

# UT5: INTEGRACIÓN DE CONTENIDO INTERACTIVO. BOOTSTRAP

DIW-DAW2

## INDICE

## Contenido

1. INTRODUCCIÓN.....	3
2. INSTALACIÓN DE BOOTSTRAP .....	3
2.1. Métodos de instalación.....	3
2.2. Instalación de Bootstrap 5 .....	4
3. GRID DE BOOTSTRAP .....	6
3.1.1. Ejemplo 2: columnas anidadas .....	10
3.2. Row-column.....	15
3.3. Alineado de columnas verticalmente.....	17
3.4. Alineación de columnas horizontalmente.....	19
3.5. Gutters .....	21
3.5.1. Otras clases para el GRID de Bootstrap 5 .....	23
4. TIPOGRAFÍAS.....	28
5. IMÁGENES.....	36
6. TABLAS .....	37
7. ACORDEON .....	40
8. ALERTAS .....	41
9. MIGAS DE PAN O BREADCRUMBS.....	42
10. BOTONES.....	43
11. CARRUSEL.....	46
12. FORMULARIOS .....	47
13. SELECTBOX.....	49
13.1. CHECKBOX.....	50
13.2. RADIOBUTTON .....	50
13.3. SWITCH.....	50
13.4. CHECKBOX INLINE .....	51
13.5. CHECKBOX ESTILO BOTÓN .....	51
13.6. RADIOBUTTON ESTILO BOTÓN.....	52
13.7. RANGE .....	52
13.8. Etiquetas flotantes .....	52
13.9. Validación de formularios .....	53
14. VENTANA MODAL EN BOOTSTRAP .....	53

15. FLEXBOX.....55

## 1. INTRODUCCIÓN

En esta unidad vamos a fijar como principal objetivo el diseño web a través de Bootstrap. Se verán sus componentes, elementos interactivos de una forma introductoria a este framework de diseño web. Un framework es una serie de herramientas y de elementos que ya han sido programados y que nosotros podremos utilizar de una forma sencilla, es decir, que Bootstrap ya tiene componentes y elementos programados y que nosotros podremos poner en nuestra página web sin tener que preocuparnos de crear el código HTML, luego el código CSS y por último JavaScript.

Ventajas:

- Con Bootstrap sólo trabajamos con código HTML y con clases que los desarrolladores de Bootstrap hicieron por nosotros de tal manera que podremos realizar sitios de una forma más rápida.
- Los sitios web que vaya creando van a ser adaptables a dispositivos móviles.
- Bootstrap nos da lo que se conoce como la grid de Bootstrap, es decir, la estructura en columnas que podemos hacer adaptables a los dispositivos móviles.
- Con Bootstrap podremos modificar los estilos y crear nuestro propio estilo modificando los colores.
- Bootstrap se enfoca bastante en crear sitios responsive.

Herramientas que se necesitan para trabajar con Bootstrap:

- Navegador. Principalmente Chrome y Firefox. Podríamos utilizar cualquiera pero estos son los que tienen herramientas para desarrolladores muy útiles y que utilizaremos más adelante.
- Editor de código (Visual Studio Code)

## 2. INSTALACIÓN DE BOOTSTRAP

### 2.1. Métodos de instalación

Así que bien, pues vamos a aprender Bootstrap 5 desde cero.

- lo primero que tenemos que hacer es descargarlo:

[Bootstrap · The most popular HTML, CSS, and JS library in the world. \(getbootstrap.com\)](https://getbootstrap.com/)

En esta página tienes varios enlaces, tienes el enlace de docs que sería documentación y aquí tienes toda la documentación de Bootstrap.

Aquí vas a tener un enlace que dice Get started y otro que dice Download.

Vamos a presionar de download que nos va a llevar a esta página, donde nos va a mencionar que tenemos varias formas de descargar Bootstrap y te las voy a explicar.

1. Descargar Bootstrap compilado, nos va a dar los archivos CSS y JavaScript compilados simplemente para que los importemos en nuestro proyecto, como cualquier archivo HTML o CSS.

## Compiled CSS and JS

Download ready-to-use compiled code for **Bootstrap v5.1.3** 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

2. Source Files, que es lo que nos va a permitir es descargar Bootstrap, pero esta versión no va a estar compilada.

Esta opción es un poco más avanzada y suele ser para personas que quieren editar Bootstrap por ejemplo a ti no te gusta algo y lo quieres editar, bueno, bajas estos archivos, editas el código y luego lo compilas.

## Source files

Compile Bootstrap with your own asset pipeline by downloading our source Sass, JavaScript, and documentation files. This option requires some additional tooling:

- [Sass compiler](#) for compiling Sass source files into CSS files
- [Autoprefixer](#) for CSS vendor prefixing

Should you require our full set of [build tools](#), they are included for developing Bootstrap and its docs, but they're likely unsuitable for your own purposes.

Download source

3. Y después tenemos esta otra opción que también vamos a ver en este capítulo que es mediante la CDN. La CDN lo que nos va a permitir es que los archivos CSS y JavaScript de Bootstrap ya están alojados en una página web que en este caso se llama Jéés Deliver y lo único que tenemos que hacer nosotros es copiar estas líneas de código que como puedes ver son los estilos CSS y lo que será el archivo de JavaScript y simplemente los pegamos en nuestra página. La desventaja de esto es que nosotros necesitamos una conexión de Internet para que esto funcione, porque estos archivos están alojados en una página que no es la nuestra.

### CDN via jsDelivr

Skip the download with [jsDelivr](#) to deliver cached version of Bootstrap's compiled CSS and JS to your project.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-...">
```

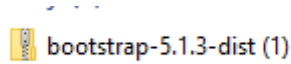
If you're using our compiled JavaScript and prefer to include Popper separately, add Popper before our JS, via a CDN preferably.

```
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js" integrity="sha384-...">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-...">
```

## 2.2. Instalación de Bootstrap 5

Vamos a utilizar el primer método:

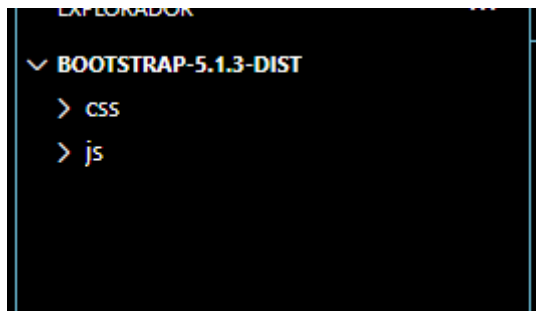
- Descargar el fichero .zip. Lo voy a guardar en una carpeta que denomina Bootstrap y lo descomprimos.



Una vez descomprimida la carpeta, dentro de esta carpeta tenemos otras dos carpetas, la carpeta de CSS y la carpeta de JavaScript.



Añadimos estas dos carpetas a nuestro proyecto en Bootstrap.



Añadimos un archivo html:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10
11 </body>
12 </html>
```

Es obligatorio  
para poder  
trabajar en  
responsive

Entonces es muy importante que la utilices porque de otra forma la parte adaptable no te va a funcionar.

Bueno, pues queremos agregar Bootstrap.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Bootstrap5</title>
8   <link rel="stylesheet" href="css/bootstrap.min.css">
9 </head>
10 <body>
11   <h1>Esto es una prueba</h1>
12   <button class="btn btn-primary">Empezar</button>
13 </body>
14 </html>
```

Hay que insertar los archivos css de bootstrap

Al botón se le puede poner una clase que sería BTN y luego puedo agregarle la clase btn-primary para indicar que quiero aplicar estilos Bootstrap.

El segundo paso es agregar los archivos JavaScript para poder utilizar algunos componentes JavaScript en Bootstrap.

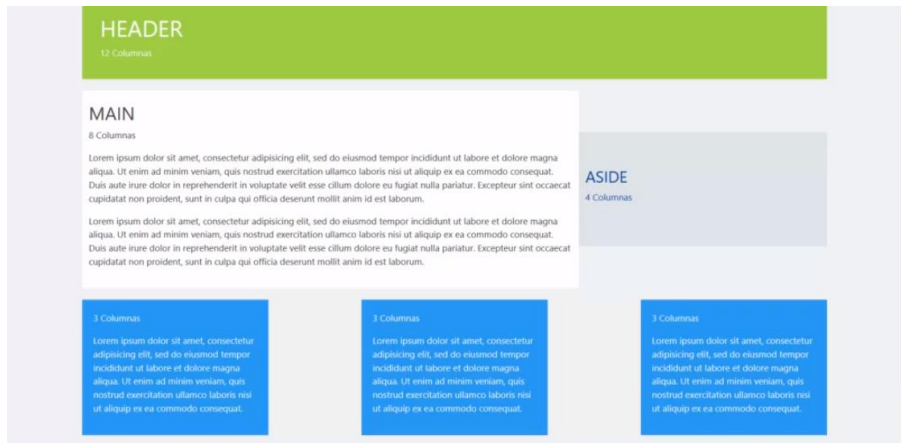
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Bootstrap5</title>
8   <link rel="stylesheet" href="css/bootstrap.min.css">
9 </head>
10 <body>
11   <h1>Esto es una prueba</h1>
12   <button class="btn btn-primary">Empezar</button>
13   <script src="js/bootstrap.min.js"></script>
14 </body>
15 </html>
```

### 3. GRID DE BOOTSTRAP

El grid de Bootstrap es el sistema de columnas y tiene muchos nombres, y nos permite hacer sitios responsive, es decir, sitios que se adaptan a diferentes tamaños de dispositivo móvil.

Bootstrap trabaja con un sistema de 12 columnas.

Esto significa que nosotros tenemos 12 columnas disponibles para asignarle a cualquier elemento.



Breakpoints que hay que tener en cuenta en Bootstrap para poder trabajar con distintos dispositivos:

	Muy pequeño <576 px	Pequeño ≥576px	Medio ≥768px	Grande ≥992px	Extra grande ≥1200px
Ancho máximo del contenedor	Ninguno (automático)	540px	720px	960px	1140px
Prefijo de clase	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
# de columnas	12				
Ancho del canalón	30px (15px en cada lado de una columna)				
encajable	sí				
Ordenación de columnas	sí				

1. Creamos una carpeta denominada GRID y dentro un archivo grid.html. Vamos a realizar la página de la figura anterior:

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Grid</title>
8   <link rel="stylesheet" href="../css/bootstrap.min.css">
9 </head>
10 <body>
11   <script src="../js/bootstrap.bundle.min.js"></script>
12 </body>
13 </html>

```

Lo que necesitamos es seguir unas pequeñas reglas que Bootstrap nos da que tenemos que respetar para poder hacer sitio responsive.

La primera regla es que todo nuestro contenido lo tenemos que encerrar dentro de contenedores que para ello tenemos que agregar un div con la clase de container.



Esta es la primera clase que vamos a ver y esta clase lo que nos va a permitir es que va a ser el contenedor principal y dentro vamos a poner todo el contenido de nuestro sitio.

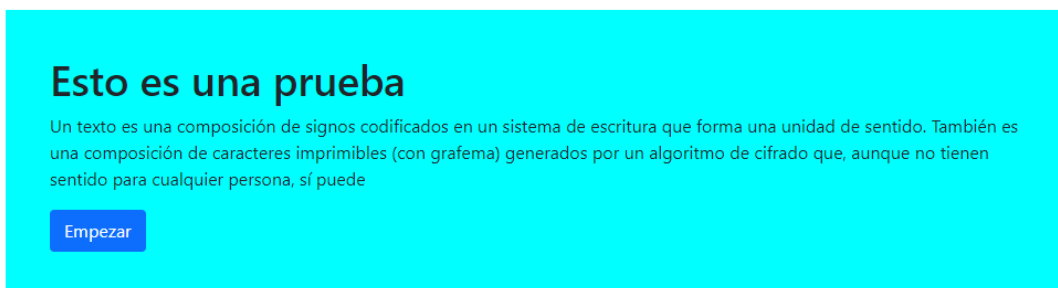
La siguiente regla que nos dice Bootstrap es que tenemos que poner dentro de un contenedor lo que son filas (grupo de columnas) y dentro de las filas tenemos que poner columnas.

Resumiendo:

- El container es el contenedor principal y en él van a ir todas las filas y columnas y después tenemos el container que también. Sirve para poder centrar nuestra página.
- Las filas son grupos horizontales de columnas.
- Y, por último, tenemos nuestras columnas. Dentro de las columnas ya podemos poner lo que queramos.

Sigamos con un ejemplo:

```
<link rel="stylesheet" href="../../css/bootstrap.min.css">
<style>
  .row > div{
    background-color: aqua;
    margin: 10px 0;
    padding: 40px;
  }
</style>
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col">
        <h1>Esto es una prueba</h1>
        <p>Un texto es una composición de signos codificados en un sistema de escritura
          que forma una unidad de sentido. También es una composición de caracteres impri
          de cifrado que, aunque no tienen sentido para cualquier persona, sí puede</p>
        <button class="btn btn-primary">Empezar</button>
      </div>
    </div>
  </div>
```



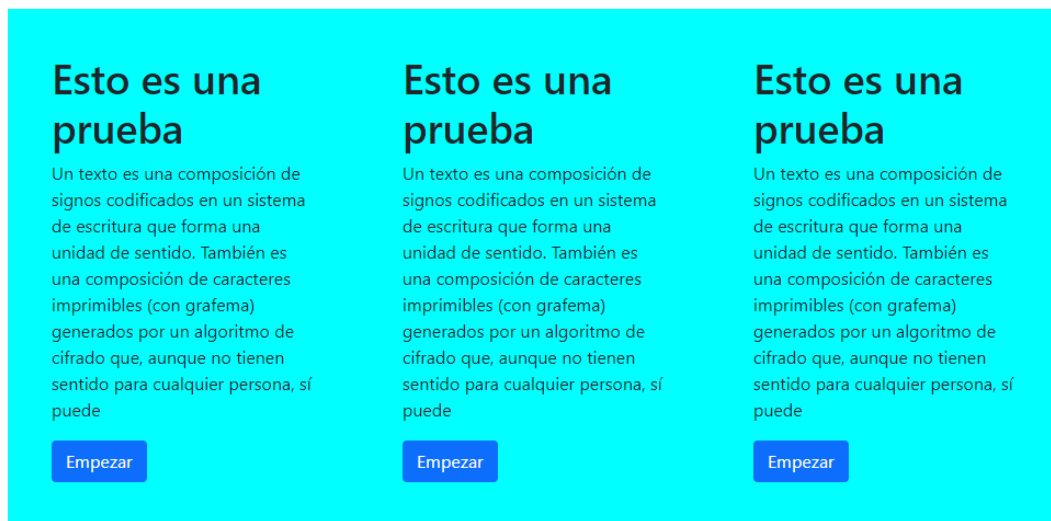
Si repito el código de la columna tres veces el resultado será el siguiente:

Pero

lo

más

interesante



En este ejemplo, las tres columnas ocupan los mismo que son 4 columnas. Podemos hacer que cada columna tenga un tamaño distinto (6,6 y 12):

```
<div class="row">
  <div class="col-6">
    <h1>Esto es una prueba</h1>
    <p>Un texto es una composición de signos codificados en un sistema de escritura
      que forma una unidad de sentido. También es una composición de caracteres impri
de cifrado que, aunque no tienen sentido para cualquier persona, sí puede</p>
    <button class="btn btn-primary">Empezar</button>
  </div>
  <div class="col-6">
    <h1>Esto es una prueba</h1>
    <p>Un texto es una composición de signos codificados en un sistema de escritura
      que forma una unidad de sentido. También es una composición de caracteres impri
de cifrado que, aunque no tienen sentido para cualquier persona, sí puede</p>
    <button class="btn btn-primary">Empezar</button>
  </div>
  <div class="col-12">
```

Yo quiero poder controlar estos tamaños de mis columnas dependiendo del dispositivo móvil. En Bootstrap se puede variar el tamaño de la columna en función del tamaño del dispositivo

Entonces nosotros tenemos el tamaño de columna, que es un tamaño extra pequeño extra small, que es normalmente para tamaños de smartphone en un formato vertical y aquí tiene la medida de donde aplica para cuando el tamaño de pantalla es menor a 576 píxeles.

```

<div class="row">
  <div class="col-12 col-md-6">
    <h1>Esto es una prueba</h1>
    <p>Un texto es una composición de signos codificados en un sistema de escritura
      que forma una unidad de sentido. También es una composición de caracteres impri
      de cifrado que, aunque no tienen sentido para cualquier persona, sí puede</p>
    <button class="btn btn-primary">Empezar</button>
  </div>
  <div class="col-12 col-sm-6">
    <h1>Esto es una prueba</h1>
    <p>Un texto es una composición de signos codificados en un sistema de escritura
      que forma una unidad de sentido. También es una composición de caracteres impri
      de cifrado que, aunque no tienen sentido para cualquier persona, sí puede</p>
    <button class="btn btn-primary">Empezar</button>
  </div>
</div>

```

Hay una opción para div que es container-fluid que lo que hace es eliminar los márgenes a la derecha y a la izquierda.

### 3.1.1. Ejemplo 2: columnas anidadas

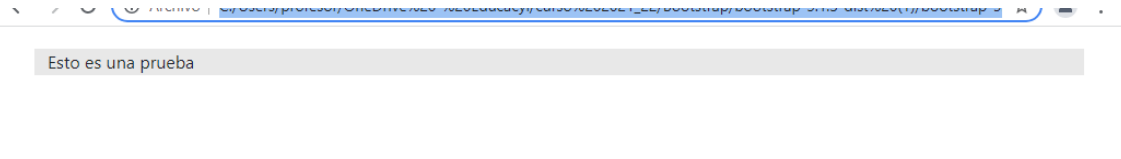
Partimos del siguiente archivo columnasanidadas.html:

```

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>Columnas anidadas</title>
  <style>
    .container{
      margin-top: 20px;
    }
    .row {
      background: #e8e8e8;
    }
    .columna{
      background: #f2f2f2;
      text-align: center;
      border: 1px solid #8d8d8d;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col-12">
      </div>
    </div>
  </div>
  <script src="../js/bootstrap.min.js"></script>
</body>
</html>

```

Si pongo en nuestra columna el texto :” Esto es una prueba” el resultado que tengo al ejecutar mi documento html es el siguiente:

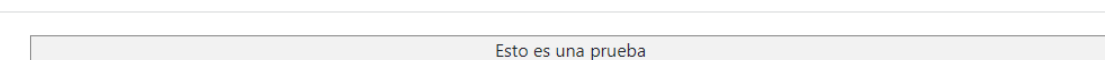


Ahora voy a crear una clase columna propia en la columna Bootstrap:

El código modificado es el siguiente:

```
<body>
  <div class="container">
    <div class="row">
      <div class="columna col-12">
        Esto es una prueba
      </div>
    </div>
  </div>
</body>
<script src=".../bootstrap.min.js"></script>
```

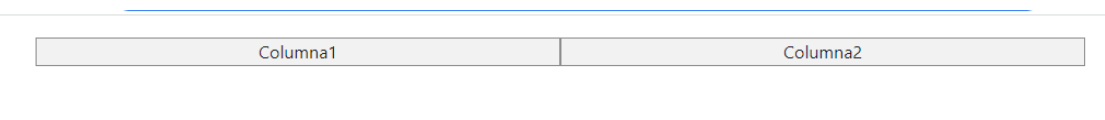
Y el resultado es:



Voy a crear dos columnas y para distinguirlas voy a poner Columna1, columna2 y cada columna puede abarcar 6 y 6 el código que tendría sería el siguiente:

```
<body>
  <div class="container">
    <div class="row">
      <div class="columna col">
        Columna1
      </div>
      <div class="columna col">
        Columna2
      </div>
    </div>
  </div>
  <script src=".../bootstrap.min.js"></script>
</body>
</html>
```

El resultado sería el siguiente:



Se pretende crear columnas anidadas, esto significa que nosotros podemos tener columnas dentro de otras columnas.

Por ejemplo, ¿qué pasa si dentro de columna 1 yo quiero agregar otras dos columnas?

Lo único que tienes que hacer es aquí, dentro de lo que sería tu contenido, agregar otra fila, y dentro de esta fila ya puedes agregar otras columnas.

El código resultante sería:

```
<body>
  <div class="container">
    <div class="row">
      <div class="columna col">
        Columna1
        <div class="row">
          <div class="col">
            Columna 11
          </div>
          <div class="col">
            Columna 12
          </div>
        </div>
      </div>
      <div class="columna col">
        Columna2
      </div>
    </div>
  </div>
</body>
```

Y el resultado sería el siguiente:

Columna1		Columna2
Columna 11	Columna 12	

Para distinguirlo mejor voy a dar estilos a estas columnas y voy a hacer lo mismo en Columna2. Creo una clase llamada columna-interior a la cual voy a dar estilo con css.

```

<body>
  <div class="container">
    <div class="row">
      <div class="columna col">
        Columna1
        <div class="row">
          <div class="columna-interior col">Columna 11</div>
          <div class="columna-interior col">Columna 12</div>
        </div>
      </div>
      <div class="columna col">
        Columna2
        <div class="row">
          <div class="columna-interior col">Columna 21</div>
          <div class="columna-interior col">Columna 22</div>
        </div>
      </div>
    </div>
  </div>

```

El estilo a columna-interior sería:

```

.columna-interior{
  background: #e8e8e8;
  border: 1px solid #8d8d8d;
  padding: 10px;
}

```

El resultado sería el siguiente:

Columna1		Columna2	
Columna 11	Columna 12	Columna 21	Columna 22

También podemos modificar el tamaño de las columnas internas de tal forma que podría realizar lo siguiente:

Por ejemplo dentro de las columnas internas de Columna 1 puedo hacer que columna 11 tenga un tamaño de 4 y columna 12 de 8. Pero también puedo decir, que el tamaño de columna11 sea de 4 y a partir de un tamaño de pantalla mediano sea 6.

Cuando el tamaño de la pantalla supera el tamaño mediano columna11 y columna 12 ocupan 6 pero cuando disminuyo la pantalla pasan a ocupar 4 y 8 respectivamente.

Columna 11	Columna 12	Columna 21	Columna 22
Columna1		Columna2	

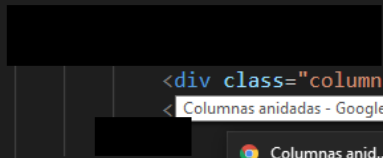
  

Columna1		Columna2	
Columna 11	Columna 12	Columna 21	Columna 22

Por ejemplo, qué pasa si esto lo queremos hacer responsive?

Para ello, voy a considerar que a partir de un tamaño mediano el tamaño sea 6 pero que si tengo un tamaño muy pequeño sea 12. Esto se codifica de la siguiente manera:

```
<body>
  <div class="container">
    <div class="row">
      <div class="columna col-12 col-md-6">
        Columna1
        <div class="row">
          <div class="columna-interior col-12 col-md-6">Columna 11</div>
          <div class="columna-interior col-12 col-md-6">Columna 12</div>
        </div>
      </div>
      <div class="columna col-12 col-md-6">
        Columna2
        <div class="row">
          <div class="columna-interior col">Columna 21</div>
          <div class="columna-interior col">Columna 22</div>
        </div>
      </div>
    </div>
  </div>
```



De esta forma lo que va a suceder es que ahora estas columnas las estoy trabajando de forma responsive.

```
<body>
  <div class="container">
    <div class="row">
      <div class="columna col-12 col-md-6">
        Columna1
        <div class="row">
          <div class="columna-interior col-12 col-md-6">Columna 11</div>
          <div class="columna-interior col-12 col-md-6">Columna 12</div>
        </div>
      </div>
      <div class="columna col-12 col-md-6">
        Columna2
        <div class="row">
          <div class="columna-interior col-12 col-md-6">Columna 21</div>
          <div class="columna-interior col-12 col-md-6">Columna 22</div>
        </div>
      </div>
    </div>
  </div>
</body>
```

Con este código conseguimos que si la pantalla es muy pequeña cada columna ocupe 12 y si es mediana, grande o extragrande ocupen 6.

Columna1		Columna2	
Columna 11	Columna 12	Columna 21	Columna 22

Tamaño superior a mediano

Columna1					
Columna 11					
Columna 12					
Columna2					
Columna 21					
Columna 22					

Tamaño más pequeño a mediano

3.2. Row-column

Vamos a crear un fichero html llamado rowcolumn donde el resultado va a ser una fila con 6 columnas:



```
<body>
  <div class="container">
    <div class="row">
      <div class="col">Columna #1 </div>
      <div class="col">Columna #2 </div>
      <div class="col">Columna #3 </div>
      <div class="col">Columna #4 </div>
      <div class="col">Columna #5 </div>
      <div class="col">Columna #6 </div>
    </div>
  </div>
  <script src="../../js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

El resultado será:

---

Columna #1	Columna #2	Columna #3	Columna #4	Columna #5	Columna #6
------------	------------	------------	------------	------------	------------

El estilo css que se aplicará será:

```
<style>
  .row{
    background: #e8e8ee;
    margin-top: 20px;
  }
  .col{
    background: #f2f2f2;
    text-align: center;
    padding: 10px;
    border: 1px solid #4d5061;
  }
</style>
```

Resultado de las 6 columnas:

---

Columna #1	Columna #2	Columna #3	Columna #4	Columna #5	Columna #6
------------	------------	------------	------------	------------	------------

En este caso en una fila tengo 6 columnas todas del mismo tamaño. En Bootstrap 5 hay una clase nueva que es **row-cols** que lo que hace es decir el número de columnas por fila que quiero poner, por ejemplo:

```

    </style>
</head>
<body>
  <div class="container">
    <div class="row row-cols-3">
      <div class="col">Columna #1 </div>
      <div class="col">Columna #2 </div>
      <div class="col">Columna #3 </div>
      <div class="col">Columna #4 </div>
      <div class="col">Columna #5 </div>
      <div class="col">Columna #6 </div>
    </div>
  </div>
</body>

```

En cada fila 3  
columnas

çEl resultado es:

Columna #1	Columna #2	Columna #3
Columna #4	Columna #5	Columna #6

Si pusiera row-cols-4 el resultado sería el siguiente:

Columna #1	Columna #2	Columna #3	Columna #4
Columna #5	Columna #6		

### 3.3. Alineado de columnas verticalmente

En este caso voy a crear un grid con tres columnas. Al crear en mi fichero html las tres columnas puedo escribir el siguiente código:

**.col\*3{Columna #}\$** y de esta forma se crea en el fichero las tres líneas de código referente a las tres columnas:

```

<body>
  <div class="container">
    <div class="row">
      <div class="col">Columna #1</div>
      <div class="col">Columna #2</div>
      <div class="col">Columna #3</div>
    </div>
  </div>
  <script src="../../js/bootstrap.bundle.min.js"></script>
</body>

```

Para alinear las columnas verticalmente utilizaremos la clase align-items-start, align-items-center y align-items-end de la siguiente manera:

Voy a considerar una altura de la fila de 150 px.

```
<body>
  <div class="container">
    <div class="row align-items-start">
      <div class="col">Columna #1</div>
      <div class="col">Columna #2</div>
      <div class="col">Columna #3</div>
    </div>
    <div class="row align-items-center">
      <div class="col">Columna #1</div>
      <div class="col">Columna #2</div>
      <div class="col">Columna #3</div>
    </div>
    <div class="row align-items-end">
      <div class="col">Columna #1</div>
      <div class="col">Columna #2</div>
      <div class="col">Columna #3</div>
    </div>
  </div>
</body>
```

El resultado en el navegador será el siguiente:

Columna #1	Columna #2	Columna #3
Columna #1	Columna #2	Columna #3
Columna #1	Columna #2	Columna #3

También se puede alinear cada columna de forma independiente. Para ello, se utiliza en el elemento que se quiera aplicar la clase align-self-start, align-self-center y align-self-end.

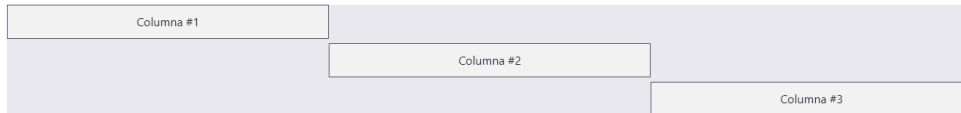
El código podría ser el siguiente:

```

<div class="row">
  <div class="col align-self-start">Columna #1</div>
  <div class="col align-self-center">Columna #2</div>
  <div class="col align-self-end">Columna #3</div>
</div>

```

El resultado sería el siguiente\_



### 3.4. Alineación de columnas horizontalmente

En este caso se va a realizar una alineación horizontal de columnas. Para ello, parto de un archivo llamado alineaciónhorizontalmente.html:

```

<body>
  <div class="container">
    <div class="row">
      <div class="columna col-4">Columna $</div>
    </div>
  </div>
  <script src="../js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

Es para crear un contenedor columna de tamaño 4.

Si le doy al tabulador el resultado es el siguiente:

```

23 <body>
24   <div class="container">
25     <div class="row">
26       <div class="columna col-4">Columna 1</div>
27     </div>
28   </div>
29   <script src="../js/bootstrap.bundle.min.js"></script>
30 </body>

```

A la fila voy a añadir 2 columnas:

```

<div class="container">
  <div class="row">
    <div class="columna col-4">Columna 1</div>
    <div class="columna col-4">Columna 2</div>
  </div>
</div>

```

Columna 1	Columna 2	
-----------	-----------	--

Para alinear horizontalmente se pueden utilizar las clases `justify-content-start` (alinear a la izquierda), `justify-content-center` (alinear al centro) y `justify-content-end` (para alinear a la derecha):

```
<body>
  <div class="container">
    <div class="row justify-content-start">
      <div class="columna col-4">Columna 1</div>
      <div class="columna col-4">Columna 2</div>
    </div>
    <div class="row justify-content-center">
      <div class="columna col-4">Columna 1</div>
      <div class="columna col-4">Columna 2</div>
    </div>
    <div class="row justify-content-end">
      <div class="columna col-4">Columna 1</div>
      <div class="columna col-4">Columna 2</div>
    </div>
  </div>
```

El resultado será:

Columna 1	Columna 2	
	Columna 1	Columna 2
	Columna 1	Columna 2

En este caso, los huecos quedan a un lado u otro de las columnas, pero a veces nos gustaría que el espacio se encuentre entre las columnas para ello hay las siguientes clases: `justify-content-around` (coloca un espacio delante y detrás de cada columna), `justify-content-between` (el espacio queda entre las dos columnas) y `justify-content-evenly` (los espacios vacíos se reparten equitativamente entre las columnas);

```
<div class="row justify-content-around">
  <div class="columna col-4">Columna 1</div>
  <div class="columna col-4">Columna 2</div>
</div>
<div class="row justify-content-between">
  <div class="columna col-4">Columna 1</div>
  <div class="columna col-4">Columna 2</div>
</div>
<div class="row justify-content-evenly">
  <div class="columna col-4">Columna 1</div>
  <div class="columna col-4">Columna 2</div>
</div>
```

El resultado será:



Todas estas clases se parecen mucho a Flexbox sin Bootstrap.

### 3.5. Gutters

Los Gutters son espaciados que Bootstrap agrega por defecto a nuestras columnas que a veces queremos quitar y otras veces agregar a nuestro gusto.

Vamos a considerar el siguiente código en un archivo thegutters.html:

```
grid > thegutters.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>The gutters</title>
8   <link rel="stylesheet" href="../css/bootstrap.min.css">
9   <style>
10     .row{
11       background: #e8e8e8;
12       margin-top: 20px;
13     }
14     .contenido{
15       background: #f2f2f2;
16       text-align: center;
17       padding: 10px;
18       border: 1px solid #4d5061
19     }
20   }
21 </style>
22 </head>
23 <body>
24   <div class="container">
25     <div class="row">
26       <div class="col-6">
27         <div class="contenido">Columna 1</div>
28       </div>
29       <div class="col-6">
30         <div class="contenido">Columna 2</div>
31       </div>
32       <div class="col-6">
33         <div class="contenido">Columna 3</div>
34       </div>
35       <div class="col-6">
36         <div class="contenido">Columna 4</div>
37       </div>
38     </div>
39   </div>
40   <script src="../js/bootstrap.bundle.min.js"></script>
41 </body>
```

El resultado es el siguiente:

Columna 1	Columna 2
Columna 3	Columna 4

Aparecen huecos entre las columnas que a lo mejor no nos interesan. Para quitar esos huecos utilizaremos la clase g-0:

```

<body>
  <div class="container">
    <div class="row g-0">
      <div class="col-6">
        <div class="contenido">Columna 1</div>
      </div>
      <div class="col-6">
        <div class="contenido">Columna 2</div>
      </div>
    </div>
  </div>

```

El resultado sería el siguiente:

Columna 1	Columna 2
Columna 3	Columna 4

Pero tenemos más opciones además de g-0 que serían gx-5 para un gutter en el eje X y añade un espaciado de 5 y gy-5 (en el eje Y y añade un espaciado de 5).

```

<div class="container">
  <div class="row gx-5 gy-5">
    <div class="col-6">
      <div class="contenido">Columna 1</div>
    </div>
    <div class="col-6">
      <div class="contenido">Columna 2</div>
    </div>
    <div class="col-6">
      <div class="contenido">Columna 3</div>
    </div>
    <div class="col-6">
      <div class="contenido">Columna 4</div>
    </div>
  </div>
</div>

```

Columna 1	Columna 2
Columna 3	Columna 4

### 3.5.1. Otras clases para el GRID de Bootstrap 5

#### 1. Ordenar columnas.

Existe una clase llamada order que nos permite ordenar columnas. Existen varias opciones:

- Order-1 a order-5. Lo que hace es poner la columna en la posición que indica el número. Sólo se puede ordenar hasta la posición 5. Primero coloca las columnas que no tienen número de orden y luego empieza a colocar las columnas a las que hemos puesto order-x



- Order-first. La columna a la que se aplica se coloca al principio.
- Orden-last. La columna se coloca al final.

Un ejemplo aparece en el fichero ordenar.html

```
<style>
  .row{
    background: #e8e8ee;
    margin-top: 20px;
    height: 150px;
  }
  .col{
    background: #f2f2f2;
    text-align: center;
    padding: 10px;
    border: 1px solid #4d5061;
  }
</style>
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col">Columna #1</div>
      <div class="col">Columna #2</div>
      <div class="col">Columna #3</div>
      <div class="col">Columna #4</div>
    </div>
  </div>
</body>
```

El resultado:

a

Columna #1	Columna #2	Columna #3	Columna #4
------------	------------	------------	------------

Voy a invertir el orden de las columnas:

```
<div class="container">
  <div class="row">
    <div class="col order-4">Columna #1</div>
    <div class="col order-3">Columna #2</div>
    <div class="col order-2">Columna #3</div>
    <div class="col order-1">Columna #4</div>
  </div>
</div>
```

El resultado:

Columna #4	Columna #3	Columna #2	Columna #1
------------	------------	------------	------------

También puedo hacer que si la pantalla es pequeña y muy pequeña el orden sea el invertido y si la pantalla tiene un tamaño mediano el orden sea normal:

```
<body>
  <div class="container">
    <div class="row">
      <div class="col order-4 order-md-1">Columna #1</div>
      <div class="col order-3 order-md-2">Columna #2</div>
      <div class="col order-2 order-md-3">Columna #3</div>
      <div class="col order-1 order-md-4">Columna #4</div>
    </div>
  </div>
</body>
```

Resultado pantalla grande:

Columna #1	Columna #2	Columna #3	Columna #4
------------	------------	------------	------------

Resultado pantalla pequeña

Columna #4	Columna #3	Columna #2	Columna #1
------------	------------	------------	------------

2. Clase offset. Esta clase permite desplazar una columna.  
Supongamos el archivo offset.html con el siguiente código:

```
<div class="row">  
  <div class="columna col-3 offset-3">Columna</div>  
</div>
```

Se desplaza la columna 3 columnas.

Columna #1	Columna #2	Columna #3	Columna #4
	Columna		

Podemos hacer que cuando la pantalla sea extra-pequeña la columna se desplace 3 y de pequeña en adelante se desplace de 6.

```
<div class="row">  
  <div class="columna col-3 offset-3 offset-sm-6">Columna</div>  
</div>
```

El resultado será el siguiente cuando la pantalla sea pequeña, mediana, etc.

	Columna	
--	---------	--

Si la pantalla fuera extra pequeña:

Columna #4	Columna #3	Columna #2	Columna #1
------------	------------	------------	------------

	Columna	
--	---------	--

Se pueden aplicar offset a varias columnas. Para ello, se tiene el siguiente ejemplo:

```
<div class="row">  
  <div class="columna col-3 offset-3">Columna1</div>  
  <div class="columna col-3 offset-3">Columna2</div>  
</div>
```

	Columna1		Columna2
--	----------	--	----------

Ejercicio:

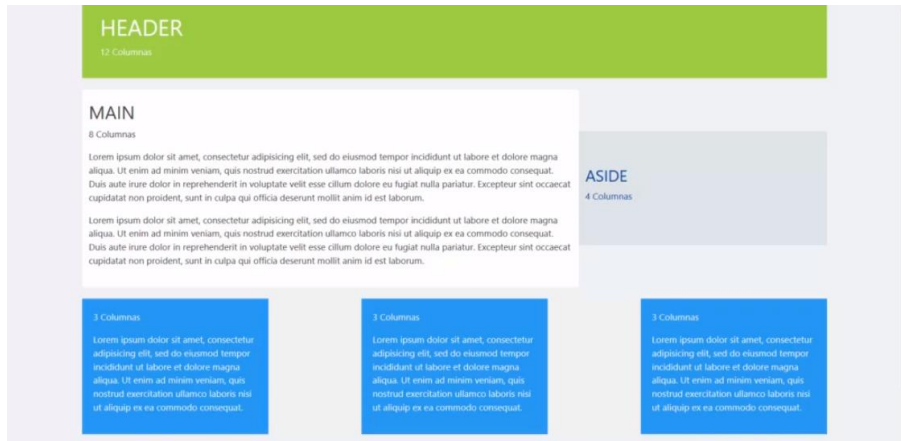
Diseñar la página web con Bootstrap. Consta de un header o un encabezado que tiene un tamaño de 12 columnas, tiene un main o un contenido principal que tiene un tamaño de 8 columnas junto con aside de cuatro columnas.

Después, debajo vamos a tener otra sección donde vamos a tener tres columnas y cada uno va a tener un tamaño de tres columnas y el espacio restante lo vamos a dividir entre cada una de ellas.

Esta es la estructura básica y también queremos que tenga un tamaño responsive.

- Es decir, se pretende que cuando bajemos el tamaño, estas columnas, por ejemplo, que eran de tres pasen a ser de cuatro.
- Y el espacio vacío, pues obviamente va a dejar de existir y va a pasar a ser parte de las propias columnas.
- También en la parte de arriba, cuando llegue a un tamaño pequeño o a lo mejor mediano, quiero que el contenido principal tenga 12 columnas.

- Luego el aside, quiero que tenga 12 columnas y cada uno de estos elementos quiero que tenga 12 columnas.  
De esta forma voy a llegar a lo que sería el tamaño de móvil donde se va a haber perfecto.



Además, esta página consta de un footer de 12 columnas donde tiene que aparecer tu nombre.

## 4. TIPOGRAFÍAS

En Bootstrap los encabezados tienen los siguientes estilos:

```
<body>
  <div class="container">
    <div class="row">
      <div class="col">
        <h1>Encabezado H1</h1>
        <h2>Encabezado H2</h2>
        <h3>Encabezado H3</h3>
        <h4>Encabezado H4</h4>
        <h5>Encabezado H5</h5>
        <h6>Encabezado H6</h6>
      </div>
    </div>
  </div>
```

El resultado será:

# Encabezado H1

## Encabezado H2

### Encabezado H3

#### Encabezado H4

##### Encabezado H5

###### Encabezado H6

Podemos añadir textos secundarios a estos encabezados. Para ello, voy a utilizar una etiqueta small utilizando una clase text-muted que lo que hace es cambiar el color.

```
<body>
  <div class="container">
    <div class="row">
      <div class="col">
        <h1>Encabezado H1 <small class="text-muted">Texto secundario</small> </h1>
        <h2>Encabezado H2<small class="text-muted">Texto secundario</small></h2>
        <h3>Encabezado H3 <small class="text-muted">Texto secundario</small></h3>
        <h4>Encabezado H4 <small class="text-muted">Texto secundario</small></h4>
        <h5>Encabezado H5 <small class="text-muted">Texto secundario</small></h5>
        <h6>Encabezado H6 <small class="text-muted">Texto secundario</small></h6>
      </div>
    </div>
  </div>
```

# Encabezado H1 Texto secundario

## Encabezado H2 Texto secundario

### Encabezado H3 Texto secundario

#### Encabezado H4 Texto secundario

##### Encabezado H5 Texto secundario

###### Encabezado H6 Texto secundario

Existe una clase que nos permite poner los párrafos (p) con el mismo estilo que los encabezados. Estas clases son h1, h2, h3, h4, h5 y h6

```
<div class="col">
  <p class="h1">Párrafo H1 <small class="text-muted">Texto secundario</small> </p>
  <p class="h2">Párrafo H2<small class="text-muted">Texto secundario</small></p>
  <p class="h3">Párrafo H3 <small class="text-muted">Texto secundario</small></p>
  <p class="h4">Párrafo H4 <small class="text-muted">Texto secundario</small></p>
  <p class="h5">Párrafo H6 <small class="text-muted">Texto secundario</small></p>
  <p class="h6">Párrafo 45 <small class="text-muted">Texto secundario</small></p>
</div>
```

---

Encabezado H1 Texto secundario

Encabezado H2 Texto secundario

Encabezado H3 Texto secundario

Encabezado H4 Texto secundario

Encabezado H5 Texto secundario

Encabezado H6 Texto secundario

---

Párrafo H1 Texto secundario

Párrafo H2 Texto secundario

Párrafo H3 Texto secundario

Párrafo H4 Texto secundario

Párrafo H6 Texto secundario

Párrafo 45 Texto secundario

En los banners promocionales de los productos que ofrece una página web:

```
<div class="col">
  <h1 class="display-1">Display 1 </h1>
  <h2 class="display-2">Display 2</h2>
  <h3 class="display-3">Display 3</h3>
  <h4 class="display-4">Display 4 </h4>
  <h5 class="display-5">Display 5</h5>
  <h6 class="display-6">Display 6 </h6>
</div>
```

Y el resultado será:

---

Display 1

Display 2

Display 3

Display 4

Display 5

Display 6

El formato de un párrafo lo puedo cambiar con la clase lead:

Esto es un párrafo de prueba

Esto es un párrafo de prueba

```
<div class="col">
  <p>Esto es un párrafo de prueba</p>
  <p class="lead">Esto es un párrafo de prueba</p>
</div>
```

```
<hr>
<div class="col">
  <figure class="text-center">
    <blockquote class="blockquote">
      <p>Esto es una segunda prueba</p>
    </blockquote>
  </figure>
</div>
```

Encabezado H1

Texto secundario

Encabezado H2

Texto secundario

Encabezado H3

Texto secundario

Encabezado H4

Texto secundario

Encabezado H5

Texto secundario

Encabezado H6

Texto secundario

Párrafo H1

Texto secundario

Párrafo H2

Texto secundario

Párrafo H3

Texto secundario

Párrafo H4

Texto secundario

Párrafo H6

Texto secundario

Párrafo 45

Texto secundario

Display 1

Display 2

Display 3

Display 4

Display 5

Display 6

Esto es un párrafo de prueba

Esto es un párrafo de prueba

Esto es una segunda prueba



Encabezado H1

Texto secundario

Encabezado H2

Texto secundario

Encabezado H3

Texto secundario

Encabezado H4

Texto secundario

Encabezado H5

Texto secundario

Encabezado H6

Texto secundario

Párrafo H1

Texto secundario

Párrafo H2

Texto secundario

Párrafo H3

Texto secundario

Párrafo H4

Texto secundario

Párrafo H6

Texto secundario

Párrafo 45

Texto secundario

Display 1

Display 2

Display 3

Display 4

Display 5

Display 6

Esto es un párrafo de prueba

Esto es un párrafo de prueba

```
<figure class="text-center">
  <blockquote class="blockquote">
    <p>Esto es una segunda prueba</p>
  </blockquote>
  <figcaption class="blockquote-footer">
    Esto es una <cite title="Titulo"> prueba</cite> tercera
  </figcaption>
</figure>
```

Esto es una segunda prueba

— Esto es una prueba tercera

Etiquetas para párrafos:

Esto es una prueba

~~Esto es una prueba 1~~

~~Esto es una prueba 1~~

Esto es una prueba 1

Estos es una prueba 1

**Esto es una prueba**

*Esto es una prueba 1*

Para los párrafos se pueden conseguir los efectos anteriores con Bootstrap utilizando clases:

```
<div class="col-6">
  <p class="fw-bold">Esto es una prueba 2</p>
  <p class="fw-normal">Esto es normal</p>
  <p class="fw-light">Esto es texto ligero</p>
  <p class="fst-italic">Esto es itálica </p>
  <p class="fst-normal">Este es fst</p>
</div>
```

---

## Esto es una prueba 2

Esto es normal

Esto es texto ligero

*Esto es itálica*

Este es fst

Para conseguir textos subrayados:

```
<div class="col-6">
  <p class="fw-bold">Esto es una prueba 2</p>
  <p class="fw-normal">Esto es normal</p>
  <p class="fw-light">Esto es texto ligero</p>
  <p class="fst-italic">Esto es itálica </p>
  <p class="fst-normal">Este es fst</p>
  <p class="text-decoration-underline"> Texto subrayado </p>
  <p class="text-decoration-line-trough"></p>
  <a href="#" class="text-decoration-none">Enlace</a>
</div>
```

---

## Esto es una prueba 2

Esto es normal

Esto es texto ligero

*Esto es itálica*

Este es fst

Texto subrayado

[Enlace](#)

Sólo nos falta centrar y alinear el texto:

```
<div class="col-12">
  <hr>
  <p class="text-start">Texto alineado a la izquierda</p>
  <p class="text-center">Texto centrado</p>
  <p class="text-end">Texto alineado a la derecha</p>
</div>
```

```
<div class="col-12">
  <hr>
  <p class="text-star">Texto alineado a la izquierda</p>
  <p class="text.center">Texto centrado</p>
  <p class="text.end">Texto alineado a la derecha</p>
</div>
```

OLA DE DEPURACIÓN TERMINAL Filtro

Más ejemplos para utilizar las clases anteriores en función del tamaño de la pantalla:

```
<div class="col">  
  <p class="text-start text-sm-end"> Texto alineado a la derecha en dispositivos muy pequeños</p>  
  <p class="text-start text-md-end"> Texto alineado a la derecha en dispositivos medianos</p>  
  <p class="text-start text-lg-end"> Texto alineado a la derecha en dispositivos grandes</p>  
  <p class="text-start text-xl-end"> Texto alineado a la derecha en dispositivos extra grandes</p>  
</div>
```

Para trabajar con textos en mayúsculas, minúsculas y letras capitales:

```
<div class="col">  
  <p class="text-lowercase">Minúsculas. Esto es un ejemplo de Minúsculas</p>  
  <p class="text-uppercase">Mayúsculas. Esto es un ejemplo de MAYUSCULAS</p>  
  <p class="text-capitalize">Letras capitales. Esto es un ejemplo de letras capitales</p>  
</div>
```

Para modificar el tamaño de la fuente de un párrafo se utilizará la clase fs:

---

Tamaño de fuente 1

Tamaño de fuente 2

Tamaño de fuente 3

Tamaño de fuente 4

Tamaño de fuente 5

Tamaño de fuente 6



```
<div class="col">
  <p class="fs-1">Tamaño de fuente 1</p>
  <p class="fs-2"> Tamaño de fuente 2</p>
  <p class="fs-3"> Tamaño de fuente 3</p>
  <p class="fs-4"> Tamaño de fuente 4</p>
  <p class="fs-5"> Tamaño de fuente 5</p>
  <p class="fs-6"> Tamaño de fuente 6</p>
</div>
```

Para cambiar el color del párrafo y ponerle un fondo se podrían utilizar los siguientes efectos:

```
<div class="col-6">
  <p class="text-primary">Texto primario</p>
  <p class="text-secondary">Texto secundario</p>
  <p class="text-success"> Texto success</p>
  <p class="text-danger"> Texto danger</p>
  <p class="text-dark">Texto dark</p>
  <p class="text-body"> Texto body</p>
  <p class="text-muted">Texto muteado</p>
</div>
<div class="col-6">
  <p class="text-warning bg-dark">Esto es una prueba</p>
  <p class="text-info bg-dark">Esto es una prueba</p>
  <p class="text-light bg-dark">Esto es una prueba</p>
  <p class="text-white bg-dark">Esto es una prueba</p>
</div>
>
```

```
src="/js/bootstrap.bundle.min.js"></script>
```

Esto es una prueba

Esto es una prueba

Esto es una prueba

Esto es una prueba

5 palabras | Español (España) | Accesibilidad: es necesario investigar

Para obtener una opacidad del 50% se podría utilizar la siguiente clase:

```
<p class="text-black-50">Opacidad del 50%</p>
```

Opacidad del 50%

## 5. IMÁGENES

Para utilizar imágenes si ponemos el siguiente código:

```
<body>
  <div class="container">
    <div class="row">
      <div class="col-6">
        
      </div>
    </div>
  </div>
  <script src="./js/bootstrap.bundle.min.js"></script>
</body>
```

Si ejecutamos el fichero se puede comprobar que la imagen no ocupa 6. Para obtener el comportamiento adecuado se debe utilizar la clase `img-fluid`.

```
<body>
  <div class="container">
    <div class="row">
      <div class="col-6">
        
      </div>
    </div>
  </div>
  <script src="./js/bootstrap.bundle.min.js"></script>
</body>
```

De esta forma se obtiene la imagen que ocupa una columna de 6:



Si se tienen 2 imágenes y queremos trabajar con float (por ejemplo una imagen dejarla a la izquierda y la otra a la derecha) tendríamos dos métodos para trabajar:

- Crear columnas y luego trabajar con flex-box
- O utilizar las clases float-start y float-end:

```
<div class="col-12">
  
  
</div>
```

Y el resultado será:



Para agregar imágenes que debajo tengan un texto con una descripción:

```
<div class="row">
  <div class="col-12">
    <figure class="figure">
      
      <figcaption class="figure-caption">
        <p class="text-center"> Bienvenido al mundo real</p>
      </figcaption>
    </figure>
  </div>
</div>
```

Hay que utilizar la clase figure dentro de la etiqueta figure y la clase figure-caption para poder insertar el texto justo debajo de la imagen. La clase rounded es para rodear las esquinas de la imagen.

## 6. TABLAS

Para crear tablas en html:

```
<div class="container">
  <div class="row">
    <div class="col">
      <table>
        <thead>
          <tr>
            <th>Nombre</th>
            <th>Edad</th>
            <th>Pais</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>Maria</td>
            <td>34</td>
            <td>España</td>
          </tr>
          <tr>
            <td>Sofía</td>
            <td>43</td>
            <td>Italia</td>
          </tr>
          <tr>
            <td>Andrés</td>
            <td>54</td>
            <td>Portugal</td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</div>
```

El resultado es el siguiente:

---

**Nombre Edad Pais**

Maria 34 España

Sofía 43 Italia

Andrés 54 Portugal

Sin embargo si utilizo la clase table el aspecto de la tabla cambia:

```
<div class="col">
  <table class="table">
    <thead>
      <tr>
```

El resultado sería:

Nombre	Edad	País
Maria	34	España
Sofía	43	Italia
Andrés	54	Portugal

Otras clases que se pueden utilizar:

- Table-striped. Es una clase que permite poner otro color las filas impares.
- Table-hover. Cuando se pasa el cursor el aspecto de la fila varía.
- Table-sm. Compacta la tabla.

Nombre	Edad	País
Maria	34	España
Sofía	43	Italia
Andrés	54	Portugal

El código será:

```
<div class="col">  
<table class="table table-striped table-hover table-sm">  
<thead>
```

Para añadir una pequeña leyenda a la tabla:

```
table <class="table table-striped table-hover table-sm">  
<caption>Tabla de mis clientes</caption>  
<thead>
```

Para modificar los colores se puede utilizar la clase table-dark:

```
<thead>  
<tr class="table-dark">  
<th>Nombre</th>  
<th>Edad</th>  
<th>País</th>  
</tr>
```

Para hacer la tabla responsive, la introduzco dentro de un div y utilizo la clase table-responsive:



```
<div class="container">
  <div class="row">
    <div class="col">
      <div class="table-responsive">
        <table class="table table-striped table-hover table-sm">...
      </table>
    </div>
  </div>
</div>
```

Para activar una celda se puede utilizar la clase table-active y esa celda se marcará de otro color. Podemos indicar el color que queramos con las clases:

- Table-primary
- Table-secondary
- Table-success
- Table-danger
- Table-warning
- Table-info
- Table-light
- Table-dark

Otras clases para tablas:

- Border-danger (tabla con bordes de una forma especial)
- Table-borderless (tabla sin bordes)

```
<thead>
  <tr class="table-dark">
    <th class="active table-danger">Nombre</th>
    <th>Edad</th>
    <th>Pais</th>
  </tr>
```

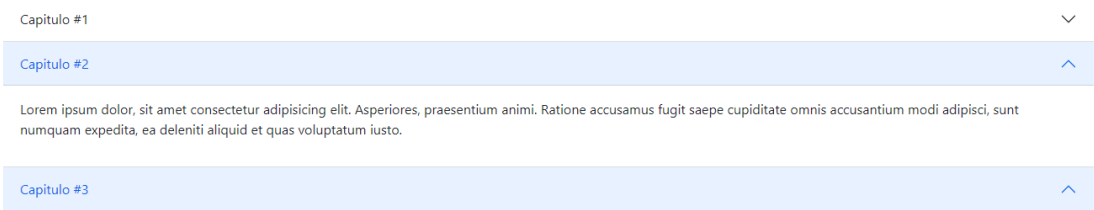
Si activamos una celda y se pone la clase table-danger el resultado es el siguiente:

Nombre	Edad	Pais
Maria	34	España
Sofia	43	Italia
Andrés	54	Portugal

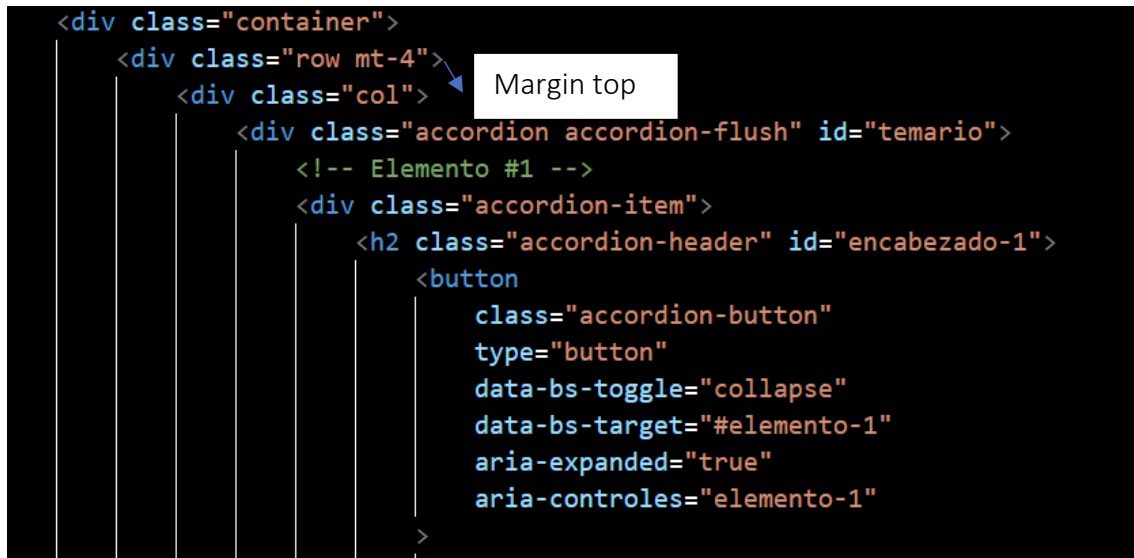
Tabla de mis clientes

## 7. ACORDEON

En esta actividad se pretende realizar un acordeón como el de la figura:



Su código sería el siguiente:



```

<div class="container">
  <div class="row mt-4">
    <div class="col">
      <div class="accordion accordion-flush" id="temario">
        <!-- Elemento #1 -->
        <div class="accordion-item">
          <h2 class="accordion-header" id="encabezado-1">
            <button
              class="accordion-button"
              type="button"
              data-bs-toggle="collapse"
              data-bs-target="#elemento-1"
              aria-expanded="true"
              aria-controls="elemento-1"
            >

```

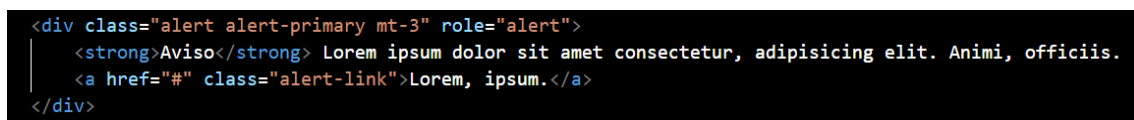
Se crea el primer elemento, el elemento encabezado de Capítulo 1. Para ello, utilizo la clase accordion. La clase accordion-flush lo que consigue es quitar los bordes. En la parte donde pone Capítulo1 se considera como un botón. Para identificar este elemento primero tengo que utilizar la clase accordion-item. El div que pone Capítulo1 es un botón que tiene una clase accordion-button,

## 8. ALERTAS

Muchas veces en nuestras páginas web vamos a querer mostrar alguna advertencia o alguna alerta a los usuarios.

Se tiene que crear un div, al cual simplemente, le vamos a poner la clase de alert y de esta forma ya estamos creando el componente de una alerta.

Para ello se utiliza la clase alert y se podría crear la alerta con colores, como por ejemplo primary.



```

<div class="alert alert-primary mt-3" role="alert">
  <strong>Aviso</strong> Lorem ipsum dolor sit amet consectetur, adipisicing elit. Animi, officiis.
  <a href="#" class="alert-link">Lorem, ipsum.</a>
</div>

```

Esto es importante otra vez para dispositivos de lectura y para dispositivos para personas con alguna discapacidad, por ejemplo visual. Esta alerta, se puede repetir para tener varios colores, entonces tenemos este color que sería el primary, y lo que voy a hacer es que la voy a repetir para tenerla más veces, porque tenemos el color primary, tenemos el secondary que es el secundario, tenemos el de success para a lo mejor mostrar algún aviso de que tuvo éxito alguna operación. Luego tenemos el de Danger para mostrar el de que hubo algún error. Después tenemos el de Warning para mostrar advertencias y tenemos aquí también el de info. Vamos a poner otro más, el Delight y me faltó uno más que sería el de Dark.

**Aviso** Lorem ipsum dolor sit amet consectetur, adipisicing elit. Animi, officiis. [Lorem, ipsum.](#)

**Aviso** Lorem ipsum dolor sit amet consectetur, adipisicing elit. Animi, officiis. [Lorem, ipsum.](#)

**Aviso** Lorem ipsum dolor sit amet consectetur, adipisicing elit. Animi, officiis. [Lorem, ipsum.](#)

Ahora estos componentes de alerta muchas veces vamos a querer, por ejemplo, que tengan un botón de cerrar para que el usuario cuando le dé un clic lo pueda cerrar. Podemos hacer una alerta que tenga ese botón para cerrarlo.

Otro ejemplo:

```
<div class="alert alert-success mt-3 alert-dismissible fade show" role="alert">
  <h3>Encabezado</h3>
  <hr>
  <strong>Aviso</strong> Lorem ipsum dolor sit amet consectetur, adipisicing elit.
  <a href="#" class="alert-link">Lorem, ipsum.</a>
  <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
</div>
```

Vamos a tener nuestra alerta, que como puedes ver tiene un encabezado. Esto simplemente es para demostrar que puedes poner lo que quieras. Tenemos nuestro texto y tenemos el botón.

## 9. MIGAS DE PAN O BREADCRUMBS

En este apartado se verá cómo crear “migas de pan” o breadcrumbs con Bootstrap. Las migas de pan es una pequeña navegación que sirve para indicarle al usuario en qué parte de su sitio web se encuentra.

Para ello, al documento html hay que añadir un NAV porque se trata de un menú de navegación y hay que poner la clase breadcrumb.

```
<div class="container">
  <div class="row">
    <div class="col">
      <nav class="breadcrumb mt-3">
        <a href="#" class="breadcrumb-item">Inicio</a>
        <a href="#" class="breadcrumb-item">Productos</a>
        <a href="#" class="breadcrumb-item">Tablets</a>
        <span class="breadcrumb-item active">Ipads</span>
      </nav>
    </div>
  </div>
</div>
```

Esto podría ser un enlace  
<a></a>

El resultado sería:

[Inicio](#) / [Productos](#) / [Tablets](#) / [Ipads](#)

## 10. BOTONES

Para crear botones en Bootstrap vamos a utilizar la etiqueta `button` y la clase `btn`. Para colorear estos botones podemos poner `btn-primary`, `btn-secondary`, `btn-success`, etc. Los colores que se establecen con las fuentes, alertas, acordeones, etc. En el código que se muestra a continuación aparecen todos los colores posibles:

```
<div class="row">
  <div class="col">
    <button type="button" class="btn btn-primary">Primario</button>
    <button type="button" class="btn btn-secondary">Secundario</button>
    <button type="button" class="btn btn-success">Exito</button>
    <button type="button" class="btn btn-danger">Peligro</button>
    <button type="button" class="btn btn-warning">Precaucion</button>
    <button type="button" class="btn btn-info">Informacion</button>
    <button type="button" class="btn btn-light">Claro</button>
    <button type="button" class="btn btn-dark">Oscuro</button>
  </div>
</div>
```

El resultado es el siguiente:



Se podría poner un botón con apariencia de enlace para ello se utiliza la clase `btn-link`

[Oscuro](#)

```
<button type="button" class="btn btn-link">Oscuro</button>
```

Para conseguir otro aspecto de los botones se podría utilizar la clase `btn-outline-primary`, `btn-outline-secondary`, etc.

```
<div class="col-12">
  <hr class="mt-3">
  <button type="button" class="btn btn-outline-primary">Primario</button>
  <button type="button" class="btn btn-outline-secondary">Secundario</button>
  <button type="button" class="btn btn-outline-success">Exito</button>
  <button type="button" class="btn btn-outline-danger">Peligro</button>
  <button type="button" class="btn btn-outline-warning">Precaucion</button>
  <button type="button" class="btn btn-outline-info">Informacion</button>
  <button type="button" class="btn btn-outline-light text-dark">Claro</button>
  <button type="button" class="btn btn-outline-dark">Oscuro</button>
  <button type="button" class="btn btn-outlin-link">Oscuro</button>
</div>
```

El resultado es el siguiente:



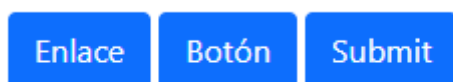
La clase disabled se utiliza para deshabilitar un botón:

```
<div class="col-12">
  <hr class="mt-3">
  <button type="button" class="btn disabled btn-outline-primary">Primario</button>
  <button type="button" class="btn disabled btn-outline-secondary">Secundario</button>
  <button type="button" class="btn disabled btn-outline-success">Exito</button>
  <button type="button" class="btn disabled btn-outline-danger">Peligro</button>
  <button type="button" class="btn disabled btn-outline-warning">Precaucion</button>
  <button type="button" class="btn disabled btn-outline-info">Informacion</button>
  <button type="button" class="btn disabled btn-outline-light text-dark">Claro</button>
  <button type="button" class="btn disabled btn-outline-dark">Oscuro</button>
  <button type="button" class="btn disabled btn-outlin-link">Oscuro</button>
</div>
```

El resultado son botones desactivados.

Se pueden tener distintos botones sin ser la etiqueta enlace, es decir, podremos convertir a button un enlace, una etiqueta input de tipo botón y submit, Esto se aprecia en el código:

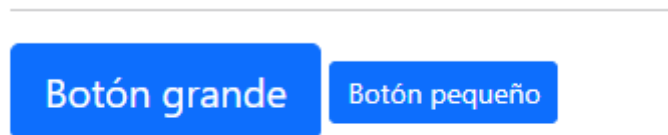
```
<div class="col-12">
  <hr class="mt-3">
  <a href="#" class="btn btn-primary">Enlace</a>
  <input type="button" class="btn btn-primary" value="Botón">
  <input type="submit" class="btn btn-primary" value="Submit">
</div>
```



Se podría modificar el tamaño de botón:

```
<div class="col-12">
  <hr class="mt-3">
  <button type="button" class="btn btn-lg btn-primary">Botón grande</button>
  <button type="button" class="btn btn-sm btn-primary">Botón pequeño</button>
</div>
```

El resultado sería:



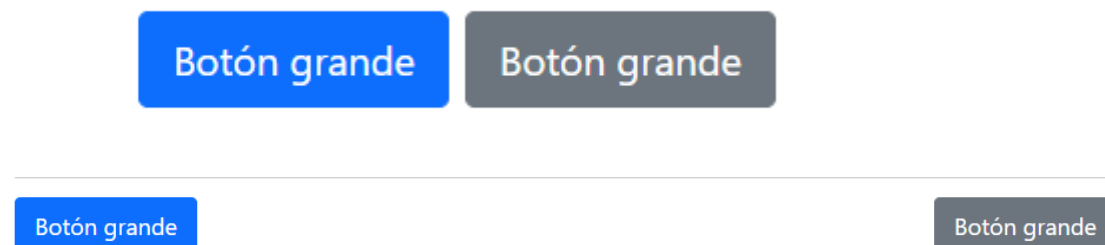
Para hacer que dos botones abarquen el 100% del espacio disponible:

```
</div>
<div class="col-12">
  <hr class="mt-3">
  <div class="d-grid gap-2 col-12">
    <button type="button" class="btn btn-lg btn-primary">Botón grande</button>
    <button type="button" class="btn btn-lg btn-secondary">Botón grande</button>
  </div>
</div>
```

Para hacer esto, se introducen la clase d-grid y gap-2 (que es para poner una separación entre los botones).

Cambiar el posicionamiento de los botones:

```
<div class="col-12">
  <hr class="mt-3">
  <div class="d-flex gap-2 justify-content-between">
    <button type="button" class="btn btn-lg btn-primary">Botón grande</button>
    <button type="button" class="btn btn-lg btn-secondary">Botón grande</button>
  </div>
</div>
```



```

<div class="col-12">
  <hr class="mt-3">
  <div class="d-flex gap-2 justify-content-end">
    <button type="button" class="btn btn-lg btn-primary">Botón grande</button>
    <button type="button" class="btn btn-lg btn-secondary">Botón grande</button>
  </div>

```

También se puede poner justify-content-center.

## 11. CARRUSEL

Para crear un carrusel debemos crear un contenedor al que le asignaremos la clase carousel slide y un data-bs-ride= carousel. Además, hay que asignar un id para identificar el div.

Para la parte interior del carrusel se tiene que crear otro div con la clase carousel-inner. Dentro de este div podremos crear cada uno de los ítems que forman el slide.

Al final el código quedaría de la siguiente forma:

```

<div class="row slider">
  <div class="col">
    <div class="carousel slide" data-bs-ride="carousel" id="slider">
      <div class="carousel-inner">
        <div class="carousel-item active">
          
        </div>
        <div class="carousel-item">
          
        </div>
        <div class="carousel-item">
          
        </div>
      </div>
    </div>
  </div>

```

Es obligatorio para que se carguen bien las imágenes que uno de los carousel-item tenga la clase active.

Para crear un botón para el lado derecho y otro para el lado izquierdo:

```

<button class="carousel-control-prev" type="button" data-bs-target="#slider">
  <span class="carousel-control-prev-icon" aria-hidden="true"></span>
  <span class="visually-hidden">Anterior</span>
</button>
<button class="carousel-control-next" type="button" data-bs-target="#slider">
  <span class="carousel-control-next-icon" aria-hidden="true"></span>
  <span class="visually-hidden">Siguiente</span>
</button>

```

La clase visually-hidden estará el span oculto y se utiliza para usuarios que no pueden ver los botones y podrán escuchar Anterior y Siguiente.

Para poner indicadores para saber en que parte del slide nos encontramos:

Con

```
<div class="carousel-indicators">
  <button
    type="button"
    data-bs-target="#slider"
    data-bs-slide-to="0"
    class="active"
    aria-current="true"
    aria-label="Slideshow 1"
  ></button>
  <button
    type="button"
    data-bs-target="#slider"
    data-bs-slide-to="1"
    aria-label="Slideshow 2"
  ></button>
  <button
    type="button"
    data-bs-target="#slider"
    data-bs-slide-to="2"
    aria-label="Slideshow 3"
  ></button>
```

Con data-bs-slide-to=2 se está diciendo a que imagen se va. Siempre se empieza por 0.

## 12. FORMULARIOS

Vamos a crear el siguiente formulario:



Correo electrónico

Te enviaremos un mensaje de confirmación

Mensaje

Este formulario estaría formado por unas etiquetas, un input, una caja de texto y un botón.

La clase para dar formato en Bootstrap a inputs y textarea es **form-control**.

La clase para las etiquetas es **form-label**.

El código para obtener los elementos de la figura sería el siguiente:

Para que aparezca debajo del input el comentario “Te enviaremos un mensaje de confirmación” se utilizará un div asociado al aria-describedby. Si no quisiéramos que apareciera ese comentario no sería necesario poner en el input el atributo aria-describedby.

```

<form action="">
  <div class="mb-3">
    <label for="correo" class="form-label">Correo
    electrónico</label>
    <input type="text" class="form-control" id="correo" aria-
    describedby="ayuda-correo" placeholder="Correo">
    <div id="ayuda-correo" class="form-text">
      Te enviaremos un mensaje de confirmación
    </div>
  </div>
  <div class="mb-3">
    <label for="mensaje" class="form-label">Mensaje</label>
    <textarea name="mensaje" class="form-control" id="mensaje">
    </textarea>
  </div>
  <button type="submit" class="btn btn-primary">Enviar comentario
</button>
</form>

```

Se utilizarán las clase form-select-lg y form-select-sm para que el selectbox sea el formato grande o pequeño.

```

<div class="container">
  <div class="row mt-3">
    <select name="pais" id="pais" class="form-select mb-3" aria-label="Selección de paí:
      <option value="mexico">México</option>
      <option value="españa">España</option>
      <option value="colombia">Colombia</option>
    </select>

    <select name="pais" id="pais" class="form-select form-select-lg mb-3" aria-label="Se
      <option value="mexico">México</option>
      <option value="españa">España</option>
      <option value="colombia">Colombia</option>
    </select>

    <select name="pais" id="pais" class="form-select form-select-sm mb-3" aria-label="Se
      <option value="mexico">México</option>
      <option value="españa">España</option>
      <option value="colombia">Colombia</option>
    </select>
  </div>
</div>

```

Se puede utilizar el atributo múltiple para que en el elemento select puedan elegir más de un elemento. Con size se indica el tamaño de la selección.

Con la opción disabled el selectbox estará desahabilitado.

```

<select multiple size="3" name="pais" id="pais" class="form-select mb-3" aria-label:
  <option value="mexico">México</option>
  <option value="españa">España</option>
  <option value="colombia">Colombia</option>
  <option value="peru">Peru</option>
  <option value="argentina">Argentina</option>
</select>

<select disabled name="pais" id="pais" class="form-select mb-3" aria-label="Selecció
  <option value="mexico">México</option>
  <option value="españa">España</option>
  <option value="colombia">Colombia</option>
</select>

```

### 13.1. CHECKBOX

La clase para un input para un checkbox será **form-check**. Estos inputs para los checkbox se debería estar dentro de un div. Si se quiere utilizar una etiqueta

```

<div class="col-4">
  <div class="form-check">
    <input type="checkbox" name="checkbox-1" class="form-check-input" id="checkbox-1">
    <label for="checkbox-1" class="form-check-label">Checkbox 1</label>
  </div>
  <div class="form-check">
    <input type="checkbox" name="checkbox-2" class="form-check-input" id="checkbox-2">
    <label for="checkbox-2" class="form-check-label">Checkbox 2</label>
  </div>
  <div class="form-check">
    <input type="checkbox" name="checkbox-3" class="form-check-input" id="checkbox-3">
    <label for="checkbox-3" class="form-check-label">Checkbox 3</label>
  </div>
</div>

```

### 13.2. RADIOBUTTON

```

<!-- Radio Buttons -->
<div class="col-4">
  <div class="form-check">
    <input type="radio" name="mi-radio-button" class="form-check-input" id="radio-1">
    <label for="radio-1" class="form-check-label">Radio 1</label>
  </div>
  <div class="form-check">
    <input type="radio" name="mi-radio-button" class="form-check-input" id="radio-2">
    <label for="radio-2" class="form-check-label">Radio 2</label>
  </div>
  <div class="form-check">
    <input type="radio" name="mi-radio-button" class="form-check-input" id="radio-3">
    <label for="radio-3" class="form-check-label">Radio 3</label>
  </div>
</div>

```

Para crear los radiobutton también tengo que crear un div con la clase **form-select** y dentro tendré el input con la clase **form-check-input**.

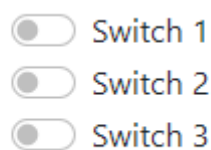
### 13.3. SWITCH

```

<!-- Switch -->
<div class="col-4">
  <div class="form-check form-switch">
    <input type="checkbox" name="switch-1" class="form-check-input" id="switch-1">
    <label for="switch-1" class="form-check-label">Switch 1</label>
  </div>
  <div class="form-check form-switch">
    <input type="checkbox" name="switch-2" class="form-check-input" id="switch-2">
    <label for="switch-2" class="form-check-label">Switch 2</label>
  </div>
  <div class="form-check form-switch">
    <input type="checkbox" name="switch-3" class="form-check-input" id="switch-3">
    <label for="switch-3" class="form-check-label">Switch 3</label>
  </div>
</div>

```

Con este código se consigue el siguiente elemento:



### 13.4. CHECKBOX INLINE

Para trabajar con un checkbox en inline:

```

-- Checkbox Inline -->
iv class="col-12 mt-4">
  <div class="form-check form-check-inline">
    <input type="checkbox" name="checkbox-inline-1" class="form-check-input" id="checkbox-inline-1">
    <label for="checkbox-inline-1" class="form-check-label">Checkbox 1</label>
  </div>
  <div class="form-check form-check-inline">
    <input type="checkbox" name="checkbox-inline-2" class="form-check-input" id="checkbox-inline-2">
    <label for="checkbox-inline-2" class="form-check-label">Checkbox 2</label>
  </div>
  <div class="form-check form-check-inline">
    <input type="checkbox" name="checkbox-inline-3" class="form-check-input" id="checkbox-inline-3">
    <label for="checkbox-inline-3" class="form-check-label">Checkbox 3</label>
  </div>
</div>

```

### 13.5. CHECKBOX ESTILO BOTÓN

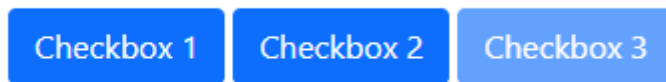
```

-- Checkbox estilo boton -->
iv class="col-6 mt-4">
  <input type="checkbox" name="checkbox-button-1" class="btn-check" id="checkbox-button-1">
  <label for="checkbox-button-1" class="btn btn-primary">Checkbox 1</label>

  <input type="checkbox" name="checkbox-button-2" class="btn-check" id="checkbox-button-2">
  <label for="checkbox-button-2" class="btn btn-primary">Checkbox 2</label>

  <input type="checkbox" name="checkbox-button-3" class="btn-check" id="checkbox-button-3" disabled>
  <label for="checkbox-button-3" class="btn btn-primary">Checkbox 3</label>

```



### 13.6. RADIOBUTTON ESTILO BOTÓN

```
<!-- Radio estilo boton -->
<div class="col-6 mt-4">

  <input type="radio" name="radio-button" class="btn-check" id="radio-button-1">
  <label for="radio-button-1" class="btn btn-primary">Radio 1</label>

  <input type="radio" name="radio-button" class="btn-check" id="radio-button-2">
  <label for="radio-button-2" class="btn btn-primary">Radio 2</label>

  <input type="radio" name="radio-button" class="btn-check" id="radio-button-3">
  <label for="radio-button-3" class="btn btn-primary">Radio 3</label>

</div>
```

### 13.7. RANGE

Para dar estilo a este elemento de tipo range de un formulario se debe usar la clase form-range.

```
<div class="container">
  <div class="row mt-3">
    <div class="col-6">
      <label for="rango" class="form-label">Presupuesto</label>
      <input type="range" class="form-range" id="rango" min="0" max="1000" step="100">
      <div class="d-flex justify-content-between">
        <p class="form-text">$0</p>
        <p class="form-text">$1000</p>
      </div>
    </div>
  </div>
</div>
```



### 13.8. Etiquetas flotantes

El efecto de las etiquetas flotantes es que cuando se pulsa sobre ella se hace más pequeña y se podrá escribir.

```

<div class="container">
  <div class="row mt-3">
    <div class="col-6">
      <div class="form-floating mb-3">
        <input type="text" name="nombre" id="nombre" class="form-control" placeholder="">
        <label for="nombre">Nombre</label>
      </div>

      <div class="form-floating mb-3">
        <textarea name="notas" id="notas" class="form-control" placeholder="Notas"></textarea>
        <label for="notas">Notas</label>
      </div>

      <div class="form-floating mb-3">

```

### 13.9. Validación de formularios

Para validar o invalidar un formulario en Bootstrap existen clases para poder realizarlo de forma rápida y sencilla.

```

</head>
<body>
  <div class="container">
    <div class="row mt-5">
      <div class="col-6">
        <form action="">
          <label for="nombre" class="form-label">Nombre</label>
          <input type="text" class="form-control is-valid" placeholder="No">
          <div class="valid-feedback">Correcto!</div>
        </form>
      </div>

      <div class="col-6">
        <form action="">
          <label for="correo" class="form-label">Correo</label>
          <input type="email" class="form-control is-invalid" placeholder="No">
          <div class="invalid-feedback">Por favor introduce un correo válido</div>
        </form>
      </div>
    </div>
  </div>

```

Para validar el formulario tenemos que añadir a la clase del elemento (en este caso, caja de texto) la clase `is-valid` o `is-invalid` para la no validación.

El resultado del correo anterior sería el siguiente:

<p>Nombre</p> <div style="border: 1px solid #28a745; padding: 5px; display: inline-block;">Nombre</div> <div style="color: #28a745; font-size: 1.2em; margin-left: 10px;">✓</div> <p>Correcto!</p>	<p>Correo</p> <div style="border: 1px solid #dc3545; padding: 5px; display: inline-block;">correo@correo.com</div> <div style="color: #dc3545; font-size: 1.2em; margin-left: 10px;">!</div> <p>Por favor introduce un correo válido.</p>
--	---

La clase `invalid-feedback` es para establecer un feedback con el usuario si lo que ha escrito es correcto o no.

## 14. VENTANA MODAL EN BOOTSTRAP

Podemos crear o abrir ventanas sin utilizar JavaScript al pulsar un botón para ello se debe utilizar las siguientes clases:

- Lo primero que tengo que hacer es crear un botón:

```
<div class="container">
  <div class="row mt-5">
    <div class="col">
      <button
        class="btn btn-primary btn-lg"
        data-bs-toggle="modal"
        data-bs-target="#mi-modal-1"
      >
        Abrir Modal 1
      </button>
    </div>
  </div>
</div>
```

A este botón con data-bs-toggle="modal" le decimos que es modal y que cuando pulsamos se abra una ventana emergente cuyo contenido sea el id mi-modal-1. Para establecer la ventana emergente tengo que crear un div con id mi-modal-1

```
<div
  class="modal fade"
  id="mi-modal-1"
  tabindex="-1"
  aria-hidden="true"
  aria-labelledby="label-modal-1"
>
  <!-- Caja de dialogo -->
  <div class="modal-dialog">
    <!-- Contenido de la caja -->
    <div class="modal-content">
      <!-- Encabezado de la caja -->
      <div class="modal-header">
        <h5 class="modal-title">Titulo del modal</h5>
        <button class="btn-close" data-bs-dismiss="modal" aria-label="Cerrar"></button>
      </div>

      <!-- Cuerpo de la caja -->
      <div class="modal-body">
        <h1>Hola Mundo</h1>
      </div>

      <!-- Pie de pagina de la caja -->
      <div class="modal-footer">
        <button type="button" class="btn btn-light" data-bs-dismiss="modal">Cancelar</button>
        <button type="button" class="btn btn-primary">Acepto</button>
      </div>
    </div>
  </div>
</div>
```

Dentro de esa caja puedo crear un encabezado, un body y footer. Para ello, creo una caja con la clase modal-dialog cuyo contenido será otro div con la clase modal-content.

Para crear el encabezado dentro de este div debo crear otro div con la clase modal-header, para crear el cuerpo otro div con la clase modal-body y para el footer otro div con la clase modal-footer.

Dentro de la cabecera puedo poner el botón de cerrar con `button` y la clase `btn-close`.

Y el footer puedo crear los dos botones de Aceptar y Cancelar.

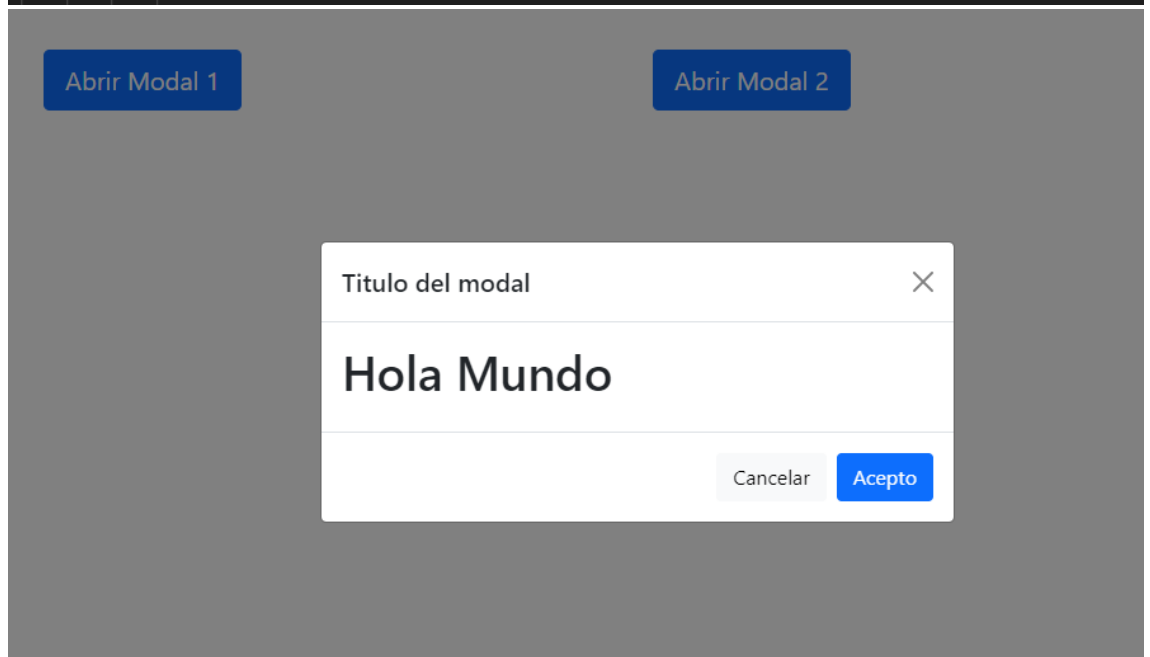
Para centrar la ventana emergente puedo utilizar la clase `modal-dialog-centered`:

Si quiero que la ventana no se cierre cuando pincho fuera de la ventana debo utilizar `data-bs-backdrop="static"`

El código sería el siguiente:

```
<div
  class="modal fade"
  id="mi-modal-1"
  tabindex="-1"
  aria-hidden="true"
  aria-labelledby="label-modal-1"
  data-bs-backdrop="static"

  <!-- Caja de dialogo -->
  <div class="modal-dialog modal-dialog-centered modal-dialog-scrollable">
    <!-- Contenido de la caja -->
    <div class="modal-content">
      <!-- Encabezado de la caja -->
      <div class="modal-header">
        <h5 class="modal-title">Titulo del modal</h5>
        <button class="btn-close" data-bs-dismiss="modal" aria-label="Cerrar"></button>
      </div>
      <!-- Cuerpo de la caja -->
```



## 15. FLEXBOX

La primera clase con la que se debe utilizar Bootstrap es `d-flex`. Esto significa que los elementos hijo del contenedor `d-flex` se van a comportar como elementos flex.



```

<div class="bg-primary text-white p-3 d-flex">
  <p class="mb-0">Display Flex <strong>.d-flex</strong></p>
</div>

<br>
<div class="bg-primary text-white p-3 d-inline-flex">
  <p class="mb-0">Display Inline Flex <strong>.d-inline-flex</strong></p>
</div>

```

Con la clase d-flex el contenedor ocupa el 100% y sin embargo con la clase d-inline-flex el contenedor abarca lo que ocupa.

Las clases más importantes Flex-Direction en BootStrap para Flex-Box:

- Flex-row: Es la clase que se aplica por defecto lo que hace es colocar cada caja una a continuación de la otra.
- Flex-row-reverse. Se cambia el flujo de posición de las cajas.
- Flex-column. Cada caja ocupa el 100%. Por lo tanto, va a colocar nuestros contenedores uno debajo del otro.
- Flex-column-reverse. Realizará lo mismo que en el caso anterior pero va a colocar las cajas en orden inverso.

```

<p>.flex-row</p>
<div class="contenedor d-flex flex-row">
  <div>Caja 1</div>
  <div>Caja 2</div>
  <div>Caja 3</div>
</div>

<p>.flex-row-reverse</p>
<div class="contenedor d-flex flex-row-reverse">
  <div>Caja 1</div>
  <div>Caja 2</div>
  <div>Caja 3</div>
</div>

<p>.flex-column</p>
<div class="contenedor d-flex flex-column">
  <div>Caja 1</div>
  <div>Caja 2</div>
  <div>Caja 3</div>
</div>

<p>.flex-column-reverse</p>
<div class="contenedor d-flex flex-column-reverse">
  <div>Caja 1</div>
  <div>Caja 2</div>
  <div>Caja 3</div>
</div>

```

Además, podemos hacer que todas estas clases de Flex-Box sean responsive:

### Variantes

```

.flex-row
.flex-row-reverse
.flex-column
.flex-column-reverse
sm | md | lg | xl | xxl

```

```
.flex-*-row  
.flex-*-row-reverse  
.flex-*-column  
.flex-*-column-reverse
```

Todas las clases vistas anteriormente se podrían hacer responsive con las opciones sm, md, lg, xl y xxl.

- Para alinear el contenido de forma horizontal utilizaremos las siguientes clases:
  - Justify-content-start (para alinear a la derecha)
  - Justify-content-center (alinear al centro)
  - Justify-content-end (alinear a la izquierda)
  - Justify-content-between (Separa equitativamente los elementos)
  - Justify-content-around (coloca un espacio a la derecha e izquierda del contenedor)
  - Justify-content-evenly (el espacio entre los elementos o contenedores es equitativo). La forma de trabajar es igual que la anterior, colocar un espacio a la derecha e izquierda del contenedor.

Un ejemplo de cómo utilizar estas clases:

```
<p>.justify-content-start</p>  
<div class="contenedor d-flex justify-content-start">  
  <div>Caja 1</div>  
  <div>Caja 2</div>  
  <div>Caja 3</div>  
</div>  
  
<p>.justify-content-center</p>  
<div class="contenedor d-flex justify-content-center">  
  <div>Caja 1</div>  
  <div>Caja 2</div>  
  <div>Caja 3</div>  
</div>  
  
<p>.justify-content-end</p>  
<div class="contenedor d-flex justify-content-end">  
  <div>Caja 1</div>  
  <div>Caja 2</div>  
  <div>Caja 3</div>  
</div>
```

Estas clases también se pueden hacer responsive:

```

<!-- Variantes:
    .justify-content-start
    .justify-content-end
    .justify-content-center
    .justify-content-between
    .justify-content-around

    sm | md | lg | xl | xxl
    .justify-content-*-start
    .justify-content-*-end
    .justify-content-*-center
    .justify-content-*-between
    .justify-content-*-around

```

- La clase flex-fill.

Se aplica a elementos hijos y va a rellenar el espacio libre

```

<div class="contenedor d-flex align-items-start">
  <div>Lorem ipsum dolor sit amet.</div>
  <div class="flex-fill">.flex-fill</div>
  <div class="flex-fill">.flex-fill</div>
</div>
</div>
</div>

```

El resultado sería:

### Fill

Lorem ipsum dolor sit amet.	.flex-fill	.flex-fill
-----------------------------	------------	------------

Esta clase tiene sus variantes en responsive.

```

<!--
    .flex-fill

    sm | md | lg | xl | xxl
    .flex-*-fill
-->

```

- Las clases con flex-box `align-items-start`, `align-items-end`, `align-items-center`, `align-items-baseline` y `align-items-stretch`.

Con estas clases colocamos los elementos en la parte de arriba, abajo, centrados, en la parte de arriba en función del texto y ocupando verticalmente el 100%.

```
<p>.align-items-start</p>
<div class="contenedor d-flex align-items-start">
  <div>Caja 1</div>
  <div>Caja 2</div>
  <div>Caja 3</div>
</div>

<p>.align-items-center</p>
<div class="contenedor d-flex align-items-center">
  <div>Caja 1</div>
  <div>Caja 2</div>
  <div>Caja 3</div>
</div>

<p>.align-items-end</p>
<div class="contenedor d-flex align-items-end">
  <div>Caja 1</div>
  <div>Caja 2</div>
  <div>Caja 3</div>
</div>
```

```
<p>.align-items-baseline</p>
<div class="contenedor d-flex align-items-baseline">
  <div>Caja 1</div>
  <div>Caja 2</div>
  <div>Caja 3</div>
</div>

<p>.align-items-stretch</p>
<div class="contenedor d-flex align-items-stretch">
  <div>Caja 1</div>
  <div>Caja 2</div>
  <div>Caja 3</div>
</div>
</div>
```

El resultado es el siguiente:

`.align-items-start`



`.align-items-center`



`.align-items-end`



•

`.align-items-baseline`



`.align-items-strech`



•

Opciones responsive:

`<!-- Variantes:`

```
.align-items-start  
.align-items-end  
.align-items-center  
.align-items-baseline  
.align-items-stretch
```

```
sm | md | lg | xl | xxl  
.align-items-*-start  
.align-items-*-end  
.align-items-*-center  
.align-items-*-baseline  
.align-items-*-stretch
```

`-->`

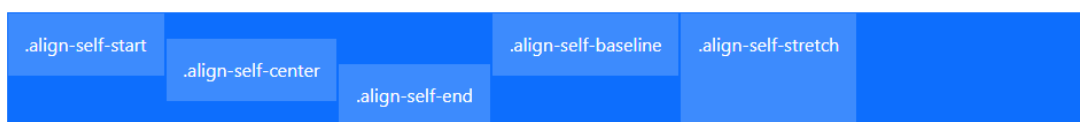
- Clases align-self-start, align-self-center, align-self-end, align-self-baseline, align-self-stretch.

Coloca cada elemento del contenedor o arriba, centrado verticalmente, abajo, arriba por texto u ocupando 100%. Dentro del mismo contenedor.

```
<div class="contenedor d-flex">
  <div class="align-self-start">.align-self-start</div>
  <div class="align-self-center">.align-self-center</div>
  <div class="align-self-end">.align-self-end</div>
  <div class="align-self-baseline">.align-self-baseline</div>
  <div class="align-self-stretch">.align-self-stretch</div>
</div>
```

Resultado:

### Align Self



Variantes responsive:

```
<!-- Variantes:
    .align-self-start
    .align-self-end
    .align-self-center
    .align-self-baseline
    .align-self-stretch

    sm | md | lg | xl | xxl
    .align-self-*-start
    .align-self-*-end
    .align-self-*-center
    .align-self-*-baseline
    .align-self-*-stretch
-->
```