EL DISEÑO WEB FLUIDO

Antes de la aparición del CSS, las páginas web se hacían a base de tablas. La estructura web o layout era de proporciones fijas, lo que significaba que se veía igual en todos los navegadores.

Se llegó así al estándar del grid de 960px, que dio pie a la creación del famoso [960 Grid System](http://960.gs/).

Así, las páginas web en su mayoría se diseñaban con un ancho de 960px, lo que daba una apariencia bastante unificada de las diferentes páginas web.

Con la aparición de pantallas de mayor resolución, el grid de 960px se fue quedando pequeño, con lo que empezaron a aparecer otros que permitían diseñar para una resolución mayor. El problema entonces era que había usuarios que seguían utilizando pantallas de baja resolución, con lo que tenían que utilizar la barra lateral horizontal para desplazarse por el contenido, lo cual no proporcionaba precisamente una gran experiencia de usuario.

Finalmente, con el lanzamiento del iPhone y el uso masivo de Internet desde los dispositivos móviles se vio que este sistema de diseño web fijo no era precisamente la mejor solución. Si, por ejemplo, seguías utilizando el grid fijo de 960px y navegabas con el iPhone necesitabas hacer un constante uso del zoom para poder ampliar los contenidos que te interesaba leer. Y al contrario, si utilizabas una pantalla de 27 pulgadas.

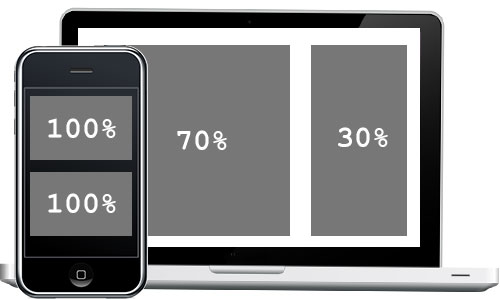
En mi opinión, el diseño web basado en un grid fijo está en vías de extinción. Igual que en su momento murieron las tablas para dar paso a los divs.

QUÉ ES EL DISEÑO WEB FLUIDO

El diseño web líquido o fluido toma su nombre del inglés fluid layout. Básicamente el diseño web líquido se adapta al ancho de ventana que tenemos disponible del navegador, de modo que cambia las proporciones en función del ancho de ventana. Así, un diseño web de dos columnas puede pasar a ser uno de una sola columna si, por ejemplo, el ancho de ventana es de 320px, como sucede con muchos dispositivos móviles.

La diferencia principal de la estructura web fija y la que se basa en un grid fluido es que en el primer caso utilizamos mayoritariamente píxeles para definir el espacio que ocupan los diferentes elementos, mientras que en el caso del diseño web líquido lo que utilizaremos serán **porcentajes**.

Veamos un ejemplo. Imaginemos que tenemos un layout basado en dos columnas y queremos que si el ancho de ventana es inferior a 480px pase a ser un diseño web de una sola columna. ¿Cómo lo hacemos? Muy fácil. Combinamos lo que ya sabemos de los media queries de CSS3 junto con el uso de porcentajes del layout fluido. Así, si para 960px la proporción es del 70% y el 30% para la zona central y la barra lateral respectivamente, en el caso de una sola columna ambas zonas ocuparán el 100%:



Simplificándolo mucho, a nivel de CSS podríamos hacer lo siguiente:

|  |  |  |
| --- | --- | --- |
| 1 | @media screen and (max-width: 479px) { | |
| 2 | #main { width:100%; } |

|  |  |  |
| --- | --- | --- |
| 3 | #sidebar { width: 100%;} | |
| 4 | } |

|  |  |  |
| --- | --- | --- |
| 5 | @media screen and (min-width: 480px) { | |
| 6 | #main { width:70%; } |

|  |  |  |
| --- | --- | --- |
| 7 | #sidebar { width: 30%;} | |
| 8 | } |

Y digo simplificando mucho, porque como posteriormente veremos, tendremos que tener en cuenta más anchos de ventana y no sólo que sea mayor o menor que 480px. Pero a título ilustrativo este ejemplo ya nos sirve. Este código lo que hace es que si el ancho de ventana es menor de 480px aplique el diseño de una sola columna, mientras que si es mayor entonces utilice el layout de dos columnas.

En este ejemplo vemos claro la potente combinación entre el uso de media queries de CSS3 y un diseño web fluido. Porque si bien podemos simplemente dar un ancho fijo a cada elemento a través de los media queries, es mejor utilizar un grid fluido si queremos garantizar una buena experiencia de usuario, pues los elementos se irán cambiarán de tamaño en función del ancho de ventana del navegador. Un ejemplo es la web de [Smashing Magazine](http://www.smashingmagazine.com/" \t "_blank). Te recomiendo abrirla desde tu ordenador e ir haciendo más pequeño el ancho de ventana del navegador, verás cómo absolutamente todo se ajusta al espacio disponible en pantalla.

DISEÑO WEB RESPONSIVE Y DISEÑO FLUIDO

Si buscas en Internet, verás que existe bastante confusión entre el diseño web responsive y el diseño fluido. Originariamente Ethan Marcotte dijo que para que un diseño web fuera responsive debía responder necesariamente a un layout líquido. Sin embargo, el [framework de CSS Skeleton](http://www.getskeleton.com/), disponible en, dice ser responsive cuando en sí utiliza una estructura fija. Entonces… ¿en qué quedamos?

En mi opinión, para que un diseño web se pueda considerar 100% responsive es necesario que sea también fluido, porque en otro caso lo que hace es saltar a una determinada resolución. En la misma página web de Skeleton encontrarás sitios web desarrollados con este framework. Te invito a hacer lo mismo que con el ejemplo anterior y verás lo que sucede. Si vas reduciendo o ampliando el ancho del navegador, verás que en este caso el contenido no cambia de posición hasta que salta a un layout diferente. Esto tiene una serie de ventajas, como que es más fácil de implementar, pero en palabras de Ethan Marcotte no se puede considerar plenamente responsive, pues no ofrece la mejor experiencia de usuario.

CÓMO CALCULAR LOS PORCENTAJES EN EL DISEÑO FLUIDO

Como comentaba, lo que más asusta al que nunca se ha puesto con el diseño web fluido es el tema de los porcentajes. Es aquello del más vale malo conocido que bueno por conocer, continuamos haciendo servir píxeles y no porcentajes porque nos sentimos más cómodos. Pero que no llegue la sangre al río, Ethan Marcotte creó una fórmula que nos va a facilitar enormemente la vida. Y que conste que no soy su fan, pero hay que reconocer que del tema domina y mucho. Veamos pues la fórmula:

target ÷ context = result

Que traducido sería:

**objetivo ÷ contexto = resultado**

Si recuerdas esta fórmula no tendrás ningún problema en crear de aquí en adelante sitios web basado en un diseño fluido, así que toma nota. Veamos un ejemplo práctico de cómo utilizarla.

Supongamos que tenemos un layout de 637px de ancho con dos columnas, una de 204px y la otra de 420px. Queremos convertir estas medidas a porcentajes.



Podemos definir el ancho de 637px como 100%. Ahora queremos ver cuánto ocupan los otros dos elementos. Es tan fácil como esto:

* (420px ÷ 637 px)\*100= 65.9340659%
* (204px ÷ 637px)\*100 = 32.025117%

En nuestro código CSS utilizaremos estos porcentajes, usando siempre todos los decimales. Veamos, por ejemplo, cómo quedaría la barra lateral de 204px:

|  |  |
| --- | --- |
| 1 | #sidebar {    width: 32.025117%;  } |

Veamos otro ejemplo:



Si queremos determinar el porcentaje del cuadrado que mide 100px de ancho, no nos debemos confundir y coger el contexto como los 960px, pues en este caso el contexto es el ancho de la barra lateral. Así, tendremos:

(100px ÷ 200px)\*100 = 50%

Es importante que antes de ponerte a hacer cálculos te fijes bien con respecto a qué elemento has de hacer los mismos.

FUENTES FLEXIBLES

Para que nuestro diseño web fluido esté completo no hemos de olvidar el configurar las fuentes para que se adapten también al ancho de ventana del navegador. Los porcentajes son a los píxeles lo que son los ems a las unidades que definen las fuentes.

|  |  |
| --- | --- |
| 1 | body {  font:  100%  "Times New Roman", serif; } |

Te estarás preguntando qué significa este 100%… bien, cada navegador trae un tamaño de fuente predeterminado. Al escribir 100% lo que decimos es que la fuente del body sea igual al tamaño de fuente definido por el navegador en cuestión.

Hay una fórmula que sirve para hacernos una idea aproximada de qué tamaño tiene una fuente y es la siguiente:

**100 % = 1 em**= 12 pt = 16 px

La parte en negrita siempre se cumple, pero el resto depende de si el usuario ha modificado el tamaño de fuente predeterminado del navegador, de aquí que si el usuario tiene una discapacidad visual y ha aumentado el tamaño predeterminado de la fuente del navegador a 20px, entonces 1 em será igual a 20px y no a 16px.

Con el diseño web responsive vale la pena que te acostumbres a definir las fuentes en **ems o porcentajes** y no en píxeles o puntos. Te recomiendo definir el body al 100% y así usarlo de referencia con el resto de fuentes del sitio web en cuestión.

En el caso que no quieras que se utilice como 100% el tamaño de fuente definido por el navegador, lo que puedes hacer es utilizar un [reset de CSS](http://es.wikipedia.org/wiki/Reset_CSS). El objetivo de los resets de CSS es evitar las diferencias visuales que se dan al mostrar una misma página web en diferentes navegadores.

Por ejemplo, la mayoría de navegadores establece un margen por defecto entre el contenido de la página web y su propia ventana, cuyo valor varía de un navegador a otro. Los diseñadores web para subsanar esa diferencia, suelen declarar la siguiente línea al comienzo de sus hojas de estilo:

\* { margin: 0; padding: 0; }

Esa única línea indica, y con el [selector universal](http://es.wikipedia.org/w/index.php?title=Selector_universal&action=edit&redlink=1) de CSS representado por un asterisco, que todos los elementos contenidos en el HTML al que se aplique, carecerán de márgenes. De esa manera, el diseñador se verá obligado a declarar después los márgenes necesarios en el diseño de su página web, en cada uno de los lugares donde sea necesario, sin tener que dejar ese aspecto a decisión de ningún navegador, y minimizando por tanto, las diferencias visuales entre los mismos.

Los Reset CSS pueden contener esa y otras muchas líneas de código que, en su conjunto, servirán al diseñador web para unificar su visualización entre navegadores.