

Secciones y esquema de un documento HTML5

La especificación HTML5 trae muchos nuevos elementos a los desarrolladores web, permitiéndoles describir la estructura de un documento web con semántica estandarizada. Este documento describe estos elementos y cómo usarlos para definir el esquema de cualquier documento.

A lo largo de este documento se empleará la palabra **esquema** en el sentido de esqueleto estructural o tabla de contenido, similar a cómo el índice de un libro muestra la estructura jerárquica de su contenido. Asimismo, usaremos **algoritmo de esquematizado** al proceso de construir el esquema infiriéndolo del contenido.

La estructura de un documento, por ejemplo la estructura semántica de lo que está entre `<body>` y `</body>` es fundamental para presentar la página al usuario. HTML4 usa la idea de secciones y subsecciones de un documento para describir su estructura. Una sección está definida por un elemento divisor HTML (`<div>`) con elementos de cabecera HTML (`<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, o `<h6>`) en él, definiendo sus títulos. La relación de estos elementos divisor y de cabecera HTML desemboca en la estructura del documento y su esquema.

Entonces el siguiente marcado:

```
<div class="seccion" id="zorro" >
<h1>El zorro</h1>
<p>En esta sección discutiremos sobre los aspectos menos conocidos del zorr
... bla, bla, bla ...
<div class="subseccion" id="zorro-habitat" >
<h2>Hábitat</h2>
<p>El zorro necesita un nido en donde poner sus huevos, de ahí que necesite
...bla, bla, bla...
</div>
</div>
```

genera el siguiente esquema:

- 1. El zorro
 - 1.1 Hábitat

Los elementos [`<div>`](#) no son obligatorios para definir una nueva sección. La sola presencia de un elemento de cabecera HTML es suficiente para implicar una nueva sección. Entonces,

```
<h1>El zorro</h1>
<p>En esta sección discutiremos sobre los aspectos menos conocidos del zorr
... bla, bla, bla ...
<h2>Hábitat</h2>
<p>El zorro necesita un nido en donde poner sus huevos, de ahí que necesite
...bla, bla, bla...
<h2>Costumbres</h2>
<h1>El oso</h1>
```

genera el siguiente esquema:

- 1. El zorro
 - 1.1 Hábitat
 - 1.2 Costumbres
- 2. El oso

Problemas resueltos por HTML5

La definición de la estructura de un documento en HTML 4 y su algoritmo de esquematizado es muy tosco y genera numerosos problemas:

1. El uso de [`<div>`](#) para definir secciones semánticas sin definir valores específicos para los atributos **class** hace imposible la automatización del algoritmo de esquematizado ("¿Es ese [`<div>`](#) parte del esquema de la página, definiendo una sección o subsección?" o "¿es un [`<div>`](#) sólo de presentación, sólo usado para estilo?"). En otras palabras, la especificación HTML4 es muy imprecisa de lo que es una sección y

cómo ese ámbito está definido. La generación automatizada del esquema es importante, especialmente para [tecnologías de apoyo](#), que son capaces de adaptar la forma en que presentan información a los usuarios de acuerdo a la estructura del documento. HTML5 quita la necesidad de elementos [<div>](#) del algoritmo de esquematizado introduciendo un nuevo elemento, [<section>](#), el elemento de sección HTML.

2. Mezclar varios documentos es difícil: la inclusión de un sub-documento en un documento principal significa cambiar el nivel de los elementos de cabecera HTML Headings Element para que el esquema se mantenga. Esto se resuelve en HTML5 con los recientemente introducidos elementos de seccionado ([<article>](#), [<section>](#), [<nav>](#) y [<aside>](#)) son siempre subsecciones dentro de la sección que las contiene, sin importar qué secciones son creadas por cabeceras internas.
3. Debido a que cada elemento de cabecera HTML es parte del esquema, no es posible describir un subtítulo o título secundario (por ejemplo, `<h1>Mozart</h1><h2>El genio salzburgués</h2>` crea el esquema 1.1 Mozart 1.1 El genio salzburgués). HTML5 introduce el elemento [<hgroup>](#) que oculta todos los elementos de cabecera excepto el primero de más alto rango (por ejemplo, `<hgroup><h1>Mozart</h1><h2>El genio salzburgués</h2></hgroup>` crea el esquema 1.1 Mozart).
4. En HTML4, cada sección es parte del esquema del documento. Pero los documentos no son lineales a veces. Un documento puede tener secciones especiales conteniendo información que no es parte del flujo principal, aunque esté relacionado a él, como un bloque de publicidad o una caja de explicación. HTML5 introduce el elemento [<aside>](#) permitiendo a dichas secciones no ser parte del esquema principal.
5. Otra vez, en HTML4, debido a que cada sección es parte del esquema del documento, no hay forma de tener una sección conteniendo información relacionada no al documento pero sí al sitio entero, como logos, menús, tablas de contenidos, o información de derechos de autor y notas legales. Para ese propósito, HTML5 introduce tres

elementos de sección específicos: [<nav>](#) para colecciones de enlaces, como una tabla de contenidos, [<footer>](#) y [<header>](#) información relacionada con el sitio.

De manera más general, HTML5 trae precisión a las características de seccionado y cabecera, permitiendo a los esquemas de documento ser predecibles y usados por el navegador para mejorar la experiencia del usuario.

El algoritmo de esquematizado de HTML5

Definiendo secciones en HTML5

Todo el contenido incluido dentro del elemento [<body>](#) es parte de una sección. Las secciones en HTML5 pueden ser anidadas. Además de la sección principal, definida por el elemento [<body>](#), los límites de la sección son definidos explícita o implícitamente. Las secciones definidas explícitamente son el contenido definido en las etiquetas [<body>](#), [<section>](#), [<article>](#), [<aside>](#), [<footer>](#), [<header>](#), y [<nav>](#).

Nota: Cada sección puede tener su propia jerarquía de cabeceras. Por lo tanto, incluso una sección anidada puede tener un elemento [<h1>](#). Consulte también [Definiendo cabeceras en HTML5](#).

Ejemplo:

```
<section>
  <h1>El pato</h1>
  <section>
    <h1>Introducción</h1>
    <p>En esta sección, ampliaremos nuestro concepto del pato.
  </section>
  <section>
    <h1>Hábitat</h1>
    <p>El pato, como fiero depredador, necesita un entorno con abundantes l
  </section>
  <aside>
    <p>otros estudiosos del pato
  </aside>
```

```
</section>
<footer>
  <p>(c) 2010 The Example company
</footer>
```

El bloque de HTML define dos secciones de alto nivel:

```
<section>
  <h1>El pato</h1>
  <section>
    <h1>Introducción</h1>
    <p>En esta sección, ampliaremos nuestro concepto del pato.
  </section>
  <section>
    <h1>Hábitat</h1>
    <p>El pato, como fiero depredador, necesita un entorno con abundantes l
  </section>
    <aside>
      <p>otros estudiosos del pato
    </aside>
  </section>

<footer>
  <p>(c) 2010 The Example company
</footer>
```

La primera sección tiene tres subsecciones:

```
<section>
  <h1>El pato</h1>

  <section>
    <h1>Introducción</h1>
    <p>En esta sección, ampliaremos nuestro concepto del pato.
  </section>

  <section>
    <h1>Hábitat</h1>
    <p>El pato, como fiero depredador, necesita un entorno con abundantes l
  </section>

  <aside>
    <p>Otros estudiosos del lobo
```

```
</aside>
</section>

<footer>
  <p>(c) 2010 The Example company
</footer>
```

Esto genera la siguiente estructura:

1. El pato
 - 1.1 Introducción
 - 1.2 Hábitat
 - 1.3 Section (aside)

Definiendo cabeceras en HTML5

Mientras que los elementos de seccionado en HTML definen la estructura de un documento, un esquema también necesita cabeceras para ser útil. La regla básica es simple: el primer elemento de cabecera HTML (uno de los [`<h1>`](#), [`<h2>`](#), [`<h3>`](#), [`<h4>`](#), [`<h5>`](#), [`<h6>`](#)) define la cabecera de la sección actual.

Los elementos de cabecera tienen un *rango* dado por el número del nombre del elemento, donde [`<h1>`](#) tiene el rango *más alto*, y [`<h6>`](#) tiene el rango *más bajo*. Los rangos relativos importan sólo en una sección; el esquema es determinado por las secciones, no por el rango de cabeceras de las secciones. Por ejemplo, este código:

```
<section>
  <h1>La mosca</h1>
  <p>El esta sección hablaremos de la mosca, una criatura adorable.
  ... bla, bla, bla ...
  <section>
    <h2>Hábitat</h2>
    <p>Únicamente se han divisado colonias de moscas en los montes de....
    ...bla, bla, bla ...
  </section>
</section>
<section>
  <h3>El mosquito</h3>
```

```
<p>A continuación, otra rareza a punto de extinguirse, el mosquito.  
...bla, bla, bla...  
</section>
```

Genera el siguiente esquema:

- ```
1. La mosca
 1.1 Hábitat
2. El mosquito
```

Nótese que el rango del elemento de cabecera (en el ejemplo, [<h1>](#) para la primera sección de más alto nivel, [<h2>](#) para la subsección y [<h3>](#) para la segunda sección de alto nivel) no es importante. (Cualquier rango puede ser usado como la cabecera de una sección explícitamente definida, aunque esta práctica no es recomendada).

## Seccionado implícito

Debido a que los elementos de seccionado HTML5 no son obligatorios para definir un esquema, para mantener la compatibilidad con la web existente dominada por HTML4, hay una forma de definir secciones sin ellas. Esto es llamado *seccionado implícito*.

Los elementos de cabecera ([<h1>](#) a [<h6>](#)) definen un nuevo seccionado implícito cuando ellos no son el primer encabezado de sus secciones padre. La forma en que esta sección implícita es posicionada en el esquema es definida por su rango relativo con la cabecera anterior en su sección padre. Si es de un rango más bajo que la cabecera anterior, abre una sub-sección implícita de la sección. Este código:

```
<section>
 <h1>El águila</h1>
 <p>

 <h3 class="subsec-implicita">Hábitat</h3>
 <p> ...
</section>
```

Genera el siguiente esquema:

- 1. El águila
  - 1.1 Hábitat (*impícitamente definido mediante <h3>*)

Si es del mismo rango que la cabecera anterior, cierra la sección previa (ique puede haber sido explícita!) y abre una nueva sección implícita del mismo nivel:

```
<section>
 <h1>El águila</h1>
 <p>...
 <h1 class="secc-implicita">El buitre</h1>
 <p>...
</section>
```

genera el siguiente esquema:

- 1. El águila
- 2. El buitre (*implícitamente definido por <h1>, quien al mismo tiempo cierr*

Si es de un rango más alto que el encabezamiento anterior, cierra la sección anterior y abre una nueva sección implícita al más alto nivel:

```
<body>
 <h1>Colores</h1>
 <h2>Rojo</h2>
 <p> ...
 <section>
 <h3>Amarillo</h3>
 <p> ...
 <h3>Verde</h3>
 <p> ...
 <h2>Sabores</h2>
 <p>...
 </section>
</body>
```



generando el siguiente esquema:

## 1. Colores

1.1 Rojo (*implícitamente con <h2>* )

1.2 Amarillo(*explícitamente con <section>* )

1.3 Verde (*implícitamente con <h3>, cerrando el <h3> previo*)

## 2. Sabores (*implícitamente con <h2>, cerrando el <section> previo*)

Éste no es el esquema que uno podría observando las etiquetas de encabezamiento. Para hacer que tu marcado sea entendible para una persona que eche un vistazo rápido al código, es una buena práctica

- Usar **etiquetas explícitas** para abrir y cerrar secciones.
- Acomodar el rango de cabecera al nivel de anidamiento de la sección deseada.

Sin embargo, esto no es requerido por la especificación HTML5. Si encuentra que los navegadores están generando el esquema de su documento en formas inesperadas, verifique si tiene secciones que están cerradas implícitamente por elementos de cabecera.

Una **excepción** a la regla de que ese rango de cabecera debe corresponderse con el nivel de anidamiento de sección es para secciones que pueden ser **reutilizadas** en múltiples documentos. Por ejemplo, una sección puede ser almacenada en un sistema de gestión de contenidos y ensamblada en documentos en tiempo de ejecución. En este caso, una buena práctica es comenzar en [`<h1>`](#) para el nivel de cabecera más alto de la sección reusable. El nivel anidado de la sección reusable será determinado por la jerarquía de la sección del documento en el que aparece. Las etiquetas de secciones aún son útiles en este caso.

## Sobreescribiendo seccionamiento implícito

A veces una sección necesita tener varios encabezamientos. Unos pocos

casos usuales son:

- Una sección cuyo encabezado principal es acompañado de otro encabezado , cuyo propósito es adornar o aportar cierta información al encabezado principal, pero sin oscurecerlo a la hora de formar el esquema estructural:

```
<section>
 <h1>La flauta mágica</h1>
 <h2>Ópera en dos actos del genio salzburgués</h2>
</section>
```

genera el siguiente esquema, que claramene **no** es el que se desea:

```
1. La flauta mágica
 1.1 Ópera en dos actos del genio salzburgués
```

- El encabezado secundario puede ser usado por una lista de etiquetas:

```
<section>
 <h1>Secciones y esquema del documento</h1>
 <h2>HTML, HTML5, secciones, esquema</h2>
</section>
```

genera el siguiente esquema, que **tampoco es** el que se desea:

```
1. Secciones y esquema del documento
 1.1 HTML, HTML5, secciones, esquema
```

Debido al seccionado implícito, **esto no es posible sin la ayuda del elemento de agrupamiento de cabeceras** ([<hgroup>](#) introducido en HTML5). Esto oculta todas las cabeceras del grupo, excepto la primera, permitiendo una sobreescritura del seccionado implícito. Observa las diferencias:

```
<section>
 <hgroup>
```

```
<h1>La flauta mágica</h1>
<h2>Ópera en dos actos del genio salzburgués</h2>
</hgroup>
... algún contenido ...
</section>
```

genera el siguiente esquema, que es el que se desea:

```
1. La flauta mágica
```

## Secciones desacopladas

Piensa por un momento en una sección **A** que contiene una subsección **B**. A veces es conveniente que **B** mantenga su esquema interno ( subsecciones, encabezados, etc. ), pero que no "ensucie" el esquema de **A**, dando lugar a esquemas más claros.

Existen elementos, cuyo propósito habitual es introducir contenido externo a la página: [<blockquote>](#), [<details>](#), [<fieldset>](#), [<figure>](#) y [<td>](#).

Estos elementos no se "montan" sobre el esquema global, sin importar su contenido, lo que conlleva un desacoplo o aislamiento entre las partes.

Además, el elemento de partida para formar el esquema del documento es [<body>](#), el cual por ser la raíz, se encuentra ya desacoplada de forma natural, en el sentido de que no se puede ir más arriba en el contexto.

Ejemplo:

```
<section>
 <h1>Bach</h1>
 <section>
 <h2>Introducción</h2>
 <p>...
 </section>
 <section>
 <h2>La Pasión según S. Mateo</h2>
 <p> Bla, bla, bla ...
 <blockquote>
```

```
<h1>Friedrich Nietzsche
 <p>"Esa semana he ido a escuchar tres veces la Pasión según san Mateo
</blockquote>
</section>
</section>
```

Este ejemplo resulta en el siguiente esquema:

- 1. Bach
  - 1.1 Introducción
  - 1.2 La Pasión según s. Mateo

Este esquema no contiene el esquema interno del elemento `<blockquote>` que, debido a la propia naturaleza del elemento, se desacopla. Observa bien que el elemento `<blockquote>`, a pesar de todo, contiene un esquema propio bien estructurado.

## Secciones fuera del esquema

HTML5 introduce cuatro nuevos elementos que permiten definir secciones que no pertenecen al esquema principal de un documento web:

1. El elemento sección lateral HTML (`<aside>`) define una sección que, aunque relacionada al elemento principal, no pertenece al flujo principal, como una caja de explicación o un anuncio. Tiene su propio esquema, pero no pertenece al esquema principal.
2. El elemento sección de navegación (`<nav>`) define una sección que contiene enlaces de navegación. Pueden ser muchas en un documento, por ejemplo, uno con enlaces internos a la página, como una tabla de contenidos, y otro con enlaces navegacionales. Estos enlaces no son parte del flujo principal y esquema y pueden ser inicialmente no generadas por lectores de pantalla y tecnología de apoyo similar.
3. El elemento de sección de cabecera (`<header>`) define una cabecera de página, típicamente conteniendo el logo y nombre del sitio y posiblemente un menú horizontal. A pesar de su nombre, no está

posicionado necesariamente al principio de la página.

4. El elemento HTML de pie de página ([<footer>](#)) define un pie de página, típicamente conteniendo el derecho de autor y notas legales y a veces algunos enlaces. A pesar de su nombre, no es necesario que esté posicionado al final de la página.

## Direcciones y tiempo de publicación en elementos de seccionado

El autor de un documento a veces quiere publicar alguna información de contacto, como el nombre del autor y su dirección. HTML4 permite eso vía el elemento [<address>](#), que ha sido extendido en HTML5.

Un documento puede ser hecho de diferentes secciones de diferentes autores. Una sección de otro autor diferente al autor de la página principal es definido usando el elemento [<article>](#). Consecuentemente, el elemento [<address>](#) está ahora enlazado al ancestro más cercano, [<body>](#) o [<article>](#).

De manera similar, el elemento HTML5 [<time>](#), con su atributo booleano establecido [pubdate](#), representa la fecha de publicación asociada a todo el documento, respectivamente al artículo, relacionado a su ancestro más cercano [<body>](#) o [<article>](#).

## Usando elementos HTML5 en navegadores no HTML5

Los elementos de sección y cabecera deben funcionar en la mayoría de navegadores no HTML5. Aunque no estén soportados, no necesitan una interfaz DOM especial y sólo necesitan la aplicación de CSS específico como a los elementos no conocidos se les aplica estilos como `display:inline` por defecto:

```
section, article, aside, footer, header, nav, hgroup {
 display:block;
}
```

Obviamente el desarrollador web puede aplicar estilos sobre ellos de forma diferente, pero mantener en mente que en un navegador no HTML5 la aplicación de estilos por defecto es diferente para lo que es esperado para esos elementos. También se debe notar que el elemento [`<time>`](#) no ha sido incluido, debido a que la aplicación de estilos por defecto de él en un navegador no HTML5 es el mismo que el estilo en un navegador compatible con HTML5.

Este método tiene sin embargo sus limitaciones, así como algunos navegadores no permiten la aplicación de estilos en elementos no soportados. Este es el caso de Internet Explorer (version 8 e inferior), que necesita un script específico para permitirlo:

Esta sección significa que, en el caso de Internet Explorer (8 e inferior), el scripting debe estar habilitado para mostrar seccionamiento en HTML5 y elementos de cabecera apropiadamente. Si no, estas no serán mostradas, lo que puede ser problemático dado que estos elementos probablemente definen la estructura de toda la página. Esto es el porqué de un elemento [`<noscript>`](#) explícito debe ser añadido para este caso.

```
<noscript>
 Warning !
 Because your browser does not support HTML5, some elements are simulated
 Unfortunately your browser has disabled scripting. Please enable it in o
</noscript>
```

Esto nos lleva al siguiente código para permitir el soporte de las secciones y elementos de cabecera en navegadores no HTML5, incluso para Internet Explorer (8 e inferiores), con una apropiada compatibilidad hacia atrás donde este último navegador es configurado para no usar scripting:

# Conclusión

Los nuevos elementos semánticos introducidos en HTML5 suponen la posibilidad de describir la estructura y el esquema de un documento web de una manera estándar. Representan una gran ventaja para las personas que utilizan navegadores HTML5 y necesitan la estructura para ayudarles a utilizar la página, por ejemplo las personas que necesitan la ayuda de una tecnología de apoyo. Estos nuevos elementos semánticos son fáciles de usar y con muy poco esfuerzo pueden funcionar también en navegadores no HTML5. Por lo tanto deben ser usados sin restricciones.

**HTML5 Documentation**