

Curso HTML5, CSS y JavaScript

Módulo 3: Diseño web responsive

- emilianoagustingallo@gmail.com
- <https://www.linkedin.com/in/emiliano-gallo/>



¿Qué es el diseño web responsive?

- El diseño web adaptable o adaptativo (en inglés, **Responsive Web Design**) es una técnica de diseño y desarrollo web que mediante el uso de **estructuras e imágenes fluidas**, así como de **media-queries**, consigue adaptar el sitio web al entorno del usuario.
- El diseño web responsive utiliza solo **HTML** y **CSS**.
- El diseño web responsive no es un programa o código **JavaScript**

Diseñamos para todos los tamaños

Las páginas web se pueden ver usando muchos dispositivos diferentes

- Computadoras de escritorio
- Tabletas
- Teléfonos

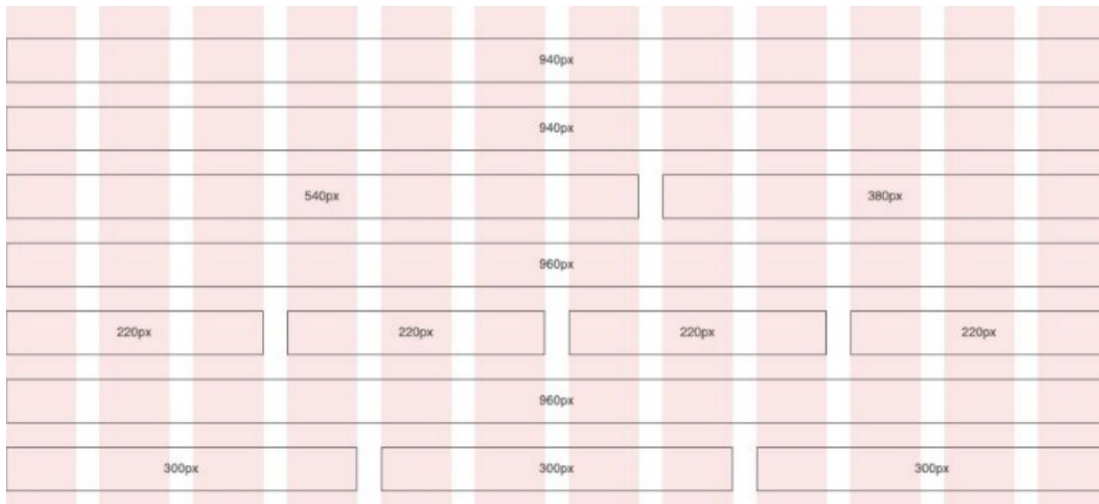
Y deben verse bien y ser fáciles de usar, independientemente del dispositivo.

Las páginas web no deben omitir información para adaptarse a dispositivos más pequeños, sino adaptar su contenido a cualquier dispositivo

Conceptos para empezar

Para poder empezar necesitamos entender ciertos conceptos, que constan en principio o en forma general de los tres siguientes

- Grillas
- Componentes flexible
- Media Queries



Ventajas del diseño responsive

Las ventajas de trabajar con diseños responsive, en sí es que cualquier usuario puede acceder y entender el contenido, lo cual genera mayor conversión ya sea de servicios, o de productos

- Mejora experiencia del usuario (usabilidad y conversión)
- Mejora mantenimiento
- Evita contenido duplicado
- Mejora el posicionamiento Web

Diseño fluido

En este tipo de diseño se usan porcentajes en lugar de píxeles, para que los elementos se adapten según el ancho de la pantalla. Si bien el resultado puede parecer atractivo en pantallas medianas, como computadores de escritorio y tablets, se producen muchos problemas en las pantallas grandes y pequeñas.

Por ejemplo, en televisores las imágenes se estiran mucho y en teléfonos los textos son difíciles de leer.

Diseño adaptable

Usa plantillas estáticas basadas en puntos de quiebre. Cuando la pantalla alcanza cierto límite de tamaño, se cambia a otro diseño.

Por ejemplo, puedes diseñar una página en tres dimensiones diferentes. Cuando un usuario visite tu sitio, los archivos adaptativos detectarán el aparato que está usando y mostrarán la plantilla adecuada.

Sin embargo, ofrece un bajo nivel de usabilidad y muestra una versión reducida de los contenidos en móviles, un medio que crece con fuerza y gana cada vez más relevancia.

Diseño responsive

En el Responsive Web Design el diseño y el contenido se adaptan a cada pantalla, entregando una experiencia de usuario muy similar en resoluciones bajas, altas o en formatos de distintas pulgadas.

Las características a tener en cuenta son

- Ancho y alto de la ventana del navegador
- Orientación del dispositivo
- Proporción entre el alto y ancho de la pantalla
- Resolución del dispositivo, es decir, la precisión del detalle en las
- Imágenes de mapa de bits.

Siempre **mobile first**

Significa diseñar para dispositivos móviles antes de diseñar para computadoras de escritorio o cualquier otro dispositivo (esto hará que la página se muestre más rápido en dispositivos más pequeños)

Esto significa que debemos hacer algunos cambios en nuestro CSS

En lugar de cambiar los estilos cuando el ancho se hace más pequeño que 768px, deberíamos cambiar el diseño cuando el ancho se hace más grande que 768px. Esto hará que nuestro diseño sea móvil primero

```
/* For mobile phones: */
[class*="col-"] {
  width: 100%;
}

@media only screen and (min-width: 768px) {
  /* For desktop: */
  .col-1 {width: 8.33%;}
  .col-2 {width: 16.66%;}
  .col-3 {width: 25%;}
  .col-4 {width: 33.33%;}
  .col-5 {width: 41.66%;}
  .col-6 {width: 50%;}
  .col-7 {width: 58.33%;}
  .col-8 {width: 66.66%;}
  .col-9 {width: 75%;}
  .col-10 {width: 83.33%;}
  .col-11 {width: 91.66%;}
  .col-12 {width: 100%;}
}
```

Porque mobile first

El 85% de la población utiliza móviles, este dato no es menor. De esta manera, en realidad , más que **Mobile First** nos centramos en **Content First**, es decir en el contenido y lo que queremos comunicar más que en la cuestión más superficial y exterior de la página

- Se simplifica el diseño
- Se pone el contenido en primer lugar y se trata de priorizar lo esencial
- No hay información de más sin tanto sentido para comprender el sitio

Ventana gráfica viewport

- El viewport es el área visible del usuario de una página web
- Varía con el dispositivo y será más pequeña en un teléfono móvil que en la pantalla de una computadora
- Antes de las tabletas y los teléfonos móviles, las páginas web estaban diseñadas solo para pantallas de computadora, y era común que las páginas web tuvieran un diseño estático y un tamaño fijo.
- Fuente <meta> tag: https://www.w3schools.com/tags/tag_meta.asp

Configurando el viewport

HTML5 introdujo un método para permitir que los diseñadores web tomen el control de la ventana gráfica, a través de la etiqueta `<meta>`

Debe incluir el siguiente elemento de ventana gráfica `<meta>` en todas sus páginas web:

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

Esto le da al navegador instrucciones sobre cómo controlar las dimensiones y la escala de la página.

La parte `width = device-width` establece el ancho de la página para seguir el ancho de la pantalla del dispositivo (que variará según el dispositivo).

La parte `initial-scale = 1.0` establece el nivel de zoom inicial cuando el navegador carga por primera vez la página.

Importancia de viewport



Without the viewport meta tag



With the viewport meta tag

Ajustar el contenido al viewport

Los usuarios están acostumbrados a desplazarse verticalmente por sitios web, pero no horizontalmente!

Por lo tanto, si el usuario se ve obligado a desplazarse horizontalmente, o alejarse, para ver toda la página web, resulta en una mala experiencia del usuario.

- NO usar elementos grandes de ancho fijo: por ejemplo, si una imagen se muestra con un ancho más ancho que la vista, puede hacer que la vista se desplace horizontalmente
- NO permitir que el contenido se base en un viewport de ancho fijo para renderizar bien: dado que las dimensiones de la pantalla y el ancho en píxeles CSS varían ampliamente entre dispositivos
- Usar **media query** para aplicar diferentes estilos para pantallas pequeñas y grandes

Media queries

Para entender el concepto de Media Query, iremos a la definición oficial del W3C que dice: **Una media query consiste en un tipo de dispositivo (media) y cero o más expresiones que comprueban el estado de las características particulares de ese dispositivo.**

Mediante el uso de media queries, la presentación del contenido puede adaptarse a un rango específico de dispositivos de salida sin necesidad de modificar el propio contenido. Una Media Query es una serie de condiciones que pueden aplicarse a nuestras hojas de estilo CSS para que, según el tamaño y resolución del dispositivo, adopte una u otra forma

Media queries

La consulta de medios es una técnica CSS introducida en CSS3.

Utiliza la regla `@media` para incluir un bloque de propiedades CSS solo si cierta condición es verdadera.

En el siguiente ejemplo, si la ventana del navegador es 600px o menor, el color de fondo será azul claro

```
@media only screen and (max-width: 600px) {  
  body {  
    background-color: lightblue;  
  }  
}
```


Media queries al importar **CSS**

También se pueden importar diferentes archivos de CSS dependiendo la condición media query

```
<link rel="stylesheet" media="screen and (min-width: 900px)" href="widescreen.css">  
<link rel="stylesheet" media="screen and (max-width: 600px)" href="smallscreen.css">
```

Uso de **landscape** y **portrait**

Describe la proporción de aspecto del dispositivo de salida. Este valor consiste de dos enteros positivos separados por una barra ("/").

Este representa la proporción de aspecto de los píxeles horizontales (primer término) a los píxeles verticales (segundo término).

@media screen and (device-aspect-ratio: 16/9){ ... }

@media screen and (device-aspect-ratio: 16/10) { ... }

También se puede utilizar el **min-aspect-ratio** donde por ejemplo la proporción debe ser al menos la misma

Puntos de quiebre breakpoints

Los breakpoints son los puntos de rotura o interrupción donde el diseño sufrirá un cambio drástico.

Según algunos autores, se deben cumplir 2 reglas:

- El contenido es el que marca el breakpoint y no el dispositivo
- El breakpoint no debe ser en píxeles, ha de ser en “em”, para darle el control al dispositivo

Se pueden crear tantos breakpoints como nos resulte necesario, es habitual crear entre 3 y 4 breakpoints, a los cuales se los llama comúnmente como small, medium, large y extra large

Puntos de quiebre breakpoints

Nos puede pasar que al crear una página web con filas y columnas, estas no se ajusten correctamente en una pantalla pequeña.

Las **media queries** pueden ayudarnos con eso. Podemos agregar un punto de interrupción donde ciertas partes del diseño se comporten de manera diferente en cada lado del punto de interrupción.

Cuando la ventana se vuelve más pequeña que 768 px, cada columna debe tener un ancho del 100%:

```
/* For desktop: */  
.col-1 {width: 8.33%;}  
.col-2 {width: 16.66%;}  
.col-3 {width: 25%;}  
.col-4 {width: 33.33%;}  
.col-5 {width: 41.66%;}  
.col-6 {width: 50%;}  
.col-7 {width: 58.33%;}  
.col-8 {width: 66.66%;}  
.col-9 {width: 75%;}  
.col-10 {width: 83.33%;}  
.col-11 {width: 91.66%;}  
.col-12 {width: 100%;}
```

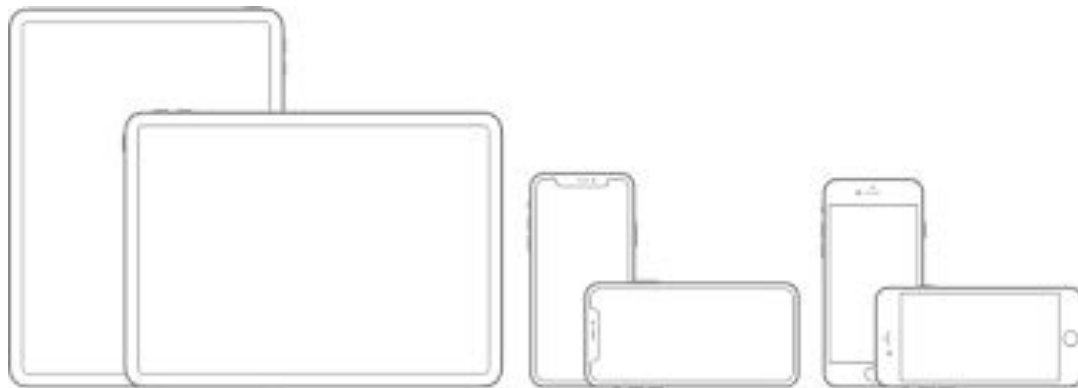
```
@media only screen and (max-width: 768px) {  
  /* For mobile phones: */  
  [class*="col-"] {  
    width: 100%;  
  }  
}
```

Uso de device-aspect-ratio

El valor de orientation es portrait (retrato) cuando el valor de la altura del medio de salida es mayor o igual al valor de la anchura de ese medio. De lo contrario el valor es landscape (paisaje).

@media screen and (orientation: landscape) { }

@media screen and (orientation: portrait) { }



Propiedad display

Especifica el comportamiento de visualización (el tipo de cuadro de representación) de un elemento.

En HTML, el valor de propiedad de visualización predeterminado se toma de las especificaciones HTML o de la hoja de estilo.

Referencia: https://www.w3schools.com/cssref/pr_class_display.asp

Ejemplo: https://www.w3schools.com/cssref/playit.asp?filename=playcss_display&preval=table

A trabajar!!



Diseñar un sitio responsive

- Diseñar un sistema de grillas de 4 columnas para dispositivos desktop y de 1 columna para dispositivos celulares y tablets
- Ubicar distintos elementos en las columnas (imagenes, textos, párrafos, etc)
- Al pasar de la vista 'desktop' a la 'mobile' los componentes tienen que acomodarse en una sola columna

