Páginas fijas, líquidas y elásticas

Cuando se diseña la estructura de una página web hay que tener en cuenta que no puede ir dirigida a ninguna resolución particular y que, en la medida de lo posible, debería ser capaz de adaptarse a todas, al menos de las más modernas con un mínimo de 800x600 píxeles.

Una mayor complejidad sería plantearse terminales de mano donde el área de visualización es muy pequeña en comparación con un monitor de por ejemplo 19 pulgadas. Pero también existe la posibilidad de contemplar estilos específicos para estos soportes, utilizando la regla de estilo @media que nos permite configurar estilo para monitores (screen), terminales de mano (handheld), impresora (print), etc

Por lo tanto controlar como se especifican las medidas de los distintos marcos es una decisión importante que se ha de tomar antes de empezar a dotar de estilo a la estructura de la página. Usualmente se habla de tres posibilidades: medidas **fijas, líquidas o elásticas**. Así lo ideal sería que toda la estructura fuese capaz de adaptarse a diferentes soportes visuales sin que la apariencia de la página no se deforme en exceso.

ESTRUCTURAS FIJAS

En las estructuras **fijas** las dimensiones de los marcos, especialmente el ancho, se expresan en valores absolutos como **píxeles** de monitor. Realmente no son del todo absolutos sino relativos a la resolución que el usuario haya efectuado en el monitor.

En la página de ejemplo [pagina-fija-1.html](http://www.wextensible.com/como-se-hace/estructura-pagina-web/ejemplos/pagina-fija-1.html) puede ver el resultado, donde ponemos un marco cabecera y un pie de página así como el contenido interior, pero con un único lateral izquierdo para no complicar el ejemplo.

El alto no suele fijarse pues es más cómodo para el usuario que el contenido se alargue verticalmente en lugar de horizontalmente. Entonces **fijamos** el ancho en una medida absoluta de 800 píxeles, medida que decíamos que era la considerada como mínima de ancho de monitor. Pero hay que tener en cuenta lo siguiente:

* Hay que dejar un espacio a la derecha que ocupará la barra de desplazamiento vertical del navegador cuando la altura de la página sea mayor que la ventana. Podemos estimar un ancho de 16 píxeles.
* En el glosario XHTML+CSS hemos expuesto la aplicación del [estilo CSS para los elementos HTML](http://www.wextensible.com/temas/xhtml-css/css-descripciones.html#css-xhtml), donde encontramos una referencia para todos los elementos en [W3C](http://www.w3.org/TR/2007/CR-CSS21-20070719/sample.html). Ahí podemos observar que se recomienda aplicar por los navegadores un **margen** de 8px al elemento <body> para separarlo de la ventana del navegador (el elemento <html>). Entonces también hemos de tener en cuenta una separación de 2x8 = 16 píxeles.

Si calculamos 800-16-16 = 768 píxeles libres para el ancho de nuestra web. Para asegurarnos en caso de que las medidas de la barra y el margen del <body> sean mayores, lo podemos dejar en 762px, de los cuáles 2px serán para los bordes derecho e izquierdo de los marcos, cada uno con un ancho de 1 píxel. Si los marcos tienen un relleno de 5px (valores de padding derecho e izquierdo), entonces nos queda un ancho libre para los marcos **cabeza** y **pie** de 762-2-10 = **750px**.

Luego tenemos que fijar el ancho del **lateral**, que elegimos en **100px**, valor no dependiente de otros marcos y que irá en función del tipo de contenido que deseemos incluir en su interior. Sumando sus rellenos y bordes, el **lateral** tiene un ancho final de 100+10+2 = **112px**. Quedará libre para el contenido 762-112 = 650px, de los cuáles hay que restar 12px para sus rellenos y bordes quedando un ancho libre para el **contenido** de 650-12 = **638px**.

Además el contenido debe separarse del lateral que flota a la izquierda, lo que se consigue dando un margen izquierdo con margin-left:112px que es lo que ocupa el lateral con sus rellenos y bordes. Decíamos que el alto no lo especificamos, por lo que el navegador lo considera con valor **auto,** con lo que alargará verticalmente los marcos para ajustar el contenido. Sin embargo hemos de especificar la altura para el marco lateral con height:100% para que la flotación de ese marco no termine en toda la altura del marco de contenido. Al final de este marco hemos de borrar la flotación para lo cual aplicamos clear:both a un bloque vacío que sólo sirve para esto.

Cuando la pantalla tiene más de 800 píxeles resultaría con una mejor estética si todo el contenido apareciera centrado horizontalmente. No es difícil conseguirlo, siendo necesario encerrar todo lo que hay en la página en otro marco <div id="total">. Le dotamos de un ancho igual que los anteriores de 762 píxeles medidos por fuera del borde, con lo que el borde del marco "total" resulta envolver todos los demás marcos. Además esta posibilidad nos da la ventaja de no tener que declarar los anchos de los marcos interiores a excepción del lateral. Esto es porque los marcos se extienden a la derecha hasta ocupar toda la longitud del marco **total** que los contiene, marco que si tiene un ancho especificado.

Para poder central el **total** le damos un valor de margen igual a **auto**. Así nos quedaría el segundo ejemplo de estructura fija centrada en la ventana que puedes ver en [pagina-fija-2.html](http://www.wextensible.com/como-se-hace/estructura-pagina-web/ejemplos/pagina-fija-2.html), lo que supone una mejora respecto al primer ejemplo.

Particularmente considero que son muchas las desventajas de la estructura fija. Veamos algunas consideraciones:

* **Dependiente del monitor:** Para pantallas con resolución inferior a 800 píxeles de ancho, el contenido a la derecha necesitara del *scroll* horizontal para acceder al mismo. Para pantallas con resolución muy superior a esos 800 píxeles, el ancho de página se reducirá demasiado resultando poco estético. Es más común este último caso pues la tendencia es usar mayores resoluciones de pantalla para poder "meter más cosas", dado que los actuales monitores lo permiten.
* **Demasiados cálculos:** Hay que controlar demasiados parámetros para conseguir que todo ajuste. Si por alguna razón modificamos el ancho inicial a otro valor diferente a 800 píxeles tendríamos que recalcular todo otra vez. Si se modifica algún padding o margin que afecte a la estructura, el esquema se nos vendrá abajo.
* **Laterales nada flexibles:** Si más tarde agregamos otro lateral derecho tendríamos que recalcular gran parte del esquema. Esto nos obliga a tener que establecer de entrada cuántos laterales queremos y sus anchos. No resultará fácil crear un esquema para luego construir páginas con cero, uno o dos laterales, algo que si es posible con las estructuras no fijas, incluso usando laterales de distinto ancho.
* **Problemas al ampliar texto con el navegador:** Los navegadores actuales como Explorer 8, Firefox 3.5, Opera 10.5 y Safari 4.0 dan la posibilidad de aumentar o disminuir el tamaño de la página de dos formas. Bien actuando sólo sobre el texto y/o haciendo un "zoom" en todo el contenido. Con la estructura fija cuando ampliamos mucho el texto no podremos controlar que no sobrepase el ancho fijo que se haya especificado.
* **Funciona bien con el zoom del navegador:** Aunque también es cierto que si el zoom se aplica sobre todo el contenido entonces las medidas fijas también serán objeto de esa ampliación o reducción, no afectando por tanto al texto.

ESTRUCTURAS LÍQUIDAS

Cuando disponemos de un elemento de bloque como el que antes usamos <div id="total"> y no especificamos nada acerca de su ancho en la propiedad width, el navegador aplica el valor **auto.** Entonces el ancho viene determinado por el contenedor donde está alojado ese marco el cuál es el elemento <body>, que a su vez se aloja en la ventana <html>. Así que el ancho de nuestro elemento <div id="total"> será siempre el de la ventana, restando los márgenes de 8px del elemento <body>. Esto quiere decir que el ancho del marco **total** se ajustará siempre al tamaño de la ventana, por lo que si además hacemos uso de las **medidas relativas en porcentajes** para los marcos que se ajustan horizontalmente como el **lateral** y **contenido,** podríamos conseguir que todo se reajuste también al **tamaño de la ventana.**

Esto es lo que se conoce como una **estructura líquida**. Hemos reutilizado el ejemplo anterior para construir la siguiente [pagina-liquida-1.html](http://www.wextensible.com/como-se-hace/estructura-pagina-web/ejemplos/pagina-liquida-1.html), cuyo código HTML es exactamente igual pero con un CSS distinto:

div#total { /\* ancho 762px \*/

border: maroon dashed 1px;

}

div#cabeza {

border: red solid 1px;

padding: 0.67%; /\* 5px/750px = 0.0067 \*/

}

div#lateral {

width: 13.12%; /\* 100px/762px = 0.1312 \*/

height: 100%;

padding: 0.67%; /\* 5px/750px = 0.0067 \*/

float: left;

border: blue solid 1px;

}

div#lateral ul{

padding-left: 16%; /\* 16px/100px = 0.16 \*/

border: green dotted 1px;

}

div#lateral ul li {

margin-left: 0;

}

div#contenido {

margin-left: 14.70%; /\* 112px/762px = 0.1470 \*/

padding: 0.67%; /\* 5px/750px = 0.0067 \*/

border: green solid 1px;

}

div#contenido p {

text-align: justify;

}

div#pie {

padding: 0.67%; /\* 5px/750px = 0.0067 \*/

border: red solid 1px;

}

div.borra-flotacion {

clear: both;

El ancho de los bordes **no admite porcentajes,** por lo que los especificamos como estaban antes de 1px. Esto puede dar lugar a desajustes por lo que habría que no considerarlos en la estructura líquida, aunque en este ejemplo los mantenemos para ver los límites de cada marco.

El ajuste de los marcos se produce con el cambio del ancho de la ventana, de tal forma que si este ancho es muy reducido puede deformar el contenido de algún lateral muy estrecho. Esto se puede arreglar incorporando la propiedad min-width al marco total, como podrá ver en el segundo ejemplo [pagina-liquida-2.html](http://www.wextensible.com/como-se-hace/estructura-pagina-web/ejemplos/pagina-liquida-2.html) que es la misma página a excepción de incorporar min-width: 650px; en el estilo de div#total. Sin embargo es necesario especificar una medida absoluta, porque las relativas en porcentajes se expresan en referencia al ancho del contenedor cuando ese ancho queda también expresamente declarado con una medida absoluta. En otro caso los porcentajes declarados en el ancho mínimo no funcionarán. Por lo tanto es un grave contratiempo pues todo es líquido menos el ancho mínimo.

Veamos las siguientes consideraciones finales de la estructura líquida:

* **Los porcentajes no se aplican a los bordes**, lo que da lugar a diferencias de ajuste si usamos bordes gruesos en los marcos.
* **Dificultad para controlar el ancho mínimo.** Aunque existen soluciones para esto, no deja de suponer un contratiempo.(con IE<8 problemas)
* **Problemas con el ampliador de texto del navegador:** Si el navegador incorpora un ajuste de tamaño de texto independiente del "zoom", cuando el texto seleccionado es muy grande y la ventana muy pequeña, el texto puede desbordar los anchos de los marcos produciéndose saltos de línea no deseados.
* **Perfecto para el zoom del navegador:** Para un tamaño de texto sin alterar, el zoom funciona a la perfección, modificando el tamaño de los marcos sin alterar las proporciones.

ESTRUCTURA ELÁSTICA

Por último tenemos la **estructura elástica** basada en que podemos declarar las dimensiones de un elemento proporcionales al tamaño de la fuente, es decir al valor de la propiedad font-size. Una característica de esta propiedad es que se hereda automáticamente en todos los elementos que permiten incluir texto. Así podríamos crear una página entera sin declarar en ningún sitio el tamaño de la fuente, en cuyo caso el navegador proporciona un tamaño (y una familia) que viene por defecto. Por ejemplo, suelen poner por defecto la fuente *Times New Roman* con un tamaño de unos *16px.* Este valor se va heredando de padre a hijo, de tal forma que todos los elementos tienen la misma propiedad si no se especifica otra cosa.

Entonces podemos usar tamaños de fuentes también relativos. Si en un elemento declaramos font-size: 2em significa que el tamaño de su fuente será **dos veces** el tamaño de la fuente del contenedor donde está albergado. Pero además esta medida relativa puede aplicarse a cualquier otra dimensión, como anchos (width), rellenos (padding), márgenes (margin) y gruesos de bordes (border-width), con lo que podemos resolver el problema que teníamos en la estructura líquida con los bordes. Es importante entender que **em** no es una medida, sino una proporción tal como lo es el porcentaje **%**.

Volvemos a tomar la página del último ejemplo tal que no es necesario modificar el HTML, aunque si cambiamos el estilo CSS. Puede ver este primer resultado en la siguiente [pagina-elastica-1.html](http://www.wextensible.com/como-se-hace/estructura-pagina-web/ejemplos/pagina-elastica-1.html). El código de estilo es el siguiente, donde los comentarios en verde se han puesto para ver la equiparación de dimensiones con los ejemplos anteriores:

body {

font-family: Times New Roman;

font-size: 1em; /\* 1em equivale a 16px por defecto \*/

}

div#total {

min-width: 40.625em; /\* 650px/16px = 40.625em \*/

border: maroon dashed 0.0625em; /\* 1px/16px = 0.0625em \*/

}

div#cabeza {

border: red solid 0.0625em;

padding: 0.3125em; /\* 5px/16px = 0.3125em \*/

}

div#lateral {

width: 6.25em; /\* 100px/16px = 6.25em \*/

height: 100%;

padding: 0.3125em;

float: left;

border: blue solid 0.0625em;

}

div#lateral ul{

padding-left: 1em; /\* 16px/16px = 1em \*/

border: green dotted 0.0625em;

}

div#lateral ul li {

margin-left: 0;

}

div#contenido {

/\* Con el ancho del lateral del ejemplo anterior: 112px/16px = 7em

Que es lo mismo que sumar el ancho del lateral más sus dos

rellenos y bordes, todo sumado en em's:

6.25em + (2 x 0.3125em) + (2 x 0.0625em) = 7em

\*/

margin-left: 7em;

padding: 0.3125em;

border: green solid 0.0625em;

}

div#contenido p {

text-align: justify;

}

div#pie {

padding: 0.3125em;

border: red solid 0.0625em;

}

div.borra-flotacion {

clear: both;

}

En primer lugar hay que establecer el tamaño de la fuente del cuerpo <body> y también conviene el nombre de la fuente. Así nos aseguramos que controlamos esta medida independientemente de la que por defecto se establezca para el navegador. En este ejemplo ponemos **Times New Roman** y **1em** para poder comparar el ejemplo con los anteriores, pero podría ser cualquier otra fuente y tamaño. Lo importante es que el resto de marcos que irán dentro del <body> heredarán esta fuente y, sobre todo, su tamaño.

Luego en div#total que es el marco total que recoge toda la página, declaramos su mínimo tamaño con min-width: 40.625em, valor que podemos estimar realizando algunas pruebas, pues dependerá de la fuente elegida en el paso anterior. En todo caso ya sabíamos que podíamos darle un min-width: 650px en el ejemplo 2 de estructura líquida con ancho mínimo. Sólo tenemos que hallar la proporción 650px/16px = 40.625em, pues 1em es equiparable a 16px para una fuente como la Times New Roman. Que sea equiparable no es lo mismo que sea igual, pues todas las fuentes no tienen el mismo tamaño aparente para 1em. En todo caso es cuestión de probar algunos anchos mínimos en em's y ver el resultado.

Si nuestros marcos no tienen bordes podemos obviar poner sus anchos con em's. También podemos hacerlos todos con medidas absolutas de 1px, aunque podrían observarse pequeños desajustes en ampliaciones de zoom. En todo caso también vamos a convertir 1px/16px = 0.0625em que aplicamos a todos los marcos.

Para los marcos interiores **cabeza, lateral, contenido** y **pie** le aplicamos un relleno de 5px que serán 5px/16px = 0.3125em. Para el **lateral** le ponemos un valor de 6.25em, resultado de la proporción 100px/16px = 6.25em. Al igual que el relleno bien podría ser cualquier valor que se adapte a nuestras necesidades, aunque lo hacemos así para poder comparar el resultado del ejemplo con las otras estructuras. Por lo tanto es muy importante entender que aunque estamos haciendo estas operaciones, bien podríamos directamente poner una cantidad en em's y mediante prueba ir buscando el valor que más nos guste.

El cálculo del margen izquierdo del **contenido** que lo separa del **lateral** lo hacemos directamente en em's: 6.25em + (2 x 0.3125em) + (2 x 0.0625em) = 7em, siendo el ancho del lateral más dos veces su relleno y 2 veces el grueso de su borde. En este caso si es necesario hacer este cálculo, pero es mucho más sencillo que los correspondientes de las estructuras fijas o líquidas.

Una ventaja de la estructura elástica es que podemos también expresar los gruesos de los bordes en **em's**. Así tenemos el siguiente ejemplo [pagina-elastica-2.html](http://www.wextensible.com/como-se-hace/estructura-pagina-web/ejemplos/pagina-elastica-2.html), cuyo código CSS queda modificado en lo que respecta al grosor de bordes, incluyendo un grosor excesivo pero necesario para ver el efecto de redimensionamiento. En el código ponemos en azul aquello que cambia con respecto al ejemplo anterior:

body {

font-family: Times New Roman;

font-size: 1em; /\* 1em equivale a 16px por defecto \*/

}

div#total {

min-width: 40.625em; /\* 650px/16px = 40.625em \*/

border: maroon dashed 0.5em; /\* 8px/16px = 0.5em \*/

}

div#cabeza {

padding: 0.3125em; /\* 5px/16px = 0.3125em \*/

border: red solid 0.5em;

}

div#lateral {

width: 6.25em; /\* 100px/16px = 6.25em \*/

height: 100%;

padding: 0.3125em;

float: left;

border: blue solid 0.5em;

}

div#lateral ul{

padding-left: 1em; /\* 16px/16px = 1em \*/

border: green dotted 0.5em;

}

div#lateral ul li {

margin-left: 0;

}

div#contenido {

/\* Con el ancho del lateral del ejemplo anterior: 112px/16px = 7em

Que es lo mismo que sumar el ancho del lateral más sus dos

rellenos y bordes, todo sumado en em's:

6.25em + (2 x 0.3125em) + (2 x 0.5em) = 7.875em

\*/

margin-left: 7.875em;

padding: 0.3125em;

border: green solid 0.5em;

}

div#contenido p {

text-align: justify;

}

div#pie {

padding: 0.3125em;

border: red solid 0.5em;

}

div.borra-flotacion {

clear: both;

}

Como se observa, el único cálculo que debemos tener en cuenta es el relacionado con el margen izquierdo del contenido, pero aún así es una operación más simple pues no involucra cálculos de proporciones sino sumas de longitudes del lateral en em's. Si actuamos con el zoom del navegador, el grosor de los bordes también resulta afectado por ese zoom de una forma adecuada. Todo el conjunto parece que ajusta mejor que con las otras estructuras.

Consideraciones finales de la estructura elástica:

* **Control total del ancho mínimo:** Podemos especificar un ancho mínimo en em's sin problemas (siempre que el navegador acepte min-width).
* **Ajuste perfecto de los bordes:** El grueso de bordes se amplía y reduce con el zoom del navegador. Este es un efecto que no siempre resulta estético, en cuyo caso simplemente basta con ignorarlos, ponerle un dimensionamiento absoluto de 1px o incluso mejor, poner bordes transparentes o del mismo color que el fondo. En todo caso es una ventaja que se pueda aplicar.
* **El ampliador de texto si funciona ahora:** Cómo el redimensionamiento se hace en función del tamaño del texto, cuando aplicamos el ampliador de texto del navegador, el reajuste se produce de manera perfecta con esta estructura, no desbordándose las líneas de los elementos del lateral (los vínculos) como pasaba con las estructuras fijas y líquidas.
* **También funciona con el zoom del navegador:** Como los anteriores, el zoom del navegador no le afecta en absoluto, es más, el efecto de ampliar el texto y aplicar el zoom es el mismo.
* **Es ideal para aplicar un script que haga de zoom:** Aunque los navegadores actuales ya lo incorporan, con esta estructura es fácil hacer un script para realizar el zoom directamente sin usar los propios del navegador. Veremos un ejemplo de esto en el siguiente tema.
* **Flexibilidad de los laterales:** Cómo los cálculos y la configuración es más simple, es más fácil configurar un esquema con un contenido y 2 laterales por ejemplo. E incluso con ambos laterales de diferente tamaño.

**Cabeceras elásticas con banner**

Para incorporar una imagen en la barra de la cabecera de una página **líquida** o **elástica** a modo de **banner**, podemos usar una imagen de fondo con background-image o el elemento <img>. Hay varias razones a tener en cuenta a la hora de incluir una imagen en esta estructura. El principal inconveniente es que la imagen ha de extenderse para pantallas grandes. Si el monitor es de *1280* píxeles de ancho, la imagen no ha de ser mucho más pequeña pues perdería calidad cuando la ventana esté maximizada. Otro detalle es que al elemento <img> no podemos declararle sus dimensiones con width o height. Esto también va a hacer que la página se cargue más lentamente pues el navegador no sabrá inicialmente que espacio reservar para esa imagen.

Con la opción CSS incorporamos una imagen con la propiedad [{background-image}](http://www.wextensible.com/temas/css3-fondo-bordes/). Este es un ejemplo de la [página elástica con banner CSS](http://www.wextensible.com/como-se-hace/estructura-pagina-web/ejemplos/pagina-elastica-banner-css.html). El estilo CSS modificado es:

div#cabeza {  
 border: red solid 0.0625em;  
 padding: 0.3125em; /\* 5px/16px = 0.3125em \*/  
 background-image: url(banner.jpg);  
 background-repeat: no-repeat;  
 background-size: cover;  
 font: 2em bold 'Arial Black';  
 color: red;  
}

Agregamos un fondo con {background} usando **cover** para extender la imagen. He agregado un color y tamaño de letra para que resalte en ese fondo. La [imagen](http://www.wextensible.com/como-se-hace/estructura-pagina-web/ejemplos/banner.jpg) ha de ser preparada antes con un tamaño de *1000×100* píxeles, quizá demasiado grande, pero con el objeto de que al extenderse no pierda calidad. Será cuestión de probar otros tamaños menores y otras proporciones con objeto de que no sea muy pesada. La altura de la cabecera queda condicionada por el contenido HTML y no por la imagen de fondo. Para que verticalmente se presente la imagen completa habría que incrementar el alto de la cabecera, pues no podemos darle valores absolutos a {background-size} pues entonces no se readaptará al contenedor.

La otra forma es mediante un elemento <img>, con el ejemplo [página elástica con banner HTML](http://www.wextensible.com/como-se-hace/estructura-pagina-web/ejemplos/pagina-elastica-banner-img.html). Ahora la imagen es también el contenido de la cabecera, por lo que ésta ajustará tanto su alto como su ancho a las medidas de la imagen. El código HTML a modificar es este:

...  
 <style type="text/css">  
 ...  
 div#cabeza {  
 border: red solid 0.0625em;  
 /\*padding: 0.3125em; \*/ /\* 5px/16px = 0.3125em \*/  
 font: 2em bold 'Arial Black';  
 color: red;  
 position: relative;  
 }  
 div#cabeza div.headvinc {  
 position: absolute;  
 left: 0.7em;  
 top: 0.7em;  
 z-index: 1;  
 }  
 div#cabeza img.banner {  
 width: 100%;  
 }  
 ...  
 </style>  
</head>  
<body>  
 <div id="total">  
 <div id="cabeza">  
 <div class="headvinc"><a href="http://www.wextensible.com"><img  
 src="/res/img/wextensible.gif"  
 width="41" height="40" alt="wextensible"  
 title="www.wextensible.com" /></a>  
 TITULO DE LA PÁGINA</div>  
 <img src="banner.jpg" class="banner" alt="banner" />  
 </div>  
 ....

Eliminamos el padding para que la imagen ocupe toda la cabecera. Resaltamos texto y le damos un **posicionamiento** con valor **relative** pues vamos a **posicionar** elementos. El contenido que antes había, es decir, el elemento <a> y el texto del título, lo metemos todo en un elemento <div class="headvinc"> para posicionarlo de forma **absoluta** arriba y a la izquierda, a *0.7em* de esa esquina. Lo ponemos en una capa más alta con z-index: 1 para situarlo sobre la imagen. Ésta irá en el fondo y al darle un tamaño de width: 100% se extenderá para ocupar toda la cabecera. Es importante notar que no podemos darle atributos width o height a la imagen, pues de otra forma tomaría esos valores. Ahora la imagen se escala en ambas direcciones presentándose completa. Aunque el alto de la cabecerá va incrementándose con el ancho de la ventana, cosa que antes no pasaba. Además el hecho de que la imagen no tenga inicialmente un tamaño declarado supone que el navegador deberá cargarla completamente para seguir renderizando el resto de la página. Y esto es una desventaja en relación con la optimización de la carga de una página, donde todas los elementos <img> deberían ser especificados en tamaño.