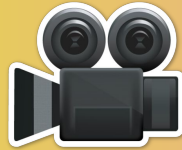


***RECUERDA PONER A GRABAR LA
CLASE***





Clase 12. DESARROLLO WEB

@MEDIA + PSEUDO CLASES



OBJETIVOS DE LA CLASE

- Repasar diseño responsive con Bootstrap.
- Conocer el uso de pseudo clases

GLOSARIO:

Clase 11

Librería: colección de elementos que suelen tener un propósito concreto.

Framework: conjunto de herramientas y código, para trabajar de acuerdo con una metodología, utilizando determinados patrones.

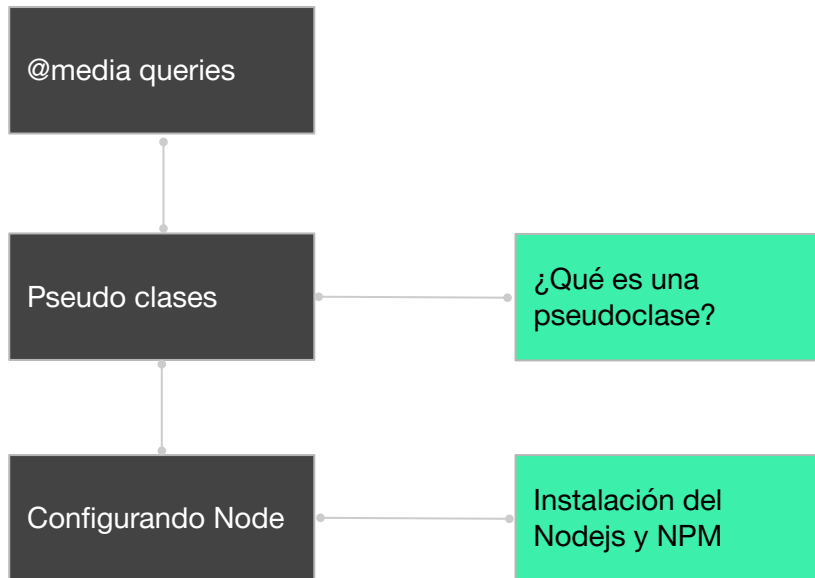
Bootstrap: Es un framework que permite crear interfaces web con CSS y JavaScript. Es responsive

Responsive design: Que adapta la interfaz del sitio web al tamaño del dispositivo en que se visualice.

MAPA DE CONCEPTOS

MAPA DE CONCEPTOS CLASE 12

¡Para
recordar!



CRONOGRAMA DEL CURSO

Clase 11



Frawework CSS + Bootstrap



PRÁCTICAS DE LO
VISTO EN CLASE



MAQUETAR CON
BOOTSTRAP



FULL BOOTSTRAP

Clase 12



@Media + Pseudo Clases



PRÁCTICAS DE LO
VISTO EN CLASE



SEGUNDA ENTREGA
DEL PROYECTO FINAL

Clase 13



SASS I



PRÁCTICAS DE LO
VISTO EN CLASE



SASS I



GUIÓN DE LA CLASE

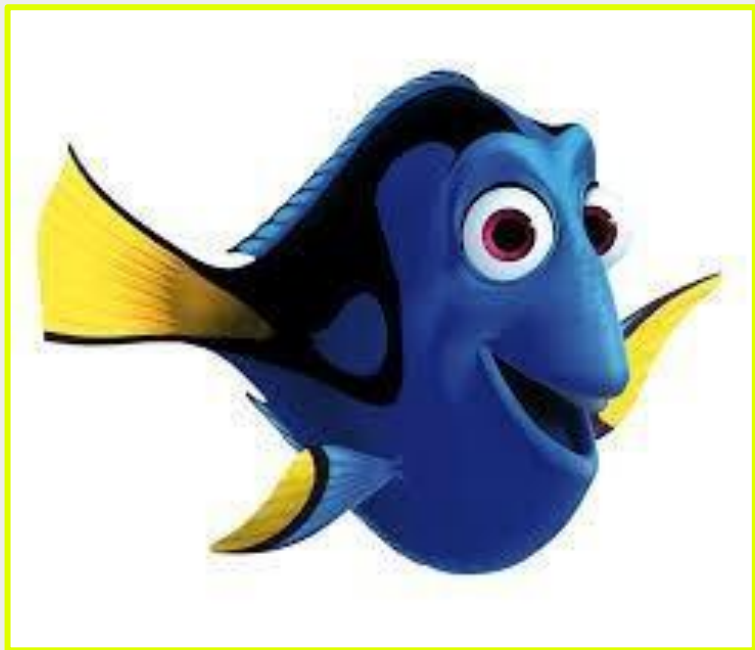
Accede al material complementario [aquí](#).



RESPONSIVE



VAMOS A REPASAR RESPONSIVE



Volvemos a tocar el tema, pues para aprovechar Bootstrap, es fundamental conocer y entender muy bien el responsive. Cualquier cambio que se realice altera el diseño en todos los dispositivos.

PÁGINA RESPONSIVE

¡Para
recordar!



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, ofreciendo al usuario la mejor experiencia posible.

RECORDEMOS

¡Para
recordar!



Responsive Web Design

Mobile First Web Design



CODER HOUSE

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 quisiera 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, podrías hacer:

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

EJEMPLO DE MEDIA QUERIES



Si quisiera variar la alineación del texto que se aplica en una clase, a partir de las pantallas tipo escritorio:

```
.miestilo {  
    text-align: center;  
}  
  
@media (max-width: 992px) {  
    .miestilo {  
        text-align: left;  
    }  
}
```

EJEMPLO DE MEDIA QUERIES



Puedes también modificar el cuerpo de texto, si lo utilizas en px para diferentes pantallas:

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

Recuerda que esto no es necesario si utilizas el font-size en “em”.

OPERADOR “ADD”



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

```
@media (max-width: 700px) and (min-width: 400px) {  
    .miestilo {  
        text-align: left;  
    }  
}
```

ORIENTACIÓN



En este caso, sólo se reproducirá el estilo si la ventana tiene un ancho de de 700px o más, y la pantalla está 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>
```

CSS

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

/* Topnav links */
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none; }

@media screen and (max-width: 600px) {
  .topnav a {
    float: none;
    width: 100%;
  }
}
```

EJEMPLO DE MEDIA QUERIES



Puedes modificar la distribución de las columnas para que en pantallas pequeñas se vean una abajo de la otra:

Desktop

Responsive Four Column Layout

Resize the browser window to see the responsive effect. On screens that are 992px wide or less, the columns will resize from four columns to two columns. On screens that are 600px wide or less, the columns will stack on top of each other instead of next to eachother.

Column 1 Some text..	Column 2 Some text..
Column 3 Some text..	Column 4 Some text..

Mobile

Responsive Four Column Layout

Resize the browser window to see the responsive effect. On screens that are 992px wide or less, the columns will resize from four columns to two columns. On screens that are 600px wide or less, the columns will stack on top of each other instead of next to eachother.

Column 1 Some text..
Column 2 Some text..
Column 3 Some text..

[Ver el código](#)

CODER HOUSE

EJEMPLO DE MEDIA QUERIES

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

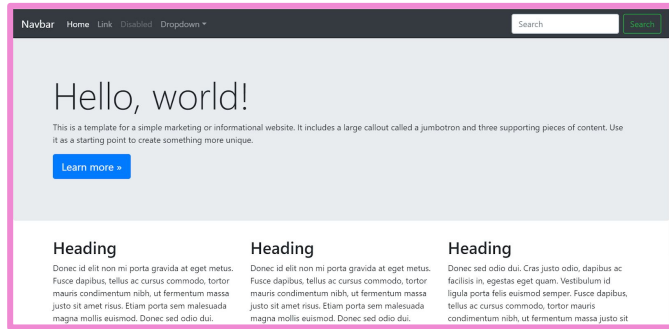
- Cambiar el tamaño y la posición de una imagen.
- Modificar la posición de cualquier elemento.
- Cambiar el tamaño de letra, la fuente o su color.
- Aplicar combinaciones de estilos avanzados.

¡Cualquier cosa que se te ocurra!

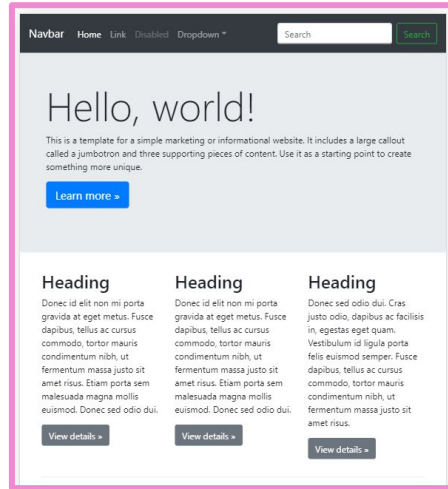


EJEMPLO

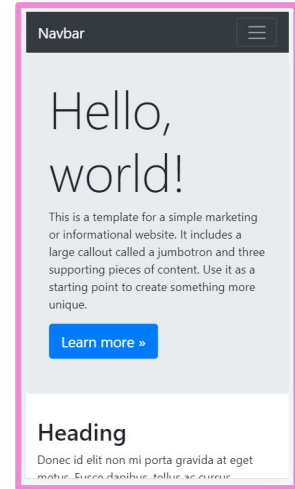
Desktop



Tablet



Mobile



CODER HOUSE

Ejemplo
en vivo



PSEUDO CLASES

¿Qué es una pseudo clase?

Una pseudoclase CSS es una palabra clave que se añade a los selectores y que especifica un estado especial del elemento seleccionado.

Por ejemplo, `:hover` aplicará un estilo cuando el usuario haga hover sobre el elemento especificado por el selector.

`selector:pseudoclase { propiedad: valor; }`



¿Qué es una pseudo clase?



Las pseudoclasas, junto con los pseudoelementos, permiten aplicar un **estilo** a un elemento no sólo en relación con el contenido del árbol de documento, sino **también en relación a factores externos** como:

- El historial del navegador (:visited, por ejemplo),
- El estado de su contenido (como :checked en algunos elementos de formulario),
- La posición del ratón (como :hover que permite saber si el ratón está encima de un elemento o no).

Teniendo como referencia...

```
<div id="abuelo">
  <p>Párrafo 1</p>
  <p>Párrafo 2</p>
  <div class="padre">
    <div class="hijo"><a title="Título hijo1">HIJO1</a></div>
    <div class="hijo"><a>HIJO2</a></div>
    <div class="hijo"><a>HIJO3</a></div>
  </div>
  <p>Párrafo 3</p>
  <p>Párrafo 4</p>
  <div class="padre">
    <div class="hijo"><a title="Título hijo4">HIJO4</a></div>
    <div class="hijo"><a>HIJO5</a></div>
    <div class="hijo"><a>HIJO6</a></div>
  </div>
  <p>Párrafo 5</p>
  <p>Párrafo 6</p>
</div>
```

Algunos ejemplos

```
a:hover{ font-weight:bold; background-color:blue; color:white;  
text-decoration:underline; }  
a:link{ text-decoration:none; color:red; }  
a:visited{ color:#999999; }  
a:active{ background-color:yellow; color:black; }
```

Ejemplos más complejos...

Otra pseudo clase muy útil pero poco utilizada, aplica a todos los elementos A, excepto a los incluidos como elementos B. Es decir, podríamos aplicar unos estilos a todos los elementos “div” y evitar que éstos se apliquen a otras capas con un id o class determinado. En el siguiente ejemplo hacemos justo esto, evitando que se apliquen los estilos a las capas “padre”.

Algunos ejemplos más complejos...

```
div:not(.padre){  
  padding-left:40px;  
  padding-right:40px;  
}
```

Otros ejemplos...

La solución a todos los problemas de maquetación de antaño con las filas y los bloques.

👉 Con `:nth-child(N)` podremos aplicar sus estilos a todos los elementos hijos cuya posición sea un número “N” respecto a un padre. Este número N no tiene que ser necesariamente un número entero para especificar una posición fija (la posición 2, por ejemplo), ya que también permite insertar fórmulas y palabras específicas.

Otros ejemplos...

```
.padre div:nth-child(2){ font-weight:bold; color:orange; }
```

¡ATENCIÓN!

Recuerda instalar Node.js para la próxima clase.



[Ver video-tutorial](#)

CONFIGURANDO NODE

Ejemplo
en vivo



INSTALACIÓN DEL NODEJS Y EL NPM

REPASANDO LA INSTALACIÓN DEL PROCESADOR

1

Instala **nodejs**.

2

Instala **npm**.

3

Ingresa al
directorio del
repositorio.

4

Inicia el npm,
con **npm
init**.

REPASANDO LA INSTALACIÓN DEL PROCESADOR

5

Instala el nodemon: **npm install -D node-sass nodemon.**

6

Crea la carpeta SCSS y CSS y sus archivos respectivos.

7

Edita el package.json e inserta los líneas.

- "build-css":
- "node-sass --include-path scss scss/prueba.scss css/pruebacss.css",
 - "watch-css": "nodemon -e scss -x \"npm run build-css\""

8

Compila con:
npm run watch-css.



SEGUNDA ENTREGA DEL PROYECTO FINAL

Hacer entrega del proyecto **(dos páginas a elección)** adaptado a la vista mobile y la vista desktop agregando animaciones. Para el mismo podrán optar entre el uso correcto del framework o sin el uso del mismo. La entrega se deberá cargar a un repositorio de github publico.

SEGUNDA ENTREGA DEL PROYECTO FINAL

Formato: link de github. Incluir apellido en el nombre del archivo (ej. "Nombre Proyecto - Apellido").

Sugerencia: activar comentarios en el archivo.

Proyecto
Final



2

Estructura avanzada de la web

>>Objetivos generales:

1. Realizar una estructura del HTML prolija, limpia, fácil de leer y que no tenga errores en sus atributos o en sus valores.

>>Objetivos específicos:

1. Maquetar dos páginas de tu sitio web. Para esto se puede usar bootstrap o grids + flexbox.
2. Utilizar todas las propiedades CSS necesarias vistas hasta el momento.

>>Se debe entregar:

- Maquetado de la web: las estructuras maquetan a la web en base al framework elegido, haciendo usos de clases utilitarias para armar grillas, elementos web y estilos propios del framework, además del HTML de contenido.
- Páginas: todas las páginas tienen el contenido estructurado y el estilo linkado. También tiene que tener agregadas las diferentes librerías de Javascript y CSS pertinentes al framework.



Estilo avanzado de la web

Formato: archivo CSS

>>Objetivos generales:

1. Crear archivos de CSS para darle estilo a la web.

>>Objetivos específicos:

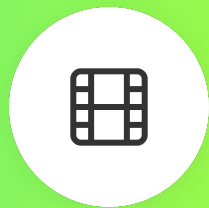
1. Agregar todas las propiedades CSS necesarias (transformaciones, animaciones y/o transiciones para otorgarle dinamismo a la web en elementos que tengan interacción con el usuario.)
2. Hacer uso de selectores de CSS para poder darle estilo propio a los elementos que ya vienen con su propia identidad del framework.

>>Se debe entregar:

- Estilo avanzado: se le mejorarán los elementos interactivos con variaciones en sus diferentes estados, ya sea de la mano de transformaciones, transiciones y/o animaciones.
- Estilo del Framework: no todos los elementos del framework van a tener una estética que condice con el sitio en el que son implementados, por lo que se usará CSS para darles un estilo acorde.
- Estructura de la web: usa etiquetas no sólo para armar contenido, sino para armar los elementos que van a conformar el layout de la web, los contenedores, etc.

¿PREGUNTAS?





***¿QUIERES SABER MÁS? TE DEJAMOS
MATERIAL AMPLIADO DE LA CLASE***

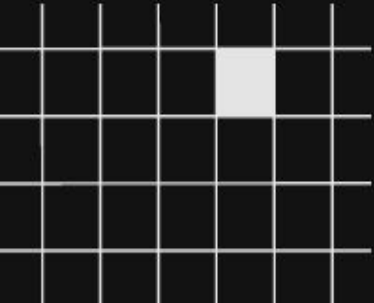


- [Themes y templates](#) | **TE LO EXPLICO CON GATITOS**
- [Íconos Material Design](#) | **Material Design Icons**



¡MUCHAS GRACIAS!

Resumen de lo visto en clase hoy:

- Aplicar nuevas herramientas de Bootstrap.
 - Aplicar un theme de Bootstrap.
 - Repasar Responsive.
- 



OPINA Y VALORA ESTA CLASE

#DEMOCRATIZANDO LA EDUCACIÓN

CODER HOUSE