

# 1. CONCEPTOS INICIALES

No es simplemente una nueva versión del lenguaje de marcación HTML, sino una agrupación de diversas especificaciones concernientes al desarrollo web. Es decir, HTML 5 no se limita sólo a crear nuevas etiquetas, atributos y eliminar aquellas marcas que están en desuso o se utilizan inadecuadamente, sino que va mucho más allá.

HTML5 es considerado el producto de la combinación de HTML, CSS y Javascript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y Javascript hace el resto.

## a) NOVEDADES

- **Estructura del cuerpo:** La mayoría de las webs tienen un formato común, formado por elementos como cabecera, pie, navegadores, etc. HTML 5 permite agrupar todas estas partes de una web en nuevas etiquetas que representarán cada una de las partes típicas de una página.
- **Etiquetas para contenido específico:** Hasta ahora se utilizaba una única etiqueta para incorporar diversos tipos de contenido enriquecido, como animaciones Flash o vídeo. Ahora se utilizarán etiquetas específicas para cada tipo de contenido en particular, como audio, vídeo, etc.
- **Fin de las etiquetas de presentación:** todas las etiquetas que tienen que ver con la presentación del documento, es decir, que modifican estilos de la página, serán eliminadas. La responsabilidad de definir el aspecto de una web correrá a cargo únicamente de CSS.

## b) API's HTML5

- **WEB STORAGE (localStorage y sessionStorage):**  
El navegador permitirá el uso de una base de datos local, con la que se podrá trabajar en una página web por medio del cliente y a través de un API. Es algo así como las Cookies, pero pensadas para almacenar grandes cantidades de información, lo que permitirá la creación de aplicaciones web que funcionen sin necesidad de estar conectados a Internet. Con el HTML clásico si necesitamos almacenar datos en el cliente (navegador) se utilizan las cookies. Con HTML5 se han agregado otras tecnologías para almacenar datos en el cliente. El objeto localStorage nos permite almacenar datos que serán recordados por el navegador para siempre, es decir no tienen una fecha de caducidad.  
La cantidad de información que podemos almacenar es muy superior a la permitida con las cookies. La información que se almacena en el localStorage a diferencia de las cookies no se envía al servidor cada vez que se solicita una página.
- **GEOLOCALIZACIÓN**  
Las páginas web se podrán localizar geográficamente por medio de un API que permita la Geolocalización.  
La geolocalización es una característica de HTML5 que nos permite acceder a ubicación geográfica del usuario del sitio web con una precisión absoluta en el caso que el dispositivo que utilice tenga GPS.  
La precisión va a disminuir si no tiene GPS y debe utilizarse la IP del proveedor de servicios de Internet.  
El navegador administra la geolocalización mediante un objeto llamado geolocation que es un atributo del objeto navigator.
- **DRAG & DROP**  
El HTML5 nos permite fácilmente implementar el concepto de drag and drop (arrastrar y soltar). Parte se resuelve con HTML y otra mediante Javascript.
- **CANVAS**  
Es un componente que permitirá dibujar, por medio de las funciones de un API, en la página todo tipo de formas, que podrán estar animadas y responder a interacción del usuario. Es algo así como las posibilidades que nos ofrece Flash, pero dentro de la especificación del HTML y sin la necesidad de tener instalado ningún plugin. (javascript)

## 2. ESTRUCTURA GLOBAL

---

La declaración del tipo de documento (DTD Document Type Declaration) se ubica en la primera línea del archivo HTML, es decir antes de la marca html.

### a) !DOCTYPE

`<!DOCTYPE HTML>` o `<!DOCTYPE html>`

Luego debemos construir la estructura del document.

`<!DOCTYPE html>`

`<html lang="es">`

`</html>`

### b) META

En la cabecera del documento. La etiqueta `<meta>` especifica cómo el texto será presentado en pantalla.

`<head>`

`<meta charset="iso-8859-1">`

`</head>`

La nueva etiqueta `<meta>` para la definición del tipo de caracteres es más corta y simple. Podemos cambiar el tipo iso-8859-1 por el necesario para nuestros documentos (UTF-8) y agregar otras etiquetas `<meta>` como description o keywords para definir otros aspectos de la página web.

`<head>`

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

`<meta name="description" content="Ejemplo de HTML5">`

`<meta name="keywords" content="HTML5, CSS3, Javascript">`

`</head>`

Hay varios tipos de etiqueta `<meta>` que pueden ser incluidas para declarar información general sobre el documento, pero esta información no es mostrada en la ventana del navegador, es solo importante para motores de búsqueda y dispositivos que necesitan hacer una vista previa del documento u obtener un sumario de la información que contiene.

### c) LINK

En la cabecera del documento.

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

En HTML5 ya no se necesita especificar qué tipo de estilos estamos insertando, por lo que el atributo type ha sido eliminado.

### d) ESTRUCTURA DEL CUERPO

Para evitar el uso abusivo de elementos DIV para estructurar una web en bloques en HTML5 aparecen etiquetas que perfeccionan esta estructuración estableciendo qué es cada sección, eliminando así DIV innecesarios. Este cambio en la semántica hace que la estructura de la web sea más coherente y fácil de entender por otras personas y los navegadores podrán darle más importancia a según qué secciones de la web facilitándole además la tarea a los buscadores, así como cualquier otra aplicación que interprete sitios web.

Las webs se dividirán en los siguientes elementos:



- **HEADER**

El elemento header debe utilizarse para marcar la cabecera de una página (contiene el logotipo del sitio, una imagen, un cuadro de búsqueda...)

El elemento header puede estar anidado en otras secciones de la página (es decir que no solo se utiliza para la cabecera de la página)

- **NAV**

El elemento nav representa una parte de una página que enlaza a otras páginas o partes dentro de la página. Es una sección con enlaces de navegación.

Desde las opciones de menú se puede ir a cada una de las secciones de la página (`href="#id"`)

No todos los grupos de enlaces en una página deben ser agrupados en un elemento nav. Únicamente las secciones que consisten en bloques de navegación más importantes son adecuados para el elemento de navegación.

- **SECTION**

El elemento section representa un documento genérico o de la sección de aplicación. Una sección, en este contexto, es una agrupación temática de los contenidos. Puede ser un capítulo, una sección de un capítulo o básicamente cualquier cosa que incluya su propio encabezado. Una página de inicio de un sitio Web puede ser dividida en secciones para una introducción, noticias, información de contacto etc. (Cada sección puede tener un id).

- **FOOTER**

El elemento footer se utiliza para indicar el pie de la página o de una sección. Un pie de página contiene información general acerca de su sección el autor, enlaces a documentos relacionados, datos de derechos de autor etc.

- **ASIDE**

El elemento aside representa una nota, un consejo, una explicación. El elemento puede ser utilizado para efectos de atracción, como las comillas tipográficas o barras laterales, para la publicidad, por grupos de elementos de navegación, y por otro contenido que se considera por separado del contenido principal de la página.

- **ARTICLE**

El elemento article representa una entrada independiente en un blog, revista, periódico etc. Cuando se anidan los elementos article, los artículos internos están relacionados con el contenido del artículo exterior. Por ejemplo, una entrada de blog en un sitio que acepta comentarios, el elemento article principal agrupa el artículo propiamente dicho y otro bloque article anidado con los comentarios de los usuarios.

## **OTRAS ETIQUETAS HTML5**

- **<HGROUP>**

Este elemento es usado para agrupar varias etiquetas de cabeceras (<h1> <h2> <h3> <h4> <h5> <h6>) cuando se van a utilizar títulos, subtítulos.

- **<FIGURE>**

Representa un contenido independiente, a menudo con un título. Por lo general, se trata de una imagen, una ilustración, un diagrama, un fragmento de código, o un esquema al que se hace referencia en el texto principal, pero que se puede mover a otra página o a un apéndice sin que afecte al flujo principal.

- **<FIGCAPTION>**

Representa un subtítulo o leyenda asociada al contenido de <figure> al cual está anidado, pudiendo ser colocado como primer o último hijo de <figure>, es importante destacar que <figcaption> es opcional.

- **<MARK>**

Para marcar un contenido de manera que pueda resaltarse y diferenciarse de los demás.

- **<SMALL>**

Este elemento representa contenido al margen, como letra pequeña (por ejemplo, descargos, restricciones legales, declaración de derechos...).

- **<CITE>**

Este elemento es usado para mostrar el título de un trabajo (libro, película, poema...).

- **<ADDRESS>**

Este elemento encierra información de contacto para un elemento <article> o el documento completo. Es recomendable que sea insertado dentro de un elemento <footer>

- **<TIME>**

Este elemento se utiliza para mostrar fecha y hora en formatos comprensibles por los usuarios y el navegador. El valor para los usuarios es ubicado entre las etiquetas mientras que el específico para programas y navegadores es incluido como el valor del atributo datetime. Un segundo atributo optativo llamado pubdate es usado para indicar que el valor de datetime es la fecha de publicación.

**Ejemplo1:** Documento html con la estructura de la página en diferentes secciones.

### 3. FORMULARIOS

---

HTML5 nos ofrece nuevas opciones a la hora de crear formularios, como son nuevos tipos de campos y validación automática del lado del cliente, sin necesidad de scripts.

#### a) NUEVOS TIPOS DE CAMPOS

- **email:** nos sirve para introducir direcciones de email  
`<input type="email" name="miemail" id="miemail">`  
El texto insertado en el campo generado será controlado por el navegador y validado como un email. Si la validación falla, el formulario no será enviado.
- **url:** para direcciones de páginas web  
Está destinado a recibir solo URLs absolutas y retornará un error si el valor es inválido.  
`<input type="url" name="miurl" id="miurl">`
- **number:** para la introducción de números  
El tipo number es sólo válido cuando recibe una entrada numérica. Existen algunos atributos que pueden ser útiles para este campo:  
**min** El valor de este atributo determina el mínimo valor aceptado para el campo.  
**max** El valor de este atributo determina el máximo valor aceptado para el campo.  
**step** El valor de este atributo determina el tamaño en el que el valor será incrementado o disminuido en cada paso.  
Por ejemplo, si declara un valor de 5 para step en un campo que tiene un valor mínimo de 0 y máximo de 10, el navegador no le permitirá especificar valores entre 0 y 5 o entre 5 y 10.  
`<input type="number" name="numero" id="numero" min="0" max="10" step="5">`
- **tel:** para números de teléfono  
Este tipo de campo es para números telefónicos. A diferencia de los tipos email y url, el tipo tel no requiere ninguna sintaxis en particular. Es solo una indicación para el navegador en caso de que necesite hacer ajustes de acuerdo al dispositivo en el que la aplicación es ejecutada. Si es un móvil se abre el teclado telefónico.  
`<input type="tel" name="telefono" id="telefono">`
- **search:** para campos de búsqueda  
El tipo search (búsqueda) no controla la entrada, es solo una indicación para los navegadores. Al detectar este tipo de campo algunos navegadores cambiarán el diseño del elemento para ofrecer al usuario un indicio de su propósito.  
`<input type="search" name="busqueda" id="busqueda">`
- **date:** para seleccionar una fecha  
Algunos navegadores muestran en pantalla un calendario que aparece cada vez que el usuario hace clic sobre el campo. El calendario le permite al usuario seleccionar un día que será insertado en el campo junto con el resto de la fecha.  
`<input type="date" name="fecha" id="fecha">`
- **time:** para seleccionar una hora  
Toma el formato de horas y minutos, pero su comportamiento depende de cada navegador en este momento. Normalmente el valor esperado tiene la sintaxis hora:minutos:segundos, pero también puede ser solo hora:minutos.  
`<input type="time" name="hora" id="hora">`
- **month:** para seleccionar un mes  
Normalmente el valor esperado tiene la sintaxis año-mes.  
`<input type="month" name="mes" id="mes">`

- **week:** para seleccionar una semana completa
- **datetime:** para seleccionar fecha y hora  
 Para insertar fecha y hora completa, incluyendo la zona horaria.
- **datetime-local:** para seleccionar una fecha y una hora sin zona horaria
- **range:** para seleccionar entre un rango de valores  
 Este tipo de campo hace que el navegador construya una nueva clase de control que no existía previamente. Este nuevo control le permite al usuario seleccionar un valor a partir de una serie de valores o rango. Normalmente es mostrado en pantalla como un puntero deslizable o un campo con flechas para seleccionar un valor entre los predeterminados. El tipo range usa los atributos min y max estudiados previamente para configurar los límites del rango. También puede utilizar el atributo step para establecer el tamaño en el cual el valor del campo será incrementado o disminuido en cada paso.
- **color:** para seleccionar un color en valor hexadecimal
- **password:** tipo contraseña
- **file:** subir archivos

## b) NUEVOS ATRIBUTOS

- **placeholder**  
 Representa una sugerencia corta, una palabra o frase para ayudar al usuario a escribir la información correcta. El valor de este atributo es presentado en pantalla por los navegadores dentro del campo, como una marca de agua que desaparece cuando el elemento es enfocado.  
 Es distinto de value porque este campo no hay que borrarlo.
- **required**  
 Nos indica que ese campo no puede ser enviado vacío, y debe cumplir las especificaciones del tipo de campo que sea, por ejemplo, en un campo tipo email debemos introducir una dirección de correo válida.
- **multiple**  
 Es otro atributo booleano que puede ser usado en algunos tipos de campo (por ejemplo, email o file) para permitir el ingreso de entradas múltiples en el mismo campo.
- **autofocus**  
 Hace que el cursor se coloque automáticamente en ese campo sin necesidad de ningún script.  
 El atributo autofocus enfocará la página web sobre el elemento seleccionado pero considerando la situación actual. No moverá el foco cuando ya haya sido establecido por el usuario sobre otro elemento.
- **maxlength**  
 En los textareas ahora podemos utilizar este atributo para limitar el número máximo de caracteres introducidos.

- **pattern:**

Para propósitos de validación. Usa expresiones regulares para personalizar reglas de validación. Algunos de los tipos de campo vistos validan cadenas de texto específicas, pero no permiten hacer validaciones personalizadas, como por ejemplo un código postal que consiste en 5 números.

El atributo pattern nos permite crear nuestro propio tipo de campo para controlar esta clase de valores no ordinarios. Puede incluso incluir un atributo title para personalizar mensajes de error.

Ejemplo:

```
<input pattern="[0-9]{5}" name="codigopostal" id="codigopostal" title="Escriba los 5 números de su código postal">
```

Mirar la página: <http://html5pattern.com>

- **form:**

Nos permite declarar elementos para un formulario fuera del ámbito de las etiquetas <form>. Hasta ahora, para construir un formulario teníamos que escribir las etiquetas <form> de apertura y cierre y luego declarar cada elemento del formulario entre ellas. En HTML5 podemos insertar los elementos en cualquier parte del código y luego hacer referencia al formulario que pertenecen usando su nombre y el atributo form.

## c) OTROS ELEMENTOS DE LOS FORMULARIOS

<fieldset>	Agrupación de campos
<b>Tipo de elemento</b>	Bloque
<b>Descripción</b>	Se emplea para agrupar de forma lógica varios campos de un formulario
<legend>	Título o leyenda de un fieldset
<b>Tipo de elemento</b>	En línea
<b>Descripción</b>	Se emplea para definir el título o leyenda de un conjunto de campos de formulario agrupados con la etiqueta fieldset

<label>	Título o leyenda de un campo de formulario
<b>Atributos específicos</b>	for="id_de_elemento" - Indica el ID del campo del formulario para el que este elemento es su título. Otros: accesskey, onfocus y onblur
<b>Tipo de elemento</b>	En línea
<b>Descripción</b>	Se emplea para definir el título o leyenda de los campos definidos en un formulario

### Ejemplo2Formulario

En este ejemplo añadir los tipos de campos que no aparecen y probarlos.

## 4. VALIDACIONES CON PATTERN

---

<http://html5pattern.com>

### 1. Validar una URL

```
/^(https?:\/\/)?([\da-z\.-]+)\.([a-z]{2,6})([\/w\?=-]*)*\/?$
```

### 2. Validar un E-mail

```
^[a-z0-9-]+(\.[a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*\.[a-z]{2,3}$
```

### 3. Comprobar la seguridad de una contraseña

```
(?=^.{8,}$)((?=.*\d)|(?=.*\W+))(?![\.\n])(?=.*[A-Z])(?=.*[a-z]).*$
```

De esta forma comprobaremos:

Contraseñas que contengan al menos una letra mayúscula.

Contraseñas que contengan al menos una letra minúscula.

Contraseñas que contengan al menos un número o carácter especial.

Contraseñas cuya longitud sea como mínimo 8 caracteres.

Contraseñas cuya longitud máxima no debe ser arbitrariamente limitada.

### 4. Validar un número de teléfono

```
^\+?\d{1,3}?[-.]?(?(\d{2,3})\)?[-.]?\d\d\d[-.]?\d\d\d\d$
```

### 5. Validar número de tarjeta de crédito

```
^((67\d{2})|(4\d{3})|(5[1-5]\d{2})|(6011))(-?\s?\d{4}){3}((3[4,7])\ d{2}-?\s?\d{6}-?\s?\d{5})$
```

### 6. Recoger ID de un vídeo de Youtube

```
/http:\/\/(?:youtu\.be\/|(?:[a-z]{2,3}\.?)youtube\.com\/watch(?:\?|#|!)v=)([w-]{11}).*/gi
```

### 7. Validar todas las etiquetas de imagen sin cerrar

```
<img([>+])(\s*[^\s])></img([>+)
```

### 8. Validar una dirección IP

```
/^(((1-9)?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5]).){3}((1-9)?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$
```

### 9. Validar código postal

```
^([1-9]{2}|[0-9][1-9]|[1-9][0-9])[0-9]{3}$
```

### 10. Validar un nombre de usuario

```
/^[a-z\d_]{4,15}$/i
```



## 5. ETIQUETAS DE AUDIO Y VIDEO

---

### a) AUDIO

Otro elemento que se agrega al HTML5 es el AUDIO. El objetivo de esta etiqueta es permitir la carga y ejecución de archivos de audio sin requerir un plugin de Flash, Silverlight o Java.

Un ejemplo de disponer el elemento audio dentro de una página sería:

```
<audio src="sonido.ogg" autoplay controls loop></audio>
```

Las propiedades que podemos utilizar con la marca audio son:

- **src:** La URL donde se almacena el archivo de audio. Si no definimos la URL la busca en el mismo directorio donde se almacena la página.
- **loop:** El archivo de audio se ejecuta una y otra vez.
- **controls:** Indica que se deben mostrar la interface visual del control en la página (este control permite al visitante arrancar el audio, detenerlo, desplazarse etc.)
- **autobuffer:** En caso de estar presente indica que primero debe descargarse el archivo en el cliente antes de comenzar a ejecutarse.

Como no hay un formato de audio universalmente adoptado por todos los navegadores el elemento audio nos permite agregarle distintas fuentes:

```
<audio controls autoplay loop>  
  <source src="sonido.ogg">  
  <source src="sonido.mp3">  
  <source src="sonido.wav">  
  <source src="sonido.au">  
</audio>
```

El elemento source indica a través de la propiedad src la ubicación del archivo de audio respectivo.

El orden que disponemos estas fuentes es importante. Primero el navegador busca la primera fuente y verifica que puede reproducir dicho archivo, en caso negativo pasa a la siguiente fuente.

Por el momento no hay un formato con soporte para todos los navegadores, dependerá del sitio que implementemos para ver si tiene sentido duplicar nuestros archivos con distintos formatos de audio.

**Ejemplo3AudioyVideo.html**

## b) VIDEO

HTML5 introdujo un elemento para insertar y reproducir video en un documento HTML.

El elemento **<video>** usa etiquetas de apertura y cierre y solo unos pocos parámetros para lograr su función.

```
<video src="video.mp4" controls>
```

Este navegador no permite tag video

```
</video>
```

Las propiedades más importantes de la marca video son:

- **src:** Dirección donde se almacena el video.
- **controls:** Se visualiza el panel de control del video: botón de inicio, barra de avance del video etc.
- **loop:** El archivo de vídeo se ejecuta una y otra vez.
- **width:** Ancho en píxeles del video.
- **height:** Alto en píxeles del video.

Los tamaños del vídeo modificarlos en el css.

Como no hay un formato de video universalmente adoptado por todos los navegadores el elemento video nos permite agregarle distintas fuentes:

```
<video width="640" height="360" controls>
```

```
  <source src="video.mp4">
```

```
  <source src="video.wmv">
```

```
  <source src="video.ogg">
```

```
</video>
```

**Ejemplo3AudioyVideo.html**