

Estructura de una Página con HTML5

HTML

`<div id="header">`

`<div id="nav">`

`<div class="article">`

`<div class="section">`

`<div id="sidebar">`

`<div id="footer">`

HTML5

`<header>`

`<nav>`

`<article>`

`<section>`

`<aside>`

`<footer>`

La nueva Estructura Semántica del HTML5

Con el nacimiento de la última versión de HTML, el llamado HTML5, se introducen una serie de etiquetas nuevas con cierto sentido semántico. Con sentido semántico me refiero a cómo la etiqueta describen el significado de su contenido. Si estás algo familiarizado con las páginas web, te habrás dado cuenta de que en la mayoría de los casos podemos distinguir en casi todas ellas ciertas partes comunes. Una cabecera (no me refiero ahora al head del principio del código Html) que suele estar en la parte alta de las webs y donde se coloca el logotipo, título, descripción y a veces algo de publicidad. También suelen tener un pie de página, con enlaces externos, normas, enlaces legales, mapa de la web y, de nuevo, publicidad. Otras partes comunes suele ser un menú o barra de navegación, etc.



Las mentes pensantes del HTML5 han decidido por tanto crear etiquetas para esas y otras nuevas secciones de las páginas web. Etiquetas con cierto significado semántico, es decir, que describan lo que contienen en su interior. Es decir Por ejemplo, se sabe lo que es una cabecera, lo que es un pie, una zona de navegación, etc, y el Html5 propone identificarlas con etiquetas concretas. De este modo, los navegadores sabrán qué partes de una página web son eso, cabeceras, pies de página, partes centrales de contenido, etc, y de ese modo podrán dar más importancia a unas u otras.

Hasta ahora solíamos usar las capas o DIVs para todo esto, verdad? Nos bastaba un div con un identificador tipo `div id="cabecera"` etc. A partir de ahora, si es una cabecera tendrá su etiqueta. Solo usaremos un DIV cuando no exista otra etiqueta más idónea para usar o, cuando pretendamos aplicar un Script o unos estilos CSS a un bloque de contenido al que no es posible aplicar una de esas nuevas etiquetas.

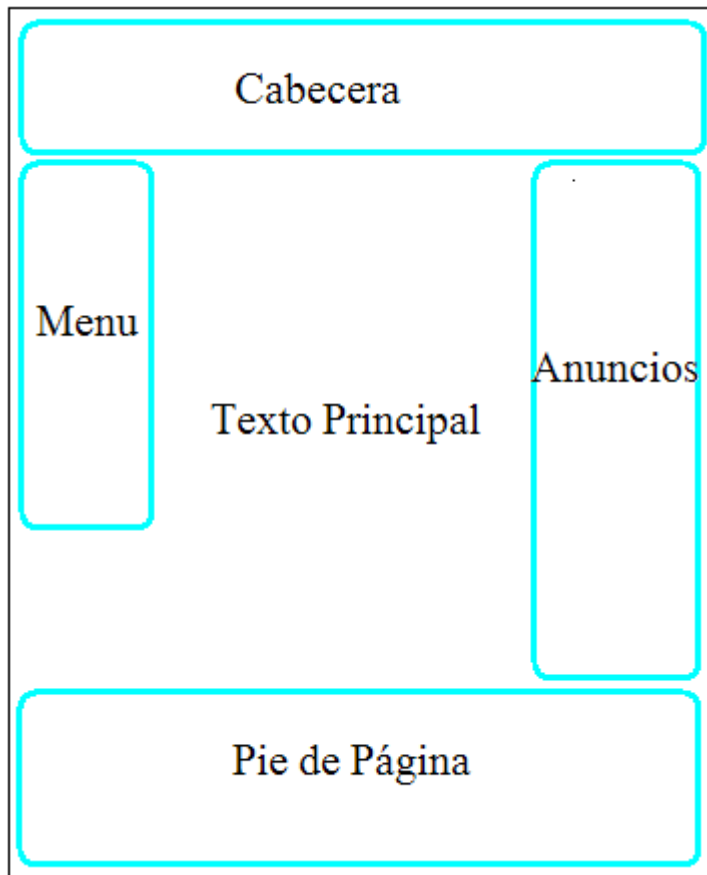
Aunque ahora en muchos casos no usemos divs, sino éstas nuevas etiquetas, no será problema aplicarles estilos css, pues también a esas etiquetas podremos colocarles un `id="nombredeestilo"` o `class="nombredeestilo"`.

A continuación os voy a detallar cada una de esas nuevas etiquetas y os daré indicaciones sobre cuándo y por qué usar unas u otras. Es algo totalmente novedoso, pero te recomiendo que empieces a usarlas desde ya. Son el futuro y la base para poder aplicar todas las nuevas tecnologías que van a ir cayendo conforme el HTML5 se vaya haciendo más y más fuerte.

Las etiquetas semánticas de HTML5

Como os decía, Html5 a definido etiquetas especiales para algunas partes de las páginas web en las que antes solíamos aplicar capas o Divs. A partir de ahora tenemos que esforzarnos por utilizar una de las nuevas etiquetas en su lugar. Si no vemos coherente aplicarle una de las nuevas etiquetas a un bloque de texto, entonces y solo entonces, usaremos un Div.

Veamos cuáles son esas nuevas etiquetas del Html5 así por encima y cómo se podrían distribuir dentro del diseño de una página web para entenderlas mejor. Posteriormente las explicamos paso a paso. La siguiente figura muestra una estructura típica de diseño de páginas web:



Ves? Una cabecera arriba, un menú de navegación a la izquierda, pie de página abajo y un bloque lateral a la derecha para colocar contenido secundario, como un calendario, publicidad o qué se yo. Es lo típico. HTML5 ha definido etiquetas para algunas de esas zonas, no por estar situadas en esas zonas, sino por el contenido que poseen dentro (semánticas, recuerda). De ese modo, han creado las siguientes etiquetas:

```
<header></header>
<hgroup></hgroup>
<nav></nav>
<article></article>
<section></section>
<aside></aside>
<figure></figure>
<footer></footer>
<address></address>
```

Vamos a verlas detenidamente y, a empezar a usarlas, que esto es el futuro, al menos hasta que nos lo cambien de nuevo!

La etiqueta SECTION en el HTML5

La etiqueta `<section>` y su correspondiente etiqueta de cierre `</section>` se utilizan para encerrar el código correspondiente a una sección genérica dentro de un documento o aplicación. Normalmente, un bloque de texto al que perfectamente le podríamos colocar un título o encabezado. Además, todo el contenido que engloba ha de guardar cierta relación entre sí.

Para verlo más claro, un ejemplo de bloque que podríamos encerrar entre etiquetas section podría ser las últimas novedades de la web o del foro. No sería idónea esta etiqueta por tanto para encerrar un bloque de publicidad, por ejemplo, o bloques de cosas diferentes unas de otras.

Dice la norma que si es algo que bien pudiera ser sindicable, tiene todas las papeletas de ser candidato para pertenecer a un section.

Recuerda que no se debe usar para englobar un bloque cualquiera de código simplemente porque necesitas encerrarlo dentro de algo para aplicarle así estilos CSS o algún Script. Si el único motivo para encerrar un bloque de código es el de poder aplicarle lo anterior, no tiene sentido aplicarle una de estas nuevas etiquetas. En su lugar se puede y debe usar un DIV de toda la vida. Veremos ejemplo más adelante si te estás liando. Solo sigue leyendo y lo iras comprendiendo al terminar de ver todas las etiquetas.

Otra regla que podemos usar para saber si un bloque de código o un elemento de una página web debe encerrarse entre esta etiqueta es el preguntarse.... si la web fuera un libro.... estaría ese elemento en el índice inicial del mismo? Si la respuesta es SI, es buen candidato para section.

Dentro de una página web por tanto, pueden existir varios section, cómo no (en un libro suelen existir varias secciones en el índice, no?). Es más, dentro de un section pueden haber otros section secundarios y así sucesivamente (o acaso en un libro no hay secciones, subsecciones, etc, en el índice?).

Si una sección es algo que podría tener un título, es lógico pensar además que lo primero que encontraremos dentro es un título `<h1>` y después una etiqueta `<p>` con el contenido de la sección.

Otra ayudita para decidir si una parte de una página web es o no un section, es tener en cuenta que los section suelen ser temáticos. Suelen englobar contenido que trata de un mismo tema. Si por ejemplo en una página de una web hay un apartado hablando de diversos productos y otra parte en la que se habla tan solo de las formas de pago, esa segunda parte podría englobarse dentro de un section. Se le podría asignar el encabezado

"Formas de Pago", su contenido trata de un tema concreto, es algo que podría estar presente en un índice de la web... decididamente es un section. Lo ves?

Dentro de un section grandote, pueden existir otros section hijo. Imagina una web de un periódico que divide las noticias de su portada en noticias de Europa, de Asia, de Estados Unidos, etc. Podríamos pensar que son secciones temáticas, a las que se les puede colocar un encabezado tipo "Noticias Europeas", etc, cierto? Por lo que se podrían tomar como section. Pero aún dentro de cada una de esas section podríamos englobar todas las noticias de deportes dentro de Europa y colocarlas en un section hijo de Europa, titulado "Noticias Deportivas en Europa". Luego podemos encontrarnos con varios section dentro de un section.

Conoces los blogs? En un blog, todo el conjunto formado por una entrada del autor, podría encerrarse dentro de un section (normalmente tienen un título y luego algunos párrafos). Luego, todo el conjunto con los comentarios escritos por los visitantes, podría formar otro section distinto, titulado "comentarios" por ejemplo. Lo vamos viendo más claro? Sigamos con otras de estas nuevas etiquetas del Html5.

La nueva etiqueta ARTICLE de Html5

Si la etiqueta `<section>` se usaba para bloques de contenido capaces de poder pertenecer a un índice, agrupando bloques de contenido de cierto tema, con la etiqueta `<article>` podemos separar cada uno de esos elementos independientes que pudieran componer ese section concreto. Un article podría considerarse por tanto como cada uno de los elementos en que podemos dividir un section. Al igual que el resto de estas nuevas etiquetas, posee etiqueta de cierre, `</article>`

Creo que podríamos saber si un bloque de código debería pertenecer a un Article si al preguntarnos si ese bloque tiene sentido al colocarlo solo, separado del resto de contenido actual, tiene o carece de sentido. Si al separarlo o independizarlo se pierde su sentido, no es un Article. Si en cambio al separarlo del resto de contenido de la página web el bloque pierde su sentido, no es candidato para pertenecer a un Article.

En una página web pueden existir varios articles, dentro de un section o incluso independientes del mismo. Pueden además contener en su interior títulos con h1 y párrafos, además de otros articles en su interior. Quizás sea fácil confundirlos con los section. Veamos si con el uso aprendemos a distinguirlos un poco, no es sencillo por ahora, verdad?

Un ejemplo real de article podría ser un mensaje de un foro, el artículo de una revista o periódico, un comentario de un usuario a una entrada de blog o incluso una entrada en un blog. La estructura podría ser la siguiente:

```
<article>
  <h1>La etiqueta Article</h1>
  <p>La etiqueta <b>article</b> suele usarse para fragmentos independientes de contenido...</p>
  ...
</article>
```

Decíamos que un article podría ser tan grande como para poder contener incluso header, footer, etc. Mirad un ejemplo:

```
<article>
  <header>
    <h1>La etiqueta Article</h1>
  </header>
  <p>La etiqueta <b>Article</b> suele usarse para fragmento independientes de contenido...</p>
  ...
  <footer>
    <p><small>Contenido publicado por Juanito</small></p>
  </footer>
</article>
```

Por más que leamos las normas del Html5, no termina de quedar muy claro cuándo usar article y cuando section. En el último apartado de esta sección trato de explicar las [diferencias entre section y article](#) que yo capto, aunque como digo, lo leais donde lo leais, veréis como las interpretaciones no son siempre las mismas.

La nueva etiqueta HEADER para las cabeceras en HTML5

Las etiquetas `<header>` y su cierre `</header>` quizás sean mucho más sencillas de entender que las dos anteriores. Se encargan de indicar a los navegadores que lo que contienen es la cabecera de la página y que dentro seguramente llevarán lo típico, algún logotipo, el título de la página, quizás una descripción de la misma y poco más. La norma define a la cabecera como un grupo de ayudas a la navegación. En definitiva, el HTML5 nos indica que todo lo que antes teníamos en la cabecera ha de ir ahora entre esas dos etiquetas, no más.

No es una etiqueta obligatoria, aunque debemos usarla si realmente nuestras páginas tienen algún bloque de código que se pueda entender como "cabecera". Además de los elementos citados, un header podría contener un menú de ayuda a la navegación, una tabla de contenidos, un formulario de búsqueda, etc. Su contenido no es tan estricto.

Al tratarse de etiquetas semánticas, nos da igual la posición que ocupen dentro de nuestra página, un header será un header esté donde esté. Quizás hayas diseñado una web en la que el título y la descripción de la web junto con algunos enlaces o ayudas a la navegación los tengas situados en una barra lateral. Ese bloque continuará siendo un encabezado o header, pues corresponde con el significado semántico de header y la posición que ocupa ya os digo que no nos importa en ese aspecto.

Por otro lado, el header no se limita a identificar la cabecera de una página web, sino que puede usarse también para recoger la cabecera de una sección o varias dentro del contenido. Si por ejemplo en una web con cabecera, hay dentro un conjunto de informaciones a las que crees que le puede venir bien un título, con breve descripción y ayudas a la navegación (un mini header, no?) podrías añadir otro par de etiquetas `<header>` y `</header>` para ese mini header.

```
.....
<header>
  <h1>TITULO DE LA WEB</h1>
```

```
<h2>Breve descripción de la página web.</h2>
<ul>
  <li><a href="inicio.html">Inicio</a></li>
  <li><a href="unaseccion.html">Una Sección</a></li>
  <li><a href="otraseccion.html">Otra Sección</a></li>
</ul>
</header>
.....
```

La etiqueta FOOTER para pies de página en HTML5

Si la etiqueta `<header>` era sencilla de entender, esta otra etiqueta `<footer>` lo es aún más. Se trata de una etiqueta para encerrar todo aquel código típico de las partes inferiores de las páginas web. Contendrán por tanto enlaces a otras webs relacionadas, al mapa de la web, a una página de comentarios sobre el copyright, una política del portal, quizás algo de publicidad, etc.

Footer significa pie y aunque está construida para contener lo dicho, no ha de tener que estar en la parte más baja de una página web para poder ser un footer. Tened en cuenta que todas etiquetas intentan informar a los navegadores sobre lo que pueden contener esos bloques, no más. No olvidemos que el aspecto, apariencia e incluso la posición dentro de la web son cosas que les importan poco a los navegadores. Esas cosas ya las leerán en la hoja de estilo. El Html5 pretende con estas etiquetas saber cuál es su contenido, información semántica, no más. Por tanto, puedes colocar luego esa "capa" en cualquier parte dentro de la web.

Un footer o pie de página, puede contener tantos elementos como sea necesario. No ha de limitarse a contener lo dicho antes. Existen páginas con verdaderos super pies de página, con una lista enorme de enlaces, con información acerca del registro en el portal correspondiente, con la típica "letra pequeña" en la que se detallan condiciones de temas expuestos en el contenido principal, etc. Un footer puede ser realmente extenso, no temas. Puede por tanto contener incluso sections, articles, y algunas de las etiquetas que veremos a continuación. Cada página es un mundo y imagino que cada uno será capaz de diferenciar cuáles etiquetas puede aplicar.

Veamos otras de estas nuevas etiquetas que nos propone Html5. Ya quedan menos.

El código HTML5 y su etiqueta de navegación NAV

Otro elemento bastante típico y común en muchas páginas web es la zona de navegación, formada por enlaces a las zonas principales de la web, o por un menú de navegación, desplegable de enlaces, etc. Por lo visto Html5 interpreta que es una zona que deberíamos identificar y ha creado para ello la nueva etiqueta `<nav>`, con su respectiva etiqueta de cierre, claro, `</nav>`.

No todos los enlaces de una página web han de pertenecer obligatoriamente a un nav. Los enlaces de políticas, de datos de contacto, mapas del sitio, copyrights, enlaces a secciones

secundarias etc, que aparecen normalmente en un pié de página no es necesario que los rodeemos con la etiqueta `<nav>`, pues la etiqueta `<footer>` ya recoge esos aspectos. Ésta etiqueta es más bien para enlaces que favorezcan la navegación por la web, que nos dirijan a zonas importantes.

El por qué HTML5 añade esta etiqueta es sencillo. Cada vez es más habitual que nuestras páginas aparezcan no solo en los navegadores que usamos en el ordenador, sino también en dispositivos móviles, teléfonos, iPads, etc. Al usar esta etiqueta el agente de usuario (el navegador o como quiera llamarse según el dispositivo usado) podría dar la opción al usuario de esconder el menú de navegación para no molestar, para tener más espacio disponible para ver el contenido importante. También podría dar la opción de hacerlo aparecer cuando el usuario quiera cambiar de sección. Esto solo sería posible si se sabe dónde empieza y dónde termina ese menú de navegación y eso es precisamente lo que conseguimos con la etiqueta nav. Hoy por hoy no conozco agentes de usuario (por ejemplo navegadores) que permitan esto, pero es necesario tener nuestra web preparada para cuando llegue ese momento. Ocurre lo mismo con el resto de etiquetas que estamos viendo. Quizás no les veamos utilidad hoy, pero estad seguros de que la tendrán, y mucha, en un futuro no muy lejano.

En la norma donde se dictan toda estas reglas del HTML5 no he leído que no pueda existir más de un bloque del tipo nav dentro de una misma página. No obstante he leído en algunos sitios que se recomienda que solo exista uno. Al final la decisión es tuya y seguro que no te equivocas al elegir.

```
<nav>
<ul>
  <li><a href="cursohtml.html">Curso de Html5</a></li>
  <li><a href="cursocss.html">Curso de CSS</a></li>
  <li><a href="cursojavascript.html">Curso de Javascript</a></li>
</ul>
</nav>
```

Agrupación de títulos con HGROUP en HTML5

El actual HTML5 ha creado una etiqueta más para agrupar titulares. En muchas ocasiones es habitual encontrarnos con el título de la página web con una breve descripción o definición o lema de la página bajo él. Es muy típico por ejemplo en los blog tipo WordPress.

Pues HTML5 ha previsto ese caso y ha decidido crear la etiqueta `<hgroup>` con su etiqueta de cierre `</hgroup>` para encerrar casos como ese, en los que se podría definir el título de la web con un `<h1>` y una descripción bajo el mismo con una etiqueta `<h2>`.

Cualquier agrupación de titulares definidos con `<h...>` seguidos deberíamos encerrarlos entre `<hgroup>` y `</hgroup>`. No, yo tampoco le veo mucho uso a esto, pero... quién manda?.

Los títulos parecen tener ahora un sentido más estricto. Da la impresión de ser más observados ahora. Tienen de hecho algunas particularidades al aparecer junto con estas nuevas etiquetas.


```

<section>
  <hgroup>
    <h1>Curso de Html5</h1>
    <h2>Aprende con nosotros las últimas novedades del Html5.</h2>
  </hgroup>
  <p>Bla bla bla bla bla bla bla bla bla bla bla...</p>
  ...
  <footer>
    <p><small>Contenido publicado por Pepito</small></p>
  </footer>
</section>

```

La etiqueta ASIDE en HTML5

Esta es la última etiqueta que nos presenta HTML5 para bloques con cierto contenido semántico dentro de una página web. Con las etiquetas `<aside>` y `</aside>`, se nos invita a rodear todo aquel contenido que no es directamente contenido principal del que estamos hablando o del que estamos tratando en esa página en concreto. La norma dice que se use para todo aquel contenido tangencial al contenido principal de la página...

Lo usaremos por tanto para todos aquellos elementos secundarios, como podrían ser los bloques publicitarios, enlaces externos, citas, un calendario de eventos, etc, siempre claro que no encontremos otra etiqueta más acorde de entre las ya comentadas.

```

<aside>
  <h1>Publicidad</h1>
  (( código de la publicidad ))
</aside>

```

Y con esta, terminamos de comentar estas nuevas etiquetas de HTML5 para representar zonas con cierto carácter o significado semántico. Existen otras nuevas etiquetas, pero ya son para resaltar textos, etc., no para agrupar bloques de elementos, etc.

Recuerda aplicarlas cuando tengas claro que los elementos contenidos son realmente lo que el nombre de la etiqueta representa. Ante la duda, no las uses sin tener claro el porqué. Si no sabes si cierto bloque de código precisa o no de una de estas etiquetas, simplemente no le pongas ninguna. Si la necesitaras para poder aplicarle estilos o para ejecutarle cierto script, usa mejor las capas o Divs de toda la vida, que para eso son.

Espero que puedas terminar de entenderlo con algunos ejemplos prácticos y muestras reales.

Diferencias entre section y article en Html5

Las diferencias entre article y section, cuándo usar una etiqueta article y cuándo utilizar la etiqueta section, no termina de quedar demasiado clara tras leer la norma que rige el Html5. Es normal confundir una con otra y no estar seguro de cuál utilizar en cada momento al escribir código Html5, verdad?. Como regla general podríamos establecer lo siguiente:

Si el fragmento de código posee significado por si mismo, si en caso de escribirlo en un papel separado del resto de la web el fragmento continua teniendo su sentido, podemos usar un `<article>`.

Si no tiene tanto sentido, pero tiene relación con lo comentado en el resto de esa página, podemos usar un `<section>`.

Si el algo que no tiene mucho que ver son el tema de esa página en concreto ya dijimos que se trataría de un `<aside>`.

Si no cumple con ninguna de las condiciones anteriores, pero necesitamos encerrarlo entre dos etiquetas para poder aplicarle estilos o scripts, entonces lo que debemos usar es un Div.

Ejemplos de código HTML5 utilizando las nuevas etiquetas semánticas

Veamos unos cuantos ejemplos de códigos Html5 reales, en los que poder ver mucho mejor el significado de todas estas definiciones tan pesaditas, de acuerdo? Vamos allá.

Partimos de la estructura básica, una página aún sin nada más que el título.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8" />
    <title>Titulo</title>
  </head>
  <body>
    (( CONTENIDO ))
  </body>
</html>
```

Vamos a añadirle elementos típicos de una página web normal y corriente y decidimos por el camino qué etiquetas podríamos colocarle a cada parte, de acuerdo? Nos olvidamos del lugar en las que colocaremos esas partes, pues como sabemos, eso lo conseguiremos con los estilos CSS. Además os recuerdo que el Html5 define el significado de cada parte, no el cómo ni dónde se muestran en el navegador, entendido?

A esta página web de ejemplo vamos a colocarle para empezar el título de la web seguido de una breve descripción de la misma. Lo acompañaremos con un buen logotipo y un recuadrito de aquellos para buscar dentro de la web. Si recordamos lo leído antes, será fácil elegir colocar todo eso dentro de un `<header>`, verdad? Dentro de ese header colocaremos el título de la web dentro de un titular tipo `<h1>` seguido de la descripción en un `<h2>`. Recordamos entonces la definición de hgroup de modo que encerramos ambas etiquetas dentro de un `<hgroup>` y listo.

El logotipo no tiene mucho significado semántico. Es solo un adornito promocional, para que los visitantes se queden con un recuerdo visual de la página. Carece por tanto de significado semántico pero necesitare encerrarlo dentro de algo para poder aplicarle estilos

posteriormente. De modo que para el logotipo usaré un simple div, al que asignaré un nombre para facilitar la aplicación de estilos más tarde. Le colocaré por tanto un `<div id="logotipo">`.

El recuadrito de búsqueda podría considerarse un elemento digno de una etiqueta `<nav>`, pero casi prefiero reservarla para el menú principal de enlaces, de modo que al igual que he hecho con el logotipo, le aplicaré un simple div de la forma `<div id="buscador">`.

También pienso colocar un anuncio de copyright, datos de contacto con el webmaster y algunos enlaces hacia las normas legales y al mapa de la web. Tiene toda la pinta de querer pertenecer a un pie de página, de modo que rodearé todo aquello entre `<footer>`.

Dentro de ese footer, se podrían distinguir la sección de datos de contacto, la del copyright y los enlaces hacia normas y mapa de la web. Crearé por tanto tres secciones para esas tres partes.

Nos queda por definir la parte principal de los contenidos. Quizás quiera presentarla en dos columnas, quizás en tres... pero eso no son aspectos semánticos, sino visuales, de modo que en caso de decidirme por una forma u otra lo trataría de conseguir aplicando divs, no etiquetas semánticas, no crees? Lo que si tengo claro es que en esa parte principal voy a mostrar por un lado noticias actuales sobre la temática de mi web. Y también mostraré los temas más populares, los más visitados o más votados por los visitantes. Además colocaré un menú hacia las distintas partes de la web y un anuncio en el que trato de vender en pdf un resumen de toda la web.

Claramente podemos adivinar un section para las novedades, otro section para los temas más populares y un nav para el menú de navegación. Sus titulares podrían ser `<h1>` con "Últimas Novedades" y "Temas Populares". También podemos colocar el anuncio del pdf en un aside, pues no llega a ser directamente contenido principal, aún estando relacionado (pero tangencialmente...).

Dentro de los section de novedades y temas populares, encontraremos diversas publicaciones. Si por ejemplo aparecen 5 novedades, cada una de ellas pertenecerá a un article, si es que hemos entendido bien su definición páginas atrás. Idem para cada uno de los temas más populares del segundo section, no? Cada uno irá dentro de un article independiente, pero dentro del section correspondiente, claro.

Por otro lado, quiero darle algo de estilos a esa parte principal, colocándole algunos bordes sombreados, etc, de modo que, como no hay ya una etiqueta que englobe toda esa parte principal, rodeo todo aquello de un div con el nombre "contenido" para poder aplicarle esos estilos más tarde, con CSS, claro.

Con eso la estructura queda bien clara y definida. Aún necesitaría centrar toda la página en el navegador. Para ello podemos aplicar ciertos estilos a la etiqueta correspondiente, podemos conseguir un centrado perfecto en cualquier navegador. El problema es que no tenemos ninguna etiqueta englobando a toda la página... de modo que, como no encuentro ninguna etiqueta semántica que lo englobe todo, coloco de nuevo un simple div y le coloco

el nombre "centrado" para conseguir ese efecto. Como ves, al ser su función solamente la de aplicar estilos, uso un div y no una etiqueta con significado semántico.

El código completo podría quedar más o menos así:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8" />
    <title>Titulo</title>
  </head>
  <body>
    <div id="centrado">
      <header>
        <hgroup>
          <h1>(( TITULO ))</h1>
          <h2>(( DESCRIPCION DE LA WEB ))</h2>
        </hgroup>
        <div id="logotipo">(( LOGOTIPO ))</div>
      </header>
      <div id="contenido">
        <nav>
          (( MENU DE NAVEGACION ))
        </nav>
        <aside>
          (( ANUNCIO DEL PDF ))
        </aside>
        <section id="izquierda">
          <h2>Ultimas Novedades</h2>
          <article>(( NOVEDAD UNO ))</article>
          <article>(( NOVEDAD DOS ))</article>
          <article>(( NOVEDAD TRES ))</article>
        </section>
        <section id="centro">
          <h2>Temas Populares</h2>
          <article>(( POPULAR A ))</article>
          <article>(( POPULAR B ))</article>
          <article>(( POPULAR C ))</article>
        </section>
      </div>
      <footer>
        <section id="copyright">(( Datos Copyright ))</section>
        <section id="contacto">(( Datos de Contacto ))</section>
        <section id="enlaces">(( MapaWeb, Política, etc. ))</section>
      </footer>
    </div>
  </body>
</html>
```