuente es el de insertar acentos en el código html por medio de los números de las entidades: al principio, en algunos navegadores las letras acentuadas y algunos caracteres especiales como la "ñ" no se visualizaban correctamente, por lo cual debían ser sustituidos por la referencia a su entidad.

La siguiente tabla muestra una lista de estos caracteres especiales:

• signo <	correspondencia en HTML	<code>&lt;</code>
• signo >	correspondencia en HTML	<code>&gt;</code>
• signo &	correspondencia en HTML	<code>&amp;</code>
• signo "	correspondencia en HTML	<code>&quot;</code>
• espacio en blanco	correspondencia en HTML	<code>&nbsp;</code>
• Espacio ASCII	correspondencia en HTML	<code>&#x0020;</code>
• Tabulador ASCII	correspondencia en HTML	<code>&#x0009;</code>
• Avance de página ASCII	correspondencia en HTML	<code>&#x000C;</code>
• Espacio de anchura cero	correspondencia en HTML	<code>&#x200B;</code>
• Retorno de carro	correspondencia en HTML	<code>&#x000D;</code>
• Á, á	correspondencia en HTML	<code>&Aacute;</code> <code>&aacute;</code>
• É, é	correspondencia en HTML	<code>&Eacute;</code> <code>&eacute;</code>
• Í, í	correspondencia en HTML	<code>&Iacute;</code> <code>&iacute;</code>
• Ó, ó	correspondencia en HTML	<code>&Oacute;</code> <code>&oacute;</code>
• Ú, ú	correspondencia en HTML	<code>&Uacute;</code> <code>&uacute;</code>
• Ñ, ñ	correspondencia en HTML	<code>&Ntilde;</code> <code>&ntilde;</code>

Para más información: <http://ascii.cl/es/codigos-html.htm>

Etiquetas (*tags*):

- **Son los textos que delimitan los distintos elementos que componen un documento.** Los elementos de HTML suelen estar formados por una marca de inicio < >, un contenido y una marca de final </ >, aunque hay algunos elementos especiales que sólo tienen marca de inicio y se llaman **elementos vacíos**.
- No son sensibles a mayúsculas y minúsculas (*aunque **es aconsejable escribir todo en minúsculas porque el próximo estándar así lo exigirá***).

- Hay dos tipos de etiquetas: etiquetas de comienzo y etiquetas de final de elemento. La mayoría de los identificadores necesitan ambas etiquetas aunque unos pocos no necesitan la de final.

1) **Etiquetas de comienzo de elemento.** Delimitadas por los caracteres "<" y ">".

<identificador>

Ej: <head>

2) **Etiquetas de final de elemento.** Delimitadas por los caracteres "</" y ">".

</identificador>

Ej: </head>

- Muchas etiquetas tienen atributos, los cuales modifican el funcionamiento de la misma. Sintaxis:

<identificador [atributos]>

Estructura general de un atributo: *Literal=valor*

Normalmente, el valor de un atributo es una cadena de caracteres entre dobles comillas. Dentro de ella no se pueden poner los siguientes caracteres " , > , & . Si es necesario ponerlos se sustituyen por " > y &

Ejemplo:

```
<a href="https://www.elmundo.es/" title="El mundo.es">
...
</a>
```

Comentarios:

- Texto introducido en un documento que no aparece en el formato final.

<!-- comentario -->

*Ej: <!-- Esto no aparecerá en la página web
aunque ocupe varias líneas -->*

1. Estructura de un documento HTML

Un documento HTML consta de las siguientes partes:

1. Identificación SGML
2. Una etiqueta de comienzo de documento <html>

3. Cabecera (iniciada por la etiqueta <head> y cerrada por </head>)
4. Cuerpo del documento (iniciada por la etiqueta <body> y cerrada por </body>)
5. Una etiqueta de fin de documento </html>

```
<!DOCTYPE html>
<html lang="es">
  <head>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

A continuación veremos cada una de estas secciones.

3. Identificación SGML

Permite identificar la DTD (la estructura de etiquetas del documento, los atributos que pueden tener las etiquetas, los eventos, etc.) adecuada para procesar el documento.

Hasta HTML 4 se especificaban tres DTDs, (ver anexo HTML4). Con HTML 5 el doctype en es más sencillo:

```
<!DOCTYPE html>
```

Podemos validar un documento HTML *online* desde la página oficial de W3C <http://validator.w3.org/>. Aquí encontraremos 3 opciones: validar una web con URL, validar un fichero o validar código HTML copiado directamente a la web.

Hoy en día, algunos navegadores disponen de complementos que ayudan a validar código y a desarrollar. *Por ejemplo Firefox* y Chrome. En Firefox tenemos el complemento **HTML validator**: se descarga de Internet y se instala a través del menú Herramientas/Complementos/Extensiones. Una vez instalado aparece un botón en la esquina inferior derecha del navegador que nos permitirá validar el código HTML asociado a la página. Otro complemento interesante de Firefox es Firebug.

4. Etiqueta <html>

Delimita y engloba toda la página web, que consta de cabecera y cuerpo.

5. Cabecera <head>

Contiene información general acerca del propio documento y su visualización.

Se identifica con la etiqueta **<head>** y finaliza por tanto con **</head>**.

En su ámbito se pueden emplear diferentes elementos referenciados por sus etiquetas, los más relevantes son:

- **<title>** [cadena de caracteres] **</title>**

Da título al documento, en la mayoría de los navegadores se visualiza en la barra de título.

- **<base href="URL">**

Indica la localización de los ficheros, gráficos, sonidos, etc. a los que se hace referencia en nuestra página web. Si no se incluye esta directiva, el navegador entiende que los elementos se encuentran en el mismo lugar que nuestra página, por lo que esta etiqueta suele usarse cuando los ficheros están en otro servidor.

`<base href="http://www.miweb.com/archivos/">`

- **<link >**

Especifica la relación entre el documento actual y un recurso externo. Los usos posibles de este elemento incluyen la definición de un marco relacional para navegación.

Este elemento es más frecuentemente usado para enlazar hojas de estilos.

`<link rel="stylesheet" href="styles.css">`

- **<meta http-equiv="VALOR1" | name="VALOR2" | charset="VALOR3" content=VALOR4>**

Las etiquetas <meta> o "*meta tags*" proporcionan información sobre el documento HTML. Son identificadores ocultos, es decir instrucciones especiales del lenguaje HTML que no son mostradas directamente en el navegador, pero que pueden ser utilizadas por navegadores, motores de búsqueda u otros servicios web.

Existen **tres tipos** de *meta tags*, con distintas funciones:

a) LAS META NAMES: se utilizan para optimizar el resultado en los motores de búsqueda. Contienen información referida al documento HTML (autor, fecha de modificación, descripción, etc). El formato es:

`<meta name="valor_name" content="valor_content">`

El atributo "name" puede tomar los siguientes valores entre otros: *author* (autor), *description* (descripción), *keywords* (palabras clave), *lang* (idioma), etc.

Ejemplos:

`<meta name="author" content="Profesor">`

`<meta name="lang" content="es">`

`<meta name="description" content="Página sobre Palencia">`

`<meta name="keywords" content="Palencia, Castilla, monumentos, ciudad, castellana, carrión">`

`<meta name="viewport" content="width=device-width, initial-scale=1.0">`

b) LAS HTTP-EQUIV: este atributo se utiliza para especificar información de las cabeceras http que regulan el diálogo entre el servidor y el navegador. Los valores establecidos por este metadato pueden ser utilizados por el servidor al entregar la página al navegador del usuario, y son utilizadas para refinar la información y dar instrucciones al navegador que las está leyendo. El formato es:

`<meta http-equiv="valor_http-equiv" content="valor_content">`

El atributo http-equiv puede tomar los siguientes valores entre otros: *default-style* (el estilo preferido), *refresh* (intervalo de refresco del documento).

Ejemplos:

Indica la hoja de estilos por defecto o el estilo por defecto a utilizar -->

`<meta http-equiv="refresh" content="3">`

<!-- La página se recarga cada tres segundos-->

c) CHARSET: especifica la codificación de caracteres utilizada. En HTML5 la codificación por defecto si no se especifica nada es UTF-8. No obstante, es el navegador el que finalmente decide la codificación a utilizar. Para evitar un visionado incorrecto en algunos navegadores o navegadores mal configurados, **es recomendable incluir siempre la etiqueta meta charset.**

```
<meta charset="UTF-8">
```

6. Cuerpo del documento <body>

Es el contenedor de la información propia del documento.

<body>

Veamos algún ejemplo:

1) Cuerpo de documento:

```
<body >
    .....
</body>
```

Veamos ahora el aspecto básico de la mínima página web:

```
<html lang="es">
<head>
    <meta name="author" content="Pepito Pérez">
    <meta name="lang" content="es">
    <meta name="description" content="Página sobre Palencia">
    <meta name="keywords" content="Palencia, Castilla, monumentos">
    <meta http-equiv="Content-Type" content="text/html";
        charset="iso-8859-1"> <!-- codificación española -->
    <title>Título de la página web que sale en el navegador</title>
</head>
<body>
    <!-- Aquí es donde va el contenido del documento, es decir,
    lo que se verá en el navegador. Ahora vamos a aprender a
    darle formato -->
</body>
</html>
```

En el ámbito del cuerpo del documento se pueden emplear distintos elementos para dar formato al contenido del mismo (párrafos, listas tablas, formularios, ...). Esto es lo que vamos a explicar en el siguiente apartado.

Elementos del cuerpo del documento:

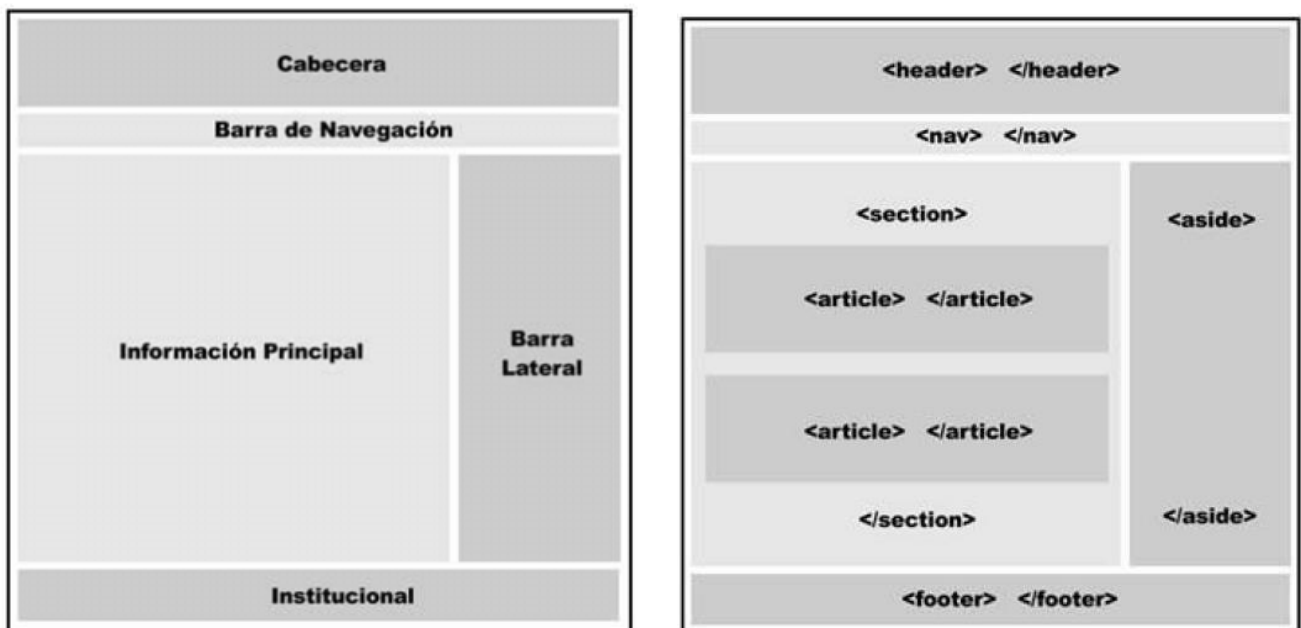
Secciones. Etiquetas semánticas.

La idea de HTML5 es que los elementos HTML, más que para formato, sirvan para dar valor semántico al contenido. Es decir, para indicar qué tipo de contenido es. Con las siguientes etiquetas se marca la semántica y con las hojas de estilo CSS que veremos en la siguiente unidad didáctica le damos formato a esa semántica. Con todo ello se consigue que la estructura de la web sea más coherente y fácil de entender.

En la imagen de la izquierda tenemos un diseño común de cualquier sitio web actual, con las diferentes secciones que lo componen.

En las versiones anteriores a HTML5 se **solía utilizar la etiqueta <div>** para realizar las divisiones de una página web, pero **actualmente es recomendado utilizar los elementos semánticos** ya que describen claramente su propósito.

En la imagen de la derecha se muestran los diferentes elementos de HTML5 para cada una de las secciones.



<header></header>

Cabecera de la página o de una sección. Por lo general siempre se define un header principal donde se incluye el logo o el nombre del sitio, pero además se pueden (y deben) definir otros elementos header dentro de los elementos section o article.

<nav></nav>

Ofrece ayuda para la navegación, usualmente menús principales o grandes bloques de enlaces. Debe ser utilizado sólo para la navegación principal del sitio y no para enlaces externos.

Es un elemento muy versátil, normalmente aparece insertado dentro de un elemento header o footer pero puede aparecer en cualquier otra parte del cuerpo siempre que mantenga su finalidad.

<section></section>

Contiene la información más relevante del documento y puede ser encontrada en diferentes formas (por ejemplo, dividida en varios bloques o columnas).

<aside></aside>

Puede contener datos relacionados con la información principal pero que no son relevantes o igual de importantes, como aclaraciones del contenido o referencias. Y también puede contener información no directamente relacionada con el contenido que lo rodea, como elementos publicitarios.

<footer></footer>

Contiene información general sobre el autor, el copyright, la fecha de última modificación o la organización.

<article></article>

Define contenido autónomo o independiente, con la intención de ser reutilizado de modo aislado, es decir, información que puede ser entendida como un todo de forma íntegra. Tiene sentido dentro del elemento <section> e incluso puede aparecer de forma independiente.

<main></main>

Determina el contenido principal del documento. Solo puede existir un elemento <main> en el documento.

<address></address> Dirección, información sobre el autor del documento, dirección, etc.

```
<address>texto</address>
```

Es muy importante tener en cuenta que estas etiquetas no indican su posición en la página Web, sino su valor semántico.

3.4.1 Títulos

Existen 6 niveles de cabeceras:

- Cabecera de nivel 1: `<h1>Texto de la cabecera</h1>`
- Cabecera de nivel 2: `<h2>Texto de la cabecera</h2>`
- Cabecera de nivel 3: `<h3>Texto de la cabecera</h3>`
- Cabecera de nivel 4: `<h4>Texto de la cabecera</h4>`
- Cabecera de nivel 5: `<h5>Texto de la cabecera</h5>`
- Cabecera de nivel 6: `<h6>Texto de la cabecera</h6>`

El formato en que se visualizan las cabeceras depende de su nivel, variando:

- Tamaño de la letra
- Tipo de resaltado
- Líneas a saltar antes y después del texto.

Ejemplos:

```
<h1>Cabecera de nivel 1</h1>
```

```
<h3>Cabecera de nivel 3</h3>
```

3.4.2 Etiquetas de contenido

Definen la estructura de un bloque:

- **`<p></p>`**: Párrafo

Etiqueta de párrafo normal. Hace que el texto que está contenido en esta etiqueta sea considerado un párrafo. A ese texto se le colocará un salto de párrafo al final. Los saltos de línea y los espacios blancos seguidos introducidos no se representan (se comprimen en un único espacio). La longitud de las líneas viene definida por el tamaño de la ventana del navegador. Para provocar un salto de línea hay que usar la etiqueta `
`. La etiqueta de finalización `</p>` no es obligatoria, aunque conviene ponerla.

```
<p>texto</p>
```

- **`<pre></pre>`**: Texto con formato previo

Conjunto de texto que se muestra como se introdujo en el formato original, incluyendo los saltos de línea y los espacios en blanco.

Se recomienda no utilizarlo en la medida de lo posible.

```
<pre>texto</pre>
```

- **`<blockquote></blockquote>`**: Anotación

Sirve para escribir una cita, el texto se presenta indentado y en un formato distinto al del párrafo normal. cita en bloque .

Sus atributos son:

- **title** Texto informativo o título del elemento. Suele mostrarse a modo de "tool tip".
- **lang** Información sobre el idioma del contenido del elemento y del valor de sus atributos.

```
<blockquote>texto</blockquote>
```

- **<div></div>**: Permite agrupar varios bloques de texto en uno solo, heredando todos ellos la alineación especificada mediante el parámetro **align**=(left|center|right|justify). Se trata de una etiqueta muy potente que demuestra su potencial cuando se usa con las hojas de estilo.

```
<div>texto</div>
```

- **<figure></figure>**: Establece una figura ilustrada como parte del documento.
- **<figcaption></figcaption>**: Representa la leyenda de una figura.
- **Listas** : las estudiaremos mas adelante, en el apartado 7

3.4.3 Elementos generales

- **
** Salto de línea. Fuerza que se parta una línea de texto independientemente del formato en que se esté trabajando. Se puede usar sin la barra final, pero en el próximo estándar será obligatorio, así es que es mejor ponerla.

```
<br>        mejor: <br/>
```

Extensiones de la etiqueta **
**: su uso está indicado para romper la secuencia de texto cuando se sitúa este alrededor de una imagen

```
propiedad de css clear:left; Busca el primer margen libre a la izquierda.
propiedad de css clear:right; Busca el primer margen libre a la derecha.
propiedad de css clear:all; Busca el primer margen libre a ambos lados.
```

- **<hr>** Línea horizontal. Representa un cambio temático entre párrafos.

```
<hr/>
```

3.4.4 Etiquetas de textos

Se utilizan para enfatizar o resaltar una zona del texto. Todas las etiquetas tienen el mismo formato:

`<etiqueta>texto</etiqueta>`

Dos tipos:

1. Estilos lógicos: asociados a distintos tipos de información.

- **``**: Superíndice
- **``**: Subíndice
- **`<cite></cite>`**: Contiene una cita o proporciona un enlace al documento original o fuente. Una dirección URI. Por defecto: lo fija el navegador
- **`<dfn></dfn>`**: Indica que aquí es donde se define el término encerrado.
- **`<code></code>`**: Designa un fragmento de código de computadora.
- **`<samp></samp>`**: Designa una muestra de la salida de un programa, script, etc.
- **`<kbd></kbd>`**: Indica texto que debe ser introducido por el usuario o la tecla que debe pulsar el usuario..
- **`<var></var>`**: Indica que el texto es una variable o un argumento de un programa.
- **`<abbr></abbr>`**: Indica una forma abreviada (p.ej., WWW, HTTP, URI, Mass., etc.). Envuelve un texto que al pasar el puntero por encima despliega un tooltip. El contenido del tooltip se detalla mediante el atributo **title**.
- **`<q></q>`**: el contenido es una cita textual de una frase
- **``**: el texto afectado ha sido eliminado en esta versión del documento
- **`<ins></ins>`**: el texto afectado ha sido añadido en esta versión del documento
- **`<time></time>`**: define una fecha y/o hora.
- **`<small></small>`**: Representa un *comentario aparte*, es decir, textos como un descargo de responsabilidad o una nota de derechos de autoría, que no son esenciales para la comprensión del documento.

- **<mark></mark>**: Representa texto resaltado con propósitos de *referencia*, es decir por su relevancia en otro contexto.
- **** para resaltar el texto ****

Los elementos de frase tienen significados particulares en documentos técnicos.

Ejemplos:

```
H<sub>2</sub>O  
E = mc<sup>2</sup>
```

```
Como dijo <cite>Harry S. Truman</cite>,  
<q lang="en-us">The buck stops here.</q>
```

```
Se puede encontrar más información en <cite>[ISO-0000]</cite>.
```

```
Por favor, utilice el siguiente número de referencia en nuestra  
correspondencia futura: <strong>1-234-55</strong>
```

2. Estilos físicos: asociados a distintos tipos de letra. Estos ya no se usan se configuran mediante CSS

3.4.5 Etiquetas de control de las fuentes de texto

- ****: Especifica texto en línea. Sirve para aplicar estilo al texto o agrupar elementos en línea.

3.4.6 Hiperenlaces o hipervínculos

La etiqueta `<a>` permite crear enlaces a otras páginas u objetos.

Atributos:

- `href="URL"`

Identifica el destino del enlace. La URL puede ser absoluta (<http://www.ya.com>), relativa (`pagina3.html`), de salto hacia un marcador interno (`#marca1`) o una mezcla entre ambas (`pagina3.html#marca1`)

Ejemplo: `La Web de Microsoft`

- `name="nombre"`

Coloca un marcador con ese identificador en la posición en la que se encuentre la etiqueta `<a>`, permitiendo hacer un enlace a otro lugar en nuestra página.

``

Ejemplo:

`<!-- Establecer un marcador dentro de una página -->`

``

`<!-- En otro lugar del documento podemos poner un enlace a esa marca así: -->`

`Documentación pública`

`<!-- En otra página web podemos poner un enlace a esa marca así: (suponemos que el marcador se encuentra en un fichero llamado página3.html) -->`

`Documentación pública`

- `target="destino"`

Indica en qué parte de la ventana se muestra el enlace. Es indispensable para utilizar con los marcos. Lo veremos más adelante.

- `title="descripción del enlace"`: es obligatorio cuando se codifica html accesible porque esta descripción es la que el navegador de voz canta a su usuario, normalmente su valor coincide con el texto del enlace

Veamos un ejemplo de hipervínculos:

```
<html lang="es">
<head>
  <title>Ejemplo hipervínculos</title>
</head>

<body>
  <h1><a name="arriba">Página de enlaces</a></h1>
  <a href="#abajo">Ir abajo</a><br/>

  <a href="ej4.html">Ir a ejemplo 4</a><br/><br/>
  <a href="http://www.google.com/">Ir a
google</a><br/>
```