

Agencia de  
Aprendizaje  
a lo largo  
de la vida

# Desarrollo Fullstack



# Les damos la bienvenida

Vamos a comenzar a grabar la clase



## Clase 10

### Bootstrap

- ▶ Frameworks
- ▶ Instalación
- ▶ Containers
- ▶ Grilla
- ▶ Clases y Predefinidos
- ▶ Componentes

## Clase 11

### CSS Avanzado

- ▶ Diseño Responsive
- ▶ Viewport
- ▶ Mobile First
- ▶ Media Queries
- ▶ Breakpoint

## Clase 12

### Javascript

- ▶ Variables
- ▶ Tipos de datos
- ▶ Prompt, alert y console.

# CSS

De todos los tamaños



# RESPONSIVE

Un mismo diseño, muchos dispositivos.

# Diseño Responsive

El **diseño web responsive** tiene como objetivo **establecer** metodologías y estándares que nos permitan **adaptar** **nuestros diseños** a **distintos tamaños** de pantalla.

El fin es **mantener la experiencia** de navegación **intacta** sin importar el dispositivo donde se esté reproduciendo el sitio web.



# Viewport

Cuando hablamos de **viewport** nos referimos a la **ventana visible de nuestro sitio** en el navegador, es decir, el **ancho** y **alto** que componen **solamente el sitio que se está proyectando**.

Este concepto **es el mismo en cualquier entorno**, ya que siempre hace referencia espectro visible, así un **sitio se proyecte** en un **celular**, una **tableta** o un **monitor de escritorio**.

Para **asegurarnos** de **mantener siempre la misma escala de pixeles** frente a los cambios de resolución de los diferentes dispositivos, debemos utilizar la etiqueta **<meta>** con el **name viewport** para la escala 1:1.

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

# Metodologías Responsive



## Responsive Web Design (desktop first)

Propone que **nuestros sitios** sean pensados para verse en una **pantalla de escritorio** y **adaptar** los estilos para que se ajusten hasta el **tamaño de un celular**.

Desktop first



## Mobile First

Se piensan los **diseños** teniendo en cuenta los tamaños de **celulares** para luego ir adaptando los estilos para que el sitio se ajuste a **tamaños de pantalla más grandes**. Pensado para proyectos donde los usuarios acceden en mayor medida desde este tipo de dispositivos.

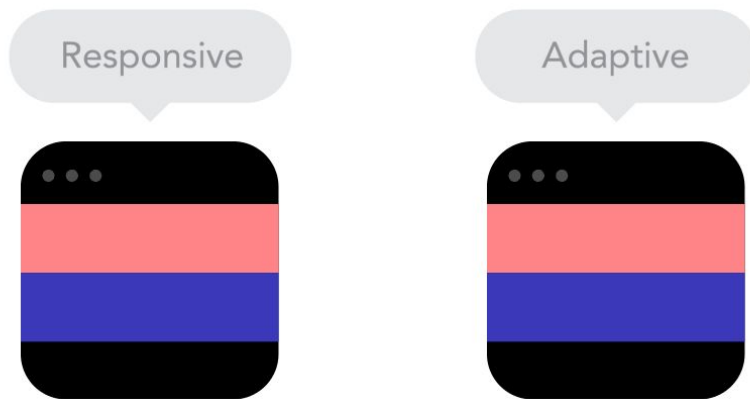
Mobile first



# Principios del diseño responsivo

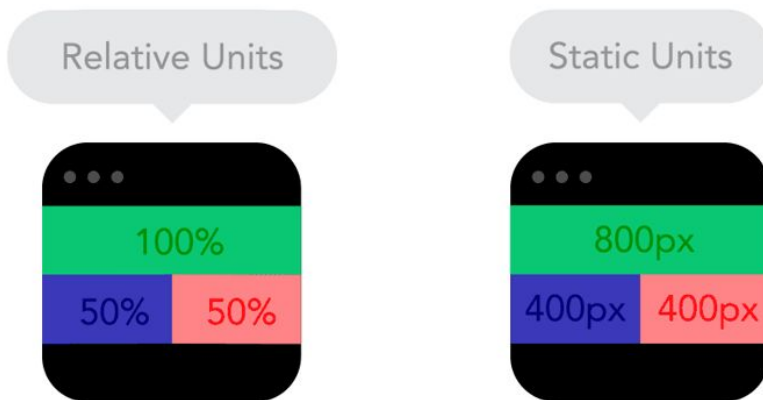
# Diseño Responsivo vs Adaptativo

El **Responsivo** es aquel donde **nuestro diseño corre de forma fluída** ante los cambios de tamaño de las distintas pantallas, mientras que el **Adaptativo** obedece a *cambios más marcados*, donde **el diseño se va adaptando** en los momentos que se **producen quiebres o rupturas** de los elementos del sitio. **Si bien ambos son diferentes, en muchos casos se complementan.**



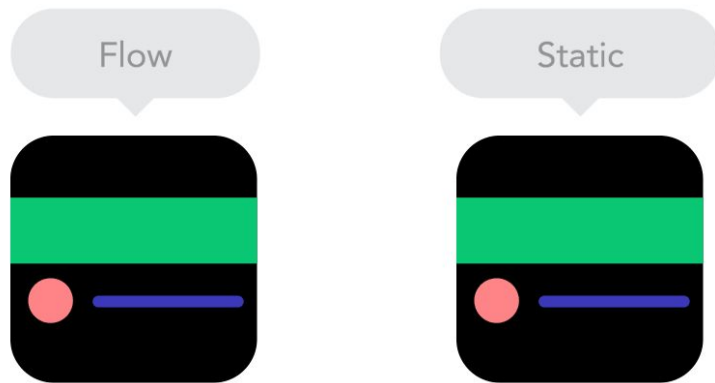
# Unidades Relativas

El uso de unidades relativas como **porcentajes**, **rem**, **em**, **vw** y **vh** nos dan la posibilidad de **generar proporciones dinámicas** que se definan en función de factores inherentes al tamaño de nuestras pantallas, de este modo **evitaremos estructuras rígidas** que favorecen los *overflows* y no logran adaptarse a los diversos cambios de resolución.



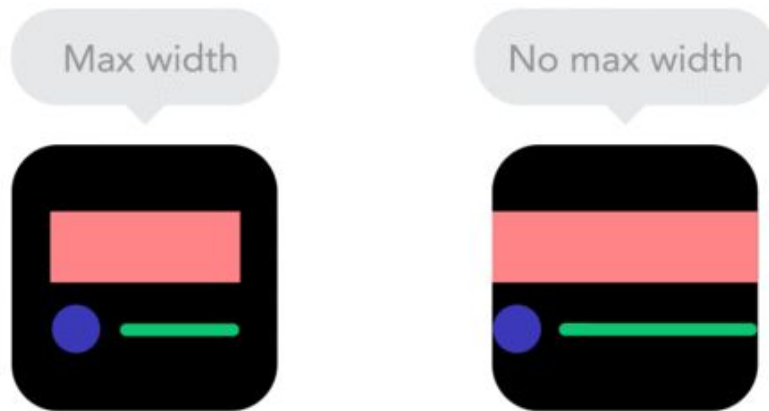
# Flujo de los elementos

Gracias al uso de unidades de medida relativas, podemos lograr que nuestros sitios se adapten uniformemente, **no solo horizontalmente**, sino **también verticalmente** haciendo que nuestras cajas fluyan de forma natural y mantengan siempre sus proporciones.



# Máximos y Mínimos

Cuando hablamos de **responsive** solemos **concentrarnos** en todo lo que sucede cuando nuestra pantalla disminuye de tamaño, pero rara vez pensamos en **qué pasaría** si alguien accede desde **pantallas grandes** como **monitores ultra wide o televisores**. Para manejar estos extremos siempre es recomendable aprovechar las propiedades `max-width` y `min-width`.



# ¿Cómo aplicamos responsive?

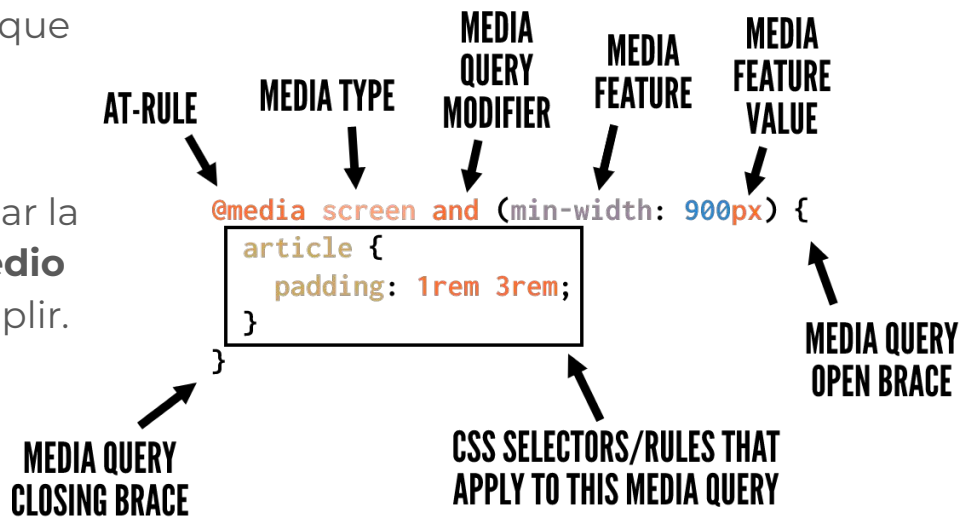
Existen múltiples maneras de **generar diseños que se adapten** a distintos tamaños de pantallas. Como **hemos visto anteriormente** en el curso podemos valernos de herramientas como **flex** o **grid** y sus propiedades que **nos permiten** manejar la forma en que **nuestras cajas fluyen frente ante** **estos cambios**.

**Otra forma** es el uso de **media queries** y **breakpoints** predefinidos para establecer qué debe hacer nuestro diseño en los **momentos que empieza a** **deformarse**.

# Media queries

Las **media queries** son un bloque de código condicional donde podremos escribir CSS que **sólo se aplicará** en caso de **cumplirse** la condición definida en **dicha regla**.

Para definir una media query debemos usar la palabra reservada **@media** seguida del **medio** donde debe aplicarse y la **condición** a cumplir.





# Tipos de Media

Los tipos de media query nos ayudan a **establecer** en qué casos debe aplicarse teniendo en cuenta el **medio** donde se **proyectará nuestro sitio**.

Según el medio elegido, estos **estilos** serán **válidos** por ejemplo **solo** en pantallas o cuando el sitio sea **impreso en papel**.

<b>screen</b>	Monitores o pantallas de ordenador. Es el más común.
<b>print</b>	Documentos de medios impresos o pantallas de previsualización de impresión.
<b>speech</b>	Lectores de texto para invidentes.
<b>all</b>	Todos los dispositivos o medios. El que se utiliza <b>por defecto</b> .

# Breakpoints

Son las **reglas condicionales** que le colocamos a nuestros media queries para que **apliquen** nuestros **estilos** según los **distintos escenarios**.

Estos **pueden adoptar diferentes valores** como **max-width** y **min-width** o **height**, landscape, portrait, **device-width**, entre otras.



# Breakpoints

Estas reglas condicionales se las conoce como **breakpoints** o puntos de quiebre y **dependiendo** el caso puede ser medidas estándar o estar basadas en necesidades puntuales de un sitio en particular.

En este ejemplo, definimos **3** comportamientos distintos **para un mismo elemento**, **dependiendo** el tamaño actual que tome la pantalla.

```
@media screen and (max-width: 480px) {  
  .element {  
    width: 100%;  
  }  
}  
  
@media screen and (max-width: 768px) {  
  .element {  
    width: 50%;  
  }  
}  
  
@media screen and (max-width: 1024px) {  
  .element {  
    width: 25%;  
  }  
}
```

# ¡Vamos a practicar!



# No te olvides de dar el presente

## **Recordá:**

- **Revisar la Cartelera de Novedades.**
- **Hacer tus consultas en el Foro.**

**Todo en el Aula Virtual.**

# Gracias