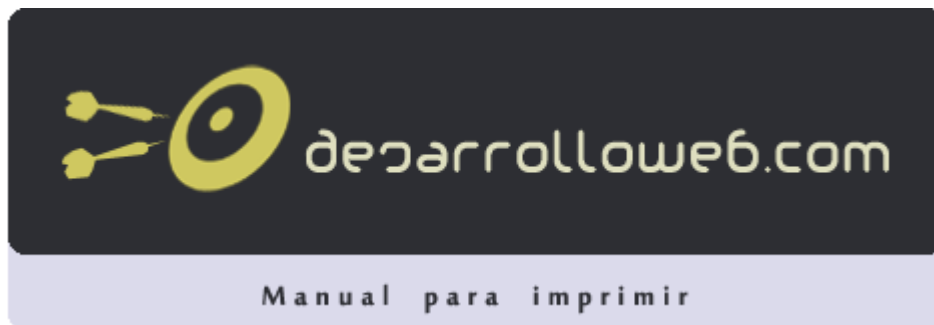


# Manual del lenguaje de marcación de HTML5

Manual que trata sobre el lenguaje HTML5, pero específicamente con las novedades del lenguaje de marcación o etiquetado, dejando aparte otras tecnologías relacionadas.



## Autores del manual

Este manual ha sido realizado por los siguientes colaboradores de DesarrolloWeb.com:

### Miguel Angel Alvarez

Director de DesarrolloWeb.com y EscuelaIT  
<http://www.desarrolloweb.com>  
(7 capítulos)

### Dairo Galeano

Desarrollador independiente  
(2 capítulos)

### Jaime Peña Tresancos

Escritor. Colaborador habitual de revistas de tecnología y experto en nuevas tecnologías y programas Microsoft  
(1 capítulo)

## Qué es HTML 5

*Veremos qué es HTML 5, su previsión de tiempo para convertirse en una especificación recomendada y las novedades más significativas que proporcionará.*

En el artículo anterior publicado en DesarrolloWeb.com, "[El futuro del desarrollo web: HTML 5](#)" explicamos las razones por las que es verdaderamente importante esta nueva versión del lenguaje de marcación HTML y vimos quiénes son los que están llevando a cabo su especificación.

Ahora convendría explicar qué es exactamente HTML 5, ya que no es simplemente una nueva versión del lenguaje de marcación HTML, sino una agrupación de diversas especificaciones concernientes a el 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.

Así pues, HTML 5 es una nueva versión de diversas especificaciones, entre las que se encuentran:

- HTML 4
- XHTML 1
- CSS Nivel 2
- DOM Nivel 2 (DOM = Document Object Model)

A la par, HTML 5 pretende proporcionar una plataforma con la que desarrollar aplicaciones web más parecidas a las aplicaciones de escritorio, donde su ejecución dentro de un navegador no implique falta de recursos o facilidades para resolver las necesidades reales de los desarrolladores. Para ello se están creando unas APIs que permitan trabajar con cualquiera de los elementos de la página y realizar acciones que hasta hoy era necesario realizar por medio de tecnologías accesorias.

Estas API, que tendrán que ser implementadas por los distintos navegadores del mercado, se están documentando con minuciosidad, para que todos los Browsers, creados por cualquier compañía las soporten tal cual se han diseñado. Esto se hace con la intención que no ocurra lo que viene sucediendo en el pasado, que cada navegador hace la guerra por su parte y los que acaban pagándolo son los desarrolladores y a la postre los usuarios, que tienen muchas posibilidades de acceder a webs que no son compatibles con su navegador preferido.

## Cuándo estará listo HTML 5

Según informan en la página de la organización WHATWG, HTML 5 se prevé esté listo como especificación de implementación recomendada en el 2012. ¿Quiere esto decir que vamos a tener que esperar hasta 2012 para aprovechar las ventajas de HTML 5? realmente no es justamente así, puesto que algunos navegadores ya implementan muchas de las características del moderno lenguaje.

Resulta que HTML 5 está formado por muchos módulos distintos, cuyo grado de especificación está en niveles dispares. Por tanto, muchas de las características de HTML 5 están ya listas para ser implementadas, en un punto de desarrollo que se encuentra cercano al que finalmente será presentado. Otras muchas características están todavía simplemente en el tintero, a modo de ideas o borradores iniciales.

De hecho, las versiones más nuevas de casi todos los navegadores, incluido el polémico Internet Explorer 8, implementan algunas de las características de HTML 5. Claro que, para que una web se vea bien en todos los sistemas, hay que utilizar sólo aquellas partes que funcionan en todos los navegadores, por lo que a día de hoy, pocas son las utilidades realmente disponibles del lenguaje, si queremos hacer un sitio web compatible. No obstante, en el peor de los casos, podemos empezar a usar a nivel experimental estas características, aunque sólo sea para frotarnos las manos en espera de incorporarlas realmente en nuestras prácticas de desarrollo habituales.

## Cuáles son las novedades de HTML 5

HTML 5 incluye novedades significativas en diversos ámbitos. Como decíamos, no sólo se trata de incorporar nuevas etiquetas o eliminar otras, sino que supone mejoras en áreas que hasta ahora quedaban fuera del lenguaje y para las que se necesitaba utilizar otras tecnologías.

- **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 uno 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.
- **Canvas:** es un nuevo 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. Puedes conocer más sobre este nuevo elemento en el [manual de canvas](#) que estamos creando en DesarrolloWeb.com
- **Bases de datos locales:** 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.
- **Web Workers:** son procesos que requieren bastante tiempo de procesamiento por parte del navegador, pero que se podrán realizar en un segundo plano, para que el usuario no tenga que esperar que se terminen para empezar a usar la página. Para ello se dispondrá también de un API para el trabajo con los Web Workers.
- **Aplicaciones web Offline:** Existirá otro API para el trabajo con aplicaciones web, que se podrán desarrollar de modo que funcionen también en local y sin estar conectados a Internet.
- **Geolocalización:** Las páginas web se podrán localizar geográficamente por medio de un API que permita la Geolocalización.
- **Nuevas APIs para interfaz de usuario:** temas tan utilizados como el "drag & drop" (arrastrar y soltar) en las interfaces de usuario de los programas convencionales, serán incorporadas al HTML 5 por medio de un API.
- **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.

Como se puede ver, existirán varios API con los que podremos trabajar para el desarrollo de todo tipo de aplicaciones complejas, que funcionarán online y offline. Quizás se entienda mejor por qué HTML 5 es un proyecto tan ambicioso y que está llevando tanto tiempo para ser elaborado.

Artículo por *Miguel Angel Alvarez*

## Grabación del Webcast de HTML5

*Vídeo de la grabación del Webcast de HTML5 con Jorge del Casar, qué es HTML5 y qué no es.*

Estamos orgullosos de presentaros un webcast realizado con una de las figuras del HTML5 en español, Jorge del Casar, que se ha brindado a ofrecernos una introducción al lenguaje de la web en su nueva versión.

En este webcast ofrecemos información interesante para todo aquel que esté deseando obtener una introducción a HTML5, en la que nos informarán sobre qué es este conjunto de tecnologías y cómo la web está cambiando a raíz de ellas.

En el webcast se nos presentará cada uno de los integrantes de esta revolución en los estándares abiertos, desde el propio lenguaje HTML en su nueva versión, las CSS3 y la serie de APIs disponibles en Javascript para trabajar con el hardware de los dispositivos u ordenadores conectados a Internet. Geolocalización, Almacenamiento local, dibujo en la página, etc.

Como se podrá comprobar, uno de los detalles a los que se dedicó más tiempo es a la web semántica, y las nuevas etiquetas

que nos permiten mejorar la forma en la que los sistemas informatizados, como motores de búsqueda, entienden las páginas web.

Además se dieron algunas claves interesantes sobre qué navegadores soportan hoy una buena gama de las nuevas características del HTML5, que podríamos utilizar con toda confianza y qué herramientas existen en Internet para poder aplicar compatibilidad para navegadores que aun no soportan estos nuevos estándares.

Además, durante el vídeo se ofrecieron unas diapositivas que nos ofrecen una demo de las posibilidades de HTML5, como son estilos impactantes basados en CSS3, transiciones o animaciones CSS3 o el propio uso de las etiquetas semánticas. Pero también algunas de las funcionalidades más destacadas de las APIs de programación Javascript, como son los Websockets, que permiten actualizar la página de los clientes web cuando el servidor lo necesite.

Las diapositivas se pueden ver ahora mismo, para poder encontrar esas aplicaciones del HTML5, en la siguiente dirección:

<http://html5paratodos.jit.su>

El vídeo de este webcast se encuentra embebido a continuación:

*Artículo por Miguel Angel Alvarez*

## Etiquetas nuevas de HTML5

*Cuáles son las etiquetas nuevas del lenguaje HTML5, con una breve descripción sobre su utilidad y clasificación.*

La nueva versión del lenguaje HTML trae consigo muchas novedades. Como vimos en el artículo [Qué es HTML5](#), el nuevo estándar está compuesto de diversas tecnologías de lo más variado, para suplir todas las necesidades de la Web actual y de los nuevos dispositivos de movilidad.

En esa línea comprensión de lo que es HTML5, cabe destacar que existen varios lenguajes que están siendo actualizados con el nuevo estándar, como HTML, CSS o Javascript. Si tuviésemos una pila de asuntos que representasen cada una de estas novedades, la del propio lenguaje HTML sería la menor, seguida de CSS y luego de Javascript. Sin embargo, hay varias cosas que debemos conocer sobre el lenguaje de etiquetado en esta nueva versión, que están cambiando enormemente la manera en la que se conciben y desarrollan los sitios web actuales.

### Novedades en HTML 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:

- Etiquetas que nos traen soporte a nuevas funcionalidades: es decir, aquellas que nos sirven para extender el HTML, dando soporte a asuntos como el vídeo o el sonido, lienzos donde diseñar dibujos, etc.
- Etiquetas que componen la web semántica: algunas etiquetas que realmente no proponen nuevas funcionalidades, sino que sirven para componer sitios indicando qué son los bloques de código de una web, en lugar de cómo se deben representar.

**Nota:** Podríamos definir otras clasificaciones o incluso subgrupos en cada uno de los anteriores items, pero de momento está bien para aclarar cuáles son las novedades clave que trae HTML5.

Dentro de cada grupo tenemos multitud de nuevas etiquetas que comentaremos brevemente a continuación.

## Elementos que dan soporte a nuevas funcionalidades

Existen numerosas novedades dentro del HTML5 que se han representado por etiquetas o elementos de diversa índole. Algunas etiquetas son realmente nuevas y otras como EMBED ya existían y se han incorporado al estándar.

Etiquetas para Multimedia: Algunos elementos nuevos servirán para integrar contenido multimedia, pues sabemos que cada día esos nuevos tipos de información están más presentes en la Web.

- AUDIO: Para insertar sonido dentro de una web.
- VIDEO: Para insertar clips de vídeo.
- EMBED: Para embeber contenido externo de otro tipo, como el traído de diversos plugins que se comercializan actualmente o se comercializarán en el futuro.
- SOURCE: Permite especificar varias fuentes diferentes cuando se insertan elementos en AUDIO y VIDEO.
- TRACK: Permite especificar varias pistas de sonido o vídeo para los elementos AUDIO y VIDEO.

**Nota:** En DesarrolloWeb.com existen diversos talleres de HTML5 que explican diversos usos de las etiquetas AUDIO y VIDEO, que podemos leer si nos pasamos por el [Taller de HTML5](#).

### Nuevos elementos de formulario:

En el caso del HTML5 y los formularios tenemos que destacar que no solamente se han creado nuevas etiquetas, sino que se ha añadido soporte a las existentes anteriormente. De momento estas son las nuevas etiquetas que nos ofrece.

- METER: Para trabajar con medidas y escalas.
- PROGRESS: Implementa barras de progreso.
- DATALIST: Extensión para crear campos con funcionalidad de autocompletar.
- KEYGEN: Genera claves pública y privada para encriptación.
- OUTPUT: Realizar y mostrar cálculos matemáticos.

**Nota:** Para ver descripciones más detalladas sobre este asunto recomendamos la lectura del artículo [Nuevos elementos de formularios en HTML5](#).

Pero hablando de novedades en formularios, no debemos dejar de remarcar que HTML5 también trae diversos nuevos usos de INPUT, que nos sirven para especificar qué tipo de información concreta queremos introducir en ellos. Ello proporciona utilidad especial a esos campos INPUT, pensando también en los dispositivos móviles y la forma con la que tratan dichos campos. Todo eso se explica con detalle en el artículo [Las mejoras de los elementos INPUT de HTML5](#).

### Dibujos completos en HTML5, lienzo de CANVAS:

Luego tenemos una utilidad nueva que merece la pena verla por separado, puesto que nos va a transformar la manera con la que se experimentará la web. Se trata de un lienzo en donde se puede dibujar cualquier cosa e incluso hacer animación compleja.

CANVAS: Una etiqueta que genera un lienzo en la página donde realizar cualquier tipo de diseño, soporta dibujo de todo tipo de formas, degradados, imágenes, etc.

**Nota:** La etiqueta CANVAS del HTML simplemente delimita un área de la página donde se puede dibujar, pero para realizar esos dibujos se tiene que utilizar el lenguaje Javascript, a través del API de Canvas. Para aprender a diseñar en un CANVAS os recomendamos la lectura del [Manual del Elemento Canvas del HTML5](#).

## Elementos para la web semántica

Como decíamos, la segunda clasificación de las nuevas etiquetas del HTML5 está relacionada con lo que se llama la "web semántica". Vamos a dedicar otros artículos a estudiar en profundidad esta web semántica y por qué es importante utilizarla ya mismo en nuestros proyectos, pero de momento veamos a continuación una lista de elementos que han sido agregados al HTML para especificar qué son y no cómo se deben mostrar en la página.

### Secciones dentro de una página:

Algunas de las nuevas etiquetas nos sirven para decir qué secciones contiene una página.

- **ARTICLE:** Especifica un artículo, es decir, una unidad de contenido.
- **SECTION:** Es una sección dentro de un documento.
- **HEADER:** La cabecera de una página.
- **FOOTER:** El pie de página o informaciones que formen el pie de una sección.
- **ASIDE:** Es una parte de la web que muestra contenido accesorio, generalmente colocado en un panel lateral.
- **NAV:** con el que colocar el navegador principal de una página web.

### Otros tipos de informaciones:

Hay otras muchas etiquetas que nos sirven para definir qué es el contenido que se escribe dentro.

- **BDI:** Define una parte del texto que debe ser entendido aparte de la línea de contenido que se esté escribiendo.
- **MENU:** una lista de opciones que formen parte de un menú.
- **COMMAND:** Uno de los elementos o botones de un menú de opciones.
- **DETAILS:** Detalles o información suplementaria que se puede ver u ocultar por el usuario.
- **SUMMARY:** Encabezamiento para detalles especificados en DETAILS.
- **FIGURE:** es un contenido que ilustre el artículo, como fotos, diagramas, ilustraciones, etc.
- **FIGCAPTION:** El pie o explicación de un FIGURE.
- **HGROUP:** Un grupo de encabezamientos, útil cuando se tiene diversos bloques de encabezamientos del mismo nivel H1, H2...
- **MARK:** Un texto o información que es remarcable.
- **TIME:** Para escribir una fecha, una hora o ambas.
- **WBR:** Define un posible salto de línea.

En el próximo artículo veremos unas explicaciones más conceptuales acerca de las etiquetas semánticas y cómo éstas deben cambiar el modo en el que se entiende la composición de una página web.

Artículo por *Miguel Angel Alvarez*

## Etiquetas semánticas del HTML5

*Qué son las etiquetas semánticas del HTML5, y por qué se recomienda usar aquellas que sirven para definir la estructura del documento en los sitios web modernos.*

El propio lenguaje HTML es una de las muchas tecnologías que se ha renovado con la llegada de HTML5. Se han incorporado numerosas etiquetas que no estaban en versiones anteriores del lenguaje de marcación, que podemos clasificar en función de su utilidad o características. Una de estas clasificaciones son las etiquetas semánticas, que ya comenzamos a explicar en el artículo anterior, y que trata sobre las [etiquetas nuevas del HTML5](#).

En el presente artículo vamos a explicar con mayor profundidad qué son las etiquetas semánticas y por qué deben ser utilizadas para mejorar la creación de un documento web. Veremos el conjunto de etiquetas semánticas que sirven para definir la estructura de un sitio y cómo nos pueden ayudar a que nuestra página sea indexada con mayor exactitud por los motores de búsqueda.

### Qué son etiquetas semánticas

**Semántica:** *Pertenciente o relativo a la significación de las palabras.* Esa es la definición de la Real Academia de la Lengua Española y es así de sencillo es entender este tipo de etiquetas del lenguaje HTML: Aquellas que dan un significado a las partes del documento.

Por decirlo de otra forma, estamos ante **etiquetas que indican qué es el contenido que contienen, en lugar de cómo**

se debe formatear al mostrar el documento HTML en un cliente web.

Para que todas nuestras dudas se despejen, podemos echar mano de un ejemplo. Pensemos en la etiqueta `STRONG` ¿Qué nos indica? Que un texto debe resaltarse con mayor fuerza con respecto a otros textos que no sean "strong", parecido a lo que nos indica la etiqueta `B`, que todavía es más concreta, al especificar que un texto debe ser colocado en negrita. Pensemos ahora en una etiqueta `DIV`, o `P`. La primera es una división y `P` es un párrafo. Todo esto no son más que maneras de decirle al navegador cómo debe formatear una página, qué debe separar en un texto, en una división normal, o en un párrafo, etc.

Pocas de las etiquetas del lenguaje de marcación anteriores a HTML5 indican al documento el significado de los elementos que contienen. O sea, pocas etiquetas indican qué son y qué función representan en el documento HTML, sino que la mayoría sirven para indicar cómo se deben ver en la página una vez renderizada por el navegador.

Pues bien, en HTML5 se han incorporado muchas etiquetas que sirven para indicar qué son y cuál es el significado de los elementos que contienen dentro. Esas son las etiquetas semánticas, que ganan mucha importancia en el marco del HTML y de la composición de un documento web por ayudar a motores de búsqueda como Google a indexar más correctamente los contenidos de un sitio.

Dentro del etiquetado semántico también tenemos varias funciones, pero para los que nos dedicamos a maquetar páginas web, hay un conjunto de ellas que nos interesan especialmente. Son las que sirven para definir el esquema principal del documento, como `HEADER`, `ARTICLE`, `FOOTER`, etc. Todas esas etiquetas semánticas nos indican qué es el contenido que engloban y cuál es su relación con el conjunto de elementos del documento HTML.

**Nota:** Hay que añadir que también existen otra serie de etiquetas semánticas nuevas en la versión 5 del lenguaje, que también dicen qué es lo que hay dentro, más que cómo debe representarse. Ejemplos de ellas los vimos en el anterior artículo, como `DETAILS`, `FIGURE`, etc. De momento nos interesan menos en lo que respecta a este artículo, pero también hay que tenerlas en cuenta.

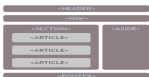
## Etiquetas semánticas estructurales

Veremos con detalle a continuación las etiquetas semánticas que sirven para definir la estructura de un documento HTML y por qué debemos incorporarlas cuanto antes a nuestras técnicas de desarrollo de sitios web.

Las etiquetas semánticas estructurales nos sirven para que los motores de búsqueda, o cualquier otro mecanismo automático que lea un sitio web, sepa con exactitud qué partes de su contenido corresponden a cada una de las partes típicas de un sitio. Generalmente, en cualquier documento tenemos una cabecera, un cuerpo y un pie de página, elementos que definen la estructura representados por diversas etiquetas.

De este modo, HTML5 tiene una etiqueta `HEADER`, que debe englobar aquellos elementos correspondientes a la cabecera de la página. Tiene un `FOOTER`, que debe englobar todas las etiquetas que componen el pie. Luego, para el cuerpo realmente no existe una etiqueta en concreto, sino que se representará por diversas otras como `SECTION`, `ARTICLE` o `ASIDE`. Existirán además otras etiquetas que representan elementos típicos como la barra de navegación, etiqueta `NAV`.

Observando esas etiquetas semánticas estructurales, cualquier sistema podría procesar la página y saber cómo está estructurada y a qué parte de las típicas de una web corresponde cada contenido. Podrá saber qué contenidos se han colocado en la cabecera, elemento `HEADER`, cuáles son los temas de los que se habla en una web, los `ARTICLES`, o cuáles son los enlaces a las secciones principales del sitio, colocados en la etiqueta `NAV`.



Existe pues una forma lógica de colocar todos los contenidos dentro de un sitio y unas etiquetas determinadas para contenerlos, atendiendo a la estructura que nosotros queremos comunicar a los sistemas que puedan procesar el documento. La imagen que acompaña a este texto expresa una de las posibles formas de ordenar el documento con las etiquetas que nos proporciona HTML5.

En la imagen se pueden ver las etiquetas estructurales más habituales, pero hay otras que podríamos utilizar dentro de las etiquetas semánticas. Además, la posición de estos elementos no tiene por qué ser así, incluso podrían anidarse de otras maneras. Quizás en nuestro sitio queramos colocar el navegador en la cabecera, en el lateral derecho o dentro del `SECTION`. Sería independiente siempre que utilizemos la etiqueta `NAV` para contenerlo. Asimismo, podrías tener más de un `SECTION` en el cuerpo de la página, sueltos o colocados dentro de otras etiquetas como `ASIDE`. Esos `SECTION` adicionales podrían tener otros `ARTICLE` o incluso, dentro de un `SECTION`, hacer otro esquema de cabecera y pie

anidando otros HEADER y FOOTER.

Como vemos, las posibilidades de agrupar, ordenar y colocar estas etiquetas para definir la estructura correría por nuestra cuenta, pudiendo generar distintos tipos de estructuras según las necesidades de cada proyecto. No obstante, siempre hay algunas reglas básicas que sí se deben cumplir, como que las unidades de contenido estén en los ARTICLE o que solo exista un navegador con NAV. En casos comunes, que deseemos tener otros navegadores, podríamos optar por echar mano de otras etiquetas semánticas como MENU, asignando al navegador principal la etiqueta NAV y a otros secundarios el MENU.

**Nota:** Realmente, las etiquetas estructurales están ahí y nosotros somos los que podemos utilizarlas a nuestro gusto, siempre que respetemos la significación que se les ha dado. Hay discusiones de sobra para interpretarlas de una manera o de otra y para decir cómo deberían ser utilizadas. Podemos usar el sentido común para imaginar lo que pueda o no ser correcto.

## Las etiquetas semánticas no definen un estilo predeterminado

Quizás este punto está de más, porque se debe entender a raíz de lo dicho hasta ahora, pero conviene apuntarlo bien para no llevarnos a engaño. Las etiquetas semánticas no tienen un estilo predeterminado que el navegador nos vaya a asignar. Es decir, porque HEADER signifique que es una cabecera, el navegador no va en ningún caso a posicionar el elemento en la parte de arriba del documento. Lo mismo con FOOTER, que no lo colocará en la parte de abajo, o ASIDE al lateral.

Somos nosotros los que, mediante CSS, debemos asignar los estilos que queramos se aplique a cada uno de esos elementos del documento HTML. Vuelvo a insistir en que las etiquetas semánticas no indican la forma como se deben mostrar, sino el significado que tienen dentro del contexto del documento.

En este sentido, cabe recordar que desde hacer tiempo se viene trabajando para separar el contenido de la forma y las etiquetas semánticas suponen un paso más hacia ese objetivo. Por ello, todos los estilos y posicionamiento deben quedar a cargo de las CSS.

## Por qué debemos utilizar las etiquetas semánticas y cómo nos ayudan

A algunos les bastará saber que Google recomienda su uso para estar convencidos de utilizarlas, pero las razones de fondo no son caprichosas de una empresa en concreto. Hay muchas compañías grandes detrás de estas nuevas etiquetas semánticas y estructurales y se han creado para facilitar la vida, no solo de Google, sino de cualquier otro sistema que pueda procesar un documento HTML.

La idea con estas etiquetas es añadir información a la Web, como metadatos semánticos que puedan ser interpretados por cualquier sistema informático. Con ello se puede facilitar que cualquier máquina que lee un documento HTML y no está apreciando cómo se presenta en pantalla, sino simplemente su código fuente, pueda saber a qué parte típica pertenece cada pedazo de código, para entender mejor qué es lo que se está tratando en cada caso.

En lo que respecta al motor de Google, o cualquier otro robot de indexación, será capaz de leer el documento y saber qué parte es la cabecera, que parte contiene los enlaces a las principales secciones del sitio, qué parte contendrá las unidades de contenido, etc.

Así Google sabrá mejor de qué trata tu página, o cuales son los distintos temas que se están desarrollando en estos momentos. Sabrá qué enlaces debe prestar especial atención para recorrer tu sitio completo y cuáles de esos enlaces son menos importantes. En definitiva, las máquinas sabrán identificar mejor qué hay en una página, y cuáles son las partes más importantes de entre todo el contenido que tiene.

Por poner ejemplos más concretos, Google será capaz de ver tus ARTICLE para saber qué post has publicado y cuál es el contenido más interesante que estás tratando en estos momentos. Podrá ver tu NAV para saber cómo organizas las secciones de tu sitio. Los POST los puede indexar de manera preferente como nuevos artículos que publicas con cierta periodicidad.

Un ejemplo interesante es lo que Google podrá hacer con tus enlaces del NAV. Si te has fijado, cuando buscas por un sitio en concreto a partir de su nombre, muchas veces aparece ese sitio en primer lugar con una serie de sub-enlaces con las secciones principales. Pues con tu navegador colocado en NAV estarías diciéndole a Google que esas son las secciones principales del sitio, que debe mostrar como páginas alternativas cuando buscan por él en el buscador.



## Conclusión

Estas etiquetas semánticas del HTML5 son un paso adelante en cuanto a crear una web con un contenido mejor formado, más útil y procesable con mayor facilidad por cualquier sistema informático. Hoy puede que solo nos interese como una manera de optimizar los sitios para una mejor indexación en buscadores, pero sentarán las bases para conseguir en el futuro un ecosistema web más saludable.

En el próximo artículo explicaremos algunos detalles sobre la compatibilidad de las etiquetas semánticas con los navegadores que se usan en la actualidad y cómo resolver casos indeseables que ocurren cuando los navegadores no las interpretan correctamente.

Artículo por *Miguel Ángel Álvarez*

## Etiquetas semánticas de HTML5 y compatibilidad

*Navegadores como Internet Explorer principalmente, no son compatibles con las etiquetas semánticas del HTML5. Qué podemos hacer para aumentar la compatibilidad con los elementos estructurales.*

Las etiquetas que han sido incorporadas en el HTML5 alteran la manera en la que se maqueta un sitio web, aportando metadatos semánticos que nos indican qué son y qué función realizan en el documento HTML completo. Son un gran avance para lo que se llama la Web Semántica y permiten que los sitios web se puedan analizar correctamente por sistemas informáticos automatizados, como los robots indexadores.

Debido a las ventajas que hoy ofrecen, y a la recomendación de uso que grandes empresas como Google fomentan, conviene tenerlas en cuenta a la hora de generar el código de las páginas de un sitio. Ahora bien, ¿Qué pasa con los navegadores antiguos que no entienden HTML5? ¿Qué pasa con las numerosas versiones de Internet Explorer que aún usan una ingente cantidad de internautas?

El problema ante estos navegadores es que no son compatibles con etiquetas semánticas y estructurales como HEADER, FOOTER, ARTICLE, etc. Por ello, una página que se vea en uno de esos clientes web no compatibles estará totalmente descuadrada. No aceptará estilos CSS sobre esos elementos ni procesará aquellas órdenes que realicemos sobre esos elementos del DOM, con lo que la experiencia de usuario para las personas que visiten el sitio desde navegadores antiguos será totalmente desvirtuada.

En el presente artículo de DesarrolloWeb.com veremos con mayor detalle cuáles son esos problemas de compatibilidad de las etiquetas nuevas del HTML5 con navegadores antiguos y cuáles son las posibles soluciones que podemos aportar con facilidad.

**Nota:** Si quieres saber más acerca de los nuevos elementos semánticos del nuevo lenguaje de la web, te recomendamos la lectura del artículo [Las etiquetas semánticas de HTML5](#).

Podemos decir con alivio que hoy la mayoría de los navegadores es compatible con las etiquetas estructurales y semánticas. Navegadores como Firefox, Chrome, Opera o Safari hace tiempo que incorporaron esta parte del estándar que viene con HTML5. Además, como estos navegadores se actualizan con suma facilidad, hasta el punto que muchas veces el proceso es totalmente transparente para el usuario, podemos tener certeza que navegamos con un software que está en la última línea en cuanto a compatibilidad con los nuevos estándares. Sin embargo, esta historia no es tan bonita cuando hablamos de Internet Explorer y hay que recordar que ¡hoy continúa siendo el navegador mayoritario!

## Internet Explorer y las etiquetas semánticas

Internet Explorer a partir de la versión 9 ya es compatible con diversas tecnologías relacionadas con el HTML5 y, entre ellas, con las etiquetas semánticas. Sin embargo, se trata de un navegador que no se actualiza por sí solo, sino que hay que hacerlo "a mano" y a menudo los usuarios no encuentran sencillo el procedimiento.

**Nota:** Podemos decir que ese modo de trabajo es una elección de Microsoft, que opina que los navegadores no deberían cambiar de versión de manera automática, para no producirse problemas derivados de esas actualizaciones y sitios o servicios web que, de la noche a la mañana, dejan de funcionar porque la nueva versión los reconoce de otra manera. Esta estrategia parece que no tiene mucha tendencia a cambiar por parte de la compañía, que siente que de esa manera está protegiendo a miles de corporaciones que utilizan su sistema operativo en millones de ordenadores.

Esa dificultad de actualización, unida a que Internet Explorer 9 no es compatible con sistemas todavía hoy populares como Windows XP, provocan que hoy muchas personas no puedan visitar correctamente un sitio que utiliza características del HTML5 como las etiquetas semánticas.

Hasta el propio Microsoft está haciendo esfuerzos para eliminar del mapa a Internet Explorer 6. A pesar de ello, hoy continúa siendo el navegador de aproximadamente un 7% de los usuarios. La versión 7 es utilizada por un 4% de los navegantes, mientras que Internet Explorer 8 es el navegador más usado en la actualidad con el 26% de los usuarios de la Web ([datos de abril de 2012](#)). Es por ello que resulta casi inviable olvidarnos de todas estas personas cuando desarrollamos un sitio web.

## Efectos indeseables en navegadores no compatibles con HTML5

Podemos ver un ejemplo de sitio web diseñado con etiquetas semánticas del HTML5 para ver cómo se renderiza en un navegador no compatible con ellas, pues es la mejor manera de entender qué es lo que pasa.



La imagen anterior corresponde con el navegador Google Chrome, pero realmente se vería exactamente igual en otros clientes típicos como Firefox u Opera.

El problema, como venimos diciendo, es que hemos compuesto el sitio de EscuelaIT utilizando etiquetas semánticas del HTML, pero Internet Explorer no las reconoce.

```
<HEADER>
  contenido de la cabecera
  <NAV>Navegador de secciones</NAV>
</HEADER>
...

```

Luego hemos aplicado los estilos a estos elementos con CSS:

```
header {
  height: 190px;
  padding: 0 10%;
  background-image: linear-gradient(left top, #F4F4F4 30%, #E0E0E0 87%);
  clear: both;
}
header nav {
  margin: 1em 0;
  padding: 0.8em 0;
  border-radius: 6px;
}

```

Como estos elementos no son reconocidos por Internet Explorer, los estilos CSS no son aplicados y ocurre que todas las etiquetas quedan sin formato.

Claro que podríamos haberlo solucionado cambiando el código HTML, y utilizando elementos DIV con clases o identificadores, para luego aplicarle estilos de la manera tradicional.

```
<DIV id="header">
  contenido de la cabecera
  <DIV id="nav">Navegador de secciones</DIV>
</DIV>

#header{
  ...
}

```

```
}  
#nav{  
  ...  
}
```

Pero esta manera de trabajar, aunque se vería correctamente en todos los navegadores y nos ahorraría muchos problemas, no tendría las ventajas de las etiquetas semánticas y los robots de indexación no entenderían tan bien su contenido, entre otras muchas situaciones.

## Hacer que navegadores antiguos soporten las etiquetas semánticas

Por unos pocos navegadores que no son compatibles con las posibilidades del HTML5 es una pena que no podamos incorporar las ventajas que ellas nos traen. Sin embargo, existen diversos modos de proveer compatibilidad con HTML5 en navegadores poco actualizados.

Lo cierto es que los métodos para mejorar las características de los navegadores son bastante sencillos de incorporar. Simplemente debemos realizar algunas optimizaciones en el código de nuestro sitio web, para cargar algunas librerías que, mediante Javascript, ofrezcan soporte a navegadores que no tienen de manera nativa compatibilidad con las características del HTML5, en este caso las etiquetas semánticas.

Existen varias librerías para hacer esto, de las cuales voy a comentar brevemente un par de ellas.

**Modernizr:** Una librería para compatibilidad con HTML5 muy completa, que contiene un componente llamado "html5shiv" que permite que los navegadores comprendan las etiquetas semánticas. Simplemente tenemos que cargar la librería y listo!

<http://modernizr.com/>

**ie7-js:** Un script Javascript para hacer compatible a Internet Explorer, desde las versiones 5 a la 8, con los navegadores modernos, lo que incluye algunos asuntos como los elementos semánticos de HTML5, pero también otros fallos comunes de algunas versiones de los exploradores, como el uso de los PNG transparentes.

<http://code.google.com/p/ie7-js/>

**Nota:** Explicaremos con detalle ambas librerías en los siguientes artículos a publicar en este manual, pero de momento quiero informar que sobre Modernizr tenemos un completo tutorial en DesarrolloWeb.com: [Manual de Modernizr](#).

## Conclusión

Estamos ante etiquetas que incrementan las posibilidades de nuestros documentos HTML, para que tengan una estructura más clara y sean mejor indexados por los motores de búsqueda. Hoy, aunque no todos los navegadores entienden las etiquetas semánticas estructurales, se recomienda desarrollar sitios web que las utilicen.

Para los casos en que navegadores antiguos no entiendan esas etiquetas, existen librerías Javascript que nos pueden solucionar la vida y aportar de una manera sencilla soporte a HTML5.

*Artículo por Miguel Ángel Álvarez*

## IE7-js compatibilidad de Internet Explorer con navegadores modernos

*Libería Javascript ie7-js, para hacer que Internet Explorer se comporte como los navegadores modernos y acepte los estándares del HTML5.*

Estamos ante una librería bien sencilla de implementar, pensada para acabar con los dolores de cabeza que Internet Explorer nos produce a muchos desarrolladores cuando se trata de hacer que un sitio web se vea igual en este navegador que en otros clientes web modernos.

La librería ofrece básicamente compatibilidad con Microsoft Internet Explorer ¡¡desde la versión 5.5!! hacia adelante, por medio de unos *scripts* Javascript muy sencillos de instalar. Realmente será copiar y pegar un pequeño código que incluye el script, con un condicional para que solamente lo cargue cuando sea necesario.

**Nota:** No ofrece compatibilidad con Internet Explorer 9, pues este navegador ya ha dado un salto de calidad, o mejor dicho, de respeto a estándares abiertos, por lo que su forma de trabajar ya es muy compatible con los navegadores de última generación.

La librería en cuestión se llama ie7-js y la podemos encontrar en la URL  
<http://code.google.com/p/ie7-js/>

### Para qué nos sirve IE7-js

Esta librería tiene varias funciones, todas enfocadas en conseguir que Internet Explorer se comporte mejor al representar nuestro sitio web. Básicamente lo que hace es adecuar la manera en la que las versiones antiguas de IE interpretan una página web, para que trabajen tal como sugieren las especificaciones de la W3C.

Para empezar, nos puede servir para algo tan sencillo como evitar tener que escribir estilos CSS y código HTML específico para Internet Explorer, en cada una de sus versiones anteriores a la 9. El resultado redundará en el ahorro de mucho tiempo, en horas de optimización de un sitio web, simplemente para conseguir que se vea correctamente en Internet Explorer.

Los distintos scripts de IE7-js nos sirven para conseguir solucionar muchas diferencias u omisiones de funcionamiento de los navegadores de Microsoft, en el código HTML y CSS, en temas como:

- Posicionamiento de capas en modo "fixed".
- Incorporación de selectores avanzados en las CSS como parent > child
- Soporte a atributos variados como min-width y max-width, así como min-height y max-height.
- Soporte a pseudo-elementos de las CSS, como :hover :first-child ::after, :focus, :last-of-type...
- Interpretación errónea del modelo de caja
- Elementos semánticos del HTML5, como article, header, footer, aside, audio, video, canvas...
- Estilos CSS para opacidad de elementos.
- Soporte para transparencia de PNG en IE6.

La lista es interminable y se puede ver en la propia documentación de la librería: [Propiedades, selectores y demás reparaciones que realiza IE7-js](#).

### Cómo cargar la librería IE7-js

La manera de incorporarla a una página web es muy sencilla. Para ello simplemente tienes que escribir un código en la cabecera de tu web que ellos mismos te proporcionan. Además, los scripts están separados en diversos bloques que debes cargar opcionalmente, dependiendo del grado de compatibilidad que quieras asegurar hacia adelante.

## Compatibilidad de IE5 e IE6 con IE7:

Si deseas que Internet Explorer 6 (o incluso IE 5) funcionen igual que Internet Explorer 7, debes escribir el siguiente código.

```
<!--[if lt IE 7]>
<script src="http://ie7-js.googlecode.com/svn/version/2.1(beta4)/IE7.js"></script>
<![endif]-->
```

**Nota:** Como puedes observar en el código anterior, se coloca el script dentro de un comentario HTML, por lo que los navegadores modernos no interpretarán el código en ningún caso. Luego, dentro del comentario, se inserta un condicional que solamente Internet Explorer identifica [if lt IE 7]. Eso indica a IE que lo que hay dentro del condicional solo tiene que ser interpretado por IE menor que 7. Por ello ese script sólo lo ejecutarán versiones anteriores a IE7. De este modo, se asegura que solo los navegadores IE5 e IE6 van a cargar el script de compatibilidad con IE7, para renderizar el documento HTML tal como lo haría la versión 7 del navegador de Microsoft.

## Compatibilidad de IE7, o menor, con IE8:

Si deseas que Internet Explorer 7, así como las versiones anteriores, se comporten igual que Internet Explorer 8, debes escribir el siguiente código.

```
<!--[if lt IE 8]>
<script src="http://ie7-js.googlecode.com/svn/version/2.1(beta4)/IE8.js"></script>
<![endif]-->
```

## Compatibilidad de IE8, o menor, con IE9:

Por último, ofrecen el script completo que aseguraría la compatibilidad con cualquier navegador menor o igual que IE8 (es decir, desde IE5 hasta IE8), para que se comporte igual que Internet Explorer 9.

```
<!--[if lt IE 9]>
<script src="http://ie7-js.googlecode.com/svn/version/2.1(beta4)/IE9.js"></script>
<![endif]-->
```

De este modo, cubrimos todos los Internet Explorer antiguos y hacemos que se comporten como la versión 9, que ya funciona bastante bien.

**Nota:** Por si no ha quedado claro, no es necesario incluir todos los scripts, sino uno de ellos, dependiendo del grado de compatibilidad que queremos alcanzar. Por ejemplo, si quieres que todos los Internet Explorer interpreten el HTML y CSS como IE9, solamente tendrías que incluir el último script.

## Sobre soporte a transparencia de PNG

Puede que hayas observado en Internet Explorer 6 que los archivos con imágenes PNG que incorporan transparencias se ven incorrectamente. Este es un asunto desagradable que muchos de nosotros obviamos, por no querer prestar atención a un navegador tan antiguo. Sin embargo, para otros desarrolladores la transparencia PNG en Internet Explorer 6 puede ser un asunto crítico.

Este script lo soluciona también, lo que es de agradecer, pero debemos tener en cuenta varios detalles. Hay uno que resulta importante señalar.

Solamente soluciona el problema de la transparencia para los archivos que hayas nombrado como \*-trans.png. Por ejemplo miarchivo-trans.png. Esto se realiza por razones de rendimiento, pero si lo deseamos, podemos activar a todos los archivos de formato PNG con la declaración de una variable.

```
var IE7_PNG_SUFFIX = ".png";
```

Ese código lo tienes que ejecutar, con un script Javascript, antes de cargar las librerías.

## Conclusión

De momento eso es todo. Yo he utilizado este script con éxito en una web donde quería que me interpretara las etiquetas estructurales de HTML5 y los resultados han sido excelentes. Obviamente, soluciona muchos otros problemas que nos pueden venir bien, pero ya solo por ese detalle merece la pena incorporarlo.

El uso del script de compatibilidad, como se ha podido ver, es muy simple. No importa cuan complicado puede ser por dentro, lo único que tenemos que hacer es incluirlo en la cabecera y desarrollar nuestra web sin importarnos todos esos asuntos relativos a las versiones antiguas de Internet Explorer que tanto nos molestan y nos dan trabajo.

Artículo por Miguel Angel Alvarez

## Compatibilidad con etiquetas semánticas del HTML5 con html5shiv

*El componente html5shiv de las librerías Modernizr nos ayuda a crear webs compatibles con las etiquetas nuevas de HTML5 que funcionan en todos los navegadores.*

Modernizr está ocupando un lugar destacado entre las herramientas habituales de cualquier desarrollador que quiera realizar webs de última generación, compatibles con la mayoría de los navegadores web, incluso los más antiguos. Existen diversas utilidades que nos pueden facilitar mucho la vida cuando realizamos un sitio web en el que queremos incorporar características avanzadas de HTML5, como venimos relatando en el [Manual de Modernizr](#).

En esta ocasión os queremos hablar de uno de los componentes de Modernizr, que nos ofrece soporte a etiquetas que han aparecido nuevas en la versión más reciente del lenguaje de la web, como son las etiquetas semánticas y etiquetas estructurales del HTML5.

### Script html5shiv para habilitar HTML5 en Internet Explorer

El componente que queremos comentar en este caso, que forma parte de Modernizr, pero que también podemos obtener por separado se llama html5shiv. Incorpora, gracias a la inclusión de un script Javascript, soporte para los nuevos elementos o etiquetas de HTML5.

Lo podemos encontrar de manera predeterminada en las descargas que realizamos con Modernizr, pero también lo podemos descargar de manera independiente en <https://github.com/aFarkas/html5shiv>

También lo encontraremos disponible comprimido y listo para usar en IE, en otros dos repositorios diferentes de [Google Code](#) y <http://code.google.com/p/html5shiv/>

El script no solo provoca que versiones antiguas de Internet Explorer entiendan las etiquetas nuevas del HTML5, sino que también incorpora unos estilos predeterminados para esos nuevos elementos, como los ARTICLE o SECTION.

Pero, aunque el caso más relevante de incompatibilidad con HTML5 es el de Internet Explorer, porque los navegadores no se actualizan por sí mismos, otros clientes web también tienen bajo soporte a las nuevas etiquetas incorporadas en la última versión del lenguaje de la web. Es por ello que este proyecto también ofrece soporte a Safari 4.x, Safari basado en iOS 3.x, y Firefox 3.x.

### Usar html5shiv de manera independiente

Si queremos utilizar esta librería por separado (o sea, sin formar parte del paquete de *scripts* Modernizr) basta con que insertemos el siguiente código en la cabecera de nuestro documento HTML:

```
<!--[if lt IE 9]>
<script src="dist/html5shiv.js"></script>
<![endif]-->
```

Esto hace que, mediante un condicional que solo lee Internet Explorer, se cargue el script únicamente para los navegadores que lo necesitarían.

**Nota:** En este caso además vemos que el condicional provocará que el script sólo lo lean los IE con versión menor que la 9. Esto es porque IE9 ya es capaz de entender las etiquetas que han sido incorporadas en HTML5.

## Usar html5shiv como parte de Modernizr

Si estamos pensando en usar diversas capacidades del HTML5 y CSS3, además de las etiquetas nuevas del lenguaje, podemos considerar descargarlo como uno de los paquetes incorporados a la super-librería Modernizr. Ello nos puede ayudar a compatibilizar muchas otras características de la nueva versión del lenguaje para la web.

En este caso tendríamos que descargar Modernizr asegurándonos que se incluye html5shiv. Para la tranquilidad del lector, estas librerías ya se encuentran activas de manera predeterminada en la descarga de Modernizr, por lo que realmente no tenemos que hacer mucho.

De todos modos, en la página de descarga podemos fijarnos que esté seleccionada html5shiv, que se encuentra en el recuadro "Extra".



¡Así de sencillo! Con esto habremos descargado Modernizr y al ponerlo en marcha en una página web tendremos automáticamente soporte a las etiquetas de HTML5, cuando de manera nativa no estén disponibles en nuestro navegador.

**Nota:** Puedes leer el artículo [Primeros pasos con Modernizr](#) para aprender a cargar las librerías en una página web, aunque es básicamente incluir el script Javascript que acabas de descargar.

## Conclusión

Con este proyecto, cualquier sitio web realizado con las etiquetas nuevas en HTML5 podrá verse de manera correcta en todos los navegadores. Se trata de un script muy sencillo de usar, que nos evitará la pregunta típica que algunos nos hacemos todavía: ¿desarrollo pensando en HTML5 o no?

La verdad es que scripts como este nos solucionarán dudas y más que un quebradero de cabeza cuando estamos dando nuestros primeros pasos en HTML5.

*Artículo por Miguel Ángel Álvarez*

## Nuevos elementos de formularios en HTML5

*Conociendo algunos de los nuevos elementos con los que podemos contar en los formularios web a partir de HTML5.*

Durante los últimos meses HTML 5 está en boca de todos. Se ha generando gran expectativa alrededor de un sin número de tecnologías que nos trae el nuevo lenguaje de la web, no solo por parte de los desarrolladores, sino también de usuarios, debido a que las nuevas especificaciones del HTML mejoran sensiblemente la experiencia de la navegación.

Con HTML5 se pretende facilitar el desarrollo aplicaciones web, las cuales siguen unos parámetros de compatibilidad, tanto usuarios de ordenadores de escritorio, como para los que usan o navegan a través desde un dispositivo móvil. En este artículo vamos a tratar una de esas utilidades que tienen como objetivo mejorar las capacidades de las aplicaciones web. Se trata de los nuevos elementos para formularios HTML.

Los formularios en HTML5 se extienden para dar respuesta a diversas necesidades que se habían detectado en el pasado y en concreto con la llegada de los dispositivos de movilidad. Smartphones, tablets, etc. ganan cada día más aficionados, debido a su practicidad y portabilidad. Hoy estos dispositivos tienen gran potencia, a nivel de hardware y software, y ante estos nuevos avances la web debía responder con algunas herramientas para mejorar la usabilidad, en este caso de los formularios HTML clásicos.

Con las nuevas especificaciones del lenguaje de la web, se va mejorando en gran medida la experiencia de navegación e interacción del usuarios. Con el uso de dispositivos móviles táctiles, no se cuenta con un teclado físico, por eso las aplicaciones y sistemas operativos implementan lo que se llama el teclado virtual, que le permita al usuario realizar la entrada de texto de una forma similar a la que se utiliza en los ordenadores de escritorio.

A lo largo de este artículo y algunos otros que vendrán a futuro, estaremos hablando y viendo de primera mano todas las bondades con las que cuentan los formularios web. En algunos casos se trata de elementos nuevos para los formularios, en otros casos simplemente se reforman elementos existentes para ofrecer alguna ventaja. En general al desarrollador le caen como anillo al dedo, pues permiten ahorrarse diversos trabajos que teníamos que realizar con algunas tecnologías auxiliares como javascript.

**Nota:** Con toda esta evolución del lenguaje de la web, se hace un poco más fácil la escritura de aplicaciones web, pues recordemos que con HTML5 no solo se crean nuevas etiquetas, sino que también hay una evolución de tecnologías auxiliares como el mismo lenguaje del cliente Javascript, además del DOM, y una muy importante nueva versión del popular CSS. Todos éstos contribuyen a una nueva forma de concebir a la web tanto para los que desarrollamos, como para los que exploran la web.

## Los nuevos elementos o etiquetas con las que podemos contar en un formulario HTML5

En esta sección veremos 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 se difícil encontrarle una utilidad dentro de nuestras aplicaciones web, sin embargo, mencionaremos los cinco nuevos elementos aunque no vayamos a trabajar con todos en esta serie de artículos sobre los formularios y sus nuevas herramientas y elementos.

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

Los nuevos elementos son los siguientes:

- meter
- progress
- detalist
- keygen
- output

## Algunos detalles sobre los cinco nuevos elementos de los formularios

Como hemos podido ver anteriormente son cinco nuevas etiquetas, con las que podemos contar en los formularios web. A continuación vamos a hacer mención de cada una, además de los posibles usos que podemos dar en nuestras aplicaciones web, con el propósito de que todos ustedes se hagan una idea sobre lo que podemos lograr con estas nuevas etiquetas.



**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.

De momento, dejamos por aquí este listado de nuevos elementos de formulario, citando a los lectores a un próximo artículo, donde veremos el uso de algunos de estos nuevos elementos.

*Artículo por Dairo Galeano*

## Las mejoras de los elementos INPUT de HTML5

*Una breve descripción de las nuevas características presentes en los elementos de formulario INPUT en HTML5.*

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.

## Doce tipos nuevos de INPUT para mejorar el trabajo con formularios

Para los que en algún momento sintieron que los INPUT disponibles en los formularios clásicos se quedaban cortos, HTML5 tiene la respuesta. A decir verdad, muchos habremos podido pensar eso en alguna ocasión, sobre todo cuando hemos tratado de desarrollar aplicaciones dirigidas a dispositivos móviles. HTML5 está pensado para ser usado en múltiples entornos, y por tal razón se han creído doce nuevos tipos de INPUT.

**Nota:** Aunque pueda parecer que no tiene demasiada importancia la creación de todos estos tipos de INPUT nuevos, podemos decir que es un gran paso adelante. Aun más ahora, que se pueden desarrollar aplicaciones nativas para dispositivos, con ayuda de algunos framework como Titanium o Phonegap, del que publicaremos un manual en Desarrolloweb.com muy pronto. Incluso, como exploraremos también en este sitio, con HTML5 vamos a poder crear aplicaciones nativas en Windows 8. Como puede verse con el nuevo lenguaje de la web se nos abren nuevos horizontes y todas estas nuevas implementaciones serán muy utilizadas en todos esos nuevos entornos de aplicaciones.

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.

## Más allá de los nuevos INPUT: nuevos atributos

Con todos los nuevos tipos INPUT se logra un gran avance con respecto a los formularios clásicos de la web. Algunas de las ventajas pueden ser observables en dispositivos touch (con pantallas táctiles).

**Nota:** Si quieres ver los nuevos INPUT funcionando en navegadores de escritorio, recomendamos usar la última versión de Google Chrome (la versión 13 en adelante debería funcionar). Firefox también puede servir, puesto que se actualiza bastante rápido.

Si bien los nuevos INPUT son revolucionarios, también existen varias ventajas que nos traen diversos nuevos atributos, venidos también a raíz de la evolución de los formularios de HTML5. Con esos atributos nuevos podemos hacer cosas realmente interesantes sin mayores problemas. En este artículo vamos a realizar un ejemplo con dos de ellos.

### Atributo placeholder:

Sin duda habrás visto en muchas ocasiones un efecto sobre los campos INPUT, en páginas como Twitter, donde hay un texto que indica la información que el usuario debe ingresar. Al situar el foco encima del campo, ese texto desaparece para que el usuario escriba lo que desee. Pues bien, ese efecto se consigue con el atributo placeholder de una manera inmediata. Podemos ponerlo en práctica con un código como el siguiente.

```
<input type="text" placeholder="Ingrese el nombre">
```

Solo hay asignar al atributo placeholder el texto que deseamos que aparezca en el INPUT, y una vez se sitúe el foco en el campo de texto, simplemente desaparece.

### Atributo autofocus:

El efecto logrado mediante el atributo autofocus es igual al de las páginas de búsqueda como Google, donde al cargar la página el foco de la aplicación ya se encuentra en un INPUT. Su uso sería tan simple como este.

```
<input type="search" style="border-color:#12c5a1" autofocus>
```

Para mayor claridad os dejamos un ejemplo básico con estos atributos asignados a campos INPUT.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Prueba1</title>
<style>
input{
border-radius:7px;
border-color: #cfc;
}
</style>
</head>
<body>
<input type="text" placeholder="Ingrese el nombre">
<input type="search" style="border-color:#12c5a1" autofocus>
</body>
</html>
```

En futuros artículos usaremos más utilidades de los formularios HTML5.

Artículo por *Dairo Galeano*

## Nuevos atributos HTML5 para Rich-text editing

*Explicaciones generales de los nuevos atributos HTML5 contenteditable y spellcheck y herramientas básicas que podemos usar a partir de HTML5 para edición de contenidos enriquecidos Rich-text editing.*

HTML5 viene con muchas mejoras, como bien sabemos. En este artículo vamos a ver dos atributos nuevos que nos sirven para editar de una manera cómoda elementos de la página. Es decir, cómo realizar elementos como etiquetas cuyo contenido se puede marcar como editable por el usuario. Además, veremos otro atributo que jugará muy a favor de la creación de un contenido correcto gramaticalmente, que es el llamado spellcheck. Ambos atributos, junto con otras de las nuevas características de HTML5 que repasaremos a continuación, serán la base para hacer con Javascript contenido editable enriquecido por medio de HTML, osea, editar contenido y que además tenga un formato básico, como negritas, itálicas, etc.

En este artículo nos dedicaremos simplemente a conocer las nuevas características del HTML5 para hacer contenido editable, dejando la parte del Javascript para un futuro [artículo donde usaremos el método execCommand para crear una interfaz de usuario en Javascript que nos permita asignar formato e los elementos](#).

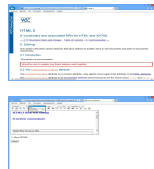
En el presente artículo trataremos los siguientes puntos:

- El nuevo atributo contenteditable
- El nuevo atributo spellcheck
- La edición simple de textos con formato HTML
- Los atajos del teclado y el menú emergente en las áreas de edición



### Referencias oficiales

- [www.w3.org/TR/2008/WD-html5-20080610/editing.html](http://www.w3.org/TR/2008/WD-html5-20080610/editing.html)
- [www.w3.org/TR/2011/WD-html5-20110525/editing.html#contenteditable](http://www.w3.org/TR/2011/WD-html5-20110525/editing.html#contenteditable)
- [developer.mozilla.org/en-US/docs/Rich-Text\\_Editing\\_in\\_Mozilla](http://developer.mozilla.org/en-US/docs/Rich-Text_Editing_in_Mozilla)



### Lo básico de "contenteditable"

Mediante este atributo podremos realizar ediciones de texto con todo el potencial del lenguaje HTML en el interior de nuestras propias páginas web. El texto en sí esta etiquetado con el mismo lenguaje de marcas que el documento HTML de la página web. Esto es, las áreas de texto pueden convertirse en áreas de edición, de manera que podremos realizar los formateos y gestiones tales que –entre otros–:

- Resaltados, negrilla, itálica, subrayado, ...
- Encabezados, títulos y subtítulos
- Manejo de estilos de todo tipo
- Copiado y pegado de textos
- Hipervínculos
- Viñetas, lista y lista numerada
- Fuentes y tamaños de fuente
- Colores de fuente y fondos
- ...

Por el contrario, no dispondremos de facilidades para el trabajo con imágenes ni de forma directa, acceso a archivos –abrir y guardar–.

El atributo `contenteditable` se mimetizaba, en versiones anteriores de HTML, mediante bibliotecas de JavaScript y similares, pero no es hasta su aparición en HTML 5 cuando ha comenzado a cobrar gran importancia por su simplicidad y potencia, siendo soportado por la totalidad de los principales navegadores de Internet –véase la tabla adjunta–

Chrom e	Firefox	I. Explorer	Opera	Safari	Android
6.0	3.5	6.0	8.0	3.2	3.0

La sintaxis es del todo simple:

```
<elemento contenteditable="true|false|inherit" [...]>
```

- True: su contenido será editable
- False: no lo será
- Inherit: será editable si el elemento padre lo es –si hereda la editabilidad–

Ejemplos de elementos sobre los que se puede aplicar son todos los que atañen a contenidos de texto, como párrafos, secciones, divisiones, etc. En los dos breves listados sintácticos que siguen se ilustra su uso.

```
<p contenteditable="true">
Escribir aquí ...
</p>
```

Aquí aplicado a una sección que incluye otros elementos como cabecera y párrafo:

```
<section contenteditable="true">
<h1>Titulo</h1>
<p>Escribir aquí...</p>
</section>
```

### Listado 1: Ejemplo simple de uso del atributo `contenteditable`

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" lang="es-es">
<title>contenteditable</title>
</head>
<body>

<section contenteditable="true">
<h1>Edición</h1>
<p>Escribir aquí...</p>
</section>

</body>
</html>
```



## El atributo "spellcheck"

Se trata de otro atributo nuevo de HTML 5, de una potencia extraordinaria. Es también un atributo global. Se realizará una comprobación ortográfica y gramatical de los textos contenidos dentro de la etiqueta que tenga el atributo, en función del idioma definido para la página HTML. De encontrarse un error, como en los procesadores de textos, se harán las propuestas oportunas o se permitirá incluir el término en un diccionario personal.

Con una sintaxis muy similar a `contenteditable`, puede y será muy recomendable utilizarse conjuntamente, como veremos en

nuestro ejemplo:

```
<elemento spellcheck="true|false" [...]>
```

- True: su contenido será comprobable
- False: no lo será

El atributo spellcheck es actualmente soportado por la totalidad de los principales navegadores de Internet –véase la tabla adjunta-, aunque, eso sí, en sus versiones más recientes.

Chrom e	Firefox	I. Explorer	Opera	Safari	Android
10.0	9.0	10.0	11.0	4.0	2.3

**Listado 2: Ejemplo simple de uso del atributo contenteditable con spellcheck**

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" lang="es-es">
<title>contenteditable</title>
</head>
<body>

<section contenteditable="true" spellcheck="true">
<h1>Titulo</h1>
<p>Escribir aquí...</p>
</section>

</body>
</html>
```



## Usemos estilos

Podremos, lógicamente, dar la apariencia deseada a nuestra área de edición mediante la utilización de estilos. En el ejemplo que sigue:

1. Marcamos un ancho
2. Marcamos un alto
3. Fijamos un borde
4. Marcamos un área libre entre el borde y el contenido
5. Fijamos que se realice un desplazamiento en el interior cuando se sobrepasan los límites, mediante barras de desplazamiento
6. Fijamos un color de fondo

**Listado 3: Ejemplo simple de uso del atributo contenteditable con spellcheck, después de aplicarle estilos para el formato del área de edición**

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" lang="es-es">
<title>contenteditable spellcheck="true"</title>
<style>
#textBox {
width: 600px;
height: 300px;
border: 2px #000000 solid;
padding: 10px;

```

```
overflow: scroll;
background-color: yellow;
}
</style>
</head>
<body>

<section id="textBox" contenteditable="true">
<h1 style="color:red">Título</h1>
<p>Escribir aquí...</p>
</section>

</body>
</html>
```



## Atajos del teclado y menú emergente

No hay un estándar definido de atajos de teclado, pero en todo caso podremos aprovechar los propios del navegador como los de cortar, pegar, seleccionar todo, ... Otros son de uso común como los de negrilla e itálica y otros más como los de subrayado dependerán del navegador en sí.

- Ctrl+B: Negrilla
- Ctrl+I: Itálica
- Ctrl+C: Copiar
- Ctrl+V: Pegar
- Ctrl+E: Seleccionar todo

Sí existe un menú emergente con opciones básicas, también dependen algunas del sistema operativo y del correspondiente navegador. Un ejemplo se muestra en la figura adjunta.



Ahora que conocemos todas estas cosas, estamos en condición de construir [una interfaz de usuario en Javascript que nos permita asignar formato e los elementos mediante el método execCommand](#).

*Artículo por Jaime Peña Tresanco*