



**Clase 6**

# **Responsive Design**





# ***OBJETIVO DE LA CLASE***

- Incorporar Responsive + Mobile First

# Términos a tener en cuenta:

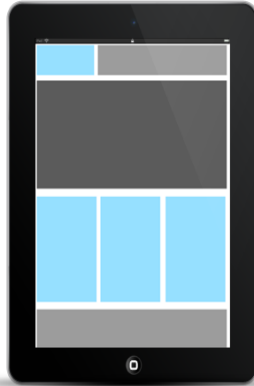
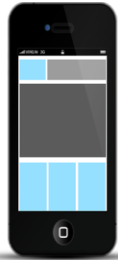
**Diseño responsive:** se refiere a la idea de que un sitio web debería mostrarse igual de bien en todo tipo de dispositivo, desde monitores de pantalla panorámica hasta teléfonos móviles.

El diseño responsive se logra a través de "Media Queries" de CSS. Pensemos en las Media Queries como una forma de aplicar condicionales a las reglas de CSS.

**Mobile First:** significa crear el código primero para los dispositivos más pequeños que los usuarios probablemente tengan, como teléfonos o tabletas. Implica trabajar en el dispositivo más pequeño, y luego acumular desde allí todo en el mismo código y el mismo proyecto, en lugar de hacer uno nuevo para cada tamaño de pantalla.

**Meta viewport:** una etiqueta `<meta> viewport` da al navegador las instrucciones sobre cómo controlar las dimensiones, y el ajuste a escala de la página.

# MOBILE FIRST



# MOBILE FIRST

Antes de hablar de “Mobile-First” debemos hacer una referencia al llamado Diseño Responsive. **Se refiere a la idea de que un sitio web debería mostrarse igual de bien en todo tipo de dispositivo,** desde monitores de pantalla panorámica hasta teléfonos móviles.

Es un enfoque para el diseño y desarrollo web que elimina la distinción entre la versión amigable para dispositivos móviles de un sitio web y su contraparte de escritorio. Con un diseño responsive **ambos son lo mismo.**



# Mobile First

"Mobile First" significa **crear el código primero para los dispositivos más pequeños** que los usuarios probablemente tengan, como teléfonos o tabletas. Trabajar en el dispositivo más pequeño y luego acumular desde allí todo en el mismo código y el mismo proyecto, en lugar de uno nuevo para cada tamaño de pantalla.



# Mobile First

Se recomienda:

1. Primero trabajar el código para que se reproduzca perfectamente en un teléfono.
2. Luego ajustar para que se ejecute en una tableta.
3. Luego en un dispositivo de escritorio.

# Mobile First

Cualquier estilo dentro del siguiente Media Querie se ejecutará cuando el tamaño de la pantalla sea de al menos 768px de ancho -tablet portrait iPad Mini- pero no cuando el tamaño de la pantalla sea menor:

## Estilos Mobile

```
@media screen and (min-width: 768px) {  
  .body {  
    background-color: #000000;  
  }  
}
```





# Media Queries

El diseño responsive se logra a través de "Media Queries" de CSS. Pensemos en las Media Queries como una forma de aplicar condicionales a las reglas de CSS. Estas últimas **le dicen al navegador qué reglas debe ignorar o aplicar dependiendo del dispositivo del usuario.**

# ***BREAK POINTS - Categorías***

Tamaño	Dispositivo
320px	Para dispositivos con pantallas pequeñas, como los teléfonos en modo vertical
480px	Para dispositivos con pantallas pequeñas, como los teléfonos, en modo horizontal
600px	Tabletas pequeñas, como el Amazon Kindle (600×800) y Barnes & Noble Nook (600×1024), en modo vertical
768px y 1023px	Tabletas de diez pulgadas como el iPad (768×1024), en modo vertical
1024px	Tabletas como el iPad (1024×768), en modo horizontal, así como algunas pantallas de ordenador portátil, netbook, y de escritorio
1200px	Para pantallas panorámicas, principalmente portátiles y de escritorio

# Meta viewport

Las páginas optimizadas para diferentes dispositivos deben incluir la etiqueta `<meta> viewport` en el encabezado del documento HTML. Una etiqueta `<meta> viewport` da al navegador las instrucciones sobre cómo controlar las dimensiones y el ajuste a escala de la página.

- Usa la etiqueta `<meta> viewport` para controlar el ancho y el ajuste de la ventana de visualización del navegador.
- Incluye `width=device-width` para hacer coincidir el ancho de la pantalla en píxeles independientes del dispositivo.
- Incluye `initial-scale=1` para establecer una relación de 1:1 entre los píxeles CSS y los píxeles independientes del dispositivo.

# *Meta viewport*

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

El uso del valor de *width=device-width* indica a la página que debe hacer coincidir el ancho de la pantalla en píxeles independientes del dispositivo. Esto permite que la página realice el reprocesamiento del contenido para adaptarlo a diferentes tamaños de pantalla.

# *Meta viewport*

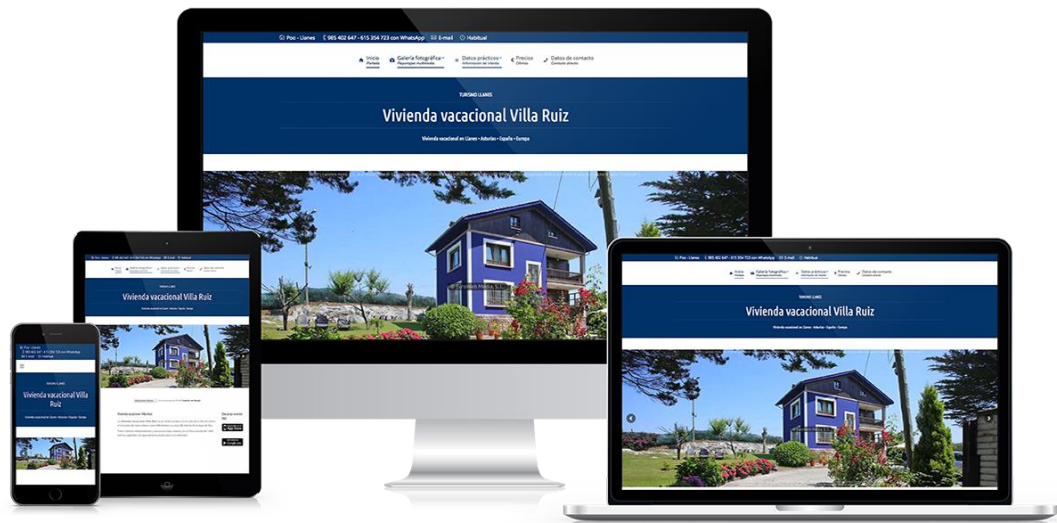




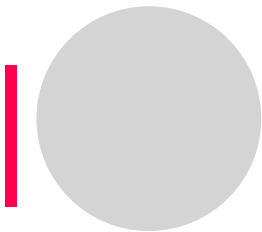
**RESPONSIVE**



# Página Responsive



El sistema **detecta automáticamente** el ancho de la pantalla y a partir del mismo **adapta todos los elementos de la página**, desde el tamaño de letra hasta las imágenes y los menús, para ofrecer al usuario la mejor experiencia posible.



Responsive Web Design

Mobile First Web Design





# MEDIA QUERIES

Recordemos los rangos que define Bootstrap:

- Pantallas extra pequeñas (móviles)  $< 576\text{px}$
- Pantallas pequeñas (\_sm, tablets \_en vertical)  $\geq 576\text{px}$
- Pantallas medianas (md, para tablets en horizontal)  $\geq 768\text{px}$
- Pantallas grandes (lg, tamaño escritorio)  $\geq 992\text{px}$
- Pantallas extra grandes (xl, escritorio grande)  $\geq 1200\text{px}$

[Breakpoints en Bootstrap](#)

# Ejemplo de Media Queries

Si por ejemplo queremos que en las pantallas extra pequeñas (xs) el color de fondo que aplica la clase *.miestilo* sea rojo y para el resto de tamaños sea verde, podríamos hacer:

```
.miestilo {  
    background-color: green;  
}  
  
@media (min-width: 768px) {  
    .miestilo {  
        background-color: red;  
    }  
}
```

# Ejemplo de Media Queries

O si por ejemplo queremos variar la alineación del texto que se aplica en una clase a partir de las pantallas tipo escritorio:

```
.miestilo {  
    text-align: center;  
    font-size: 40px;  
}  
  
@media (min-width: 992px) {  
    .miestilo {  
        text-align: left;  
        font-size: 10px;  
    }  
}
```

# Ejemplo de Media Queries

Podemos también modificar el cuerpo de texto si lo utilizamos en px para diferentes pantallas:

```
@media (min-width: 600px) {  
  div.example {  
    font-size: 80px;  
  }  
@media (max-width: 600px) {  
  div.example {  
    font-size: 30px;  
  }  
}
```

Recordemos que  
esto no es necesario  
si utilizamos el font-  
size en “em”

# Operador “and” “Y”

Podemos sumar diferentes indicaciones con las Media Queries, se utilizan con el operador “and”. En este caso el estilo que definimos se reproducirá en pantallas que van de 400px a 700px:

```
@media (min-width: 768px) and (max-width: 1024px) {  
  .miestilo {  
    text-align: left;  
  }  
}
```

# Orientación

En este caso solo se reproducirá el estilo si la ventana tiene un ancho de de 700px o más y la pantalla esta en formato horizontal.

```
@media (min-width: 700px) and  
(orientation: landscape) {  
  .miestilo {  
    text-align: left;  
  }  
}
```

# Ejemplo de Media Queries

Modificación de menú de posición horizontal en pantallas grandes a vertical en pantallas chicas

HTML

```
<div class="topnav">
  <a href="#">Home</a>
  <a href="#">Nosotros</a>
  <a href="#">Contacto</a>
</div>
```

```
/* Top navigation bar */
.topnav {
  overflow: hidden;
  background-color: #333;}
```

```
/* Topnav links */
.topnav a {
  background-image: url('imagen1')
  display: none;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none; }
```

```
@media screen and (min-width: 600px) {
  .topnav a {
    display: block;
    width: 100%;
    background-image: url('imagen2')
  }
```

CSS

# MEDIA QUERIES

Podemos llegar a hacer cosas muy avanzadas y personalizar completamente el aspecto de una web según el tamaño del dispositivo.

- Cambiar el tamaño y la posición de una imagen.
- Cambiar la posición de cualquier elemento.
- Cambiar el tamaño de letra, la fuente o su color.
- Aplicar combinaciones de estilos avanzados.

**¡Cualquier cosa que se les ocurra!**