

TEMA 4.1

HTML 5

LM
CFGS DAW

Autor: Pascual Ligeró

Revisado por:

Fco. Javier Valero – franciscojavier.valero@ceedcv.es

2019/2020

Versión:191013.2326

Licencia



CC BY-NC-SA 3.0 ES Reconocimiento - NoComercial - CompartirIgual (by-nc-sa)

No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

NOTA: Esta es una obra derivada de la original realizada por Pascual Ligeró.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:



Importante



Atención



Interesante

ÍNDICE DE CONTENIDO

1. DIFERENCIAS HTML – HTML 5.....	4
1.1 Estructura de la página.....	4
1.2 Nuevas Etiquetas.....	5
2. Etiquetas Semánticas.....	5
2.1 Elemento header.....	5
2.2 Elemento footer.....	6
2.3 Elemento section.....	6
2.4 Etiqueta nav.....	7
2.5 Etiqueta article.....	8
2.6 Etiqueta aside.....	8
2.7 Etiqueta address.....	8
2.8 Etiqueta figure y figcaption.....	9
3. Objetos Multimedia.....	10
3.1 <video>.....	10
3.2 <audio>.....	10
4. Formularios.....	11
4.1 Nuevos Elementos.....	11
4.2 Nuevos tipos INPUT.....	12
5. Bibliografía.....	13

UD04. HTML 5

1. DIFERENCIAS HTML – HTML 5

Se considera el inicio del estándar del año 2012. Poco a poco se ha convertido en el HTML más utilizado. Se creó fuera del **World Wide Web Consortium** en la entidad llamada WHATWG (**Web Hypertext Application Technology Working Group**), auspiciado por **Apple** y **Mozilla** entre otras empresas a las que se le han ido añadiendo muchas más.

La versión definitiva de la 5ª revisión del estándar se publicó en octubre de 2014. Al no ser reconocido en viejas versiones de navegadores por sus nuevas etiquetas, se recomienda al usuario actualizar su navegador a la versión más nueva, para poder disfrutar de HTML5.

Ya dentro del lenguaje HTML específicamente, las novedades que nos trae en su versión 5 son una serie de etiquetas útiles en la web actual, algo bien sencillo de aprender y de aplicar a los sitios web, así como el propio lenguaje de marcación es también sencillo de asimilar. Las podemos clasificar en dos partes:

1.1 Estructura de la página

En el caso de HTML5 la estructura es muy parecida pero simplifica muchas etiquetas:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" >
    <title>Título</title>
  </head>
  <body>
  </body>
</html>
```

Como diferencias más notables respecto a HTML:

- La etiqueta **DOCTYPE** está simplificada
- La etiqueta **meta** es mucho más sencilla para indicar el código del archivo (Unicode)
- El lenguaje se indica sólo con el atributo **lang** del elemento **html**.
- El resto de elementos no varía

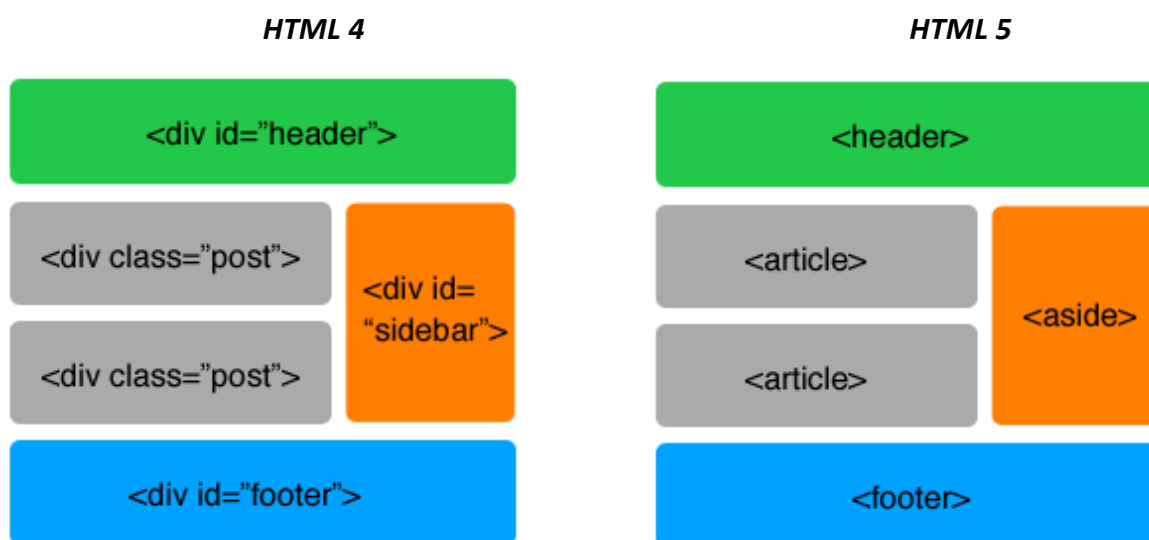
1.2 Nuevas Etiquetas

La idea en los nuevos estándares de HTML (especialmente en 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. Eso vale para casi todos los elementos ya conocidos como **p** (párrafo), **abbr** (abreviatura), **strong** (remarcado fuerte, negrita), etc.

La idea es marcar la semántica con HTML y el formato con las hojas de estilo CSS.

A este respecto hay una serie de elementos HTML aparecidos con HTML 5 que no dan ningún formato al texto pero permiten remarcarle dándole un significado y además permitiendo posteriormente que ese contenido tenga un formato especial mediante CSS.

Todos estos elementos no son comprendidos excepto por navegadores compatibles con HTML 5.



2. ETIQUETAS SEMÁNTICAS

2.1 Elemento header

Permite marcar una cabecera. Esto no tiene que ver con el elemento **head**, se trata de una serie de párrafos que se marcan de esta forma para indicar que pertenecen a la cabecera de la página.

Realmente una página puede tener varios elementos **header**, al nivel de la etiqueta **body** indica que su contenido es la cabecera de la página completa (por ejemplo donde se muestra el título general, logotipo,...). Pero dentro de una etiqueta, por ejemplo, **article** indicaría que su contenido es la cabecera del artículo. No se puede meter un header dentro de otro header ni dentro del elemento **foot** o **address**.

2.2 Elemento footer

Similar al anterior, sirve para marcar el pie de una página, sección, artículo etc. Dependiendo del contexto en el que se coloque servirá para unas cosas u otras. En el caso del elemento footer que se coloque al nivel del elemento **body**, servirá para agrupar los elementos de pie de página que suelen ser información sobre el autor, copyright, términos de uso de la página, etc.

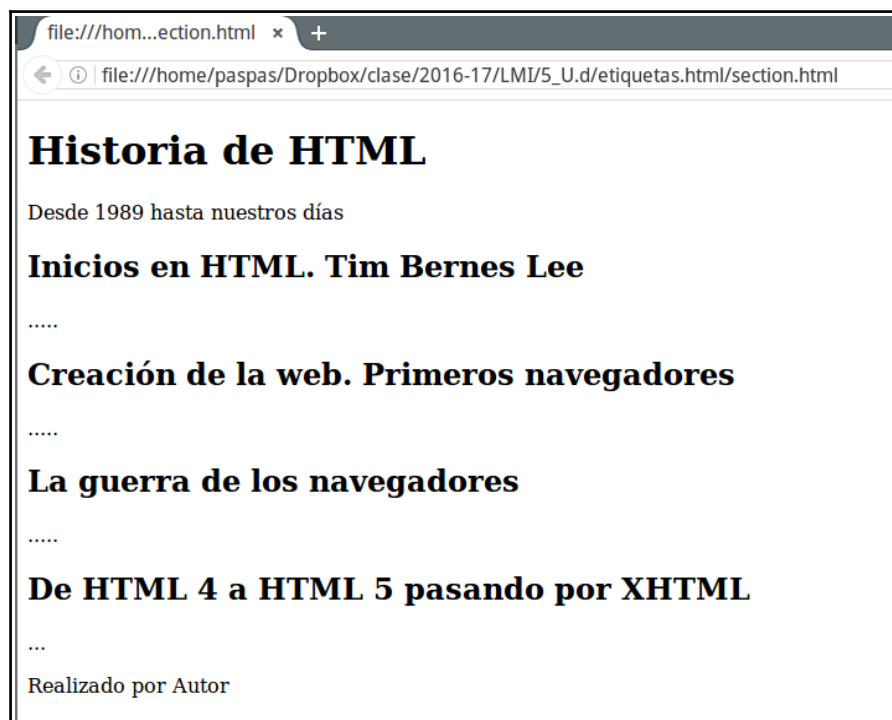
2.3 Elemento section

Con la directiva **<section>** se puede delimitar una área dentro de un documento Html, área que queremos resaltar por cualquier motivo. Es una directiva nueva en Html y funciona igual como la directiva **<div>**. Su introducción en Html se debe a que existían demasiadas directivas **<div>** en los documentos html. Su presencia, estructura mejor el código html de punto de vista estético y visual, disminuyendo en este modo la presencia de los **<div>**. Lógicamente, por la delimitación de un área con la directiva **<section>**, a este último tendría que dar un nombre utilizando los atributos "id" o "class" con el fin de su identificación posterior para formatearla. En el ejemplo que hay debajo, el párrafo esta delimitado por la directiva **<section>** llamada "seccionA" Ejemplo de página dividida en secciones:

```
<!DOCTYPE html>
<html lang="es-ES">
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>

  <body>
    <header>
      <h1>Historia de HTML</h1>
      <p>Desde 1989 hasta nuestros días</p>
    </header>
    <section id="seccionA">
      <h2>Inicios en HTML. Tim Bernes Lee</h2>
      <p>.....</p>
    </section>
    <section id="seccionB">
      <h2>Creación de la web. Primeros navegadores</h2>
      <p>.....</p>
    </section>
    <section id="seccionC">
      <h2>La guerra de los navegadores</h2>
      <p>.....</p>
    </section>
    <section id="seccionD">
      <h2>De HTML 4 a HTML 5 pasando por XHTML</h2>
```

```
        <p>...</p>
    </section>
    <footer>
        <p>Realizado por Autor</p>
    </footer>
</body>
</html>
```



2.4 Etiqueta nav

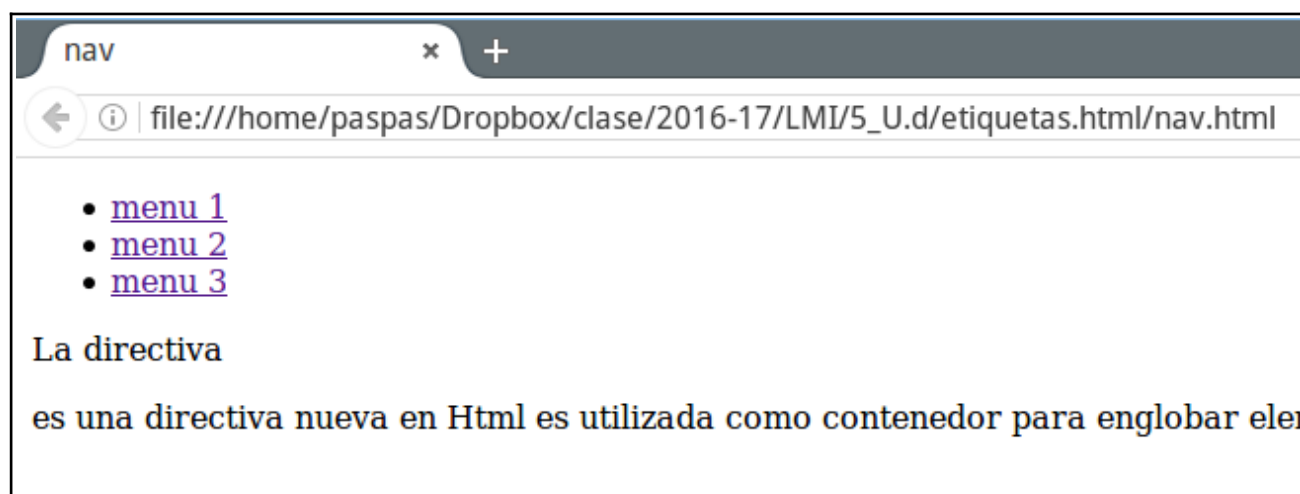
La directiva `<nav>` es una directiva nueva en Html es utilizada como contenedor para englobar elementos de navegación en un documento html. Los elementos de navegación pueden ser: el menú principal de navegación, el menú lateral de navegación, etc... Su presencia delimita las áreas de navegación de la pagina web, proporcionando una mejor vista del código html.

```
<!DOCTYPE HTML>
<html>
    <head>
        <title>nav</title>
    </head>
    <body>
        <nav>
            <ul>
```

```
<li><a href="">menu 1</a></li>
<li><a href="">menu 2</a></li>
<li><a href="">menu 3</a></li>

</ul>

</nav>
<p>La directiva <nav> es una directiva nueva en Html es utilizada
como contenedor para englobar elementos de navegación en un
documento html.
</p>
</body>
</html>
```



2.5 Etiqueta article

La directiva <article> nueva también en Html, delimita o define el contenido de una página, puede ser contenido externo, un artículo de prensa de un proveedor externo, o un texto de un web o (blog), un texto de un foro, o cualquier otro contenido.

2.6 Etiqueta aside

Delimita una sección en la pagina web donde se puede añadir contenido adicional que se considera independiente, barra lateral, publicidad o enlaces para navegación.

También permite aclaraciones sobre un texto, etc....

2.7 Etiqueta address

Representa la información de contacto de una sección, por ejemplo un artículo, una cita de un libro, o de la página si está en body.

2.8 Etiqueta figure y figcaption

Figure engloba los elementos relacionados con una imagen o un elemento multimedia.

Figcaption permite indicar el título de la imagen

```
<article>
  <hgroup>
    <h2>Palencia</h2>
    <h3>Paisajes</h3>
  </hgroup>
  <figure>
    
    <figcaption>
      Vista de la Montaña Palentina desde Camporredondo
    </figcaption>
  </figure>
</article>
```

Palencia

Paisajes



Vista de la Montaña Palentina desde Camporredondo

3. OBJETOS MULTIMEDIA

3.1 <video>

Permite insertar vídeos y animaciones en la web

Atributo	Uso
src	URL al vídeo que se desea mostrar
width	Anchura del vídeo en nuestra página
height	Altura del vídeo en nuestra página
autoplay	Usa el valor fijo autoplay para indicar que el vídeo se inicia automáticamente en cuanto se descargue
loop	Usa el valor fijo loop para indicar que el vídeo se ejecuta automáticamente una y otra vez
controls	Con valor controls indica que el vídeo mostrará los controles de reproducción (pausa, play,...)
preload	Recomendación sobre cómo debemos realizar la descarga. Posibilidades: <ul style="list-style-type: none">• auto. El vídeo se descarga en cuanto se carga la página• none. El vídeo no se descarga. Cuando el usuario pulse play, se descargará.• metadata. Descarga los metadatos del vídeo, no el vídeo en sí
poster	Permite indicar la dirección URL a una imagen que se mostrará mientras el vídeo no se está reproduciendo. Si no se usa este atributo, se usa el primer fotograma del vídeo como póster.

3.2 <audio>

Se trata de un elemento comparable al video, pero en este caso para incorporar audio a la página.

Atributo	uso
src	URL al vídeo que se desea mostrar.
autoplay	Usa el valor fijo autoplay para indicar que el vídeo se inicia automáticamente en cuanto se descargue.
loop	Usa el valor fijo loop para indicar que el vídeo se ejecuta automáticamente una y

otra vez.

controls Con valor **controls** indica que el vídeo mostrará los controles de reproducción (pausa, play,...).

preload Recomendación sobre cómo debemos realizar la descarga.

Posibilidades:

- **auto**. El vídeo se descarga en cuanto se carga la página.
- **none**. El vídeo no se descarga. Cuando el usuario pulse play, se descargará.
- **metadata**. Descarga los metadatos del vídeo, no el vídeo en sí.

4. FORMULARIOS

Vamos a ver cuáles son los cinco nuevos elementos que podemos integrar dentro de cualquier formulario web. Con estos nuevos elementos obtenemos mayor interactividad en nuestras aplicaciones web, realizando tareas que normalmente se llevaban a cabo con ayuda de códigos Javascript un tanto complejos. Ahora, con las nuevas especificaciones de HTML5, es posible realizar las mismas tareas, con la inmensa ventaja de usar etiquetas nativas, que generan campos destinados a algunos trabajos en particular, evitándonos usar elementos clásicos a los que asignar funcionalidades con programación.

Debemos aclarar que algunos elementos, de los que se han incorporados en las nuevas especificaciones de HTML5, pueden parecer muy poco prácticos. En ocasiones puede que sea difícil encontrarle una utilidad dentro de nuestras aplicaciones web, sin embargo, mencionaremos los cinco nuevos elementos.

Nota: La potencia de estos nuevos elementos radica en su uso simplificado. Algunos incluso disponen de nuevas API de programación Javascript, que se integran muy fácilmente en cualquier entorno de desarrollo que utilice tecnología HTML5.

4.1 Nuevos Elementos

- **Etiqueta METER:**

Esta nueva etiqueta se usa para representar escalas de medidas conocidas, de ahí su nombre: Meter, el cual tiene relación directa con medida. Se puede emplear para representar escalas de medición conocidas como longitud, masa, peso, entre otras.

- **Etiqueta PROGRESS:**

Con esta etiqueta no hay que hacer mayor explicación, pues como su nombre lo indica, se usa para crear barras de progreso. Podemos emplearlas o usarlas en procesos de larga duración, como la descarga de archivos, para indicar a cualquier usuario de nuestra aplicación el progreso de la operación que se está realizando.

- **Elemento DATALIST:**

Es una extensión que sirve para crear campos de autocompletado. Sirve para especificar una lista de datos u opciones que se pueden utilizar para sugerir el autocompletado de elementos como INPUT. Por tanto, para utilizar un DATALIST, tenemos que combinar ese elemento con otros elementos de formulario al que le colocamos atributos nuevos para asociar el DATALIST para hacer el autocompletado.

- **Etiqueta KEYGEN:**

Usada para generar pares de claves, clave pública y privada. Al enviar el formulario al servidor por cualquiera de los métodos HTTP, en el cliente se guarda una clave privada la clave pública se empaqueta y se envía al servidor.

- **Etiqueta o elemento OUTPUT:**

Muestra el resultado de un cálculo matemático, su uso básico puede ser tan básico como el de mostrar una simple suma de dos números.

4.2 Nuevos tipos INPUT

Cuando uno habla de formularios, se refiere también a la interacción entre los usuarios y una aplicación web, basada en la introducción de datos de entrada. En muchas ocasiones, la entrada de datos puede convertirse en una tarea tediosa para los desarrolladores, quienes tienen que buscar que métodos son los mejores para introducir datos de una manera intuitiva y validarlos por medio de la programación.

Es frecuente ver, en aplicaciones web medianamente bien realizadas, rutinas de validación de datos que permiten al usuario rellenar los formularios con la información correcta. El trabajo para la validación de los datos generalmente se realiza con un lenguaje del lado del cliente como Javascript, aunque algunos prefieren lenguajes de servidor para validar datos.

Nota: No siempre es recomendable validar los datos en el servidor, por el abuso de transacciones HTTP que pueden producirse. Por ello, el mejor método para la validación es JavaScript en el lado del cliente, o una mezcla entre ambas arquitecturas, cliente y servidor. Lamentable hay quienes no reconocen las bondades del lenguaje de programación del lado del cliente estándar, Javascript, aunque esto poco a poco va pasando a la historia y ahora más con el HTML5.

Gracias a HTML5 los desarrolladores hemos ganado unas importantes herramientas para validar datos en un formulario, de una forma más fácil, con menos rutinas de código. Incluso en algunos casos no es necesario siquiera hacer nada de JavaScript, pues existen mecanismos para verificar la corrección de los datos, con solo un poco de HTML.

A continuación mencionamos cada uno de los doce nuevos INPUT que están presentes en la quinta especificación del lenguaje HTML, con una breve explicación de cada uno.

- **INPUT tel:** Este tipo de input viene predispuesto con un formato para escribir números telefónicos. En realidad no hace ninguna validación, pero sí se puede implementar una con la nueva API de validación de JavaScript.

- **INPUT number pre formateado:** Sirve para escribir solo números. En algunos navegadores, cuando se ejecuta el evento onSubmit no se hace el envío en caso que el campo number esté lleno de caracteres que no sean numéricos.
- **INPUT search:** Además de proporcionar un campo de entrada, se le agrega un icono de búsqueda para distinguirlo de un campo de navegación.
- **INPUT color:** Este input nos brinda una paleta de colores donde el usuario puede escoger un color de forma dinámica. Es lo que llamamos un colorpicker, con la particularidad que nos lo ofrece el propio navegador.
- **INPUT range:** Proporciona un control que se desliza, cambiando automáticamente el valor del campo.
- **INPUT URL:** Este tipo de entrada viene con un formato para URL absoluta.
- **INPUT email:** Tiene la capacidad de aceptar únicamente direcciones de correo electrónico. Además, se pueden enviar varios email separados por comas, si tiene especificado el atributo multiple.
- **INPUT datetime:** Para obtener fecha del conjunto de la zona horaria UTC.
- **INPUT date:** Para introducir una fecha que no haga parte de del conjunto horario.
- **INPUT month:** Para introducir meses del año.
- **INPUT week:** Ofrece una utilidad para escribir y captar información de semanas.
- **INPUT time:** Obtiene información con horas, minutos y segundos.
- **Tipo datetime-local:** Recibe la hora local del dispositivo.

5. BIBLIOGRAFÍA

<http://www.desarrolloweb.com/articulos/nuevas-etiquetas-html5.html>

<http://www.desarrolloweb.com/manuales/manual-html.html#manual168>