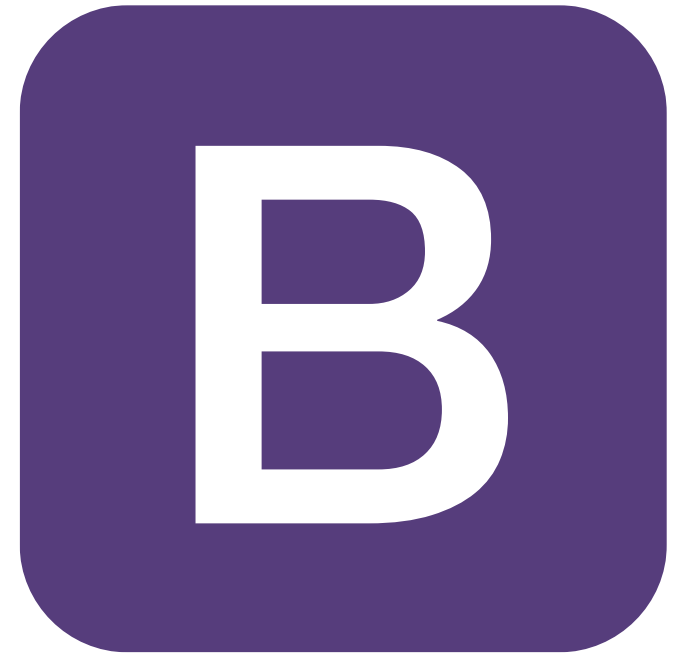


BOOTSTRAP 5



INTRODUCCIÓN A BOOTSTRAP

INTRODUCCIÓN

¿Qué es el diseño web responsivo o adaptativo (RWD)?

- filosofía de diseño y desarrollo cuyo objetivo es adaptar la apariencia de las [páginas web](#) al dispositivo que se esté utilizando para visitarlas.
- Es cierto que hay nuevos actores → aproximación de usuario a la pantalla

INTRODUCCIÓN

CONTENT IS LIKE WATER



“ You put water into a cup it becomes the cup.
You put water into a bottle it becomes the bottle.
You put it in a teapot, it becomes the teapot. ”

Josh Clark (originally Bruce Lee) - Seven deadly mobile myths
Illustration by Stéphanie Walter

INTRODUCCIÓN

- CONCEPTOS BÁSICOS

ViewPort

El área visible de una página web. Cambia según el dispositivo.

Breakpoint

Punto en el que cambian propiedades de nuestra página web. Normalmente atendiendo a su anchura...

Media Query

Estilos CSS que se aplican de forma condicional atendiendo a ciertas condiciones. Normalmente relativas al tamaño/resolución del dispositivo

INTRODUCCIÓN

- **EL VIEWPORT: AREA VISIBLE DE NUESTRA WEB**

- `<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">`
-

INTRODUCCIÓN

- **LOS BREAKPOINT:** PUNTOS DE ANCHURA DE PANTALLA EN LOS QUE CAMBIA LA DISPOSICIÓN DE LOS ELEMENTOS



INTRODUCCIÓN

- BREAKPOINTS:
- EJEMPLO
- https://www.w3schools.com/css/tryit.asp?filename=tryresponsive_mediaquery_breakpoints

INTRODUCCIÓN

MEDIA QUERIES: REGLAS CSS QUE NOS PERMITEN CAMBIAR EL ESTILO DE NUESTRA PÁGINA ATENDIENDO DIVERSOS PARÁMETROS, FUNDAMENTALMENTE LA ANCHURA Y TIPO DE DISPOSITIVO.

```
@media only screen and (min-width: 576px) {  
  .example {background: green;}  
}  
  
@media only screen and (min-width: 768px) {  
  .example {background: blue;}  
}
```


INTRODUCCIÓN

- NO ES UNA TAREA FÁCIL, PERO BOOTSTRAP NOS VA A ECHAR UNA MANO.



INTRODUCCIÓN

- Bootstrap es un [framework](#) CSS desarrollado por Twitter en 2010, para estandarizar las herramientas de la compañía.
- Inicialmente, se llamó Twitter Blueprint y, un poco más tarde, en 2011, se transformó en código abierto y su nombre cambió para Bootstrap. Desde entonces fue actualizado varias veces y ya se encuentra en la versión 5.3.
- El framework combina [CSS](#) y JavaScript para estilizar los elementos de una página HTML. Permite mucho más que, simplemente, cambiar el color de los botones y los enlaces.

INTRODUCCIÓN

- Esta es una herramienta que proporciona interactividad en la página, por lo que ofrece una serie de componentes que facilitan la comunicación con el usuario, como menús de navegación, controles de página, barras de progreso y más.
- Además de todas las características que ofrece el framework, **su principal objetivo es permitir la construcción de sitios web responsive para dispositivos móviles**

INTRODUCCIÓN

- Hay un archivo principal llamado **bootstrap.css**, que contiene una definición para todos los estilos utilizados. Básicamente, la **estructura del framework se compone de dos directorios**:
 - **css**: contiene los archivos necesarios para la estilización de los elementos y una alternativa al tema original;
 - **js**: contiene la parte posterior del archivo **bootstrap.js** (original y minificado), responsable de la ejecución de aplicaciones de estilo que requieren manipulación interactiva.
- Para asignarle una característica a un elemento, simplemente le asignamos la clase correspondiente en la propiedad “class”.

INTRODUCCIÓN

HTML



CSS3



SAAS



.

INTRODUCCIÓN

- **INSTALACIÓN:** Para instalar BootStrap podemos optar por dos opciones:
- **La primera:** usar los enlaces que nos proporcionan en la documentación via CDN (Network delivery content):
 - El primero de ellos es el CSS de la librería
 - Y posteriormente los enlaces de los scripts si vamos a usar componentes, muchos de los cuáles no son elementos estáticos y van a requerir scripts para definir su comportamiento.
- **La segunda:** descargar la hoja de estilos y los scripts y referenciarlos en nuestra web.

INTRODUCCIÓN

- Desde esta página puedes descargar los enlaces:
- <https://getbootstrap.com/docs/5.0/getting-started/introduction/>
- Tienes una plantilla inicial con la que trabajar de forma inmediata.

MAQUETACIÓN RESPONSIVE Y LAYOUT:

1. CONTENEDORES

- **¿QUÉ ES UN CONTENEDOR?:** ES LA ETIQUETA QUE CONTIENE EL RESTO DE NUESTRA PÁGINA WEB. NORMALMENTE ES UN DIV, QUE POR DEFECTO ES LA ETIQUETA QUE SIRVE PARA MAQUETAR EN HTML. BOOTSTRAP NOS PROPORCIONA DOS TIPOS DE CONTENEDORES:
 - CONTAINER
 - CONTAINER-FLUID

MAQUETACIÓN RESPONSIVE Y LAYOUT:

1. CONTENEDORES

- **container** : Ocupará diferentes anchos dependiendo del tamaño de la pantalla. Puede ocupar toda la pantalla o dejar unos márgenes a izquierda a derecha aunque, en este caso, siempre estará centrado.

```
<body> <div class="container"> </div> </body>
```

- **container-fluid**: Ocupará todo el ancho (100%) de lo que podemos ver en el navegador.

```
<body> <div class="container-fluid"> </div> </body>
```

MAQUETACIÓN RESPONSIVE Y LAYOUT:

1. CONTENEDORES

- **IMPORTANTE:** Aunque los contenedores se pueden anidar unos dentro de otros son pocas las páginas que son tan complejas para necesitar contenedores anidados.

MAQUETACIÓN RESPONSIVE Y LAYOUT:

2. EL GRID DE BOOTSTRAP

- Como ya hemos hablado en el capítulo anterior es muy difícil que nos encontremos con páginas cuyos layouts sean tan complejos como para necesitar más de un contenedor. Así que, si nos ponemos de acuerdo en eso, en que sólo vamos a necesitar un contenedor, podemos fijar una estructura general que va a permanecer fija para todos nuestros layouts:
- En ese contenedor vamos a tener filas para ir creando nuestro proyecto (row).

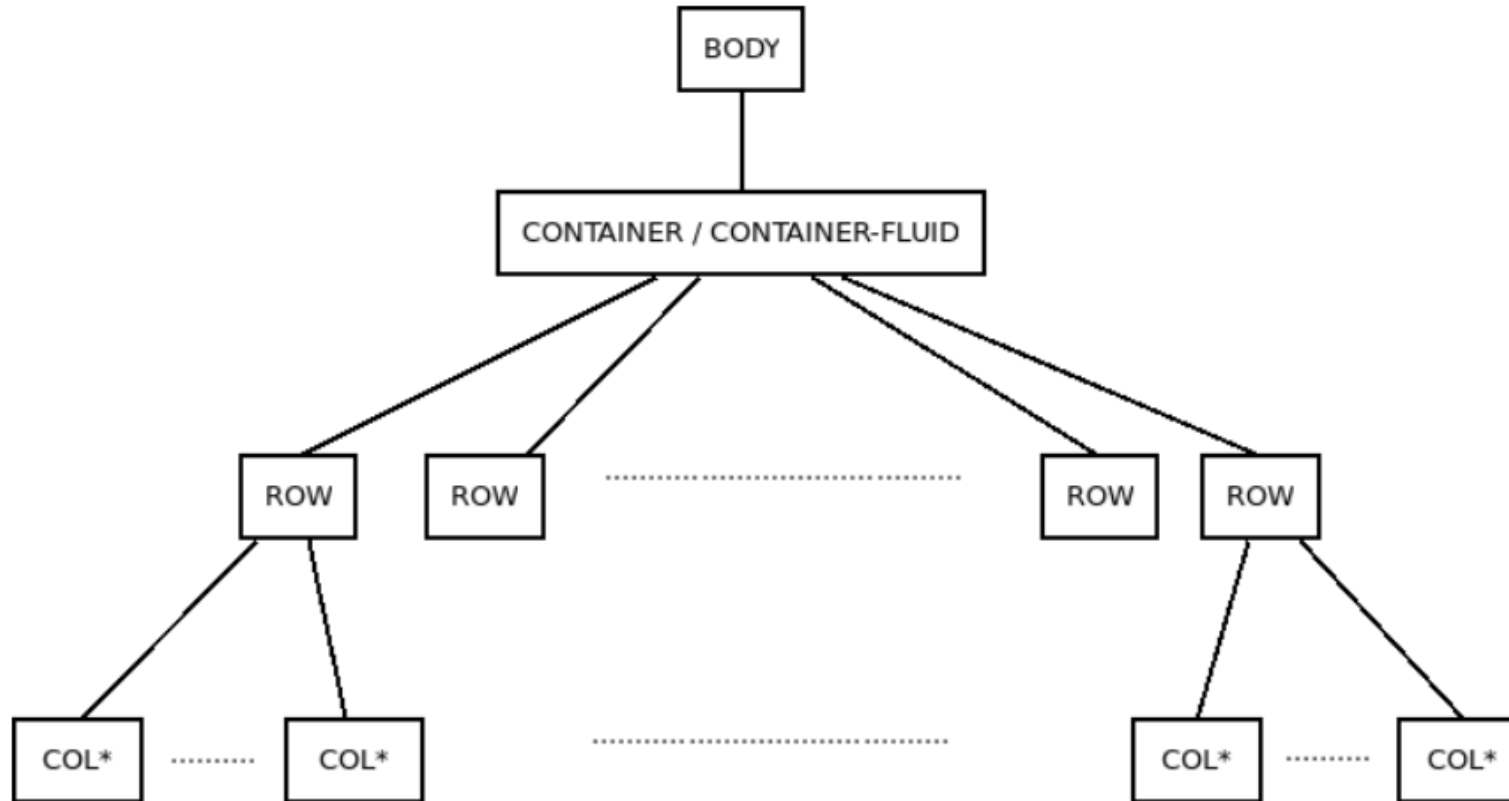
MAQUETACIÓN RESPONSIVE Y LAYOUT:

2. EL GRID DE BOOTSTRAP

- Y dentro de cada fila de cada row, en principio podemos tener hasta 12 columnas, aunque ya veremos posteriormente que dado a que Bootstrap utiliza contenedores flex podemos usar las propiedades de estos contenedores flex para poner más columnas aunque, en realidad, es difícil que tengamos que utilizar más.
- Todas estas columnas están pegadas unas a otras y tiene siempre un padding a izquierda y a derecha de 15px del que hablaremos posteriormente.
- Otra cosa importante es que no tenemos que olvidarnos de utilizar `class="row"` ya que si nuestra estructura de columnas no está dentro de un contenedor con esa clase perderemos todas las propiedades de los contenedores flex, que son los sustentan la maquetación en Bootstrap.

MAQUETACIÓN RESPONSIVE Y LAYOUT:

2. EL GRID DE BOOTSTRAP



MAQUETACIÓN RESPONSIVE Y LAYOUT:

3. BREAKPOINTS

¿Qué es un breakpoint?

- En el mundo del diseño responsive un **breakpoint** es un tamaño de pantalla (en pixels) donde se produce un cambio en la disposición de los elementos que conforman nuestra página web.
- Existen multitud de tipos de dispositivos, multitud de pantallas, pantallas que giran etc. Por lo tanto la elección de los distintos breakpoints no es algo trivial

MAQUETACIÓN RESPONSIVE Y LAYOUT:

3. BREAKPOINTS

- Sin embargo, tenemos la suerte de que los muchachos de [Twitter](#), usando su experiencia y los datos que sus millones de usuarios les proporcionan, has decidido que la elección más adecuada son 4 breakpoints:
 - En 576px
 - En 768px
 - En 992px
 - En 1200px

MAQUETACIÓN RESPONSIVE Y LAYOUT:

3. BREAKPOINTS

- Teniendo en cuenta estos 4 breakpoints podemos diferenciar entre 5 tipos de pantallas
 - **Pantallas extra pequeñas:** Entre 0 y 576px
 - **Pantallas pequeñas:** Entre 576px y 768px
 - **Pantallas medianas:** Entre 768px y 992px
 - **Pantallas grandes:** Entre 993px y 1200px
 - **Pantallas extra-grandes:** A partir de 1200px

MAQUETACIÓN RESPONSIVE Y LAYOUT:

3. BREAKPOINTS

	Extra pequeñas	Pequeñas	Medianas	Grandes	Extra Grandes
Anchura máxima	576px	768px	992px	1200px	> 1200px
Prefijo de la clase	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
Número de columnas	12				
Anchura del Gutter	30px (1px a cada lado de cada columna)				
Anidar	Sí				
Reordenar columnas	Sí				

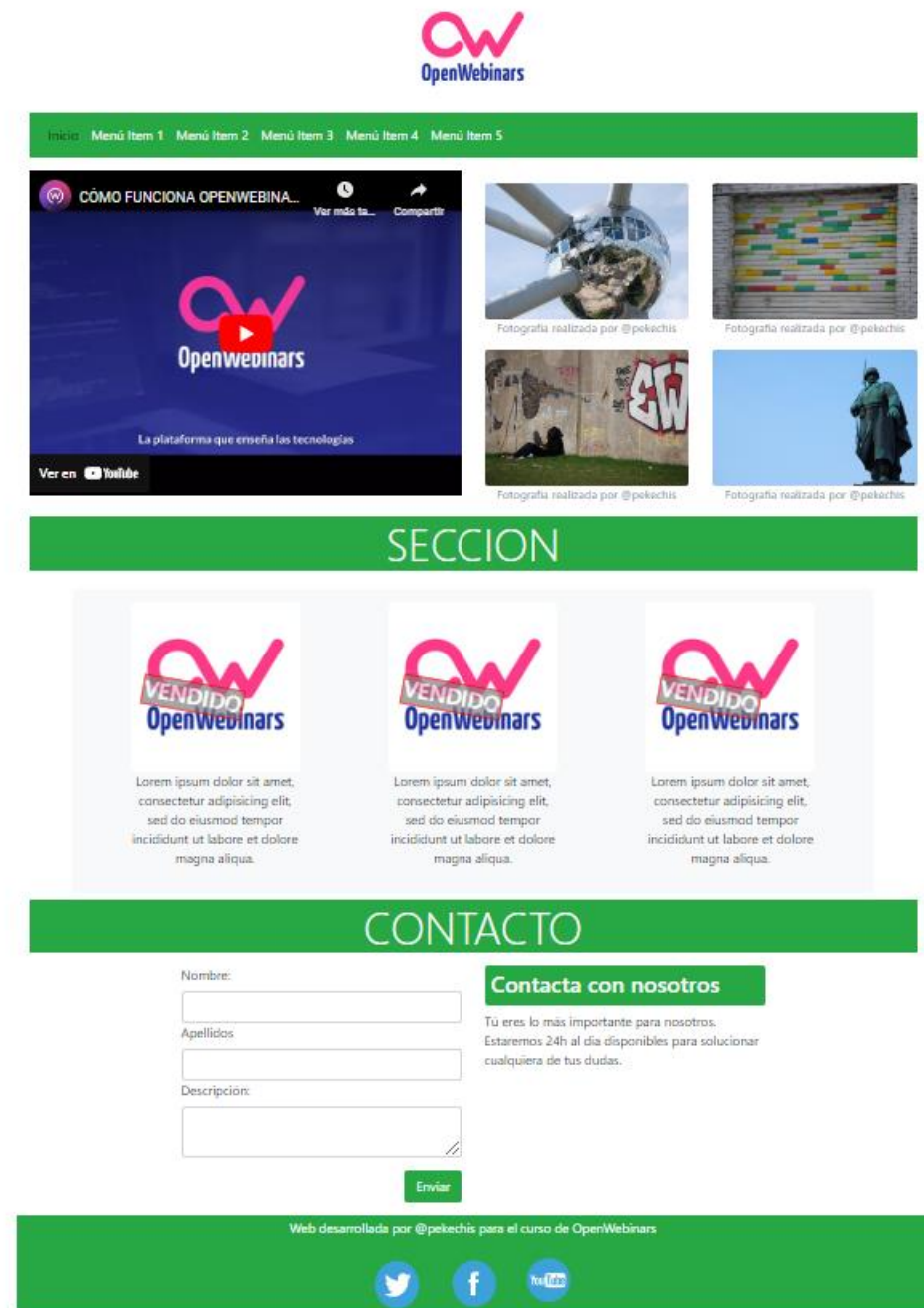
MAQUETACIÓN RESPONSIVE Y LAYOUT:

3. BREAKPOINTS

- De **especial importancia** son los prefijos CSS de las clases **.col-*** que son los que nos van a permitir establecer la anchura de los diferentes elementos que compongan nuestro layout. En definitiva son esas clases las que me proporciona Bootstrap 4 para poder maquetar.
- En el próximo apartado veremos cómo hacerlo

4. Presentación del proyecto:

<https://openwebinars.net/academia/aprende/bootstrap-4-layout/3166/>



MAQUETACIÓN RESPONSIVE Y LAYOUT:

5. COLUMNAS: DEFINIR TAMAÑO

- En apartados anteriores vimos cuál era la estructura general de una página con Bootstrap 5, vimos también cuáles eran los breakpoint establecidos y que, en principio, cada *row* puede contener hasta 12 columnas.
- En este apartado vamos a ver como nosotros podemos establecer el tamaño de cada uno de los componentes de dicha *row* o filas y que además, podemos establecer distintos tamaños dependiendo del tamaño de nuestra pantallas.

MAQUETACIÓN RESPONSIVE Y LAYOUT:

5. COLUMNAS: DEFINIR TAMAÑO

- **col-X**: hará que ese contendor X/12 columnas en pantallas extra pequeñas.

<!-- col-6 ocupará justo la mitad en pantallas extra pequeñas -->

<div class="col-6"> </div>

- **col-sm-X**: hará que ese contendor X/12 columnas en pantallas pequeñas.

<!-- col-sm-4 ocupará un tercio de la pantalla en pantallas pequeñas --> <div

class="col-sm-4"> </div>

- Y así sucesivamente para *col-md-X* para pantallas medianas, *col-lg-X* para pantalla grandes y *col-xl-X* para pantallas extragrandes.

MAQUETACIÓN RESPONSIVE Y LAYOUT:

5. COLUMNAS: DEFINIR TAMAÑO

- Una cosa importante que hay que tener en cuenta es que si establecemos un tamaño para un cierto tamaño de pantallas ese tamaño se va a mantener para pantallas de mayor tamaño pero para pantallas menores ese div ocupará toda la pantalla, es decir si tenemos:
- `<div class="col-md-6"> </div>`
- Este div ocupará la mitad de la pantalla para pantallas de 768px (recordar los breakpoints) o superiores pero para tamaños menores ocupará todo el ancho de la fila que lo contiene

MAQUETACIÓN RESPONSIVE Y LAYOUT:

5. COLUMNAS: DEFINIR TAMAÑO

- Podemos también fijar a la vez anchos diferentes para distintos tipos de tamaños de pantallas. Para ello añadiremos varias clases a un mismo elemento, por ejemplo:

```
<div class="col-sm-6 col-md-4 col-xl-3"> </div>
```

- En este caso ese div:
 - Ocupará toda la fila para pantallas extrapequeñas (<576px)
 - Ocupará la mitad de la fila para pantallas en 576px y 768px
 - Ocupará un tercio de la fila para pantallas entre 768px y 992px
 - Ocupará un cuarto para pantallas mayores de 992px

MAQUETACIÓN RESPONSIVE Y LAYOUT:

5. COLUMNAS: DEFINIR TAMAÑO

- También es importante recordar que si, en un *row* nos pasamos de 12 columnas ese elemento que provoca que nos pasemos de las 12 columnas, “*pasará*” debajo de los elementos de la fila actual ocupando el tamaño establecido. Ejemplo, si tenemos la siguiente estructura:

```
<div class="row">  
  <div class="col-md-6"> </div>  
  <div class="col-md-4"> </div>  
  <div class="col-md-4"> </div>  
</div>
```

El tercer div irá abajo ya que $6+4+4 = 14$ que es mayor que 12.

MAQUETACIÓN RESPONSIVE Y LAYOUT:

5. COLUMNAS: DEFINIR TAMAÑO

[MODIFICAR ANCHO Y ALTO DE UN ELEMENTO.](https://getbootstrap.com/docs/5.0/utilities/sizing/)

<https://getbootstrap.com/docs/5.0/utilities/sizing/>

MAQUETACIÓN RESPONSIVE Y LAYOUT:

6. COLUMNAS: TAMAÑO AUTO

- Es una de las características que deriva directamente del uso de contenedor flex.
- Usando elementos en los que no especifiquemos el número de columnas, el espacio que haya en la fila se va a distribuir de manera uniforme.
 - **col** : Para todo tipo de pantallas
 - **col-sm** : De pantallas pequeñas en adelante ($\geq 576\text{px}$)
 - **col-md** : De pantallas medianas en adelante ($\geq 768\text{px}$)
 - **col-lg** : De pantallas grande en adelante ($\geq 992\text{px}$)
 - **col-xl** : Para pantallas de 1200px en adelante

MAQUETACIÓN RESPONSIVE Y LAYOUT:

6. COLUMNAS: TAMAÑO AUTO

- Veamos el primer ejemplo:

```
<div class="row">  
  <div class="col red"> </div>  
  <div class="col black"> </div>  
</div>
```

En este caso en cualquier tipo de pantallas, esos dos elementos se repartirán a igual manera el ancho del contenedor principal.

MAQUETACIÓN RESPONSIVE Y LAYOUT:

6. COLUMNAS: TAMAÑO AUTO

- Debemos de tener cuidado porque en caso de que lo que contengan esas partes ocupe más de la mitad no se respetará esa proporción. Veamos este caso con un ejemplo:

```
<div class="row">  
  <div class="col red"> <h1>PRIMERA PARTE</h1> </div>  
  <div class="col black"> <h1>SEGUNDA PARTE</h1> </div>  
</div>
```

MAQUETACIÓN RESPONSIVE Y LAYOUT:

6. COLUMNAS: TAMAÑO AUTO

- Si queremos que el reparto equitativo suceda para determinado de pantallas:

```
<div class="row">  
  <div class="col-md yellow"></div>  
  <div class="col-md pink"></div>  
  <div class="col-md yellow"></div>  
  <div class="col-md pink"></div>  
  <div class="col-md yellow"></div>  
</div>
```

- En este caso todo se divide en 5 partes para pantallas mayores de 768px pero en cuanto la pantalla es más pequeña que eso, todos los elementos pasarán a ocupar todo el centro de la pantalla.

- Este funcionamiento abre la puerta a *rows* (filas) que tengan más de 12 columnas, por ejemplo una fila con 14 componentes:

```
<div class="row">  
  <div class="col-md blue"></div>  
  <div class="col-md pink"></div>  
  <div class="col-md blue"></div>  
  <div class="col-md pink"></div>  
  <div class="col-md blue"></div>  
  <div class="col-md pink"></div>  
  <div class="col-md blue"></div>  
  <div class="col-md pink"></div>  
  <div class="col-md blue"></div>  
  <div class="col-md pink"></div>  
  <div class="col-md blue"></div>  
  <div class="col-md pink"></div>  
  <div class="col-md blue"></div>  
  <div class="col-md pink"></div>  
</div>
```

MAQUETACIÓN RESPONSIVE Y LAYOUT:

6. COLUMNAS: TAMAÑO AUTO

- Bootstrap 5 nos permite también ajustar el tamaño de las columnas al contenido de las mismas usando `col-{breakpoint}-auto`. Así por ejemplo:

- `<div class="row">`
- `<div class="col-lg-auto yellow">`
- Anchura ajustada
- `</div>`
- `<div class="col-lg-auto green">`
- al contenido de las celdas
- `</div>`
- `</div>`

MAQUETACIÓN RESPONSIVE Y LAYOUT:

6. COLUMNAS: TAMAÑO AUTO

- **IMPORTANTE:** Debemos tener mucho cuidado cuando mezclamos tamaños automáticos junto tamaños, definidos por el usuario porque cuanto no quepan las columnas (y puede que no sumen 12) pasarán a la siguiente fila y se repartirán el espacio de la siguiente fila.

MAQUETACIÓN RESPONSIVE Y LAYOUT:

7. SALTOS DE LÍNEA

- Si mientras estamos construyendo nuestro grid y queremos que haya un salto de línea podemos forzarlo de la siguiente manera:

```
<div class="w-100"></div>
```

- Con esa estructura aunque no hayamos ocupado las 12 columnas de nuestro grid podremos pasar a la siguiente fila.

MAQUETACIÓN RESPONSIVE Y LAYOUT:

7. SALTOS DE LÍNEA

- Ejemplo:

```
<div class="row">  
  <!-- Ocupamos 5/12 de la fila -->  
  <div class="col-md-3"></div>  
  <div class="col-md-2"></div>  
  <!-- Forzamos el salto de línea -->  
  <div class="w-100"></div>  
  <!-- Ocupamos otra vez 5/12 pero en una nueva línea -->  
  <div class="col-md-3"></div>  
  <div class="col-md-2"></div>  
</div>
```

MAQUETACIÓN RESPONSIVE Y LAYOUT:

7. SALTOS DE LÍNEA

- Esto mismo se podría haber conseguido utilizando varias *rows* (filas).
- Podemos provocar también estos cambios en función del tamaño de la pantalla

MAQUETACIÓN RESPONSIVE Y LAYOUT:

8. EL GUTTER

- Por defecto Bootstrap introduce separación entre el contenido de las columnas a derecha e izquierda. Es lo que se denomina *gutter*
- Si éste no es el efecto que queremos en nuestra maquetación podemos evitarlo añadiendo al elemento que tenga la clase `g-0`.
- Así de esta manera las columnas o elementos de la fila no tendrán ese padding con la posibilidad de que el contenido de la misma queden pegados.
- Ver ejemplos.

MAQUETACIÓN RESPONSIVE Y LAYOUT:

8. EL GUTTER

```
<div class="row g-5">  
  <div class="col-md-6"><h3>Elemento con gutter</h3></div>  
  <div class="col-md-6"><h3>Elemento con gutter</h3></div>  
</div>  
  
<div class="row g-0">  
  <div class="col-md-6"><h3>Elemento sin gutter</h3></div>  
  <div class="col-md-6"><h3>Elemento sin gutter</h3></div>  
</div>
```

MAQUETACIÓN RESPONSIVE Y LAYOUT:

9. DESPLAZAMIENTO DE COLUMNAS

- Tal y como hemos visto en apartados anteriores podemos dar diferentes tamaños a los distintos componentes o columnas de una *row* o fila.
- Adicionalmente podemos establecer no sólo el tamaño de la columna si no también la posición a partir de la cuál se va a posicionar.
- Para conseguir eso debemos añadir la **clase *offset-X*** donde X es el número de columnas a la izquierda que desplazaremos el elemento que queremos posicionar.

MAQUETACIÓN RESPONSIVE Y LAYOUT:

9. DESPLAZAMIENTO DE COLUMNAS

Este desplazamiento puede ser también establecido de manera diferente para cada tamaño de pantalla:

- **offset-sm-X** : Para desplazamientos para pantallas entre 576px y 768px.
- **offset-md-X** : Para desplazamientos para pantallas entre 768px y 992px.
- **offset-lg-X** : Para desplazamientos para pantallas entre 992 y 1200px.
- **offset-xl-X** : Para desplazamientos para pantallas mayores de 1200px.

MAQUETACIÓN RESPONSIVE Y LAYOUT:

9. DESPLAZAMIENTO DE COLUMNAS

- Ejemplo:

```
<div class="container-fluid">  
  <div class="row">  
    <div class="col-sm-4 offset-md-1 red"></div>  
    <div class="col-sm-4 offset-sm-4 offset-md-2 blue"></div>  
  </div>  
</div>
```


MAQUETACIÓN RESPONSIVE Y LAYOUT:

10. ALINEAMIENTO DE LAS FILAS

Como ya hemos descrito en apartados anteriores, Bootstrap 5 utiliza contenedores flex. Precisamente utilizando las propiedades de estos elementos podremos fijar la alineación de los componentes de las *rows* o filas.

Podremos establecer:

- La alineación vertical de los elementos de la fila.
- La alineación horizontal de los elementos de la fila.

Tanto para establecer la alineación horizontal como la vertical deberemos añadir una nueva clase al contenedor *row* o fila.

MAQUETACIÓN RESPONSIVE Y LAYOUT:

10. ALINEAMIENTO DE LAS FILAS

Alineación vertical

Si queremos alinear los distintos elementos que componen una fila de manera vertical podremos hacerlo añadiendo a la *row* fila una de estas tres clases:

- **align-items-start** : Los elementos de la fila se alinearán verticalmente con el borde superior de la fila.
- **align-items-center** : Los elementos de la fila se alinearán centrados verticalmente.
- **align-items-end** : Los elementos de la fila se alinearán verticalmente con el borde inferior de la fila.

MAQUETACIÓN RESPONSIVE Y LAYOUT:

10. ALINEAMIENTO DE LAS FILAS

Alineación horizontal

Para fijar la alineación horizontal de los elementos en la *row* (fila) dispondremos de las siguientes clases:

- **justify-content-start** : Los elementos empezarán en la parte izquierda y ocuparán el tamaño horizontal establecido. Es la opción por defecto.
- **justify-content-center** : Los elementos de la fila se centrarán horizontalmente.
- **justify-content-end** : Los elementos de la fila se alinearán al final (la derecha normalmente) de la fila.
- **justify-content-around** : El espacio restante se distribuirá de manera equitativa alrededor de los elementos empezando y acabando con espacios libres.
- **justify-content-between** : El espacio restante se distribuirá de manera equitativa entre los distintos elementos estando uno de ellos totalmente a la derecha y otro totalmente a la izquierda.

MAQUETACIÓN RESPONSIVE Y LAYOUT:

10. ALINEAMIENTO DE LAS FILAS

- Con todas estas posibilidades y lo visto anteriormente es más que suficiente para casi cualquier Layout pero existen más posibilidades que podremos descubrir en la documentación oficial.

[Documentación oficial](https://getbootstrap.com/docs/5.3/getting-started/introduction/)

<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

MAQUETACIÓN RESPONSIVE Y LAYOUT:

11. ORDENACIÓN DE COLUMNAS

- Podemos cambiar el orden natural de las columnas gracias a la clase `order-x`

```
<div class="row">
```

```
  <div class="col-md-4 order-3 green">
```

Primer elemento. Posición Tercera

```
</div>
```

```
  <div class="col-md-4 order-1 yellow">
```

Segundo elemento. Posición Primera

```
</div>
```

```
  <div class="col-md-4 order-2">
```

Primer elemento. Posición Segunda

```
</div>
```

```
</div>
```

MAQUETACIÓN RESPONSIVE Y LAYOUT:

12. MARGENES ENTRE COLUMNAS

Márgenes entre columnas

- Para establecer los márgenes entre las columnas usaremos la clas `m{lado}-{tamaño}` o si queremos distinguir según los distintos tamaños de pantalla:
- `m{lado}-sm-{tamaño}` Para pantallas entre 576px y 768px.
- `m{lado}-md-{tamaño}` Para pantallas entre 768px y 992px.
- `m{lado}-lg-{tamaño}` Para pantallas entre 992px y 1200px.
- `m{lado}-xl-{tamaño}` Para pantallas de más de 1200px.

MAQUETACIÓN RESPONSIVE Y LAYOUT:

12. MARGENES ENTRE COLUMNAS

- Pudiendo ser lado:
 - **t** para margen superior (top).
 - **b** para margen inferior (bottom).
 - **s** para margen izquierdo (start). Versiones anteriores **l**
 - **e** para margen derecho (en). Versiones anteriores **r**
 - **x** para los márgenes superior e inferior.
 - **y** para los márgenes izquierdo y derecho.
 - En blanco si es para todos los lados.

MAQUETACIÓN RESPONSIVE Y LAYOUT:

12. MARGENES ENTRE COLUMNAS

Y pudiendo ser tamaño de tamaño:

- **0** : No hay margen
- **1** : 0.25rem
- **2** : 0.25rem
- **3** : 1rem
- **4** : 1.25rem
- **5** : 3rem
- **auto** : Para clases que establecen una margen auto

MAQUETACIÓN RESPONSIVE Y LAYOUT:

12. MARGENES ENTRE COLUMNAS

Centrado horizontal

- Relacionado con el tema de los márgenes hay que destacar que Bootstrap introduce una nueva clase ***mx-auto*** que permite centrar horizontalmente un elemento dentro de su contenedor

MAQUETACIÓN RESPONSIVE Y LAYOUT:

12. MARGENES ENTRE COLUMNAS

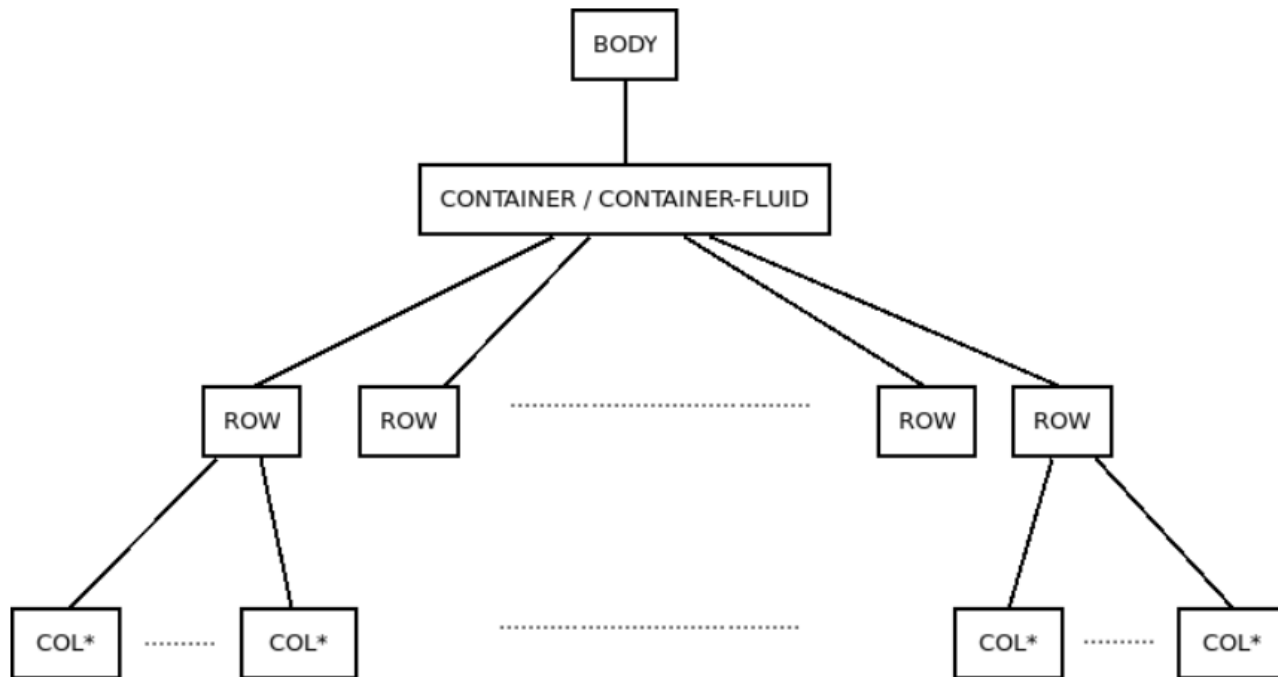
Paddings

- La notación de las clases para establecer los paddings en las columnas es muy similar a la notación de las clases para establecer márgenes entre las columnas. Simplemente tenemos que cambiar la *m* por la *p*

MAQUETACIÓN RESPONSIVE Y LAYOUT:

13. ANIDANDO FILAS

- Hasta ahora estamos maquetando en base a la estructura:



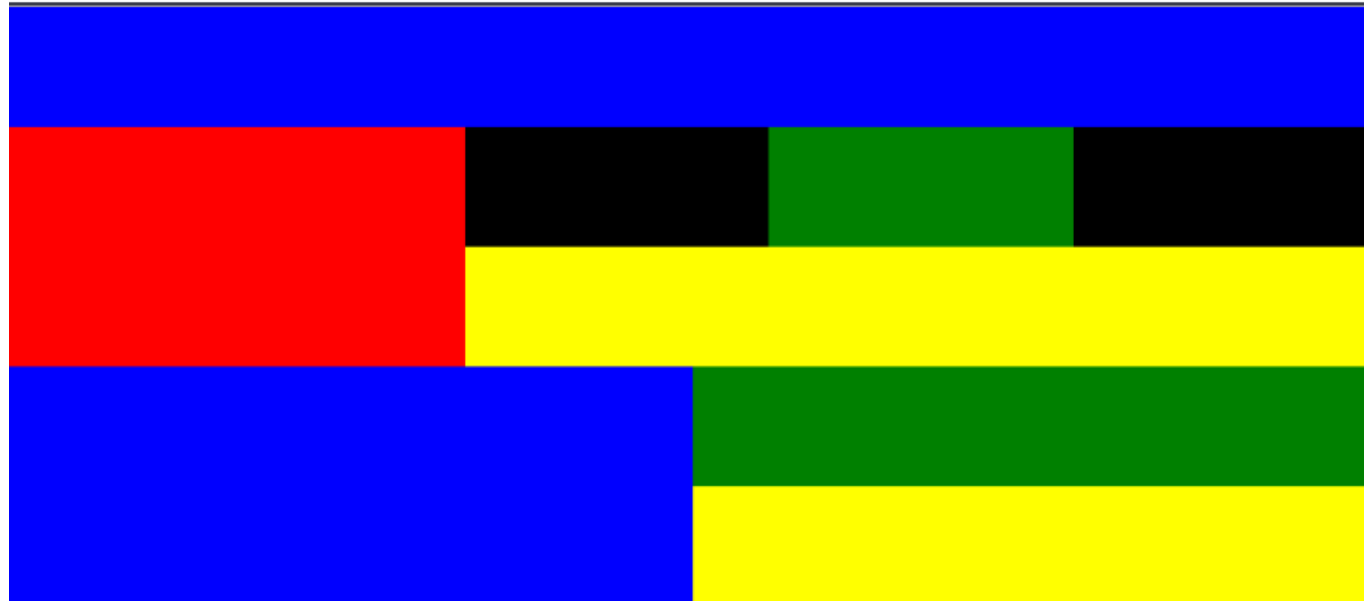
MAQUETACIÓN RESPONSIVE Y LAYOUT:

13. ANIDANDO FILAS

- Esta estructura establece una jerarquía *contenedor -> row -> col*, es decir, todas las filas tienen elementos que única y exclusivamente pertenecen a dicha fila.
- Sin embargo, los layout reales son más complejos que eso y para conseguir esas estructuras complejas vamos a tener que anidar filas (*row*) dentro de filas. Debemos de tener en cuenta que cada una de esas filas hijas podrá tener, de igual manera, hasta 12 columnas.
- En la siguiente imagen tenemos un ejemplo de ese tipo de layouts con las distintas filas señaladas.

MAQUETACIÓN RESPONSIVE Y LAYOUT:

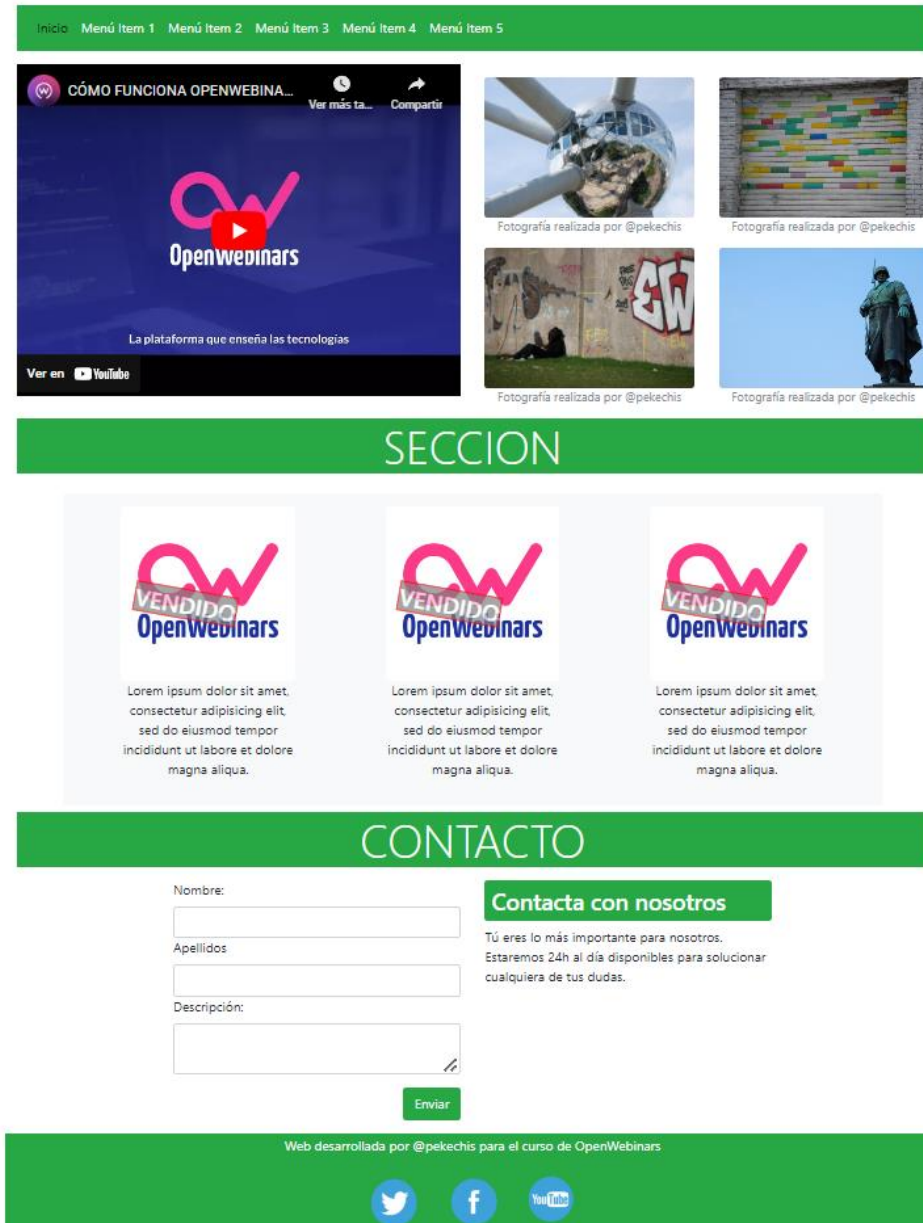
13. ANIDANDO FILAS



```

• <div class="container">
•   <div class="row">
•     <div class="col blue"></div>
•   </div>
•   <div class="row">
•     <div class="col-md-4 red grande"></div>
•     <div class="col-md-8">
•       <div class="row">
•         <div class="col black"></div>
•         <div class="col green"></div>
•         <div class="col black"></div>
•       </div>
•       <div class="row">
•         <div class="col yellow"></div>
•       </div>
•     </div>
•   </div>
•   <div class="row">
•     <div class="col-md-6 blue grande"></div>
•     <div class="col-md-6">
•       <div class="row">
•         <div class="col green"></div>
•       </div>
•       <div class="row">
•         <div class="col yellow"></div>
•       </div>
•     </div>
•   </div>
• </div>

```



Recuerda el proyecto del curso.

<https://openwebinars.net/academia/aprende/bootstrap-4-layout/3187/>

MAQUETACIÓN RESPONSIVE Y LAYOUT:

14. MAQUETACIÓN DEL PROYECTO

- Una vez visto el vídeo intenta una maquetación básica del mismo:



CONTENIDO DE LAS PÁGINAS: ELEMENTOS DE TEXTO

- BOOTSTRAP nos proporciona estilos propios:

TIPOS DE FUENTE

Bootstrap utiliza sus propios tipos de fuente. “Native Font stack” para que sea óptima en todos los navegadores

`<h>` y `<p>`

Así nos olvidamos de los incómodos márgenes y paddings por defecto que los navegadores aplican a estas etiquetas

- Use a native font stack that selects the best font-family for each OS and device.
- Y clases específicas, veámoslas:

CONTENIDO DE LAS PÁGINAS: ELEMENTOS DE TEXTO

PARA ENCABEZADOS

- Clases .hx (simulan hx)
- Clases displayX (más grandes y destacados)

CONTENIDO DE LAS PÁGINAS: ELEMENTOS DE TEXTO

- PARA LISTAS

- Clase `.list-unstyled` #Lista sin viñeta
- Clases `.list-inline` / `.list-inline-item` # alinear elementos horizontalmente

CONTENIDO DE LAS PÁGINAS: ELEMENTOS DE TEXTO

- PARA ALINEAR ELEMENTOS

- text-center /text-start/text-end
- text-truncate #Cortar texto si sobresale del contenedor

Y MÁS CLASES (similares a las etiqueta html)

- blockquote / blockquote-footer
- mark
- small

CONTENIDO DE LAS PÁGINAS: ELEMENTOS DE TEXTO

- **Los elementos de texto no son responsivos en BootStrap,** tenemos que gestionarlo mediante media queries
- No es del todo malo...el texto no cambia tanto respecto a las pantallas

CONTENIDO DE LAS PÁGINAS: IMÁGENES

- Para aquellos que no tengan claro la diferencia entre imágenes y figuras podemos resumirlo de la siguiente manera:
- Una **imagen** es únicamente el elemento gráfico el cual hemos añadido mediante las etiquetas *img* o *picture*.

``

CONTENIDO DE LAS PÁGINAS: IMÁGENES Y FIGURAS

- Una **figura** (etiqueta *figure*) es un conjunto compuesto de una imagen (etiqueta *img*) y de un texto descriptivo sobre la imagen (etiqueta *caption*). Esta etiqueta es una de las novedades en HTML5. Tradicionalmente es lo que se usa en *libros* para hacer posteriormente un índice de figuras.

```
<figure>  <figcaption>Texto descriptivo de la  
imagen</figcaption> </figure>
```

- Bootstrap, nos proporciona una serie de clases para tratar y dar estilos a este tipo de contenidos.

CONTENIDO DE LAS PÁGINAS: IMÁGENES Y FIGURAS

IMÁGENES

En relación a las imágenes las clases de interés son:

- **img-fluid** : Que nos permite convertir una imagen en responsiva. Al añadir esa clase a una imagen ésta ocupará toda la anchura de su elemento padre manteniendo sus proporciones (aspect ratio). Si hemos utilizado la etiqueta *picture* debemos añadir esta clase a la *img* que contiene y no a *picture*.
- **img-thumbnail** : Para añadir un marco redondeado a la imagen. Para esto podríamos también usar las clases relativas a bordes que veremos posteriormente en el capítulo sobre utilidades.
- **rounded** . Si queremos que las esquinas sean redondeadas.

CONTENIDO DE LAS PÁGINAS: IMÁGENES Y FIGURAS

FIGURAS

En relación a las figuras las clases de interés para dar estilos con Bootstrap son:

- **figure** : Clase a añadir a la etiqueta *figure*.
- **figure-img** : Clase a añadir a la etiqueta *img* que contiene la figura.
- **figure-caption** : Clase añadir a la etiqueta *figcaption* que contiene la figura. El texto se podrá adicionalmente alinear de distintas maneras usando *text-justify*, *text-left*, *text-right* o *text-center*.

IMPORTANTE : Si queremos que esto siga siendo responsivo debemos añadir *.img-fluid* a la imagen de la figura.

CONTENIDO DE LAS PÁGINAS: IMÁGENES

Responsive images

- Podemos hacer imágenes responsivas con .img-fluid.

```
<img src="" class="img-fluid" />
```

- Con ello obtenemos:
 - max-width: 100%;
 - height: auto;

CONTENIDO DE LAS PÁGINAS: IMÁGENES

[DOCUMENTACIÓN OFICIAL](#)

<https://getbootstrap.com/docs/5.0/content/images/>

CONTENIDO DE LAS PÁGINAS:

TABLAS

- Mediante el uso de clases Bootstrap podemos dar distintos estilos a las tablas y hacerlas responsivas.
- Estas clases relacionadas con tablas son muchas y en este apartado únicamente vamos a ver las más destacadas. Para más información os invito a visitar la documentación

CONTENIDO DE LAS PÁGINAS:

TABLAS

Haciendo que la tabla sea tabla Bootstrap

- Simplemente debemos añadir la clase ***table*** y adicionalmente la clase ***table-dark*** si queremos que se inviertan los colores de fondo y de letra.

```
<table class="table"> ..... </table>
```

CONTENIDO DE LAS PÁGINAS: TABLAS

Personalízalas con colores:

Class	Heading	Heading
Default	Cell	Cell
Primary	Cell	Cell
Secondary	Cell	Cell
Success	Cell	Cell
Danger	Cell	Cell
Warning	Cell	Cell
Info	Cell	Cell
Light	Cell	Cell
Dark	Cell	Cell

CONTENIDO DE LAS PÁGINAS:

TABLAS

```
<table class="table">
  <thead>
    <tr>
      <th scope="col">#</th>
      <th scope="col">First</th>
      <th scope="col">Last</th>
      <th scope="col">Handle</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>Mark</td>
      <td>Otto</td>
      <td>@mdo</td>
    </tr>
    <tr>
      <th scope="row">2</th>
      <td>Jacob</td>
      <td>Thornton</td>
      <td>@fat</td>
    </tr>
    <tr>
      <th scope="row">3</th>
      <td colspan="2">Larry the Bird</td>
      <td>@twitter</td>
    </tr>
  </tbody>
</table>
```

Las clases se pueden utilizar para colorear tablas, filas o celdas individuales. Ejemplo
<thead class="table-info">

CONTENIDO DE LAS PÁGINAS: TABLAS

Clases para la cabecera de las tablas

- Para dar estilos a las cabecera de las tablas tenemos las clases *light* o *thead-dark* a la etiquetas de las filas (*tr*) o a las etiquetas (*thead*).

```
<thead class="thead-light">  
  <tr>  
    <th>...</th>  
    .      ....  
  </tr>  
</thead>
```


CONTENIDO DE LAS PÁGINAS:

TABLAS

Alternando colores

- Si queremos que las filas de las tablas tengan colores alternativos para poder distinguirlas mejor debemos añadir la clase ***table-striped*** a la etiqueta *table*. Si queremos las columnas alternativas la clase será ***table-striped-columns***.
- Esto funcionará también para tablas oscuras (clase *table-dark*).

```
<table class="table table-striped">
```

```
.....
```

```
</table>
```

CONTENIDO DE LAS PÁGINAS:

TABLAS

Destacar una fila al situarnos sobre ella

- Si queremos que conforme pase el puntero de ratón por encima de una fila (que no sea la cabecera) esa fila de destaque cambiando ligeramente de color debo añadir la clase ***table-hover*** a la etiqueta *table*.

```
<table class="table table-hover">
```

```
.....
```

```
</table>
```

CONTENIDO DE LAS PÁGINAS: TABLAS

Destacar una fila

- Puedes destacar una fila utilizando la clase *.table-active*

```
<tr class="table-active">
```

```
.....
```

```
</tr>
```

CONTENIDO DE LAS PÁGINAS:

TABLAS

Bordes de las tablas

- Por defecto en Bootstrap, las tablas únicamente muestran unos ligeros bordes inferiores en las filas para separar estas. Si, por el contrario, queremos que todas las celdas muestren los 4 bordes (inferior, superior, derecha e izquierda) debemos añadir la clase ***table-bordered*** a la etiqueta *tabla*

```
<table class="table table-bordered">
```

```
.....
```

```
</table>
```

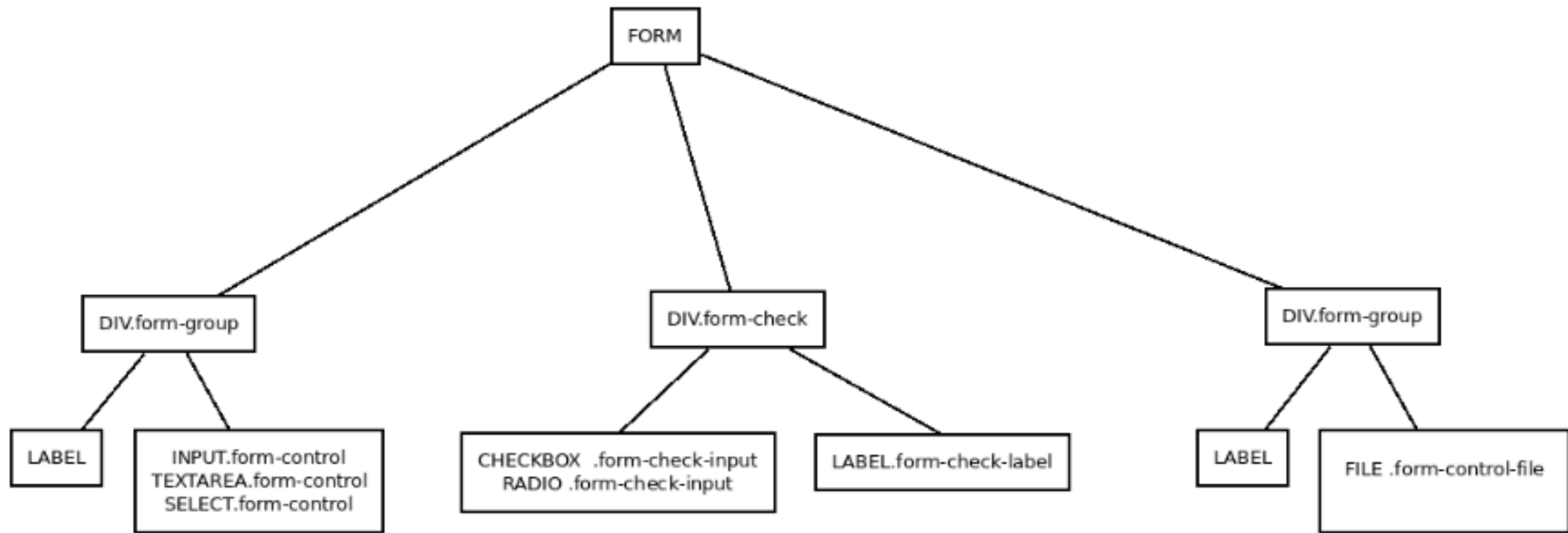
CONTENIDO DE LAS PÁGINAS: TABLAS

- <https://getbootstrap.com/docs/5.0/content/tables/>

CONTENIDO DE LAS PÁGINAS: FORMULARIOS

- Aunque en la documentación oficial los formularios se encuentran en la sección de componentes, lo cierto es que son un elemento fundamental que se incluye en páginas de registro, de consulta, de búsqueda etc. Por esa razón se ha decidido tratarlos de manera breve en este curso.
- Bootstrap proporciona una serie de clases para dar estilos a los distintos elementos de los formularios. De manera general podemos describir estas clases y la jerarquía que deben ocupar de la siguiente manera:

CONTENIDO DE LAS PÁGINAS: FORMULARIOS



CONTENIDO DE LAS PÁGINAS: FORMULARIOS

Además, se le añadirá al final un *input* de tipo *submit* o *button* con las clases correspondientes a los botones cuyos ejemplos más comunes (hay muchos más) son:

- `btn btn-primary`
- `btn btn-secondary`
- `btn btn-success`
- `btn btn-danger`
- `btn btn-warning`

CONTENIDO DE LAS PÁGINAS:

FORMULARIOS

Podemos además modificar ciertos aspectos de estos componentes del formulario. Los más interesantes son los siguientes.

- **Modificar el tamaño en altura del control.** Añadiendo clases como *form-control-lg* (grande) o *form-control-sm* (pequeños) en los *form-control*.
- **Modificar el tamaño en altura de la etiqueta del control.** Usando clases como *col-label-lg* (grandes) o *col-label-sm* (pequeños).

CONTENIDO DE LAS PÁGINAS:

FORMULARIOS

- Hacer que todos los elementos del formulario se vean en la misma línea añadiendo la clase *form-inline* a la etiqueta form.
- Hacer que las distintas opciones para elementos *radio* o *checkbox* se vean en la misma línea añadiendo al div que tenía la clase *form-check* la clase *form-check-inline*
- Añadir texto de ayuda a los diferentes elementos usando un etiqueta small dentro del *form-group* o *form-check* y dando a esa etiqueta las clases *form-text* y *text-mute*.

CONTENIDO DE LAS PÁGINAS: FORMULARIOS

- En cuanto a su disposición, los formularios por defecto ocupan en anchura lo que ocupen el contenedor padre al que corresponden pero podemos adaptar su tamaño jugando con el grid de Bootstrap que hemos visto en capítulos anteriores añadiendo clases **col-X** (o atendiendo a distintos breakpoints) al elemento que contenga la clase *form-group* o *form-check*.
- Para hacer los formularios más compactos hay una nueva clase que suprime el gutter, ***form-row*** que debe ser usada en vez de *row*.
- Bootstrap proporciona muchas más cosas en relación a los formularios, por eso, aunque con estos apuntes ya es suficiente para empezar a trabajar sin problemas, os recomiendo mirar tranquilamente la documentación.

ELEMENTOS MULTIMEDIA:

OBJETO MEDIA

- *Nos ayuda a crear elementos complejos y repetitivos en los cuales algún elemento multimedia (una imagen normalmente) se posiciona junto a otro contenido (normalmente texto)“.*
- *Se utiliza en comentarios de blogs, tweets, etc.*
- *“Ya no es soportado en Bootstrap 5” → Lo solucionamos con la clase flex y spacing*

ELEMENTOS MULTIMEDIA: OBJETO MEDIA



Jhon Carter *Posted on January 10, 2021*

Excellent feature! I love it. One day I'm definitely going to put this Bootstrap component into use and I'll let you know once I do.



Clark Kent *Posted on January 12, 2021*

Thanks, you found this component useful. Don't forget to check out other Bootstrap components as well. They're also very useful.



Jhon Carter *Posted on January 10, 2021*

Excellent feature! I love it. One day I'm definitely going to put this Bootstrap component into use and I'll let you know once I do.

ELEMENTOS MULTIMEDIA: OBJETO MEDIA

- Partiendo de este ejemplo:



Crea la estructura anidada de la parte superior del ejemplo anterior.

- `<div class="row">`
- `<div class="col border border-primary">`
- `<div class="d-flex m-4 border border-secondary">`
- `<div class="flex">`
- ``
- `</div>`
- `<div class="flex ms-3">`
- `<h5>Jhon Carter <small class="text-muted"><i>Posted on January`
`10, 2021</i></small></h5>`
- `<p>Excellent feature! I love it. One day I'm definitely going`
`to put this Bootstrap component into use and I'll let you know once I do.</p>`
- `</div>`
- `</div>`
- `</div>`
- `</div>`
- `</div>`

ELEMENTOS MULTIMEDIA: OBJETOS INCRUSTADOS

- Los elementos embebidos son algo común en todas las webs , vídeos de youtube, reproductores de canciones de spotify, presentaciones de slideshare están embebidas o incrustadas por todos los sitios.
- Desde el punto de vista más técnico nos estamos refiriendo a las etiquetas *iframe, embed, video* y *object*

ELEMENTOS MULTIMEDIA: OBJETOS INCRUSTADOS

- Para hacer estas etiquetas responsivas debemos añadir la clase ***embed-responsive*** a un elemento que la contenga y la clase ***embed-responsive-item*** a la etiqueta en cuestión. Algo así:

```
<div class="embed-responsive"> <iframe src="..." class="embed-responsive-item"></iframe> </div>
```

ELEMENTOS MULTIMEDIA: OBJETOS INCRUSTADOS

Adicionalmente podemos modificar las proporciones elementos (su *aspect ratio*) añadiendo al elemento padre clases que lo modificarán:

- *embed-responsive-21by9* : Aspect Ratio 21x9
- *embed-responsive-16by9* : Aspect Ratio 16x9
- *embed-responsive-4by3* : Aspect Ratio 4x3
- *embed-responsive-1by1* : Aspect Ratio 1x1

UTILIDADES DE BOOTSTRAP:

- BOOTSTRAP ES UN FRAMEWORK QUE NOS PROPORCIONA:
 - CLASES PARA MAQUETAR.
 - CLASES PARA GESTIONAR EL CONTENIDO: TEXTO, IMÁGENES, TABLAS...
 - CLASES PARA CREAR FORMULARIOS RESPONSIVOS
 - COMPONENTES PARA DAR DINAMISMO A MIS PÁGINAS: BARRAS DE NAVEGACIÓN, CARRUSELES DE IMÁGENES, CARDS, VENTANAS MODALES.
 - UTILIDADES PARA CAMBIAR ASPECTO, DIMENSIÓN DE LOS ELEMENTOS: BORDER, COLORES, POSICIONAR ELEMENTOS ... EVIDENTEMENTE LOS PODEMOS HACER EN NUESTRA PROPIA HOJA DE ESTILOS

FIN