

## **UNIDAD 5**

### **UTILIZACIÓN DE FRAMEWORK – BOOTSTRAP**



# Contenidos

Introducción .....	4
Frameworks responsive.....	5
Bootstrap .....	6
Funcionamiento del diseño adaptable .....	6
Herramientas del navegador para el desarrollador .....	7
Descarga e instalación .....	8
Contenido del directorio de Bootstrap.....	9
Página básica .....	9
Sistema de rejilla .....	11
Elemento contenedor.....	11
Funcionamiento del sistema de rejilla .....	11
Ejemplos: .....	12
Forzar el cambio de fila .....	15
Anidamiento de columnas.....	16
Ordenación de columnas.....	16
Ocultar elementos .....	17
Media Queries .....	17
Componentes responsive .....	19
Alertas.....	19
Colores en los links .....	20
Contenido adicional en alertas.....	23
Visualizar/Ocultar alertas .....	23
Badge .....	24
Badges dependiendo del contenido .....	25
Migas de pan / breadcrumb .....	26
Botones.....	27
Button tags .....	28
Botones con borde .....	28
Tamaños .....	29
Estado activado .....	30
Estado desactivado.....	30
Grupo de botones.....	31

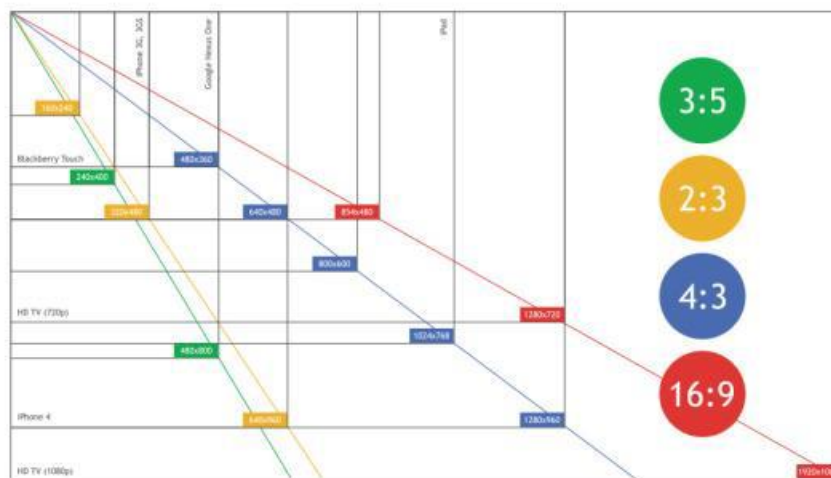
Botones toolbar .....	31
Tamaño .....	33
Anidamiento (nesting) .....	33
Botones verticales .....	34
Tarjetas (cards) .....	34
Tipos de contenidos.....	35
Bloques .....	35
Títulos, texto y enlaces .....	36
Imágenes.....	36
Encabezado y pie en cards .....	38
Tamaño .....	39
Navegación .....	40
Fondos y color de texto .....	41
Grupos de cards.....	43
Grid cards.....	45
Carousel.....	47
Ejemplo de carrusel solo con diapositivas.....	47
Ejemplo de carrusel con controles .....	48
Carrusel con controles y con indicadores.....	49
Collapse.....	50
Dropdown .....	51
Barra de navegación .....	53
Esquemas de color.....	55
Bibliografía.....	56

## Introducción

El diseño web responsive (RWD Responsive Web Design) es una filosofía de diseño y desarrollo cuyo objetivo es adaptar la apariencia de las páginas web al dispositivo que se esté utilizando para visualizarla. Hoy en día las páginas web se visualizan en multitud de tipos de dispositivos como tablets, teléfonos, ebooks, portátiles, PCs, etc. Esta tecnología pretende que con un solo diseño web tengamos una web adecuada para cualquier dispositivo.

### Variabilidad en las resoluciones de pantalla

Antiguamente el desarrollo web se basaba en la resolución estándar de 1024 x 768, pero hoy existe una amplia variedad de resoluciones, no solo en ordenadores de escritorio sino también para tablets y móviles.



A la hora de crear una web, debemos tener en cuenta a qué usuarios va destinados para poder analizar su usabilidad, ya que ya no podemos centrarnos en desarrollar una web pensando que los usuarios van a tener probablemente una única resolución de pantalla.

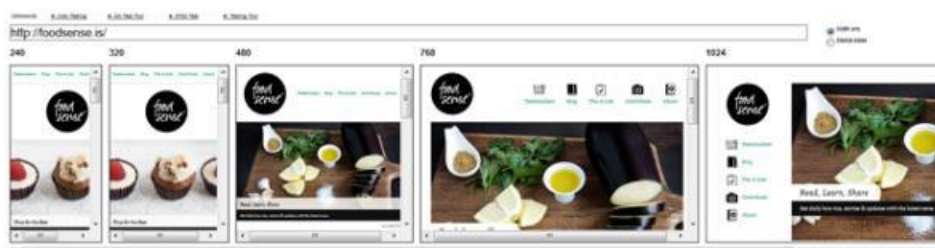
Desde hace ya unos años en el desarrollo web se ha reducido el problema con la compatibilidad entre los distintos navegadores (HTML5/CSS3, y otras en Web-Kit) por el problema de las resoluciones de dispositivos.

En la actualidad ya no es 1024x768 la resolución más utilizada, sino que es 1920x1080 y 1366x768 (<https://gs.statcounter.com/screen-resolution-stats/desktop/worldwide>).

Es fundamental tener en cuenta que en el diseño responsive, al variar tanto las posibles resoluciones en las que se verá nuestra web deberemos mostrar en primer lugar los contenidos más importantes e imprescindibles.

Como ejemplos de sitios creados de forma responsive tenemos:

### Food Sense



## Matt Kersley



## The Boston Globe



## Frameworks responsive

Como se suele decir, ¿Por qué reinventar la rueda si podemos usar algunos de los frameworks que existen en el mercado para ello? Ahorraremos tiempo, usaremos código que está muy probado y unos diseños bastante más bonitos que los que podemos encontrar de forma nativa.

Hoy en día existen en el mercado una amplia variedad de frameworks responsive, algunos de los más utilizados son:

- **Bootstrap** (<http://getbootstrap.com/>): Este framework es uno de los más populares del mercado, el cual fue desarrollado por el equipo de Twitter. Bootstrap ha sido creado para ofrecer la mejor experiencia de usuario tanto a usuarios de PC como a smartphones y tablets. Utiliza el grid responsive de 12 columnas y trae integrado decenas de complementos, plugins de JavaScript, tipografía, controladores de formularios, etc. Además, utiliza el preprocesador de CSS less.
- **Foundation** (<http://foundation.zurb.com/>): Junto con Bootstrap es uno de los frameworks más avanzados que existen en la actualidad. Ha sido desarrollado con Sass, un potente preprocesador de CSS que hace de Foundation un framework personalizable.
- **Skeleton** (<http://getskeleton.com/>): ofrece un grid responsive basado en una resolución de 960px que se ajusta al tamaño de los dispositivos móviles. Tiene poco peso e incluye una colección de archivos CSS y JS para facilitarnos el diseño de nuestra web.

- **HTML5 Boilerplate** (<https://html5boilerplate.com/>): Al igual que los demás nos ofrece un set de utilidades para construir nuestra web responsive de forma rápida y sencilla, con la ventaja de ser uno de los que menos ocupan.

En este curso, nos centraremos en Bootstrap por ser uno de los más completos, utilizados y que mejor funciona.

## Bootstrap

Como hemos comentado, Bootstrap es uno de los frameworks más populares del mercado para la creación de páginas responsive, habiendo sido desarrollado por el equipo de Twitter.

Entre los navegadores soportados está: Chrome, Firefox, Opera, Safari e Internet Explorer (a partir de la versión 8).

Está preparado para que funcione tanto en navegadores de PC, portátiles, así como tablets y Smartphone, todo gracias al sistema de rejilla dividido en columnas para el posicionamiento de los elementos de nuestra web. Además, incorpora otras muchas utilidades y complementos (formularios, botones, barras de navegación, etc.) para simplificar el desarrollo de una web responsive.

## Funcionamiento del diseño adaptable

El diseño responsive se basa en adaptar de forma dinámica el diseño web en función de la resolución de la pantalla del cliente. De modo que nuestra web será adaptada a los dispositivos móviles, sin tener necesidad de tener dos sitios separados, y al mismo tiempo también la adaptaremos para las resoluciones grandes con el objetivo de mejorar la experiencia del usuario.

Antiguamente se hacían dos diseños, uno para móviles y otro para web, sin embargo, el diseño responsive trata de estructurar o adaptar el contenido que ya tienes en el diseño original a otros formatos diferentes: móviles, tablets, y versión de escritorio como bien se muestra en la imagen.

Responsive Web Design nos permite hacer interfaces adaptadas al entorno del usuario mediante estructuras, bloques, columnas e imágenes fluidas gracias a *media-queries* de CSS.

A partir de la versión de CSS 2.1 las hojas de estilo han incluido los *media types*, lo cual nos ha facilitado, por ejemplo, proveer un estilo distinto para imprimir.

```
<link rel="stylesheet" href="core.css" type="text/css" media="screen">
<link rel="stylesheet" href="print.css" type="text/css" media="print">
```

A partir de CSS3, el W3C creó las *media queries*. Una media query nos permite inspeccionar las características del dispositivo que está renderizando nuestra web. Para utilizarlas podemos incorporar una *query* al atributo media de un link a una hoja de estilos:

```
<link rel="stylesheet" href="shetland.css" type="text/css" media="screen and (max-device-width: 480px)">
```

La query contiene dos componentes:

- type:screen, print o all
- una consulta entre paréntesis conteniendo una característica a inspeccionar (max-device-width o min-device-width) seguida por el valor al que apuntamos

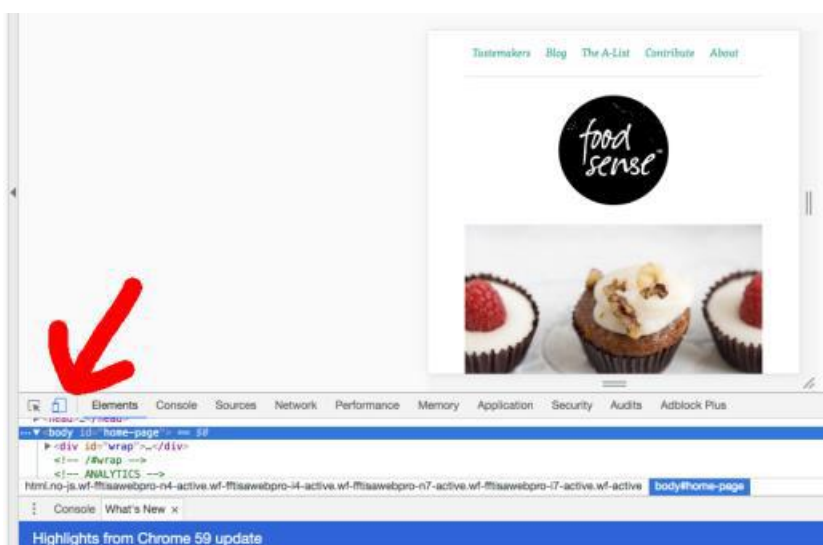
También es posible utilizarlas directamente en nuestro código CSS como parte de una regla:

```
@media only screen and (max-width: 500px) {  
  body {  
    background-color: lightblue;  
  }  
}
```

Para más información sobre las media queries podéis consultarlo aquí:  
<https://www.w3.org/TR/css3-mediaqueries/>

## Herramientas del navegador para el desarrollador

Tanto Firefox como Chrome, tiene instalado por defecto una serie de herramientas que nos permiten entre otras cosas ver la consola de mensajes, inspeccionar el código o ver la secuencia de llamadas al servidor.



Además de estas, también podemos instalar otras más avanzadas como extensiones del navegador como es Firebug.

Mediante estas herramientas podemos probar nuestra página en local y además inspeccionar el código, así como modificar los estilos en tiempo real. Usando el inspector de estas herramientas nos podemos ahorrar mucho tiempo a la hora de realizar pruebas sobre la propia página cargada, ya que de otra forma tendríamos que modificar el código directamente, recargarla y volver a probarla.

## Descarga e instalación

Para descargar Bootstrap (no es un instalador), deberemos descargarnos el zip que se encuentra en la web oficial, descargaremos la última versión disponible (pulsando el botón de *Download*)

<https://getbootstrap.com/docs/5.2/getting-started/download/>

# Download

[View on GitHub](#)

Download Bootstrap to get the compiled CSS and JavaScript, source code, or include it with your favorite package managers like npm, RubyGems, and more.

## Compiled CSS and JS

Download ready-to-use compiled code for **Bootstrap v5.2.3** to easily drop into your project, which includes:

- Compiled and minified CSS bundles (see [CSS files comparison](#))
- Compiled and minified JavaScript plugins (see [JS files comparison](#))

This doesn't include documentation, source files, or any optional JavaScript dependencies like Popper.

Download

Una vez descargado el archivo, lo descomprimos y lo guardamos en el directorio que queramos y/o nos sea fácil acceder a él.

Para poder acceder desde nuestro HTML a él, lo que haremos será enlazar el CSS, y si queremos usar las funcionalidades de javascript, también deberemos enlazar este.

Ejemplo:

```
<link href="bootstrap/bootstrap.min.css" rel="stylesheet">
<script src="javascript/bootstrap.min.js" type="text/javascript"></script>
```

Si no queremos tener descargado el archivo, y preferimos que la web se conecte al servidor de Bootstrap y lo cargue en caché usaremos las etiquetas que lo enlaza:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65" crossorigin="anonymous">
```



```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-kenU1KFdBIE4zVF0s0G1M5b4hcxpyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4"
crossorigin="anonymous"></script>
```

## Contenido del directorio de Bootstrap

En el fichero comprimido tendremos un directorio llamado *css* y otro *js* con el código de Bootstrap para CSS y JavaScript, respectivamente. En versiones anteriores, se incluían un directorio con ejemplos. Pero ahora hay otro enlace en la página de Bootstrap para descargarlos.

Para ver la compatibilidad con los distintos navegadores consulta este enlace:

<https://getbootstrap.com/docs/5.0/getting-started/browsers-devices/>

## Página básica

Bootstrap utiliza elementos HTML y propiedades CSS que requieren el uso del doctype de HTML5 para que funcionen, por lo tanto, es importante añadirlo en TODAS nuestras páginas:

```
<!DOCTYPE html>
<html>
  ...
</html>
```

Ejemplo de una plantilla básica para trabajar con Bootstrap (ver siguiente página):

```

<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap CSS -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
" rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWFspD3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">

    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- Optional JavaScript; choose one of the two! -->

    <!-- Option 1: Bootstrap Bundle with Popper -->
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.mi
n.js" integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>

    <!-- Option 2: Separate Popper and Bootstrap JS -->
    <!--
    <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"
" integrity="sha384-
IQsoLX15PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p"
crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"
integrity="sha384-
cVKIPhGWiC2Al4u+LWgxfKTRICfu0JtXr+EQDz/bglDoEyl4H0zUF0QKbrJ0EcQF"
crossorigin="anonymous"></script>
    -->
  </body>
</html>

```

## Sistema de rejilla

El sistema de rejilla de Bootstrap se basa en rejillas flexibles, las cuales se escalan al tamaño y posición adecuadas dependiendo de la pantalla en la que se esté visualizando la web.

### Elemento contenedor

El sistema de rejilla tiene que ser utilizado dentro de uno de los dos elementos contenedores que tiene Bootstrap: `container` o `container-fluid`. Es importante tener en cuenta que estos elementos se utilizan como raíz de la rejilla y no se podrán anidar unos dentro de otros.

Si lo que queremos es que el contenido de nuestra web aparezca centrado y con un ancho fijo, entonces utilizamos la clase `.container`. Ejemplo:

En cambio, si queremos que ocupe todo el ancho disponible, usamos la clase `.container-fluid`:

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

### Funcionamiento del sistema de rejilla

El Sistema de rejilla está pensado para ayudarnos en la disposición del contenido de nuestra web y su adaptación a las diferentes pantallas. Para ello, tenemos que poner el contenido dentro de celdas o columnas que irán dentro de filas. Cada fila se puede dividir hasta en 12 columnas, somos nosotros los que definimos las columnas para cada tamaño de pantalla.

El funcionamiento de las rejillas es:

- Las columnas deben ir agrupadas dentro de filas (`.row`)
- Las filas (`.row`) se deben colocar dentro de una etiqueta contenedora: `.container` (para ancho fijo) o `.container-fluid` (para poder ocupar todo el ancho), esto permitirá alinear las celdas y asignarles el espacio correcto.
- El contenido se debe disponer dentro de columnas o celdas, las cuales deben de ser el único hijo posible de las filas (`.row`), que a su vez será el único hijo posible del contenedor (`.container` ó `.container-fluid`)
- Al seguir este orden el sistema de rejilla funcionará creando el espaciado interior y los márgenes apropiados dependiendo de las dimensiones de la pantalla.
- Cada fila se puede dividir hasta un máximo de 12 columnas, pero somos nosotros los que tendremos que definir el número de columnas en el que queremos dividir cada fila y su ancho para cada tamaño de pantalla.
- Si el tamaño total de las columnas de una fila excede de 12, el sobrante se colocará en la siguiente fila.

- El tamaño de las columnas se especificará con clases CSS que Bootstrap define para cada tamaño de pantalla, por ejemplo –col-md-XX donde XX es el tamaño de la columna, que podrá tomar valores entre 1 y 12

En la siguiente tabla se recoge un resumen del funcionamiento de la rejilla de Bootstrap dependiendo del tamaño de la pantalla y las clases CSS:

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
.container	100%	540px	720px	960px	1140px	1320px
.container-sm	100%	540px	720px	960px	1140px	1320px
.container-md	100%	100%	720px	960px	1140px	1320px
.container-lg	100%	100%	100%	960px	1140px	1320px
.container-xl	100%	100%	100%	100%	1140px	1320px
.container-xxl	100%	100%	100%	100%	100%	1320px
.container-fluid	100%	100%	100%	100%	100%	100%

Es importante tener en cuenta que, al definir estas clases, no sólo se aplican para ese tamaño de pantalla, sino que para las superiores también. Por ejemplo, si no aplicamos clase para tamaños de pantalla medianos y grandes pero sí para pequeños (.col-sm-), se usará esta. Si los defines, se sobrescriben y pasan a utilizar los definidos para dichos tamaños.

## Ejemplos:

### TAMAÑO DE COLUMNAS SOLO PARA PANTALLAS DE ESCRITORIO:

Ejemplo de 3 filas. La primera de 2 columnas de tamaño desigual, la segunda 3 columnas del mismo tamaño y la tercera fila 2 columnas del mismo tamaño.

**Código:**

```
<div class="row">
  <div class="col-md-8">.col-md-8</div>
  <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
  <div class="col-md-6">.col-md-6</div>
  <div class="col-md-6">.col-md-6</div>
</div>
```

## Resultado:

.col-md-8		.col-md-4
.col-md-4	.col-md-4	.col-md-4
.col-md-6		.col-md-6

Como las columnas se han especificado únicamente mediante las clases `.col-md-*` esto se aplicará tanto en pantallas de escritorio medianas y grandes, pero no para los tamaños pequeños (tablets y móviles). En estos dos últimos casos, las columnas se ampliarán utilizando todo el ancho por lo que se mostrarán de la siguiente forma:

.col-md-8	
.col-md-4	
.col-md-4	
.col-md-4	
.col-md-4	
.col-md-6	
.col-md-6	

## TAMAÑO PARA MÓVIL Y ESCRITORIO:

Si no queremos que las columnas se apilen en tamaños pequeños, podemos indicar también la disposición que queremos para esos casos usando las clases: `.col-xs-*`, además de las que ya teníamos con `.col-md-*`. Ejemplo:

### Código:

```
<!-- En pantallas pequeñas aparecerá una columna que ocupará todo el ancho y otra que ocupará la mitad de la pantalla -->
<div class="row">
  <div class="col-xs-12 col-md-8">.col-xs-12 .col-md-8</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
</div>

<!-- En pantallas pequeñas se indica que ocupe cada columna la mitad del ancho disponible. En pantallas medianas ocuparán 1/3 -->
<div class="row">
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
</div>

<!-- Como no se indica el tamaño para pantallas grandes las columnas siempre ocuparán el 50% -->
<div class="row">
  <div class="col-xs-6">.col-xs-6</div>
  <div class="col-xs-6">.col-xs-6</div>
</div>
```

**Resultado** para pantallas md y lg (medianas y grandes respectivamente).

.col-xs-12 .col-md-8		.col-xs-6 .col-md-4
.col-xs-6 .col-md-4	.col-xs-6 .col-md-4	.col-xs-6 .col-md-4
.col-xs-6	.col-xs-6	

En el caso de pantallas pequeñas, las columnas se verían:

.col-xs-12 .col-md-8		
.col-xs-6 .col-md-4		
.col-xs-6 .col-md-4	.col-xs-6 .col-md-4	
.col-xs-6 .col-md-4		
.col-xs-6	.col-xs-6	

### TAMAÑO PARA MÓVIL, TABLET Y ESCRITORIO:

Para tener un mayor control, podemos especificar el tamaño de las columnas para las tablets, con las clases: `.col-sm-*`. Ejemplo:

**Código:**

```
<div class="row">
  <div class="col-xs-12 col-sm-6 col-md-8">.col-xs-12 .col-sm-6 .col-md-8</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
</div>
<div class="row">
  <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
  <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
  <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
</div>
```

**Resultado** en pantallas medianas y grandes:

.col-xs-12 .col-sm-6 .col-md-8		.col-xs-6 .col-md-4
.col-xs-6 .col-sm-4	.col-xs-6 .col-sm-4	.col-xs-6 .col-sm-4

**Resultado** en pantallas de tipo tablet:

.col-xs-12 .col-sm-6 .col-md-8		.col-xs-6 .col-md-4
.col-xs-6 .col-sm-4	.col-xs-6 .col-sm-4	.col-xs-6 .col-sm-4

**Resultado** en pantallas pequeñas (xs):

.col-xs-12 .col-sm-6 .col-md-8	
.col-xs-6 .col-md-4	
.col-xs-6 .col-sm-4	.col-xs-6 .col-sm-4
.col-xs-6 .col-sm-4	

Para consultar más ejemplos y/o ayuda:

<https://getbootstrap.com/docs/5.0/examples/grid/#containers>

## Forzar el cambio de fila

Si usamos la clase `.clearfix` podemos forzar el cambio de fila cuando queramos. Esta clase es realmente útil cuando las filas tienen un alto distinto, o para forzar el cambio de fila en determinados tamaños de pantalla. Por ejemplo, si añadimos la clase `.visible-xs-block` solo se producirá ese cambio de fila para pantallas pequeñas.

Ejemplo de dos columnas con distinto alto, de forma que la primera fila se coloca en una posición incorrecta al no usar el `.clearfix`:

**Código:**

```
<div class="row">
  <div class="col-xs-6 col-sm-3">
    .col-xs-6 .col-sm-3<br />Resize your viewport or check it out on your phone
  <br />
    for an example
  </div>
  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>
  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>
  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>
</div>
```

**Resultado:**

.col-xs-6 .col-sm-3 Resize your viewport or check it out on your phone for an example.	.col-xs-6 .col-sm-3
.col-xs-6 .col-sm-3	.col-xs-6 .col-sm-3

## Ejemplo con `.clearfix`

**Código:**

```
<div class="row">
  <div class="col-xs-6 col-sm-3">
    .col-xs-6 .col-sm-3<br />Resize your viewport or check it out on your phone
  <br />
    for an example
  </div>
```

```

<div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>
<div class="clearfix visible-xs-block"></div>
<div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>
<div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>
</div>

```

**Resultado:**

.col-xs-6 .col-sm-3 Resize your viewport or check it out on your phone for an example.	.col-xs-6 .col-sm-3
.col-xs-6 .col-sm-3	.col-xs-6 .col-sm-3

## Anidamiento de columnas

Una de las características que tiene Bootstrap también, es que nos permite anidar columnas. Por ejemplo, si quisiéramos que en una columna de tamaño 9, crear una nueva fila y dividirla como queramos (funciona igual, cada fila tiene 12 celdas). Ejemplo:

**Código:**

```

<div class="col-sm-9">
  Level 1: .col-sm-9
  <div class="row">
    <div class="col-xs-8 col-sm-6">Level 2: .col-xs-8 .col-sm-6</div>
    <div class="col-xs-4 col-sm-6">Level 2: .col-xs-4 .col-sm-6</div>
  </div>
</div>

```

**Resultado:**

Level 1: .col-sm-9	
Level 2: .col-xs-8 .col-sm-6	Level 2: .col-xs-4 .col-sm-6

## Ordenación de columnas

Podemos usar la clase `.order-` para controlar de forma visual el orden de nuestro contenido. Estas clases son responsive, por tanto podemos establecer el orden teniendo en cuenta que el grid tiene 12 columnas.

**Código:**



```
<div class="container">
  <div class="row">
    <div class="col">First, but unordered</div>
    <div class="col order-12">Second, but last</div>
    <div class="col order-1">Third, but first</div>
  </div>
</div>
```

**Resultado:**

First, but unordered	Third, but first	Second, but last
----------------------	------------------	------------------

Más ejemplos de columnas en <https://getbootstrap.com/docs/5.0/layout/columns/>

## Ocultar elementos

Para no tener que rediseñar nuestra página, podemos usar un diseño responsive para ocultar o mostrar ciertos elementos.

Si queremos ocultar un elemento podemos usar la clase `.d-none` o `.d-{sm,md,lg,xl,xxl}-none`.

Si lo que pretendemos es mostrar algún elemento solo para algunos tamaños, podemos ocultarlo para todos con la clase `.d-*-none` y mostrarlo con `.d-*-*`.

Ejemplo:

```
<div class="d-lg-none">Oculto solo en pantallas grandes</div>
<div class="d-none d-lg-block">Ocultar solamente en pantallas más pequeñas que la grande</div>
```

Más información en <https://getbootstrap.com/docs/5.0/utilities/display/>

## Media Queries

En la mayoría de los casos con las clases que vienen en Bootstrap nos es suficiente para diseñar nuestra web, sin embargo, en algunas situaciones a veces queremos modificar cierto comportamiento como puede ser para aplicar colores, alineación, etc. que cambie dependiendo del tamaño de la pantalla. Será en estos casos cuando utilicemos los media queries.

Una media query se define de la siguiente forma:

```
@media (min-width: TAMAÑO-EN-PÍXELES) {
  /* Estos estilos sólo se aplicarán a partir del tamaño indicado */
}
```

En este caso, los estilos que estén dentro de esta media query se aplicarán sólo a partir del tamaño indicado, pero también le podemos indicar el rango de tamaño al que aplicarse. Ejemplo:

```
@media (max-width: TAMAÑO-EN-PÍXELES) {  
  /* Estos estilos solo se aplicarán HASTA el tamaño indicado */  
}  
@media (min-width: TAMAÑO-EN-PÍXELES) and (max-width: TAMAÑO-EN-PÍXELES)  
{  
  /* Solo se aplicarán ENTRE los tamaños indicados */  
}
```

### Ejemplos de uso

Para pantallas xs, le aplicaremos un color de fondo llamado (myBackground) que será rojo, y para el resto de tamaños el fondo será verde:

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

Otro ejemplo podría ser la alineación del texto que se le aplicará cuando estemos en pantallas de escritorio:

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

Estos estilos son un ejemplo básico de lo que podemos hacer complementando el código de Bootstrap, de modo que podemos llegar a hacer cosas bastante avanzadas y personalizar la web dependiendo del tamaño del dispositivo. Otros ejemplos comunes son:

- Cambiar el tamaño y posición de una imagen entre dispositivos móviles y de escritorio
- Cambiar la posición de un elemento.

- Cambiar el tamaño de letra, fuente o color dependiendo del tamaño de la pantalla.
- Aplicar combinaciones de estilos para que su apariencia se ajuste a los elementos que contenga.
- Etc.

## Componentes responsive

Además de la rejilla, Bootstrap incluye un conjunto de componentes para facilitarnos aún más el diseño de nuestra web. Estos componentes son:

- Alertas
- Badge
- Migas de pan
- Botones
- Grupos de botones
- Tarjetas
- Carousel
- Collapse
- Desplegables
- Formularios
- Grupo de input

## Alertas

Las alertas que nos ofrece Bootstrap están disponibles para textos(.alert), así como para cuando pulsemos un botón Ejemplos básicos de alertas:

```
<div class="alert alert-primary" role="alert">
  This is a primary alert—check it out!
</div>
<div class="alert alert-secondary" role="alert">
  This is a secondary alert—check it out!
</div>
<div class="alert alert-success" role="alert">
  This is a success alert—check it out!
</div>
<div class="alert alert-danger" role="alert">
  This is a danger alert—check it out!
</div>
<div class="alert alert-warning" role="alert">
  This is a warning alert—check it out!
</div>
<div class="alert alert-info" role="alert">
```

Accesibilidad, máquinas y navegadores invidentes p.ej

```
This is a info alert—check it out!  
</div>  
<div class="alert alert-light" role="alert">  
  This is a light alert—check it out!  
</div>  
<div class="alert alert-dark" role="alert">  
  This is a dark alert—check it out!  
</div>
```

#### Resultado:

This is a primary alert—check it out!

This is a secondary alert—check it out!

This is a success alert—check it out!

This is a danger alert—check it out!

This is a warning alert—check it out!

This is a info alert—check it out!

This is a light alert—check it out!

This is a dark alert—check it out!

## Colores en los links

Usaremos la clase `.alert-link` para enlazar un hipervínculo con una alerta. Ejemplo:

### Código:

```
<div class="alert alert-primary" role="alert">
  This is a primary alert with
  <a href="#" class="alert-link">an example link</a>. Give it a click if
you like.
</div>
<div class="alert alert-secondary" role="alert">
  This is a secondary alert with
  <a href="#" class="alert-link">an example link</a>. Give it a click if
you like.
</div>
<div class="alert alert-success" role="alert">
  This is a success alert with
  <a href="#" class="alert-link">an example link</a>. Give it a click if
you like.
</div>
<div class="alert alert-danger" role="alert">
  This is a danger alert with
  <a href="#" class="alert-link">an example link</a>. Give it a click if
you like.
</div>
<div class="alert alert-warning" role="alert">
  This is a warning alert with
  <a href="#" class="alert-link">an example link</a>. Give it a click if
you like.
</div>
<div class="alert alert-info" role="alert">
  This is a info alert with <a href="#" class="alert-link">an example
link</a>. Give it a click if you like.
</div>
<div class="alert alert-light" role="alert">
  This is a light alert with <a href="#" class="alert-link">an example
link</a>. Give it a click if you like.
</div>
<div class="alert alert-dark" role="alert">
  This is a dark alert with <a href="#" class="alert-link">an example
link</a>. Give it a click if you like.
</div>
```

### Resultado:

This is a primary alert with **an example link**. Give it a click if you like.

This is a secondary alert with **an example link**. Give it a click if you like.

This is a success alert with **an example link**. Give it a click if you like.

This is a danger alert with **an example link**. Give it a click if you like.

This is a warning alert with **an example link**. Give it a click if you like.

This is a info alert with **an example link**. Give it a click if you like.

This is a light alert with **an example link**. Give it a click if you like.

This is a dark alert with **an example link**. Give it a click if you like.

## Contenido adicional en alertas

Las alertas pueden tener elementos HTML adicionales como cabeceras y párrafos.

Ejemplo:

Código:

```
<div class="alert alert-success" role="alert">
  <h4 class="alert-heading">Well done!</h4>
  <p>
    Aww yeah, you successfully read this important alert message. This
example
    text is going to run a bit longer so that you can see how spacing within
an alert works with this kind of content.
  </p>
  <hr />
  <p class="mb-0">
    Whenever you need to, be sure to use margin utilities to keep things nice
and tidy.
  </p>
</div>
```

Resultado:

### Well done!

Aww yeah, you successfully read this important alert message. This example text is going to run a bit longer so that you can see how spacing within an alert works with this kind of content.

---

Whenever you need to, be sure to use margin utilities to keep things nice and tidy.

## Visualizar/Ocultar alertas

Usando el plugin de alertas de JavaScript, es posible ocultar alguna alerta de forma inline, veamos cómo se usa:

- Asegúrate que has cargado el plugin de alertas, o compilado Bootstrap JavaScript.
- Añade un botón dismiss y la clase .alert-dismissible, el cual añade un padding extra a la derecha y lo posiciona al botón con la clase.close
- En el botón a ocultar, añade el atributo data-dismiss="alert", el cual lanza la funcionalidad de JavaScript. Asegurate de usar el elemento <button> con el comportamiento adecuado para todos los dispositivos
- Para animar las alertas cuando se oculten, añadiremos las clases .fade y .show

Código:

```
<div class="alert alert-warning alert-dismissible fade show" role="alert">
  <button type="button" class="close" data-dismiss="alert" aria-label="Close">
    <span aria-hidden="true">&times;</span>
  </button>
  <strong>Holy guacamole!</strong> You should check in on some of those fields
  below.
</div>
```

Accesibilidad, cuando la X no dice nada y queremos indicar que sería un close

In dica si no es visible o se ignora en tecnologia de asistencia

Resultado:

**Holy guacamole!** You should check in on some of those fields below. ×

## Badge

Los badges son pequeñas etiquetas que se adaptan para añadir un texto al lado de otro. Un ejemplo de badges puede ser utilizado como para resaltar algo nuevo, al lado de una cabecera. Ejemplo:

Código:

```
<h1>Example heading <span class="badge badge-secondary">New</span></h1>
<h2>Example heading <span class="badge badge-secondary">New</span></h2>
<h3>Example heading <span class="badge badge-secondary">New</span></h3>
<h4>Example heading <span class="badge badge-secondary">New</span></h4>
<h5>Example heading <span class="badge badge-secondary">New</span></h5>
<h6>Example heading <span class="badge badge-secondary">New</span></h6>
```

Resultado:



Example heading **New**

Example heading **New**

Example heading **New**

Example heading **New**

Example heading **New**

Example heading **New**

## Badges dependiendo del contenido

Código:

```
<span class="badge badge-primary">Primary</span>
<span class="badge badge-secondary">Secondary</span>
<span class="badge badge-success">Success</span>
<span class="badge badge-danger">Danger</span>
<span class="badge badge-warning">Warning</span>
<span class="badge badge-info">Info</span>
<span class="badge badge-light">Light</span>
<span class="badge badge-dark">Dark</span>
```

Resultado:

**Primary** **Secondary** **Success** **Danger** **Warning** **Info** **Light** **Dark**

Si queremos que esos badges sean más redonditos, usaremos la clase `.badge-pill`.  
Ejemplo:

Código:

```
<span class="badge badge-pill badge-primary">Primary</span>
<span class="badge badge-pill badge-secondary">Secondary</span>
<span class="badge badge-pill badge-success">Success</span>
<span class="badge badge-pill badge-danger">Danger</span>
<span class="badge badge-pill badge-warning">Warning</span>
<span class="badge badge-pill badge-info">Info</span>
<span class="badge badge-pill badge-light">Light</span>
```

```
<span class="badge badge-pill badge-dark">Dark</span>
```

Resultado:



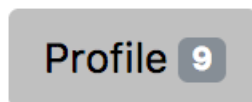
Los badges también pueden ser utilizados como botones o enlaces con un contador.

Ejemplo:

Código:

```
<button class="btn">
  Profile <span class="badge badge-secondary">9</span>
  <span class="sr-only">unread messages</span>
</button>
```

Resultado:



## Migas de pan / breadcrumb

Breadcrumb o migas de pan, nos indica dónde está la página, los separadores se añaden de forma automática con CSS.

Código:

```
<ol class="breadcrumb">
  <li class="breadcrumb-item active">Home</li>
</ol>
<ol class="breadcrumb">
  <li class="breadcrumb-item"><a href="#">Home</a></li>
  <li class="breadcrumb-item active">Library</li>
</ol>
<ol class="breadcrumb">
  <li class="breadcrumb-item"><a href="#">Home</a></li>
  <li class="breadcrumb-item"><a href="#">Library</a></li>
  <li class="breadcrumb-item active">Data</li>
</ol>
```

Resultado:

Home

Home / Library

Home / Library / Data

De forma similar, el breadcrumb lo podemos utilizar con el elemento nav. Ejemplo:

Código:

```
<nav class="breadcrumb">
  <a class="breadcrumb-item" href="#">Home</a>
  <a class="breadcrumb-item" href="#">Library</a>
  <a class="breadcrumb-item" href="#">Data</a>
  <span class="breadcrumb-item active">Bootstrap</span>
</nav>
```

Resultado:

Home / Library / Datos / Bootstrap

## Botones

Bootstrap tiene seis estilos de botones predefinidos, cada uno con su propia utilidad.

Código:

```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
```

```
<button type="button" class="btn btn-dark">Dark</button>
<button type="button" class="btn btn-link">Link</button>
```

Resultado:



## Button tags

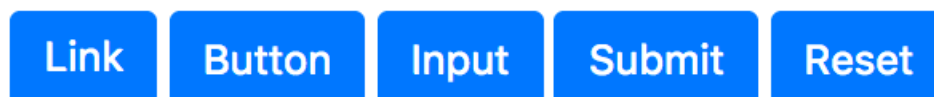
Las clases .btn se usan con el elemento <button>. Sin embargo, podemos usar estas clases con el elemento HTML <a> o <input> (aunque algunos navegadores aplican una ligera diferencia de renderizado).

Cuando usamos una clase .btn en un elemento <a> estamos usando un hipervínculo en la página más que un enlace hacia una nueva página o sección, este enlace debería de tener un role="button" para saber el propósito que tiene por ejemplo a la hora de leer información en una pantalla para sordos. Ejemplo:

Código:

```
<a class="btn btn-primary" href="#" role="button">Link</a>
<button class="btn btn-primary" type="submit">Button</button>
<input class="btn btn-primary" type="button" value="Input">
<input class="btn btn-primary" type="submit" value="Submit">
<input class="btn btn-primary" type="reset" value="Reset">
```

Resultado:



## Botones con borde

Estos botones tienen la peculiaridad de que el fondo es blanco, se mantiene el color del borde del botón y las letras tienen el color del tipo de botón indicado. Ejemplo:

Código:

```
<button type="button" class="btn btn-outline-primary">Primary</button>
<button type="button" class="btn btn-outline-secondary">Secondary</button>
<button type="button" class="btn btn-outline-success">Success</button>
<button type="button" class="btn btn-outline-danger">Danger</button>
<button type="button" class="btn btn-outline-warning">Warning</button>
```

```
<button type="button" class="btn btn-outline-info">Info</button>
<button type="button" class="btn btn-outline-light">Light</button>
<button type="button" class="btn btn-outline-dark">Dark</button>
```

Resultado:



## Tamaños

Para hacer los botones más grandes o más pequeños usaremos las clases: `.btn-lg` o `.btn-sm` respectivamente. Ejemplo:

Código para botones grandes:

```
<button type="button" class="btn btn-primary btn-lg">Large button</button>
<button type="button" class="btn btn-secondary btn-lg">Large button</button>
```

Resultado:



Código para botones pequeños:

```
<button type="button" class="btn btn-primary btn-sm">Small
button</button>
<button type="button" class="btn btn-secondary btn-sm">Small
button</button>
```

Resultado:



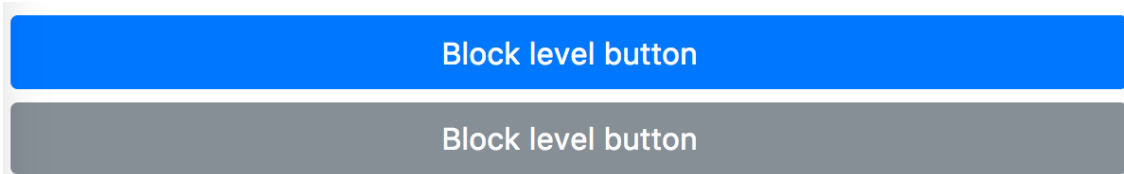
Podemos crear también botones en bloque, es decir que ocupan todo el ancho del padre. Ejemplo:

Código:

```
<button type="button" class="btn btn-primary btn-lg btn-block">Block
level button</button>
```

```
<button type="button" class="btn btn-secondary btn-lg btn-block">Block  
level button</button>
```

Resultado:



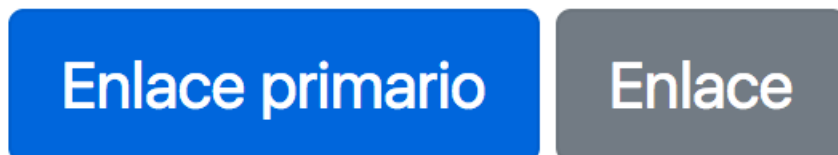
## Estado activado

Estos botones se caracterizan porque tienen el fondo más oscuro, dando la sensación de haber sido pulsados, para ello se añade el atributo `.active` junto con `aria-pressed="true"`.

Código:

```
<a href="#" class="btn btn-primary btn-lg active" role="button"  
aria-pressed="true">Enlace primario</a>  
<a href="#" class="btn btn-secondary btn-lg active" role="button"  
aria-pressed="true">Enlace</a>
```

Resultado:



## Estado desactivado

Hace que los botones parezcan inactivo, para ello se le añade el atributo `disabled` al elemento `<button>`. Ejemplo:

Código:

```
<button type="button" class="btn btn-lg btn-primary" disabled>Botón  
primario</button>  
<button type="button" class="btn btn-secondary btn-lg"  
disabled>Botón</button>
```

Resultado:

Botón primario

Botón

Para ver más funcionalidades con los botones consultar:

<https://getbootstrap.com/docs/5.2/components/buttons/>

## Grupo de botones

Bootstrap nos ofrece diferentes formas de agrupar botones, veamos un ejemplo básico usando las clases `.btn` y `.btn-group`.

Código:

```
<div class="btn-group" role="group" aria-label="Basic example">
  <button type="button" class="btn btn-secondary">Izquierda</button>
  <button type="button" class="btn btn-secondary">Medio</button>
  <button type="button" class="btn btn-secondary">Derecha</button>
</div>
```

Resultado:

Izquierda

Medio

Derecha

## Botones toolbar

Código:

```
<div class="btn-toolbar" role="toolbar" aria-label="Toolbar with button groups">
  <div class="btn-group mr-2" role="group" aria-label="First group">
    <button type="button" class="btn btn-secondary">1</button>
    <button type="button" class="btn btn-secondary">2</button>
    <button type="button" class="btn btn-secondary">3</button>
    <button type="button" class="btn btn-secondary">4</button>
  </div>
  <div class="btn-group mr-2" role="group" aria-label="Second group">
    <button type="button" class="btn btn-secondary">5</button>
    <button type="button" class="btn btn-secondary">6</button>
    <button type="button" class="btn btn-secondary">7</button>
  </div>
  <div class="btn-group" role="group" aria-label="Third group">
```

```

        <button type="button" class="btn btn-secondary">8</button>
    </div>
</div>

```

Resultado:



Podemos también establecer el grupo de botones que quereamos en nuestro toolbar.

Ejemplo:

Código:

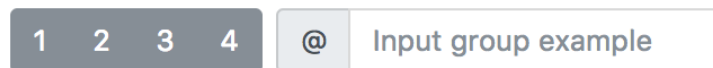
```

<div class="btn-toolbar mb-3" role="toolbar" aria-label="Toolbar with button groups">
  <div class="btn-group mr-2" role="group" aria-label="First group">
    <button type="button" class="btn btn-secondary">1</button>
    <button type="button" class="btn btn-secondary">2</button>
    <button type="button" class="btn btn-secondary">3</button>
    <button type="button" class="btn btn-secondary">4</button>
  </div>
  <div class="input-group">
    <span class="input-group-addon" id="btnGroupAddon">@</span>
    <input type="text" class="form-control" placeholder="Input group example"
      aria-label="Input group example" aria-describedby="btnGroupAddon"/>
  </div>
</div>
<div class="btn-toolbar justify-content-between" role="toolbar" aria-label="Toolbar
with button groups">
  <div class="btn-group" role="group" aria-label="First group">
    <button type="button" class="btn btn-secondary">1</button>
    <button type="button" class="btn btn-secondary">2</button>
    <button type="button" class="btn btn-secondary">3</button>
    <button type="button" class="btn btn-secondary">4</button>
  </div>
  <div class="input-group">
    <span class="input-group-addon" id="btnGroupAddon2">@</span>
    <input type="text" class="form-control" placeholder="Input group example"
      aria-label="Input group example" aria-describedby="btnGroupAddon2"/>
  </div>
</div>

```

Resultado:





## Tamaño

Al igual que los botones normales, al grupo de botones le podemos establecer un tamaño, será: `.btn-group-lg` o `.btn-group-sm`, dependiendo de si los queremos hacer grandes o pequeños. Ejemplo:

Código:

```
<div class="btn-group btn-group-lg" role="group" aria-label="...">...</div>
<div class="btn-group" role="group" aria-label="...">...</div>
<div class="btn-group btn-group-sm" role="group" aria-label="...">...</div>
```

Resultado:



## Anidamiento (nesting)

Lo usaremos cuando queremos tener un grupo de botones mezclado con un desplegable. Ejemplo:

Código:

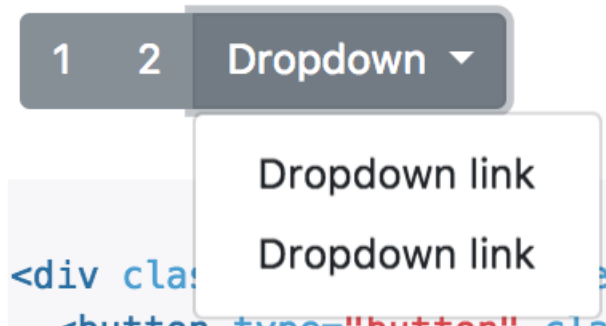
```
<div class="btn-group" role="group" aria-label="Button group with nested dropdown">
  <button type="button" class="btn btn-secondary">1</button>
  <button type="button" class="btn btn-secondary">2</button>
  <div class="btn-group" role="group">
    <button id="btnGroupDrop1" type="button" class="btn btn-secondary dropdown-
toggle"
      data-toggle="dropdown" aria-haspopup="true" ariaexpanded="false">
      Dropdown
    </button>
    <div class="dropdown-menu" aria-labelledby="btnGroupDrop1">
```

```

        <a class="dropdown-item" href="#">Dropdown link</a>
        <a class="dropdown-item" href="#">Dropdown link</a>
    </div>
</div>
</div>

```

Resultado:



## Botones verticales

Si queremos que alguno de los ejemplos anteriores en lugar de tener los botones en horizontal (por defecto) estén en vertical, tendríamos que agruparlos usando la clase: `.btn-group-vertical`.

Ejemplo:

Código:

```

<div class="btn-group-vertical">
    ...
</div>

```

## Tarjetas (cards)

Una card es un contenedor flexible y estensible. Incluye opciones para los encabezados y los pies, una amplia variedad de contenido, colores de fondo y numerosas opciones más de visualización. Las cards están hechas con un pequeño maquetado pero todo se puede personalizar. Están contruidos con flexbox, de forma que ofrecen una fácil alineación y se mezclan bien con otros componentes de Bootstrap. Veamos un ejemplo básico de un card.

Código:

```

<div class="card" style="width: 20rem">
    
    <div class="card-body">
        <h4 class="card-title">Nadal, Nº 1</h4>
        <p class="card-text">

```

Ejemplo rápido de cómo crear nuestra propia card, añadiendo una imagen, un título y algo de contenido

```
</p>  
<a href="#" class="btn btn-primary">Seguir leyendo</a>  
</div>  
</div>
```

Resultado:



## Tipos de contenidos

Las cards tienen una amplia variedad de contenido, incluyendo imágenes, texto, listados, enlaces y mucho más, veamos algún ejemplo:

### Bloques

Código:

```
<div class="card">  
  <div class="card-body">  
    Ejemplo de texto con bordes en la card.  
  </div>  
</div>
```

Resultado:

Ejemplo de texto con bordes en la card.

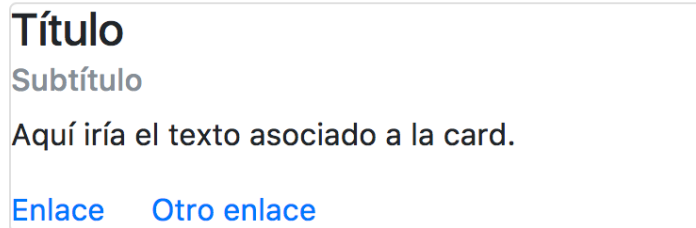
## Títulos, texto y enlaces

Para añadirle un título a la card añadiremos al `<h*>` la clase `.cardtitle`, del mismo modo para los enlaces `<a>`, usaremos: `.card-link`. Para los subtítulos usaremos en los `<h*>` la clase `.card-subtitle`, y si el subtítulo se encuentra dentro de la card y queremos alinearlos con el título usaremos: `.card-block`. Ejemplo:

Código:

```
<div class="card" style="width: 20rem">
  <div class="card-body">
    <h4 class="card-title">Título</h4>
    <h6 class="card-subtitle mb-2 text-muted">Subtítulo</h6>
    <p class="card-text">Aquí iría el texto asociado a la card</p>
    <a href="#" class="card-link">Enlace</a>
    <a href="#" class="card-link">Otro enlace</a>
  </div>
</div>
```

Resultado:



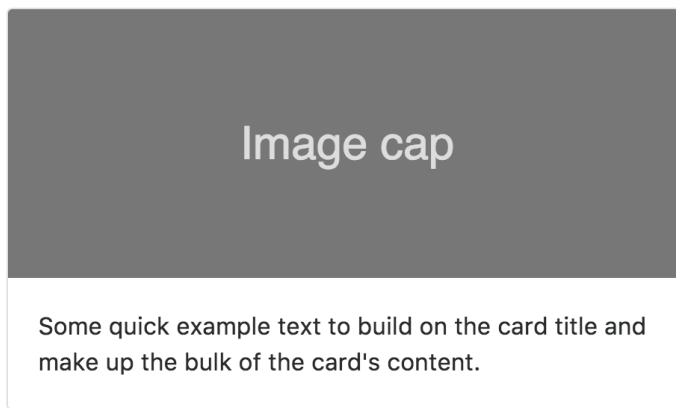
## Imágenes

Con la clase `.card-img-top` nos sitúa la imagen en la parte de arriba de la card. Con la clase `.card-text` podemos añadir texto a la imagen. Ejemplo:

Código:

```
<div class="card" style="width: 20rem">
  
  <div class="card-body">
    <p class="card-text">
      Some quick example text to build on the card title and make up the bulk of
      the card's content.
    </p>
  </div>
</div>
```

Resultado:

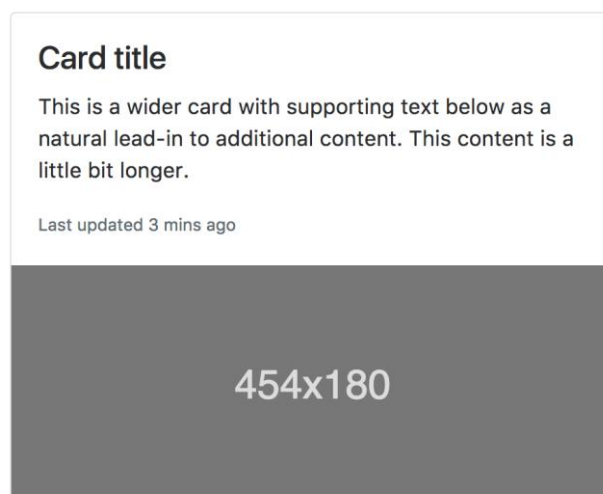


Las imágenes, no tienen por qué ir en la cabecera, al igual que los encabezados y pies, éstas las podemos situar donde queramos, veamos un ejemplo:

Código:

```
<div class="card">
  <div class="card-block">
    <h4 class="card-title">Card title</h4>
    <p class="card-text">
      This is a wider card with supporting text below as a natural lead-in to
      additional content. This content is a little bit longer.
    </p>
    <p class="card-text">
      <small class="text-muted">Last updated 3 mins ago</small>
    </p>
  </div>
  
</div>
```

Resultado:



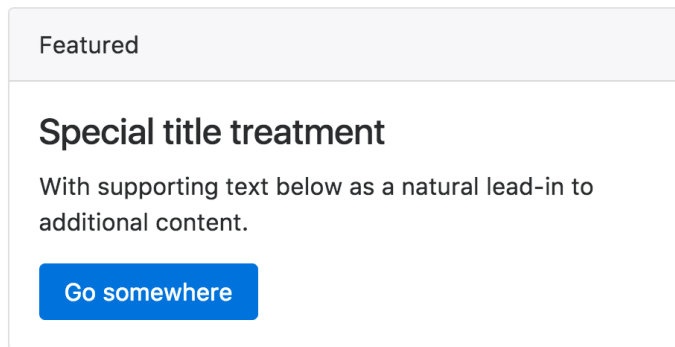
## Encabezado y pie en cards

### Encabezado

Código:

```
<div class="card">
  <div class="card-header">Featured</div>
  <div class="card-block">
    <h4 class="card-title">Special title treatment</h4>
    <p class="card-text">
      With supporting text below as a natural lead-in to additional content.
    </p>
    <a href="#" class="btn btn-primary">Go somewhere</a>
  </div>
</div>
```

Resultado:

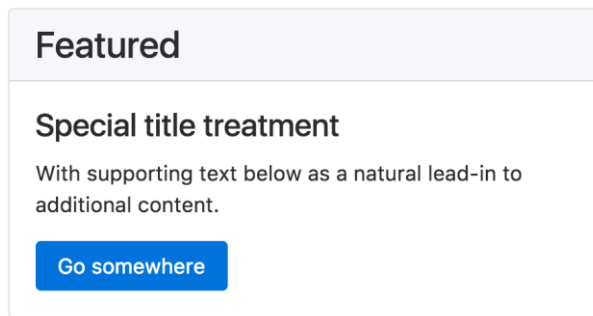


### Encabezados con estilo

Código:

```
<div class="card">
  <h3 class="card-header">Featured</h3>
  <div class="card-block">
    <h4 class="card-title">Special title treatment</h4>
    <p class="card-text">
      With supporting text below as a natural lead-in to additional content.
    </p>
    <a href="#" class="btn btn-primary">Go somewhere</a>
  </div>
</div>
```

Resultado:



## Tamaño

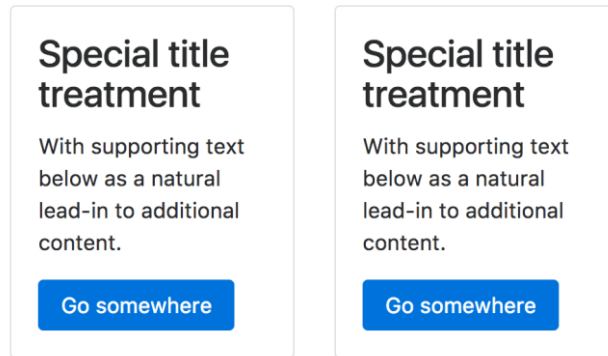
Si no se indica nada, una card ocupará el 100%, pero se puede modificar en función de nuestras necesidades con CSS, grid, grid Sass mixins o utilidades.

**Usando grid**, de forma que las agrupemos en columnas y filas.

Código:

```
<div class="row">
  <div class="col-sm-6">
    <div class="card">
      <div class="card-block">
        <h3 class="card-title">Special title treatment</h3>
        <p class="card-text">
          With supporting text below as a natural lead-in to additional content.
        </p>
        <a href="#" class="btn btn-primary">Go somewhere</a>
      </div>
    </div>
  </div>
  <div class="col-sm-6">
    <div class="card">
      <div class="card-block">
        <h3 class="card-title">Special title treatment</h3>
        <p class="card-text">
          With supporting text below as a natural lead-in to additional content.
        </p>
        <a href="#" class="btn btn-primary">Go somewhere</a>
      </div>
    </div>
  </div>
</div>
```

Resultado:



## Navegación

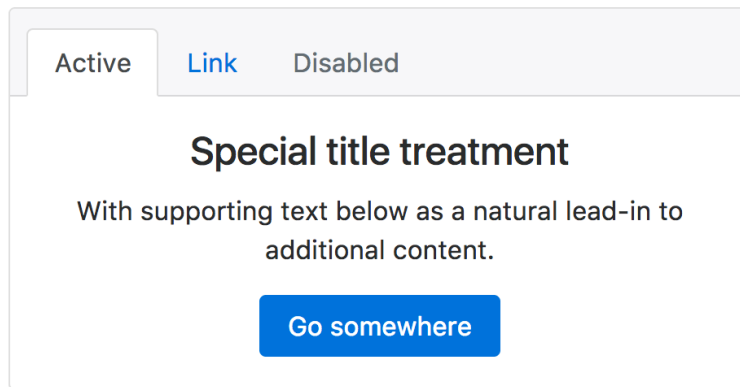
Hay varios tipos definidos. Veamos varios ejemplos.

Código:

```
<div class="card text-center">
  <div class="card-header">
    <ul class="nav nav-tabs card-header-tabs">
      <li class="nav-item">
        <a class="nav-link active" href="#">Active</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item">
        <a class="nav-link disabled" href="#">Disabled</a>
      </li>
    </ul>
  </div>
  <div class="card-block">
    <h4 class="card-title">Special title treatment</h4>
    <p class="card-text">
      With supporting text below as a natural lead-in to additional content.
    </p>
    <a href="#" class="btn btn-primary">Go somewhere</a>
  </div>
</div>
```

Resultado:





## Fondos y color de texto

Podemos de forma rápida cambiar el color de fondo de la Card, del borde así como del texto, veamos un ejemplo:

Código:

```
<div class="card text-white bg-primary mb-3" style="max-width: 18rem">
  <div class="card-header">Header</div>
  <div class="card-body">
    <h5 class="card-title">Primary card title</h5>
    <p class="card-text">
      Some quick example text to build on the card title and make up the bulk of
      the card's content.
    </p>
  </div>
</div>
<div class="card text-white bg-secondary mb-3" style="max-width: 18rem">
  <div class="card-header">Header</div>
  <div class="card-body">
    <h5 class="card-title">Secondary card title</h5>
    <p class="card-text">
      Some quick example text to build on the card title and make up the bulk of
      the card's content.
    </p>
  </div>
</div>
<div class="card text-white bg-success mb-3" style="max-width: 18rem">
  <div class="card-header">Header</div>
  <div class="card-body">
    <h5 class="card-title">Success card title</h5>
    <p class="card-text">
      Some quick example text to build on the card title and make up the bulk of
      the card's content.
    </p>
  </div>
</div>
```

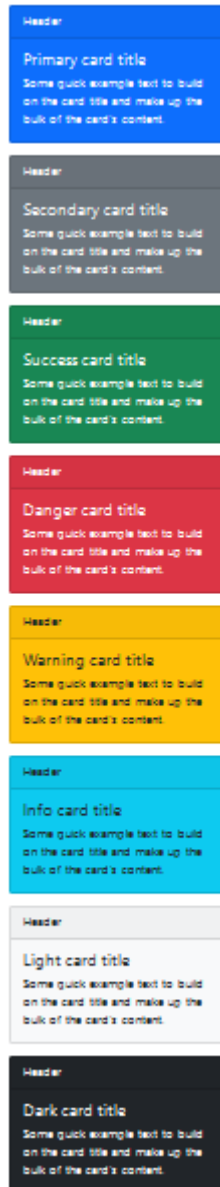
```

</div>
<div class="card text-white bg-danger mb-3" style="max-width: 18rem">
  <div class="card-header">Header</div>
  <div class="card-body">
    <h5 class="card-title">Danger card title</h5>
    <p class="card-text">
      Some quick example text to build on the card title and make up the bulk of
      the card's content.
    </p>
  </div>
</div>
<div class="card text-dark bg-warning mb-3" style="max-width: 18rem">
  <div class="card-header">Header</div>
  <div class="card-body">
    <h5 class="card-title">Warning card title</h5>
    <p class="card-text">
      Some quick example text to build on the card title and make up the bulk of
      the card's content.
    </p>
  </div>
</div>
<div class="card text-dark bg-info mb-3" style="max-width: 18rem">
  <div class="card-header">Header</div>
  <div class="card-body">
    <h5 class="card-title">Info card title</h5>
    <p class="card-text">
      Some quick example text to build on the card title and make up the bulk of
      the card's content.
    </p>
  </div>
</div>
<div class="card text-dark bg-light mb-3" style="max-width: 18rem">
  <div class="card-header">Header</div>
  <div class="card-body">
    <h5 class="card-title">Light card title</h5>
    <p class="card-text">
      Some quick example text to build on the card title and make up the bulk of
      the card's content.
    </p>
  </div>
</div>
<div class="card text-white bg-dark mb-3" style="max-width: 18rem">
  <div class="card-header">Header</div>
  <div class="card-body">
    <h5 class="card-title">Dark card title</h5>
    <p class="card-text">
      Some quick example text to build on the card title and make up the bulk of
      the card's content.
    </p>
  </div>
</div>

```

```
</div>
</div>
```

## Resultado:



También existe la posibilidad de cambiar el borde con los colores del tema: border-primary, border-secondary, border-sucess...

## Grupos de cards

Código:

```
<div class="card-group">
  <div class="card">
    
    <div class="card-body">
```

```

<h5 class="card-title">Card title</h5>
<p class="card-text">
  This is a wider card with supporting text below as a natural lead-in to
  additional content. This content is a little bit longer.
</p>
<p class="card-text">
  <small class="text-muted">Last updated 3 mins ago</small>
</p>
</div>
</div>
<div class="card">
  
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">
      This card has supporting text below as a natural lead-in to additional
      content.
    </p>
    <p class="card-text">
      <small class="text-muted">Last updated 3 mins ago</small>
    </p>
  </div>
</div>
<div class="card">
  
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">
      This is a wider card with supporting text below as a natural lead-in to
      additional content. This card has even longer content than the first to
      show that equal height action.
    </p>
    <p class="card-text">
      <small class="text-muted">Last updated 3 mins ago</small>
    </p>
  </div>
</div>
</div>

```

**Resultado:**

210x180	210x180	210x180
<b>Card title</b> <p>This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p> <p>Last updated 3 mins ago</p>	<b>Card title</b> <p>This card has supporting text below as a natural lead-in to additional content.</p> <p>Last updated 3 mins ago</p>	<b>Card title</b> <p>This is a wider card with supporting text below as a natural lead-in to additional content. This card has even longer content than the first to show that equal height action.</p> <p>Last updated 3 mins ago</p>

Si quisiéramos añadirle un footer a cada una de las cards, lo que haremos será añadirle a cada div su footer de la siguiente forma:

```
<div class="card">
  ...
  <div class="card-footer">
    <small class="text-muted">Last updated 3 mins ago</small>
  </div>
</div>
```

## Grid cards

Podemos usar el sistema de rejilla de Bootstrap y la clase `.row-cols` para indicar cuántas columnas de cards queremos mostrar.

Código:

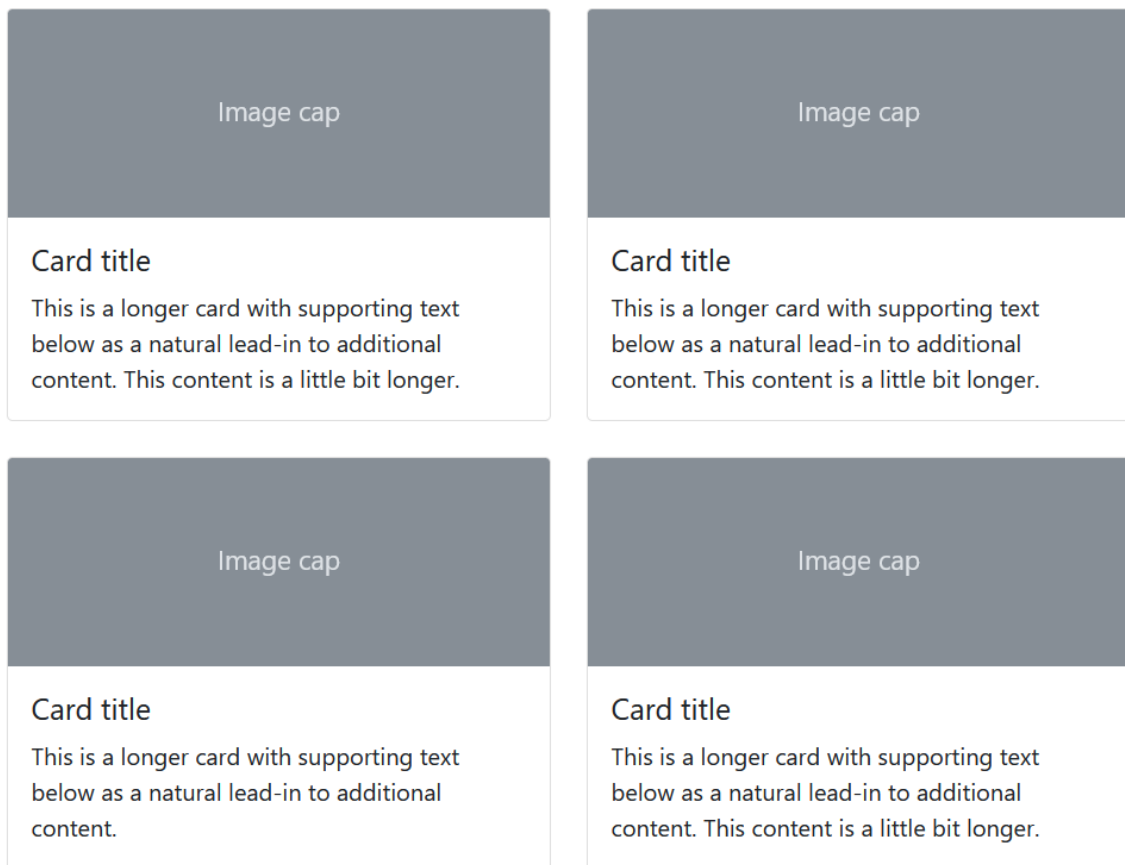
```
<div class="row row-cols-1 row-cols-md-2 g-4">
  <div class="col">
    <div class="card">
      
      <div class="card-body">
        <h5 class="card-title">Card title</h5>
```

```

        <p class="card-text">
            This is a longer card with supporting text below as a natural lead-in
            to additional content. This content is a little bit longer.
        </p>
    </div>
</div>
</div>
<div class="col">
    <div class="card">
        
        <div class="card-body">
            <h5 class="card-title">Card title</h5>
            <p class="card-text">
                This is a longer card with supporting text below as a natural lead-in
                to additional content. This content is a little bit longer.
            </p>
        </div>
    </div>
</div>
<div class="col">
    <div class="card">
        
        <div class="card-body">
            <h5 class="card-title">Card title</h5>
            <p class="card-text">
                This is a longer card with supporting text below as a natural lead-in
                to additional content.
            </p>
        </div>
    </div>
</div>
<div class="col">
    <div class="card">
        
        <div class="card-body">
            <h5 class="card-title">Card title</h5>
            <p class="card-text">
                This is a longer card with supporting text below as a natural lead-in
                to additional content. This content is a little bit longer.
            </p>
        </div>
    </div>
</div>
</div>

```

**Resultado:**



`row-cols-md-2` indica en este caso que para pantallas medianas se muestren 2 columnas.

## Carousel

Carousel es un carrusel de imágenes y/o texto u otro contenido que se desplaza de forma horizontal. Puede controlarse el tiempo de cada diapositiva, incluir controles e indicadores para mostrar qué diapositiva estamos viendo.

## Ejemplo de carrusel solo con diapositivas

Código:

```
<div id="carouselExampleSlidesOnly" class="carousel slide" data-bs-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
```

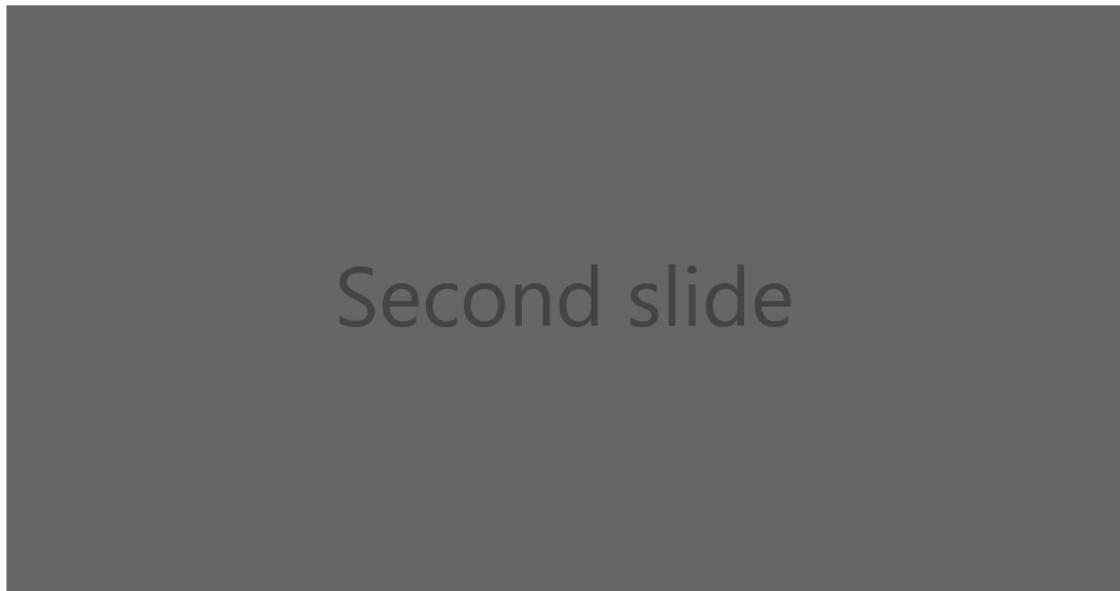
Indica que al cargar la página, sea animado (que arranque automáticamente)

```

<div class="carousel-item">
  
</div>
</div>
</div>

```

Resultado:



Al ser estático, no se aprecia. Pero el carrusel iría mostrando las tres diapositivas con un tiempo de visionado de cada una de 5 segundos (es el tiempo por defecto).

## Ejemplo de carrusel con controles

Código:

```

<div id="carouselExampleControls" class="carousel slide" data-bs-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
  <button class="carousel-control-prev" type="button" data-bs-
target="#carouselExampleControls" data-bs-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
  </button>

```



```

</button>
<button class="carousel-control-next" type="button" data-bs-
target="#carouselExampleControls" data-bs-slide="next">
  <span class="carousel-control-next-icon" aria-hidden="true"></span>
  <span class="visually-hidden">Next</span>
</button>
</div>

```

Cambia la posición de la diapositiva a un índice siguiente

Resultado:



## Carrusel con controles y con indicadores

Código:

```

<div id="carouselExampleIndicators" class="carousel slide" data-bs-ride="carousel">
  <div class="carousel-indicators">
    <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-
slide-to="0" class="active" aria-current="true" aria-label="Slide 1"></button>
    <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-
slide-to="1" aria-label="Slide 2"></button>
    <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-
slide-to="2" aria-label="Slide 3"></button>
  </div>
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
</div>

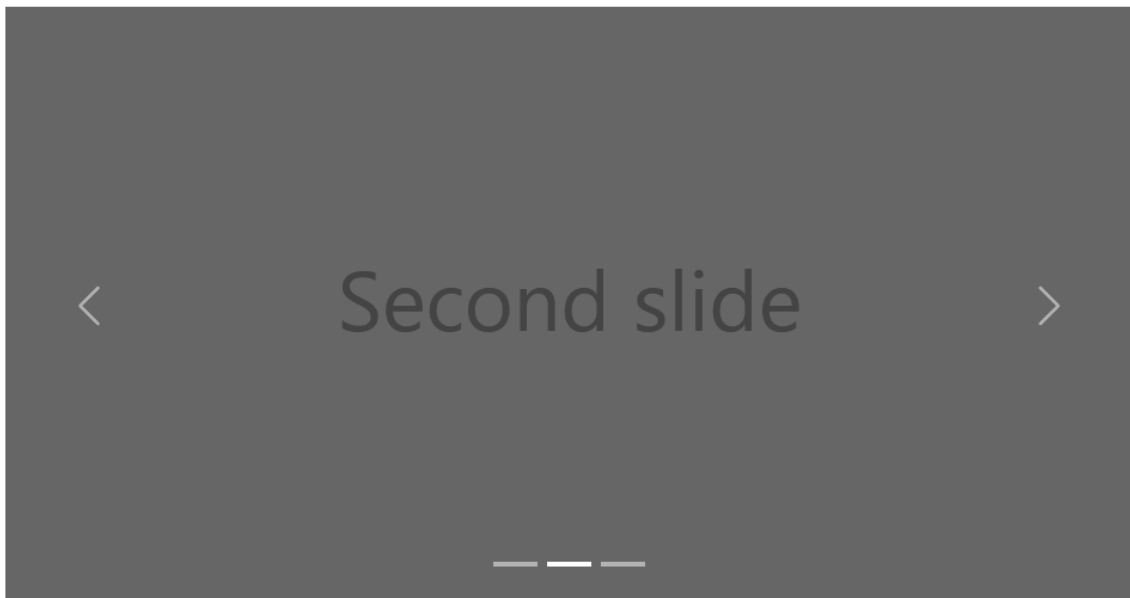
```

```

    <div class="carousel-item">
      
    </div>
  </div>
  <button class="carousel-control-prev" type="button" data-bs-
target="#carouselExampleIndicators" data-bs-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
  </button>
  <button class="carousel-control-next" type="button" data-bs-
target="#carouselExampleIndicators" data-bs-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Next</span>
  </button>
</div>

```

Resultado:



## Collapse

Collapse es un recurso que nos permite ocultar/mostrar un elemento en nuestra página.

Código:

```

<p>
  <a class="btn btn-primary" data-bs-toggle="collapse" href="#collapseExample"
role="button" aria-expanded="false" aria-controls="collapseExample">
    Link with href
  </a>

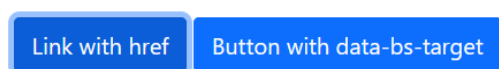
```

```

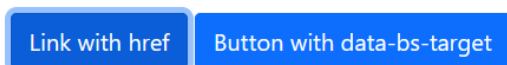
<button class="btn btn-primary" type="button" data-bs-toggle="collapse" data-bs-
target="#collapseExample" aria-expanded="false" aria-controls="collapseExample">
  Button with data-bs-target
</button>
</p>
<div class="collapse" id="collapseExample">
  <div class="card card-body">
    Some placeholder content for the collapse component. This panel is hidden by
    default but revealed when the user activates the relevant trigger.
  </div>
</div>

```

Resultado:



Al pulsar en uno de los botones:



Some placeholder content for the collapse component. This panel is hidden by default but revealed when the user activates the relevant trigger.

Si volvemos a pulsar, se ocultaría el párrafo otra vez.

## Dropdown

Dropdown nos permite crear desplegables de enlaces. Para ello, deberemos incluir la librería Popper. Para hacer pruebas, podemos agregarla mediante su CDN:

```

<script
  src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"
  integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlhm/jp3"
  crossorigin="anonymous"
></script>

```

Aunque a veces puede haber problemas entre la librería JS de Bootstrap y la de Popper, por lo que la solución sería incluir la librería conjunta:

```

<script
  src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcxpyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4"
  crossorigin="anonymous"
></script>

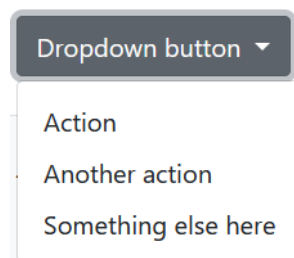
```

### Ejemplo de dropdown con un botón:

Código:

```
<div class="dropdown">
  <button class="btn btn-secondary dropdown-toggle" type="button"
id="dropdownMenuButton1" data-bs-toggle="dropdown" aria-expanded="false">
    Dropdown button
  </button>
  <ul class="dropdown-menu" aria-labelledby="dropdownMenuButton1">
    <li><a class="dropdown-item" href="#">Action</a></li>
    <li><a class="dropdown-item" href="#">Another action</a></li>
    <li><a class="dropdown-item" href="#">Something else here</a></li>
  </ul>
</div>
```

Resultado:



### Ejemplo de dropdown con un enlace:

Código:

```
<div class="dropdown">
  <a class="btn btn-secondary dropdown-toggle" href="#" role="button"
id="dropdownMenuLink" data-bs-toggle="dropdown" aria-expanded="false">
    Dropdown link
  </a>

  <ul class="dropdown-menu" aria-labelledby="dropdownMenuLink">
    <li><a class="dropdown-item" href="#">Action</a></li>
    <li><a class="dropdown-item" href="#">Another action</a></li>
    <li><a class="dropdown-item" href="#">Something else here</a></li>
  </ul>
</div>
```

Resultado:



Se puede aplicar a cualquier variante de tipo botón:

Código:

```
<!-- Example para el botón Danger -->
<div class="btn-group">
  <button type="button" class="btn btn-danger dropdown-toggle" data-bs-
toggle="dropdown" aria-expanded="false">
    Action
  </button>
  <ul class="dropdown-menu">
    <li><a class="dropdown-item" href="#">Action</a></li>
    <li><a class="dropdown-item" href="#">Another action</a></li>
    <li><a class="dropdown-item" href="#">Something else here</a></li>
    <li><hr class="dropdown-divider"></li>
    <li><a class="dropdown-item" href="#">Separated link</a></li>
  </ul>
</div>
```

Resultado para varios tipos de botones:



## Barra de navegación

Las barras de navegación requieren un contenedor `.navbar` con `.navbar-expand{sm|-md|-lg|-xl|-xxl}` para colapsar con capacidad responsive.

Entre otros, contiene los siguientes elementos:

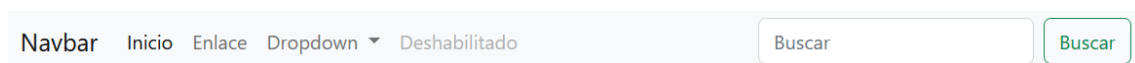
- `.navbar-brand` para el nombre de tu empresa, producto o proyecto.
- `.navbar-nav` para una navegación ligera y de altura completa (incluido soporte para menús desplegables).
- `.navbar-toggler` para usar con nuestro complemento Collapse
- `.navbar-text` para agregar cadenas de texto centradas verticalmente.
- `.collapse.navbar-collapse` para agrupar y ocultar el contenido de la barra de navegación por breakpoint determinado.

Ejemplo

## Código:

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Inicio</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Enlace</a>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown"
role="button" data-bs-toggle="dropdown" aria-expanded="false">
            Dropdown
          </a>
          <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
            <li><a class="dropdown-item" href="#">Acción</a></li>
            <li><a class="dropdown-item" href="#">Otra acción</a></li>
            <li><hr class="dropdown-divider"></li>
            <li><a class="dropdown-item" href="#">Algo más aquí</a></li>
          </ul>
        </li>
        <li class="nav-item">
          <a class="nav-link disabled">Deshabilitado</a>
        </li>
      </ul>
      <form class="d-flex" role="search">
        <input class="form-control me-2" type="search" placeholder="Buscar" aria-
label="Buscar">
        <button class="btn btn-outline-success" type="submit">Buscar</button>
      </form>
    </div>
  </div>
</nav>
```

## Resultado:



Cuando la ventana se reduzca más allá del breakpoint para las pantallas grandes (lg) se vería así:

Navbar



Puedes reemplazar el texto dentro de `.navbar-brand` con `<img>`.



## Esquemas de color

Se puede personalizar los colores de la barra de navegación gracias a la combinación de clases de temas y utilidades de `background-color`. Elige entre `.navbar-light` para usar con colores de fondo claros o `.navbar-dark` para colores de fondo oscuros. Luego, personaliza con las utilidades `.bg-*`.

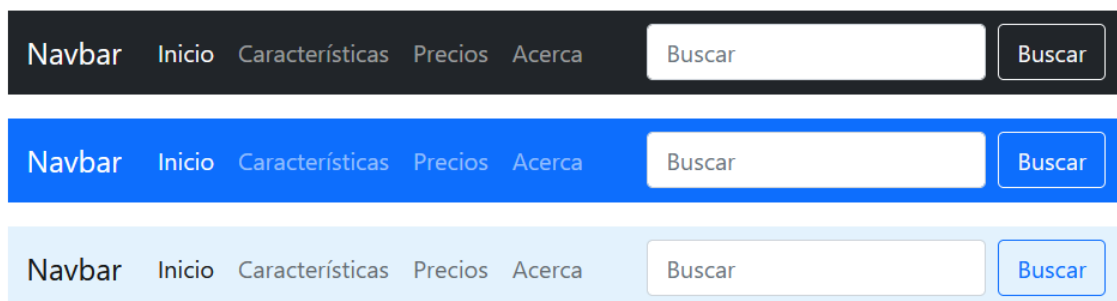
Código:

```
<nav class="navbar navbar-dark bg-dark">
  <!-- Navbar content -->
</nav>

<nav class="navbar navbar-dark bg-primary">
  <!-- Navbar content -->
</nav>

<nav class="navbar navbar-light" style="background-color: #e3f2fd;">
  <!-- Navbar content -->
</nav>
```

Resultado:



## Bibliografía

- Curso de Sass impartido para CEFIRE Específico de Formación Profesional (autora: Rosa María Medina Gómez).

Página oficial desde donde poder consultar toda la documentación y poder descargar la librería:  
<http://getbootstrap.com>

- Blog oficial de Bootstrap donde consultar las últimas novedades:  
<http://blog.getbootstrap.com/>
- Temas y plantillas gratuitas para Bootstrap: o <http://startbootstrap.com>
  - o <http://bootstrapzero.com>
  - o <http://bootswatch.com>
  - o <http://www.bootbundle.com>
- Ejemplos y trozos de código útiles para Bootstrap: <http://bootsnipp.com>
- Ejemplos de uso: <http://expo.getbootstrap.com>
- Listado de los recursos disponibles: <https://startbootstrap.com/bootstrap-resources/>