DISEÑO WEB ADAPTABLE

"No puede impedirse el viento. Pero pueden construirse molinos."

(Proverbio holandés)

¿QUÉ ES RESPONSIVE WEB DESIGN?

El uso de dispositivos móviles para navegar por la web continúa creciendo a un ritmo vertiginoso, estos dispositivos a menudo están limitados por el tamaño de la pantalla y requieren un enfoque diferente sobre cómo se presenta el contenido en la pantalla.

El RWD (Diseño web adaptivo o adaptable) es una técnica de diseño web que consiste en crear una estructura de una página web que según el tamaño de la pantalla (o ventana) en la que se visualice variará su contenido para que siempre sea visible, de esta forma la estructura de la pagina se visualizara correctamente en todos los dispositivos sin necesidad de hacer zoom y se adaptara a los giros en los dispositivos móviles. En otras palabras, el sitio web debe tener la tecnología para responder automáticamente a las preferencias del usuario. Esto eliminaría la necesidad de un diseño diferente y un nuevo desarrollo para cada aparato en el mercado.

Nota

El concepto de Responsive Design es que el layout "responde" al tamaño de pantalla del dispositivo de ahí su nombre. En el sitio http://mediaqueri.es/ podemos encontrar varias inspiraciones para realizar nuestro sitio adaptable a cualquier medio.

El origen de este término fue introducido por Ethan Marcotte (diseñador web) donde describió los pilares de este enfoque:

1) **Fluid Grid (cuadricula fluida):** En lugar de pixeles utilizar porcentajes para definir los anchos de las columnas o divs.
2) **Imágenes Flexibles:** Las imágenes no tienen anchos fijos sino un máximo (o max-width).
3) **Media Queries**: Permiten consultar al CSS basándose en la resolución.

VENTAJAS DEL DISEÑO ADAPTABLE

Entre las ventajas de este enfoque, podemos considerar lo siguiente:

Mejor experiencia del usuario: Todos los usuarios ven a la web de la mejor manera posible accediendo desde cualquier dispositivo.
Costos más bajos de desarrollo y mantenimiento: Se reducen los costos en el desarrollo ya que no es necesario realizar un web para cada dispositivo. Por otro lado, cualquier modificación en el diseño se ve reflejado en todas las plataformas.
Búsqueda: Al realizar una búsqueda en los motores aparecerá solo una URL en los resultados por lo que se ahorra en redirecciones.
Mejor SEO: Para Google, aquellas páginas que no se adapten a los distintos dispositivos son penalizadas perdiendo posicionamiento en sus búsquedas.

UNIDADES DE MEDIDA EN CSS

Los pixeles son una unidad de medida rígida. Un pixel siempre se mostrará como un pixel. Al usar pixeles para definir las dimensiones de los objetos y los estilos de fuente no solo estamos definiendo tamaños rígidos, sino que también estamos ignorando las configuraciones que cada usuario pueda tener en su navegador.

Nota

Puede suceder que un usuario con dificultad visual aumente el tamaño de la fuente de su navegador para poder leer mejor. Si hemos definidos los tamaños en pixeles ignoraremos la configuración del usuario reemplazando por nuestras definiciones. En cambio, sí hemos definido en unidades relativas los elementos se ajustarán automáticamente.

Si necesitamos nuestro diseño fluido no podemos usar pixeles. Para usar medidas relativas es muy importante no usar la propiedad font-size dentro del body de nuestro documento HTML. Para ello es recomendable usar alguna de las tres medidas que mencionaremos a continuación.

PORCENTAJES

La unidad de medida porcentual es la que se usa por defecto en los elementos HTML en donde de manera predeterminada cada elemento de bloque usa un ancho del 100%, es por eso que cuando achicamos la ventana del navegador con una página que no tenga estilos, la página se adapta, ya que siempre usará el ancho total visible. Pero cuando hay layouts más complejos es necesario trabajar en esta unidad de medida.

Supongamos, por ejemplo, que tenemos un div que contiene todos los elementos de la página y, según el diseño, este elemento debiera medir 1200 pixeles. En lugar de caer en la tentación de simplemente usar esa medida en pixeles, se recomendaría usar una medida en porcentajes, en donde el máximo ancho del elemento sean esos 1200px:

```
#wrapper{
margin: 0 auto;
width: 90%;
max-width: 1200px;
}
```

Con estas tres propiedades de CSS conseguimos que, uno, el elemento se centre en la página, dos, tenga un ancho del 90% de la ventana y, tres, su ancho nunca sea superior a 1200 pixeles.
El uso de los porcentajes también lo podemos llevar a elementos interiores del layout, en donde, por ejemplo, podemos asignar a la columna principal de contenido y a la barra lateral unas medidas de ancho del 70% y el 30% respectivamente, haciendo que sean completamente adaptables al tamaño de su elemento contenedor.

EM

Un em, es básicamente el tamaño de una letra "M" (bien podría ser cualquier otra letra) del elemento al cual se esté aplicando esta medida. Es decir, si en elemento tiene aplicado un tamaño de fuente de 16 pixeles, entonces 1 em será igual a 16px (los navegadores de manera predeterminada definen un font-size de 16px al elemento HTML, por lo tanto, por defecto 1em es igual a 16px).
La unidad em es escalable y siempre depende de su elemento padre. Por ejemplo, si el elemento body tiene un tamaño de fuente de 16px y un elemento hijo tiene una fuente con tamaño 1.3em, este texto se mostrará de un tamaño un 30% más grande que el del body (20.8px), mientras que si dentro de ese elemento tenemos otro hijo con un font-size de 1.3 em, el tamaño de fuente de este objeto sería un 30% más grande que el tamaño de su padre (27.04px).

Body = 1em (16px)
Hijo = 1.3em (16px x 1.3 = 20.8px)
Nieto = 1.3em (20.8px x 1.3 = 27.04px)

Es recomendable usar la unidad de medida em para definir los tamaños de fuente, los altos de línea y también para elementos de diseño que no requieran ser muy exactos o que requieran una medida que tenga relación con el tamaño del texto, como por ejemplo el margen entre párrafos.

REM

La unidad de medida rem es muy similar a em, con la única diferencia de que no es escalable, esto quiere decir que no depende del elemento padre, sino del elemento raíz del documento, el elemento HTML. Rem significa "Root Em", o sea, es un em basado en la raíz.

Esto significa que si el elemento HTML tiene un tamaño de fuente de 16px (como es por defecto), entonces 1rem, sería igual a 16px, y si queremos aplicar un tamaño basado en rem a cualquier elemento de la página, no importará cual sea el tamaño de fuente que tenga asociado ese elemento, ya que 1 rem siempre será igual a 16 pixeles a no ser que se modifique el elemento raíz. Por ejemplo, en este caso se define un html con el 100% es decir tomará el valor en pixeles definido por defecto en el navegador (generalmente 16 pixeles), luego el padding definido en el selector h1 tomará 5 rem que equivale a multiplicar dicho valor por 16 (el resultado final será 80 pixeles).

```
html {
  font-size: 100% /* Esto significa 16px por defecto */
}

h1 {
  font-size: 18px;
  padding: 5rem; /* 5 * 16px = 80px */
}
```

Usar rem nos permite cierta estructura para poder definir ciertas partes del layout, pero al mismo tiempo nos entrega cierta escalabilidad para respetar las configuraciones de cada usuario. Esta unidad de medida es recomendable para aplicar a elementos del layout que requieran medidas fijas y eventualmente también para textos que deseemos que tengan un tamaño de fuente que no dependa de su elemento padre. Para poder convertir una medida de pixeles a rem solo tiene que multiplicar el tamaño que quiere obtener por el número 0.0625, eso le dará el tamaño que debes usar en rem. Así es como se ha llegado a definir que 75rem es igual a 1200px:

75rem = 1200px x 0.0625

Nota

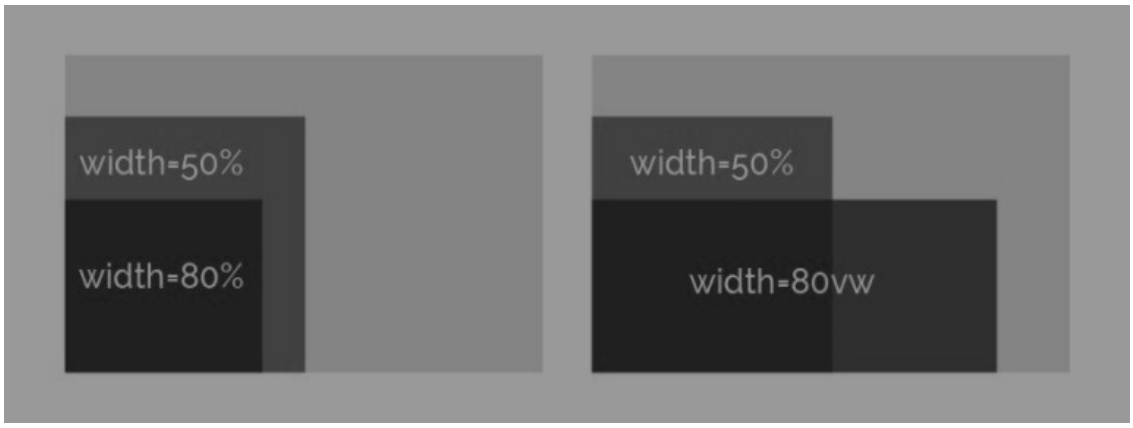
Todavía no parece haber una forma estándar de la industria de manejar la tipografía en forma responsiva algunos prefieren usar em, otros rem y algunos cuantos combinaciones de los dos.

VIEWPORT UNITS CSS

Las Viewport Units son unidades CSS relativas al viewport del navegador (donde se visualiza nuestro HTML). Gracias a estas medidas podemos definir un alto con la medida vh y un ancho con la medida vw basándonos en el tamaño de nuestro viewport. Sus valores se expresan en porcentajes (de 0 a 100).

```
#main-content {
     width:100vw; /* ancho del 100% de nuestro viewport */
    height:50vh; /* altura del 50% de nuestro viewport */
}
```

Las unidades de viewport pueden parecer similares a los porcentajes, pero son muy diferentes. En el caso de porcentajes, el ancho o alto del elemento hijo es determinado con respecto a su padre mientras que las unidades viewports siempre heredan los valores del viewport.



FLUID GRID (CUADRICULA FLUIDA)

El concepto de Fluid Grids o cuadriculas fluidas se basa en usar grillas o cuadriculas en los que se incorporan los elementos de diseño de la página. Los grids de una web pueden ser muy diversos y no existe fórmula para establecer la cantidad de columnas o el ancho de cada una de ellas. Hoy en día la mayoría de estos sistemas usa un total de 12 columnas.

Nota

El framework bootstrap, uno de los frameworks para diseño web más usado hoy en día, trabaja con cuadriculas flexibles usando doce columnas del mismo tamaño.

EL META NAME VIEWPORT

Para intentar entenderlo, primero habría que saber que significa "viewport". El viewport hace referencia al área en el cual el navegador muestra el documento html. Cuando no se diseñaban webs para móviles, los anchos de la página se hacían pensando en una pantalla de computadora. El problema estaba a la hora de ver la web en un móvil. Por ejemplo, en un iPhone. Aunque el ancho de pantalla puede ser de 320px, el viewport emula una de 980px y ajusta la web al tamaño de la pantalla. Por lo que todos los elementos de la página quedan a una escala más pequeña y dificulta la visión. Esto nos obliga a hacer zoom para verla medianamente bien. Es decir que cuando se

visualizan los sitios webs en un dispositivo móvil a menudo se reducen los contenidos, para conseguir que se ajusten al reducido espacio de la ventana del navegador.

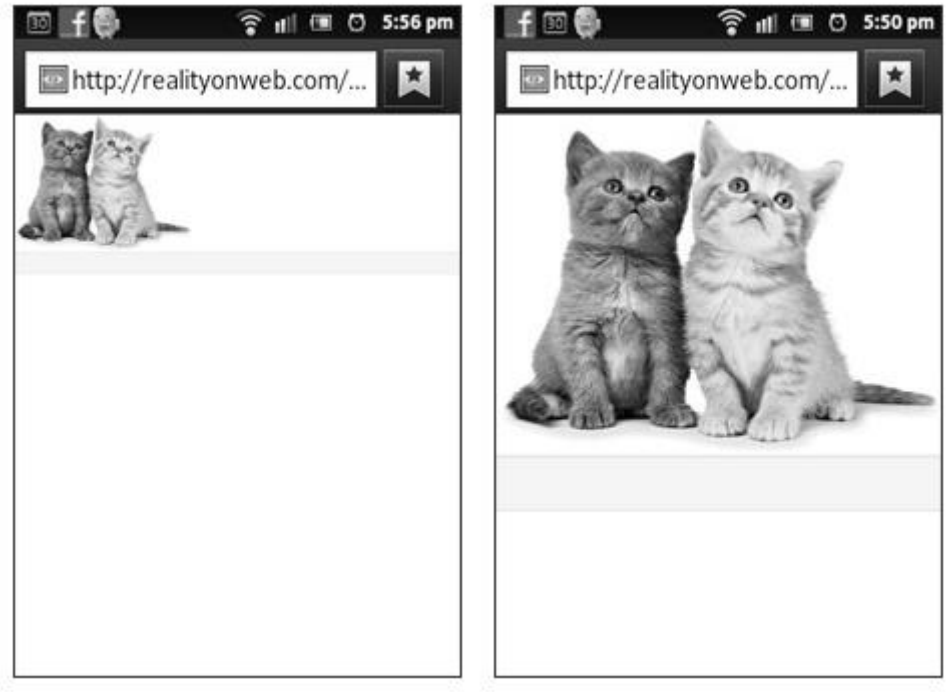


Figura 12.1 Sin viewport a la izquierda solo ocupa una parte de la pantalla, con viewport a la derecha, este caso no aplica un zoom sino que utiliza el ancho del dispositivo.

La etiqueta viewport creada por Apple indica que el ancho del sitio web se ajustará de manera automática al dispositivo móvil del usuario.
Este meta-tag debe ser incluido dentro de la etiqueta head y debe ser representado de la siguiente manera:

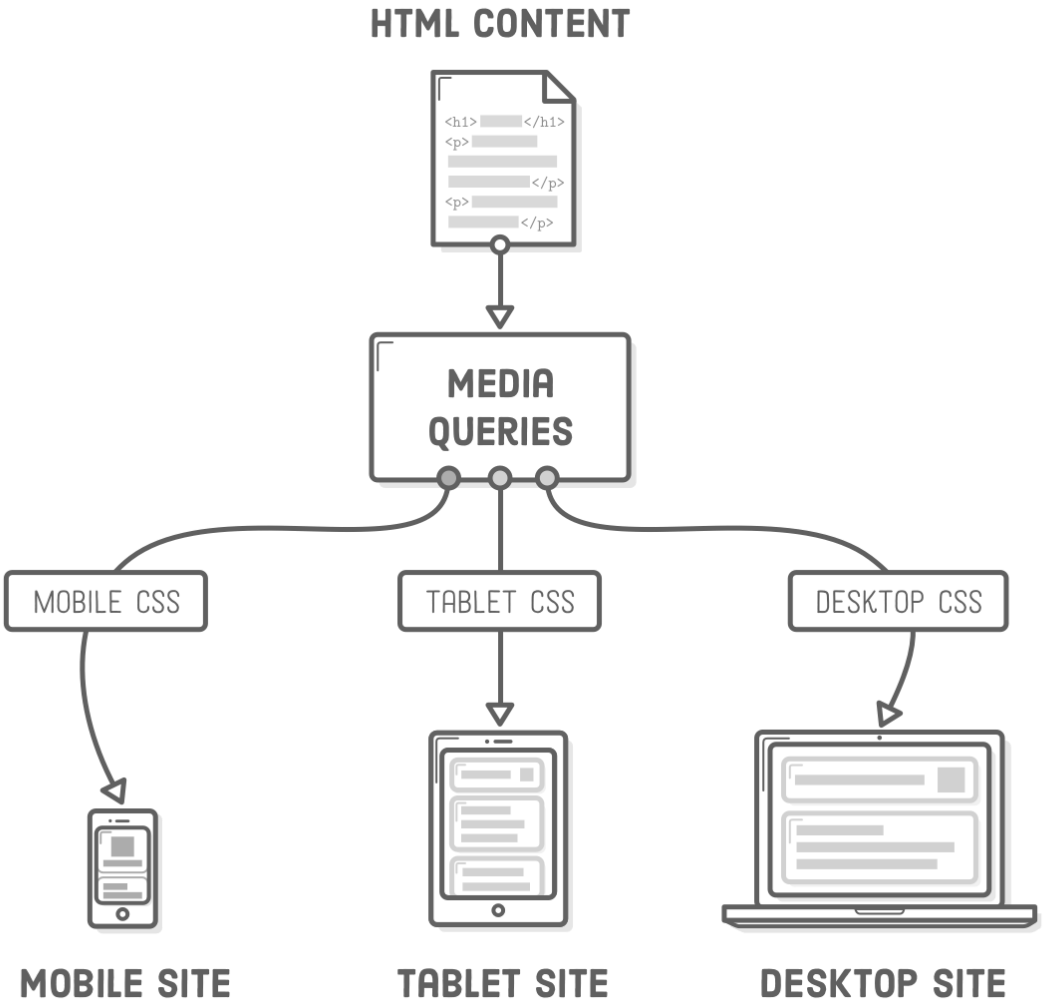
```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Se pueden usar los siguientes parámetros (separados por comas):

width=device-width: Con este valor conseguimos que el viewport sea igual a la anchura real de la pantalla del dispositivo, de modo que no se tratará de emular una pantalla mayor de lo que realmente es y veremos los píxeles reales.
initial-scale=1.0: Con este valor conseguimos que no se haga zoom sobre el documento. Es bien simple, el contenido de la web no se transformará, ni se agrandará, ni se hará menor.

USO DE MEDIA QUERIES

Los Media Queries (@media) forman parte de CSS3, son simples filtros (querys) que pueden aplicarse a los estilos CSS. Facilitan el cambio de estilos según las características del dispositivo, como el tipo de pantalla, el ancho, el alto, la orientación e incluso la resolución.



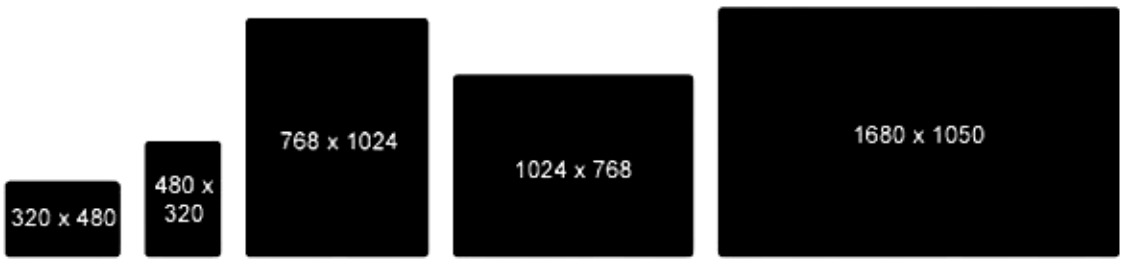
El truco está en prever la manera en que nuestro sitio se desplegará en distintos dispositivos y tamaños de ventana.
Por ejemplo, si quisiéramos aplicar un layout exclusivo para dispositivos que funcionen con un ancho de ventana menor a 980 píxeles, bastaría con utilizar el siguiente conjunto de reglas:

```
/* estilos 980 px o menos */
@media screen and (max-width: 980px){
    #contenedor{
        width: 90%;
    }
    #contenido{
        width: 60%;
    }
    #barra_lateral{
        width: 30%;
    }
}
```

Como se puede ver la instrucción se compone de dos partes: el tipo de medio utilizado (o Media Type, en este caso “screen”, que agrupa a todos los medios que se ven vía una pantalla) combinándolo mediante un “and” con el Media Query (max-width: 980px). Estamos preguntando: ¿Es un medio con pantalla y tiene un ancho de 980px o menor? Entonces le aplicamos los estilos situados entre las llaves correspondientes.
Si quisiéramos realizar lo mismo para el ancho de un dispositivo móvil podríamos hacer esto:

```
@media screen and (max-width: 480px){
    #encabezado{
        height: auto;
    }
    #contenido{
        width: auto;
    }
    #barra_lateral{
        display:none;
    }
}
```

En esta media query quitamos la barra lateral para que no ocupe espacio, asignamos un ancho automático al div de contenido y hacemos que el encabezado de nuestro sitio tenga una altura adaptable.
Si reemplazamos la palabra screen por all puede usarse para cualquier tipo de dispositivo. También podemos definir media queries que cumplan más de una condición, a través del operador and.
Así puede utilizar todos los media query que desee. Debe tener presente que los media queries pueden estar en la misma hoja de estilo o en un archivo separado, todo dependerá de la manera que quiera organizarse.
Ahora viene la pregunta del millón: ¿Cuantas medias queries vamos a definir y para que anchos?



No hay nada generalizado aunque es una buena opción definir 3 o 4 media queries genéricos: Una para móviles, otra para tablets, para PCS y dispositivos más amplios como los Smart Tv.

```
/* Pantallas más amplias */
@media (min-width: 1200px) { ... }

/* Tablets, Laptops y Desktop */
@media (min-width: 768px) and (max-width: 979px) { ... }

/* Smartphones y Tablets */
@media (max-width: 767px) { ... }

/* Smartphones */
@media (max-width: 480px) { ... }
```

Nota

En el sitio http://screensiz.es puede comparar las distintas resoluciones de teléfonos, tabletas y monitores.

IMÁGENES FLUIDAS

Una imagen tiene dimensiones fijas y si es más grande que la ventana gráfica, se formará una barra de desplazamiento. Supongamos que tenemos una imagen de 400 pixeles de ancho. Vamos a declarar que dicha imagen tenga un ancho relativo en CSS. En este caso del 100%:

```
img {width: 100%}
```

Ahora la imagen siempre se ajustará a su contenedor, es decir: Si está dentro de un artícle, ocupará el 100% del ancho de dicho artículo. Si está dentro de un aside siempre ocupará el 100% del ancho. El problema es: ¿Qué sucede cuando nuestro contenedor supera el tamaño en píxeles de la imagen? Supongamos un contenedor de 1200 píxeles.
¿Qué hace nuestra imagen fluida? Llenar el 100% del espacio. Por lo tanto, nuestra imagen de 400px pasa a convertirse en una imagen de 1200px. Es decir, sigue siendo adaptable pero ahora también pixelada. La solución ante este problema es la propiedad CSS, max-width:

```
img { max-width: 100%}
```

Mediante dicha propiedad vamos a indicar que la imagen tenga su ancho en píxeles, pero que como máximo, siempre esté al 100% de la anchura del contenedor. Dicho de forma clara:

Si el contenedor tiene 300px: nuestra imagen tendrá 300px (max-width: 100%). Si el contenedor tiene 400px: nuestra imagen tendrá 400px (max-width: 100%)
Si el contenedor tiene 1200px: nuestra imagen se queda en 400px..
Con la propiedad height normalmente no hay problema ya que nos guiamos mas por el ancho del dispositivo, lo mejor siempre para estos casos es asignarle el valor auto, es decir, delegar al navegador que calcule el alto del elemento. En la mayoría de los casos, esto hará que el alto no pierda proporción con respecto al ancho. Si el ancho de la imagen se reduce para adaptarse, el alto también lo hará. Si el ancho se queda fijo, el alto también lo hará. Finalmente, para nuestros trabajos las propiedades sobre imágenes a modificar son las siguientes:

```
img {
height: auto;
max-width: 100%;
}
```