

# **UNIDAD 5:**

# **BOOTSTRAP 4**

Diseño de Interfaces Web

2º Desarrollo de Aplicaciones Web

*José Reyes Gil Rubio*

## Tabla de contenido

COMENZANDO CON BOOTSTRAP .....	4
¿Qué es Bootstrap? .....	4
Descarga e instalación de Bootstrap .....	4
Contenedores en Bootstrap .....	5
Sistema de rejilla o grid .....	6
Ejemplo de tres columnas .....	6
Sistema de 12 columnas.....	6
Gutter .....	7
Multilínea de igual ancho .....	7
Puntos de ruptura .....	7
Ejercicio 1ª parte .....	9
Ocultar elementos.....	10
Ejercicio 2ª parte .....	10
Alineaciones .....	11
Alineación vertical dentro de una fila .....	11
Alineación vertical de cada elemento .....	11
Alineación horizontal .....	11
Orden de las columnas.....	11
Dejar columnas en blanco a la izquierda de una columna.....	12
Anidamientos .....	12
APLICANDO BOOTSTRAP 4 .....	13
Unidades porcentuales de ventana de visualización .....	13
Componentes, tipografías y utilidades.....	13
Cabecera.....	13
Espaciado.....	14
Tipografías.....	15
Colores.....	15
Tamaños .....	16
Elementos flotantes .....	16
Menú de navegación.....	17
Cards.....	17
Jumbotron .....	18

Formularios .....	18
Carrusel .....	18
Cheatsheet .....	18
COPIANDO PLANTILLAS DE EJEMPLO .....	19
OTROS ELEMENTOS.....	19
Modal .....	19
Scrollspy .....	19
Dos sticky-top menús .....	20
Tooltips.....	20
Menú dividido en dos partes con el logo en el centro.....	20
Cambio de color de menú al hacer scroll .....	21
Tarjetas con doble imagen .....	21
Sección de testimonios .....	22

## COMENZANDO CON BOOTSTRAP

### ¿Qué es Bootstrap?

Bootstrap es el framework de HTML, CSS y JavaScript más popular del mundo desarrollado por el equipo de Twitter. Definiéndolo de forma más simple, es un conjunto de estilos, efectos, configuraciones responsive, componentes,... que podemos utilizar para construir nuestra web.

Imagina que tienes los estilos CSS para un menú de navegación, un carrusel, un formulario de contacto, botones,... y esto lo podemos agregar y configurar fácilmente en nuestro sitio web.

Trabajaremos con la versión 4.4.1. Si una persona ya escribió el código HTML, CSS y JavaScript, entonces ¿para qué escribir más código? Nos permitirá crear sitios responsivos adaptados a dispositivos móviles y tablets, con menús de navegación, botones, formularios, viñetas, paginación, tarjetas y muchísimo más.

Existen multitud de plantillas con Bootstrap, así que será fácil editarlas. Utiliza un grid responsive de 12 columnas y trae integrado decenas de complementos, plugins de JavaScript, tipografías, controladores de formularios, etc.

### Descarga e instalación de Bootstrap

Creamos una nueva carpeta donde guardaremos nuestro proyecto.

Para comenzar a utilizar Bootstrap lo primero que hemos de hacer es descargarlo. Para ello en Google escribimos "Bootstrap 4". Seguimos los enlaces hasta llegar a la página de descarga de Bootstrap. Buscamos la etiqueta "Compiled CSS y JS" y procedemos a la descarga guardando el archivo en nuestra carpeta de proyecto.

Lo descomprimos y observaremos que tenemos dos carpetas con archivos CSS y JavaScript.

Abrimos Visual Studio Code y creamos un nuevo archivo index.html que guardaremos en la carpeta descomprimida de Bootstrap (donde se encuentran las carpetas CSS y JS).

Desde la web oficial de Bootstrap accedemos a la opción de "Introduction" dentro de "Getting started" y buscamos el código de "Starter template". Copiamos el código y lo pegamos en nuestro archivo HTML.

Los archivos CSS y JavaScript en este código están enlazados a archivos externos. Es tu elección dejarlos así o cambiarlos por los descargados con Bootstrap.

Si quieres cambiarlos:

- Eliminamos la línea de Bootstrap CSS y la sustituimos por:
  - `<link rel="stylesheet" href="css/bootstrap.min.css">` donde se especifica la ruta del archivo bootstrap.min.css
- Hacemos lo mismo con el script de Bootstrap JS quedando de la siguiente manera.
  - `<script src="js/bootstrap.min.js"> </script>`

Ya lo tenemos preparado todo para comenzar.

## Contenedores en Bootstrap

Container crea un contenedor que incluirá los elementos a configurar. Para crearlo haremos un div con clase container. Estos elementos se utilizan como raíz de la rejilla y **no se podrán anidar unos dentro de otros**.

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

Existen contenedores de dos tipos:

- container-fluid: Que tiene un ancho del 100%
- container: Ancho fijo.
- container-{breakpoint}: será del 100% hasta el punto de interrupción especificado.

```
<div class="container-fluid fondo">
  <p>CONTAINER-FLUID: Ancho 100%, sin márgenes.</p>
</div>
<div class="container fondo">
  <p>CONTAINER-FLUID: Ancho 100%, sin márgenes.</p>
</div>
```

## Hello, world!

CONTAINER-FLUID: Ancho 100%, sin márgenes.

CONTAINER-FLUID: Ancho 100%, sin márgenes.

En la siguiente tabla se muestra cuál es la anchura máxima del contenedor respecto al contenedor .container y .container-fluid en cada punto de interrupción.

	Extra pequeño <576px	Pequeño ≥576px	Mediano ≥768px	Grande ≥992px	Extra grande ≥1200px
.container	100%	540px	720px	960px	1140px
.container-sm	100%	540px	720px	960px	1140px
.container-md	100%	100%	720px	960px	1140px
.container-lg	100%	100%	100%	960px	1140px
.container-xl	100%	100%	100%	100%	1140px
.container-fluid	100%	100%	100%	100%	100%

Cada vez que estemos trabajando en una sección de nuestro sitio web tendrá que estar contenida dentro de estos containers.

## Sistema de rejilla o grid

Trabaja con un sistema de 12 columnas sobre flexbox que nos permite hacer el sistema de columnas mucho más flexible.

### Ejemplo de tres columnas

El sistema de cuadrícula de Bootstrap utiliza una serie de contenedores, filas y columnas para diseñar y alinear el contenido.

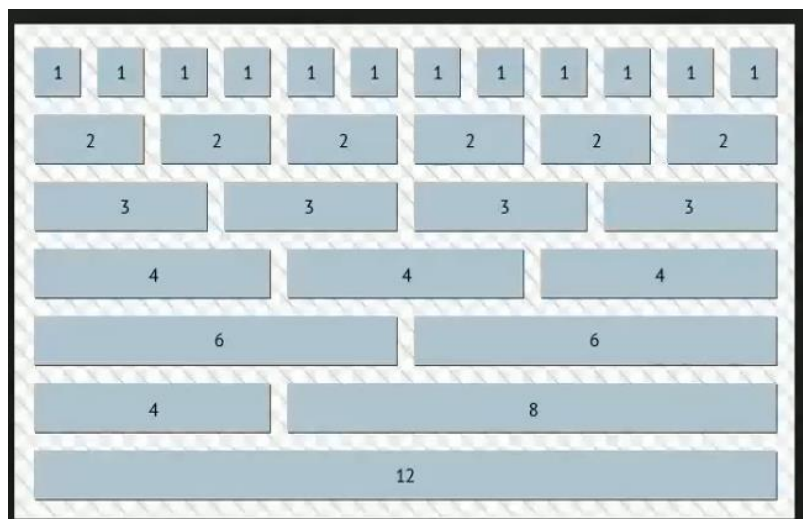
- Creamos un **.container** (contenedor): Los contenedores proporcionan un medio para centrar y rellenar horizontalmente los contenidos de su sitio.
- Creamos un **.row** (fila): Las filas son los envoltorios para las columnas.
- Creamos tres **.col** (columnas). El contenido debe colocarse dentro de las columnas y sólo las columnas que deben ser elementos secundarios inmediatos de las filas. Como trabajamos con FlexBox todas tienen el mismo tamaño si no se especifica una anchura concreta.

```
<div class="container fondo">  
  <div class="row">  
    <div class="col borde">1</div>  
    <div class="col borde">2</div>  
    <div class="col borde">3</div>  
  </div>  
</div>
```

El ejemplo anterior crea tres columnas de igual ancho utilizando nuestras clases de cuadrícula predefinidas. Esas columnas están centradas en la página con el padre .container.

### Sistema de 12 columnas

Este framework trabaja con un sistema de 12 columnas.



En la primera fila de la imagen cada espacio ocupa 1 columna. Si quisieras hacer 6 columnas iguales cada columna ocuparía dos espacios como se ve en la segunda fila de la imagen. Pero se podrían hacer multitud de combinaciones siempre ocupando un total de 12 columnas. Si el

tamaño total de las columnas de una fila excede de 12 el tamaño sobrante se colocará en la siguiente línea.

Las clases de columna indican el número de columnas que le gustaría usar de las 12 posibles por fila. Entonces, si desea tres columnas de igual ancho, puede usar `.col-4`.

```
<div class="container fondo">
  <div class="row">
    <div class="col-2 borde">1</div>
    <div class="col-6 borde">2</div>
    <div class="col-4 borde">3</div>
  </div>
</div>
```

La anchura de las columnas siempre se establece en porcentajes, por lo que siempre son fluidas y de tamaño relativo a su elemento padre.

### Gutter

Tanto las columnas como las filas tienen padding horizontal para dejar huecos entre ellas. Estos espacios se pueden utilizar indicando la clase **`.no-gutters`** sobre la fila.

```
<div class="container fondo">
  <div class="row no-gutters">
    ...
  </div>
</div>
```

### Multilínea de igual ancho

Podemos crear columnas de igual ancho que abarquen varias líneas insertando `.w-100` donde deseemos que las columnas se rompan en una nueva línea.

```
<div class="container fondo">
  <div class="row">
    <div class="col borde">columna 1</div>
    <div class="col borde">columna 2</div>
    <div class="w-100"></div>
    <div class="col borde">columna 3</div>
    <div class="col borde">columna 4</div>
  </div>
</div>
```

### Puntos de ruptura

Este sistema es 100% responsive.

A partir de aquí comenzaremos a hacer nuestros diseños usando el concepto First Mobile, es decir, primero móviles, luego tablets y finalmente para escritorio.

Existen puntos de ruptura que nos permitirán establecer límites de anchura de dispositivos:

	<b>Extra small</b> <576px	<b>Small</b> ≥576px	<b>Medium</b> ≥768px	<b>Large</b> ≥992px	<b>Extra large</b> ≥1200px
<b>Max container width</b>	None (auto)	540px	720px	960px	1140px
<b>Class prefix</b>	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
<b># of columns</b>	12				
<b>Gutter width</b>	30px (15px on each side of a column)				

- sm: para pantallas pequeñas
- md: para pantallas medianas
- lg: para pantallas grandes
- xl: para pantallas muy grandes.

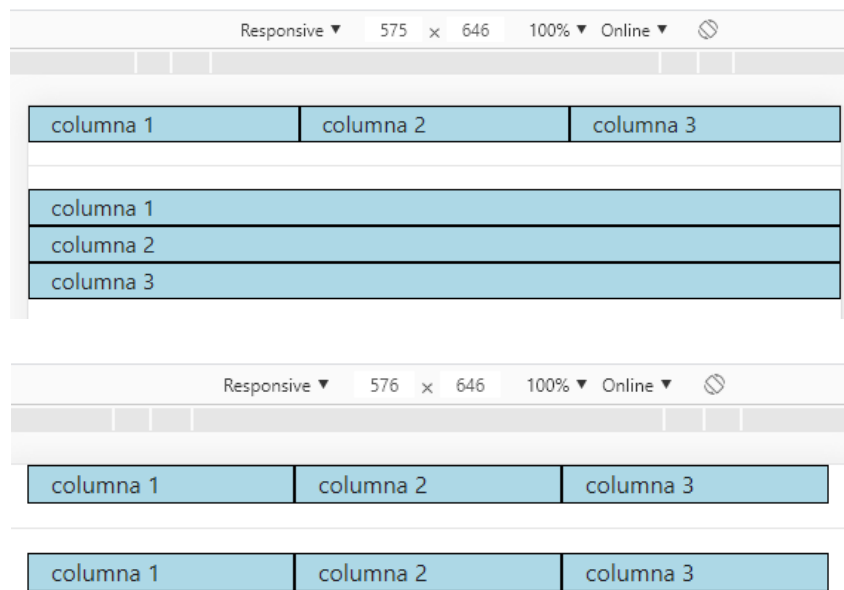
Si usamos el primer tamaño .col-sm para pantallas entre 576 y 767 píxeles y visualizamos veremos que para pantallas menores a 576 Bootstrap le da a las columnas anchura del 100%, para las mayores reparte la anchura a partes iguales.

```
<div class="container fondo">
  <div class="row">
    <div class="col borde">columna 1</div>
    <div class="col borde">columna 2</div>
    <div class="col borde">columna 3</div>
  </div>
</div>

<hr>

<div class="container fondo">
  <div class="row">
    <div class="col-sm borde">columna 1</div>
    <div class="col-sm borde">columna 2</div>
    <div class="col-sm borde">columna 3</div>
  </div>
</div>
```





En el ejemplo anterior, las medidas md, lg o xl no están definidas por lo tanto Bootstrap asignará la medida sm para cada una de ellas.

```
<div class="container fondo">
  <div class="row">
    <div class="col-sm-6 col-md-4 col-lg-3 borde">columna 1</div>
    <div class="col-sm-6 col-md-4 col-lg-3 borde">columna 2</div>
    <div class="col-sm-6 col-md-4 col-lg-3 borde">columna 3</div>
    <div class="col-sm-6 col-md-12 col-lg-3 borde">columna 3</div>
  </div>
</div>
```

¿Cómo funcionaría este código? ¿Qué pasa con xl?

### Ejercicio 1ª parte

```
<body>
  <div class="container">
    <div class="row">
      <div class="col borde color4">HEADER</div>
    </div>
    <div class="row">
      <div class="col-sm-6 col-md-3 borde color1">IZQUIERDA</div>
      <div class="col-sm-6 col-md-6 borde color2">CONTENIDO</div>
      <div class="col col-md-3 borde color3">DERECHA</div>
    </div>
    <div class="row">
      <div class="col borde color4">FOOTER</div>
    </div>
  </div>
```

- Cabecera y pie de página de ancho 100%,
- Zona central:
  - en tamaño sm dos líneas:
    - primera, 2 columnas al 50%
    - segunda, 1 columna al 100%
  - en tamaño md y superiores 3 columnas:
    - 1ª a 25%
    - 2ª a 50%
    - 3ª a 25%

### Ocultar elementos

Para ocultar un elemento haremos uso de la clase **.d-\*-none** (display-medida-none)

Para mostrar de nuevo el elemento haremos uso de la clase **.d-\*-block**. Por defecto, todos los elementos están visibles (existe también la clase **.d-\*-inline**).

```
<div class="col-sm-6 col-md-3 borde color1 d-none d-sm-block">IZQUIERDA</div>
```

Oculto el elemento para todos los dispositivos, pero lo vuelve a mostrar a partir de sm.

```
<div class="col-sm-6 col-md-3 borde color1 d-sm-none d-md-block">IZQUIERDA</div>
```

Oculto el elemento solo para tamaños sm.

### Ejercicio 2ª parte

Modificar el ejercicio anterior usando etiquetas de estructura HTML5 y ocultando para tamaños menores de sm el contenido de la izquierda.

```
<div class="container">
  <header class="row">
    <div class="col borde color4">HEADER</div>
  </header>
  <div class="row">
    <aside class="col-sm-6 col-md-3 borde color1 d-none d-sm-block">
      IZQUIERDA
    </aside>
    <main class="col-sm-6 col-md-6 borde color2">
      CONTENIDO
    </main>
    <aside class="col col-md-3 borde color3">
      DERECHA
    </aside>
  </div>
  <footer class="row">
    <div class="col borde color4">FOOTER</div>
  </footer>
</div>
```

## Alineaciones

Se realiza a través de Flex Box.

### Alineación vertical dentro de una fila

Para hacer alineación vertical de las columnas dentro de una fila en primer lugar tendremos que asignarle un alto a la fila y después le indicaremos **a la fila** el tipo de alineación con alguna de las siguientes clases:

- La alineación vertical por defecto es superior (arriba) → **.align-items-start**
- La alineación vertical centrada → **.align-items-center**
- La alineación vertical inferior (abajo) → **.align-items-end**

```
<div class="row color4 align-items-center">
```

### Alineación vertical de cada elemento

Alineación vertical de cada elemento dentro de la fila. Le indicaremos **al elemento** el tipo de alineación con alguna de las siguientes clases:

- La alineación vertical superior (arriba) → **.align-self-start**
- La alineación vertical centrada → **.align-self-center**
- La alineación vertical inferior (abajo) → **.align-self-end**

```
<div class="borde color3 col-1 align-self-end">6</div>
```

### Alineación horizontal

Alineación horizontal de las columnas dentro de una fila. Le indicaremos **a la fila** el tipo de alineación con alguna de las siguientes clases:

- La alineación horizontal centrada → **.justify-content-center**
- La alineación horizontal izquierda → **.justify-content-start**
- La alineación horizontal derecha → **.justify-content-end**
- La alineación horizontal que genera espacio al principio y al final de un tamaño la mitad que el de en medio → **.justify-content-around**
- La alineación horizontal que genera espacio entre las columnas, pero no deja margen a la izquierda ni a la derecha → **.justify-content-between**

```
<div class="row color4 justify-content-between">
```

## Orden de las columnas

Usando la siguiente clase podemos modificar el orden de las columnas:

- **.order-número**: Donde número (entre 1-12) indica la posición de ese elemento. Tenga en cuenta que si no se especifica el elemento tiene un order-0.
- Podemos aplicar el reordenamiento para los distintos tamaños de pantalla. (order-, order-sm-, order-md-, order-lg- y order-xl-).

```
<div class="borde color3 col-1 order-1">5</div>
```

## Dejar columnas en blanco a la izquierda de una columna

Usamos la siguiente clase para dejar huecos de columnas en blanco a la izquierda de la columna a la que se la aplicamos:

- offset-\*-número (offset-tamaño-número de huecos): **offset-md-3**

```
<div class="borde color3 col-1 offset-md-3">3</div>
```

Dejará 3 huecos a la izquierda de la columna a partir del tamaño md.

## Anidamientos

Dentro de una columna podemos crear una fila anidada y dentro de esa fila podemos crear varias columnas.

```
<div class="borde color3 col align-self-start">1  
  <div class="row">  
    <div class="color2 col">sub1</div>  
    <div class="color2 col">sub2</div>  
    <div class="color2 col">sub3</div>  
  </div>  
</div>
```

## APLICANDO BOOTSTRAP 4

### Unidades porcentuales de ventana de visualización

Las “unidades de longitud porcentuales de ventana de visualización” (viewport-percentage lengths) o más comúnmente llamadas viewport units son unidades css relativas a la ventana gráfica del navegador o visualizador.

Este bloque inicial o ventana gráfica puede considerarse como la capa padre de toda nuestra web, y las unidades “vh” y “vw” son las unidades en altura y anchura que hacen referencia a esta ventana:

vh (viewport-height) = Altura de la ventana gráfica.

vw (viewport-width) = Anchura de la ventana gráfica.

Sus valores se expresan en porcentaje (de 0 a 100). Ejemplos:

width:100vw; (anchura del 100% de la anchura de la ventana gráfica)

height:50vh; (altura del 50% de la altura de la ventana gráfica)

line-height:3vh; (altura de línea del 3% de altura de la ventana gráfica)

La diferencia entre las unidades porcentuales expresadas con “%” (width:100%) y las viewport-units (width:100vw) es que las primeras heredan los valores de su elemento padre, y las segundas siempre heredan los valores del bloque inicial o ventana gráfica.

### Componentes, tipografías y utilidades

Tomaremos como base a la página oficial de Bootstrap para aprender estos conceptos que los realizaremos sobre una práctica.

#### Cabecera

Colocaremos una imagen de fondo que se adaptará en función del tamaño de la ventana y sobre la que aparecerá un título centrado tanto horizontal como verticalmente.

Añadiremos esta clase en estilos CSS:

```
<style>
    .slider{
        background: url("imagenes/fruta.jpg");
        height: 100vh;
        background-size: cover;
        background-position: center;
    }
</style>
```

Asignamos nuestra imagen de fondo de cabecera, le damos una altura 100% de la altura del dispositivo. Con las dos últimas líneas le indicamos que este centrada y que se adapte al tamaño del dispositivo.

Definimos en nuestro HTML la etiqueta `<header>` e indicamos que será un contenedor de tipo fluido para que ocupe el 100%. Además le añadiremos la clase `.slider` para que coloque la imagen de fondo.

Dentro de nuestra cabecera introducimos el título con etiquetas H1. Si nos dirigimos al manual de referencia de Bootstrap en su página oficial a la sección de **Contenido** → **Tipografías** veremos las clases `display` que son las que utilizaremos en nuestro ejemplo (**`.display-3`**).

Ahora lo colocaremos centrado. Para ello utilizaremos flexbox, para ver cómo funciona nos dirigimos a su referencia en **Utilidades** → **Flex**. Aquí se nos indica que para colocar un elemento como flexbox tenemos que usar la clase **`.d-flex`** sobre el contenedor padre mostrando ahora a todos sus elementos hijos como inline.

En este mismo lugar se nos indica que para alinear horizontalmente usaremos la clase **`justify-content-center`** y para hacerlo de forma vertical usaremos la clase **`align-items-center`**.

```
<header class="container-fluid slider d-flex justify-content-center align-items-center">
  <h1 class="display-3">Bootstrap 4!</h1>
</header>
```

## Espaciado

Dentro de **Utilidades** → **Espaciado**.

Bootstrap trabaja con unidad de medida ***rem*** que es una unidad relativa que indica un índice que multiplica el tamaño por defecto de la fuente establecida, que en caso de Bootstrap es de 16px.

En nuestro proyecto crearemos una nueva sección ***main*** de tipo ***container*** para que nos dé una anchura fija en función del ancho del dispositivo.

Dentro del main añadimos un párrafo con un lorem10 y varios párrafos con lorem. Estos párrafos quedan “pegados” a nuestra cabecera. Si queremos darle márgenes o paddings usaremos las clases ***.m\*-unidad*** (márgenes) o ***.p\*-unidad*** (paddings).

Donde ***\**** actúa como modificador de esta clase y podemos usar (si no especificamos modificador actúa sobre los cuatro lados):

- t: top
- b: bottom
- l: left
- r: right
- x: Toda la línea en horizontal (izquierda y derecha)
- y: Toda la línea en vertical (arriba y abajo)

Como unidades podemos usar valores desde el 0 al 5 donde:

- 0 equivale a 0
- 1 equivale a 0.25rem
- 2 equivale a 0.5rem
- 3 equivale a 1rem
- 4 equivale a 1.5rem
- 5 equivale a 3rem

### Tipografías

Para alinear textos nos dirigiremos a **Utilidades → Text**. Por ejemplo, para centrar el texto usaremos la clase **.text-center**. Este tipo de clases permiten puntos de interrupción (sm, md, lg y xl).

Aquí tenemos también clases de transformación de texto:

- text-uppercase: Todo el texto en mayúsculas
- text-lowercase: Todo el texto en minúsculas
- text-capitalize: Primera letra de cada palabra en mayúsculas.

Otras clases para apariencia de texto:

- .font-weight-bold
- .font-weight-normal
- .font-weight-light
- .font-italic

### Colores

También podemos hacer uso de colores de texto predefinidos. Explicación en **Utilidades → Colores**:

.text-primary

.text-secondary

.text-success

.text-danger

.text-warning

.text-info

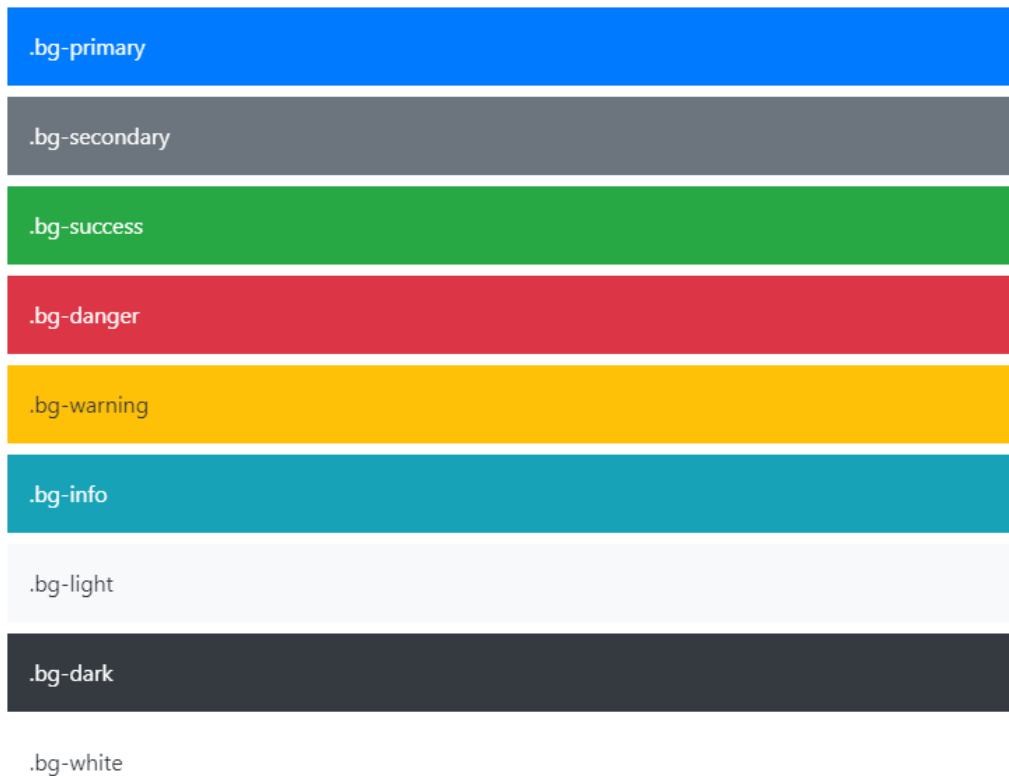
.text-light

.text-dark

.text-muted

.text-white

## Colores de fondo:



## Tamaños

Utilidades → Sizing (Tamaños)

- Con la clase **.w-porcentaje**, podemos indicar el **ancho porcentual** que queremos que ocupe el elemento.
- Con la clase **.h-porcentaje**, podemos indicar el **alto porcentual** que queremos que ocupe el elemento.

## Elementos flotantes

Podemos usar las clases:

- **.float-left**: Elemento flotante a la izquierda.
- **.float-right**: Elemento flotante a la derecha.

```
<main class="container">
  <p class="mt-5 pt-5 text-center text-uppercase font-italic font-weight-bold">
    Lorem ipsum ... Voluptatem, culpa?</p>
  <p class="text-danger">Lorem ipsum ... possimus nobis!</p>
  <p class="text-white bg-primary p-4 text-justify w-25 float-left mr-4">
    Lorem ipsum dolor ... possimus nobis!</p>
  <p>Lorem ipsum ... possimus nobis!</p>
  ...
  <p>Lorem ipsum ... possimus nobis!</p>
</main>
```



## Menú de navegación

Especificaciones en **Contenido** → **Navbar**.

Usaremos la etiqueta de HTML5 <nav> a la que le añadiríamos la clase **.navbar** (obligatorio para hacer un menú), que la tendremos que emplear siempre que deseemos realizar un menú de navegación.

Usaremos color gris oscuro para la barra con las clases **navbar-dark bg-dark**.

Metemos el logotipo con un enlace con clase **.navbar-brand** (importante) como imagen .svg, por lo que debemos especificar el ancho y alto. Además añadiremos clases para que esté alineada con nuestro texto.

Para crear los diferentes enlaces podemos usar una lista desordenada (ul), pero en este caso lo haremos con un div con clase **.navbar-nav**.

Y en lugar de elementos de lista (li) usaremos enlaces (a) con clases **.nav-item** y **.nav-link**.

Añadimos nuestro botón **Toggler** (hamburguesa) justo después de del .navbar-brand con el código copiado de la página de Bootstrap.

A continuación añadimos un div después del botón Toggler. Este div tendrá las clases **.collapse** y **.navbar-collapse** y como identificador el mismo que tiene el botón **Toggler**, lo que le indicará todo el contenido que debe colapsarse cuando hagamos uso del botón Toggler.

Con la clase **.navbar-expand-md** en la etiqueta <nav> le indicamos a partir de que elemento no aparecerá el menú con estilo hamburguesa. En nuestro ejemplo el menú horizontal aparece a partir del tamaño mediano, dejando el toggler para pequeños y extra pequeños.

Podemos agregar botones, por ejemplo, para Facebook y Youtube. Si queremos que desaparezcan cuando el menú se colapse tendremos que incluirlos dentro del div .navbar-collapse.

Alineamos los menús con diferentes clases para que aparezcan centrados.

Si queremos que el menú esté siempre visible en la parte superior de nuestra ventana usaremos la clase **.sticky-top**.

## Cards

**Componentes** → **Card**

Incluiremos un grupo de tarjetas de 3 en una nueva <section> con clase .container copiando en este caso el código de card decks que nos proporciona Bootstrap 4.

Si queremos modificar cuando aparecen las tarjetas en tres columnas debemos modificar el archivo CSS de Bootstrap, más específicamente la clase **.card-deck** que estará dentro de una media query que indica cuando se produce el cambio a tres columnas.

NOTA: Con la extensión Minify podemos guardar tras los cambios nuestro archivo CSS como .min.css.

## Jumbotron

**Componentes** → *Jumbotron*

Es un componente ligero y flexible que puede extender toda la ventana gráfica para mostrar llamadas de atención (marketing) en tu sitio.

## Formularios

**Componentes** → *Formularios*

```
<div class="form-group">  
  <label for="">Email:</label>  
  <input type="email" placeholder="Escribe tu email" class="form-control">  
</div>
```

Cada elemento del formulario lo agrupamos en un div con clase **.form-group**

Para que el input ocupe la totalidad de la ventana usamos la clase **.form-control**.

```
<div class="form-group row">  
  <label for="" class="col-2 col-form-label">Email:</label>  
  <input type="email" placeholder="Escribe tu email" class="form-control col-10">  
</div>
```

Podemos hacer limitar el ancho de los elementos del formulario haciendo uso de la rejilla de Bootstrap. Para ello, añadimos al grupo la clase **.row** (fila). La etiqueta será una columna dentro de la fila y el input otra, pudiendo establecer el número de columnas que ocupará (col-2, col-10). La etiqueta label usa otra clase propia para ella que le da estilos de CSS (**.col-form-label**)

## Carrusel

**Componentes** → *Carrusel*

## Cheatsheet

<https://hackerthemes.com/bootstrap-cheatsheet/>

## COPIANDO PLANTILLAS DE EJEMPLO

Desde la sección de Examples de la página oficial de Bootstrap podemos copiar varios ejemplos de código para usarlo como plantillas.

- Abrimos la plantilla de ejemplo que queramos copiar
- Botón derecho sobre la página → Ver código fuente de la página
- Copiamos el contenido del body trasladándolo a nuestro proyecto
- Y copiamos los estilos CSS particulares de ese ejemplo.

## OTROS ELEMENTOS

### Modal

#### Componentes → Modal

En la sección Live demo tenemos un ejemplo del que podemos copiar el código. Verás que primero tenemos un botón y luego el modal.

En nuestro botón tendremos que añadir las propiedades del botón

```
data-toggle="modal" data-target="#exampleModalCenter"
```

siendo data-target el mismo id del modal.

**NOTA:** Si añadimos esas mismas propiedades a una etiqueta <a> que encierre a una imagen también abrirá el modal al hacer clic sobre ella. El id del modal debe corresponderse con el data-target del elemento pulsado.

```
<a href="" data-toggle="modal" data-target="#exampleModalCenter"></a>
```

### Scrollspy

En primer lugar necesitamos un navbar. (Componentes → Navbar)

A continuación sustituimos las opciones de nuestro menú por las que nos muestra pills en Componentes → Navs.

Agregamos el contenido a nuestra web cada sección con un id diferente.

Añadimos estilos al <body> → position: relative;

```
body {  
  position: relative;  
}
```

Añadimos a la etiqueta body las siguientes propiedades:

```
<body data-spy="scroll" data-target="#navbar-example">
```

El identificador usado en data-target debe ser el mismo que el usado en nuestro <nav> (menú).

En cada una de la secciones en href indicamos el identificador de la sección enlazada.

```
<a class="nav-link active" href="#seccion-1">Sección 1</a>
```

## Dos sticky-top menús

Añadimos los siguiente a nuestros estilos CSS:

```
.sticky-top-2 {  
    position: -webkit-sticky;  
    position: sticky;  
    top: 56px;  
    z-index: 1019;  
}
```

Este código lo hemos obtenido del archivo Bootstrap.css.

Indicamos esta clase inventada al segundo menú sticky.

## Tooltips

Básicamente para añadir tooltips a cualquier elemento de nuestra web, botones, imágenes,... tendremos que añadir estas líneas al final de nuestro documento (encima del body).

```
<script>  
    $(function(){  
        $('[data-toggle="tooltip"]').tooltip()  
    })  
</script>
```

Y las siguientes propiedades al elemento donde queremos que aparezca el tooltip.

```
data-toggle="tooltip" data-placement="top" title="Introduce texto"
```

Donde data-placement es la posición y title el texto a mostrar.

## Menú dividido en dos partes con el logo en el centro

En primer lugar dividimos el menú en dos partes.

Copiamos el logo entre los dos menús generados.

Al primero le aplicamos la clase .d-md-none para ocultarlo en pantallas mayores a md.

Al segundo le aplicamos las clases .d-none .d-md-block para ocultarlo en pantallas menores a md.

## Cambio de color de menú al hacer scroll

Vamos agregar unas líneas de JQuery para que detecte el scroll en nuestro sitio web, estas se colocan antes de finalizar el body o bien en un archivo aparte con extensión .js

```
<script>
$(window).scroll(function() {
  if ($("#menu").offset().top > 56) {
    $("#menu").removeClass("bg-dark");
    $("#menu").addClass("bg-info");
  } else {
    $("#menu").removeClass("bg-info");
    $("#menu").addClass("bg-dark");
  }
});
</script>
```

Una pequeña explicación del código:

Estamos abriendo una función que detecta el scroll en nuestra ventana, además tenemos un condicional if el cual dice que... si (if) el top de nuestro sitio web es mayor a 56 px, se sustituye la clase «bg-dark» por la clase «bg-info» del elemento que tenga el identificador «#menu» OJO!!!!.

En caso contrario (else), se elimina la clase «bg-info» y se devuelve la clase «bg-dark».

Cabe señalar que para que esto funcione tienes que tener construido un menú (NavBar) con Bootstrap 4, y que en este caso tenga una clase distinta a «bg-info» y esté identificado con el id #menu.

Para finalizar vamos agregar una transición con Css para que el cambio de color no se realice tan brusco, esto lo coloco en la clase «bg-info» y «bg-dark».

```
<style>
.bg-info, .bg-dark {
  transition: all 1s ease;
}
</style>
```

## Tarjetas con doble imagen

Añadimos nuestras tarjetas dentro de un div.container.

Colocamos las imágenes principales.

Debajo de cada imagen principal añadimos un div con clase text-center para que la imagen esté centrada.

A la imagen secundaria le añadiremos una clase propia, .secundaria con los siguientes estilos: margin-top: -50px; ya que a la imagen secundaria le dimos una dimensión de 100x100.

En nuestra imagen secundaria añadimos la clase de Bootstrap 4 .rounded-circle (círculo) y .img-thumbnail (borde).

Ahora sólo quedaría añadir iconos de redes sociales, añadiendo previamente el link en la cabecera al sitio de donde cogeremos nuestros iconos.

## Sección de testimonios

Para realizar este ejercicio usaremos el Componente → Navs → JavaScript Behavior (Tabs)

1. Creamos un div.container con my-5
2. Añadimos la parte de tab-content obtenida de la página de Bootstrap.
3. Modificamos cada tab pane para que tengan la siguiente estructura:
  - a. div.tab-pane...
    - i. div.row
      1. div.col-4
        - a. img.img-fluid
      2. div.col-8
        - a. Texto
4. Copiamos la parte de los nav-tabs (imágenes pequeñas) de la página de Bootstrap y la pegamos al final de los tab-pane, pero dentro del container principal. Antes de pegarlo crearemos un nuevo div.row y lo incluiremos dentro.
5. A nav-tabs le damos la clase .offset-4 para desplazar 4 columnas y quede a la altura del texto y .col-8 para que ocupe el resto de las 8 columnas.
6. Cambiamos los href de cada ítem con el mismo identificador usado en los tab-pane.
7. Cambiamos el texto de los enlaces por las imágenes pequeñas.
8. Creamos una nueva clase que le aplicaremos al row que incluye a las imágenes pequeñas y a la que le indicaremos que tienen una posición relativa y un margin top negativo para subir las imágenes.
9. Arreglamos la apariencia para pantallas pequeñas con media queries y puntos de interrupción en columnas.