

UT5: INTEGRACIÓN DE CONTENIDO
INTERACTIVO. BOOTSTRAP
DIW-DAW2

INDICE

Contenido

1. INTRODUCCIÓN3
2. INSTALACIÓN DE BOOTSTRAP3
 - 2.1. Métodos de instalación3
 - 2.2. Instalación de Bootstrap 54
3. GRID DE BOOTSTRAP6
 - 3.1.1. Ejemplo 2: columnas anidadas11
 - 3.2. Row-column17
 - 3.3. Alineado de columnas verticalmente19
 - 3.4. Alineación de columnas horizontalmente21
 - 3.5. Gutters23
 - 3.5.1. Otras clases para el GRID de Bootstrap 525
4. TIPOGRAFÍAS¡Error! Marcador no definido.
5. IMÁGENES¡Error! Marcador no definido.
6. TABLAS¡Error! Marcador no definido.
7. ACORDEON¡Error! Marcador no definido.
8. ALERTAS¡Error! Marcador no definido.
9. MIGAS DE PAN O BREADCRUMBS¡Error! Marcador no definido.
10. BOTONES¡Error! Marcador no definido.
11. CARRUSEL¡Error! Marcador no definido.
12. FORMULARIOS¡Error! Marcador no definido.
13. SELECTBOX¡Error! Marcador no definido.
 - 13.1. CHECKBOX¡Error! Marcador no definido.
 - 13.2. RADIOBUTTON¡Error! Marcador no definido.
 - 13.3. SWITCH¡Error! Marcador no definido.
 - 13.4. CHECKBOX INLINE¡Error! Marcador no definido.
 - 13.5. CHECKBOX ESTILO BOTÓN¡Error! Marcador no definido.
 - 13.6. RADIOBUTTON ESTILO BOTÓN¡Error! Marcador no definido.
 - 13.7. RANGE¡Error! Marcador no definido.
 - 13.8. Etiquetas flotantes¡Error! Marcador no definido.
 - 13.9. Validación de formularios¡Error! Marcador no definido.
14. VENTANA MODAL EN BOOTSTRAP¡Error! Marcador no definido.

15. FLEXBOX **Error! Marcador no definido.**

1. INTRODUCCIÓN

En esta unidad fijaremos como 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 solo trabajamos con código HTML y con clases que los desarrolladores de Bootstrap nos hicieron para poder realizar sitios de forma más rápida.
- Los sitios web que crea serán 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 Jéies 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-...">
```

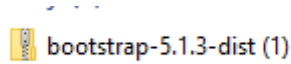
If you're using our compiled JavaScript and prefer to include Popper separately, add Popper before our JS, via a CDN preferably.

```
<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:

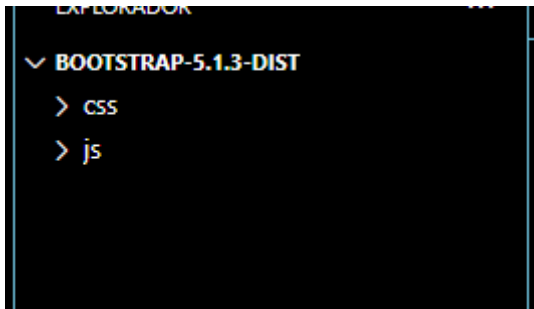
- Descargar el fichero .zip. Lo voy a guardar en una carpeta que denomina Bootstrap y lo descomprimos.



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.

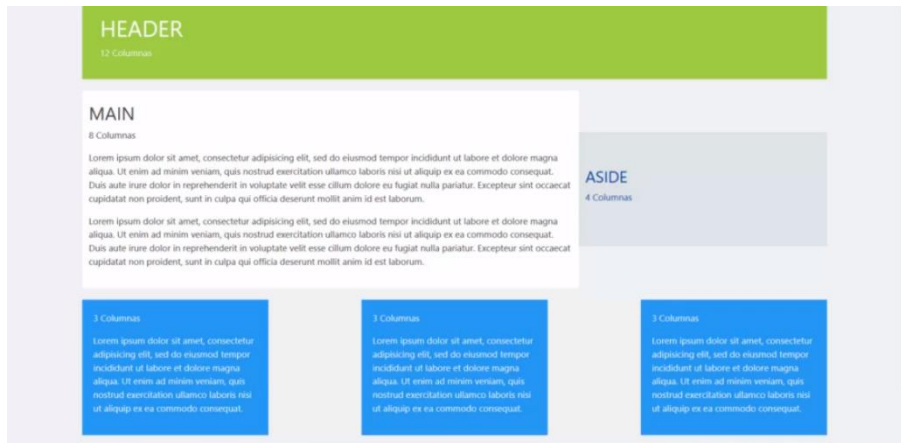
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Bootstrap5</title>
8   <link rel="stylesheet" href="css/bootstrap.min.css">
9 </head>
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.



El header ocupará 12 columnas

El main 8 columnas+aside 4 columnas=12

El tercer elemento 3 columnas de 3 columnas=9 Las otras tres columnas se reparten entre los espacios entre columnas

Breakpoints son las medidas que bootstrap va a tener en cuenta 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				

Primer proyecto con el grid de bootstrap:

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 veremos y esta clase lo que nos permitirá es que será el contenedor principal y dentro pondrá 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 row (grupo de columnas) y dentro de las filas tenemos que poner columnas (col).

Resumiendo:

- El container es el contenedor principal y en él van a ir todas las filas y columnas y después tenemos espacios en blanco en los bordes que sirven 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.

Escribe el siguiente 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>
```

Esto es una prueba

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 imprimibles (con grafema) generados por un algoritmo de cifrado que, aunque no tienen sentido para cualquier persona, sí puede

Empezar

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

Pero lo más interesante

Esto es una prueba

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 imprimibles (con grafema) generados por un algoritmo de cifrado que, aunque no tienen sentido para cualquier persona, sí puede

Empezar

Esto es una prueba

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 imprimibles (con grafema) generados por un algoritmo de cifrado que, aunque no tienen sentido para cualquier persona, sí puede

Empezar

Esto es una prueba

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 imprimibles (con grafema) generados por un algoritmo de cifrado que, aunque no tienen sentido para cualquier persona, sí puede

Empezar

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">
```

¿Cual sería el resultado?

Pero lo más interesante es poder controlar estos tamaños de columnas dependiendo del tamaño de la pantalla. 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>
```

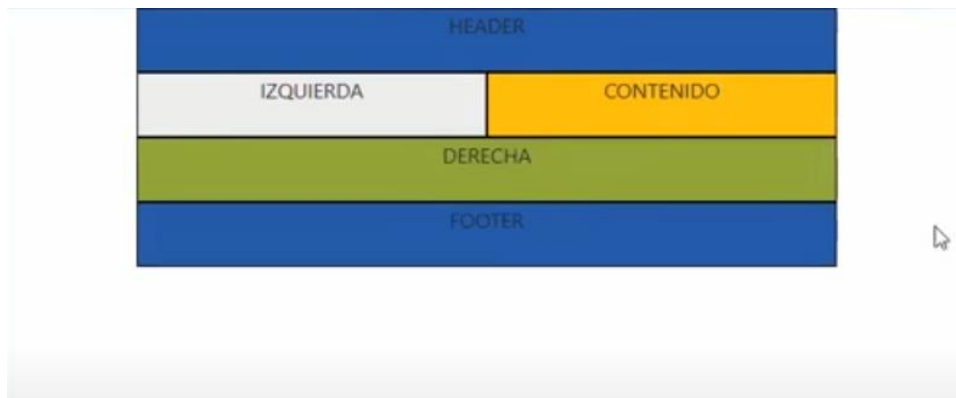
Hay una opción para div que es container-fluid que lo que hace es eliminar los márgenes a la derecha y a la izquierda.

Ejercicio 1:

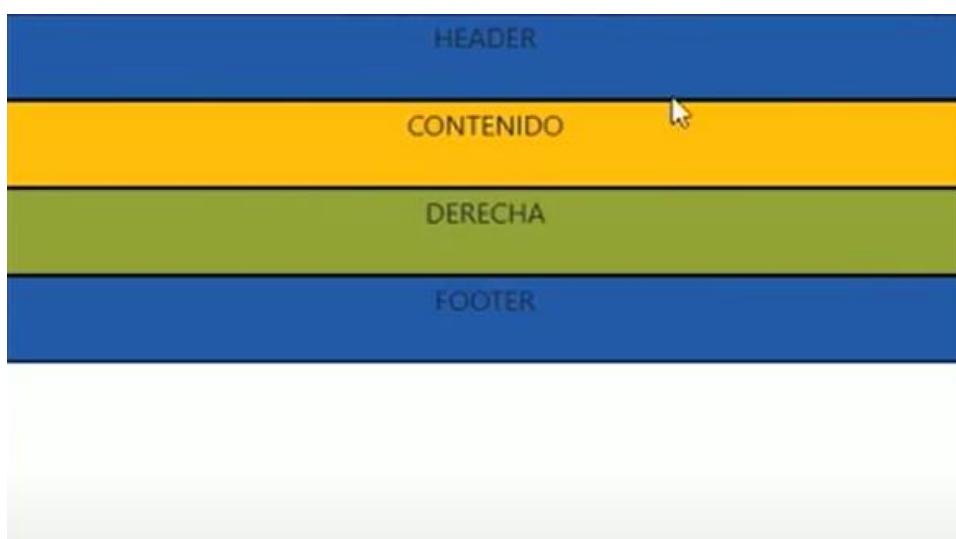
Realizar con Bootstrap la siguiente configuración de filas y columnas:



Si disminuyo la pantalla tengo que conseguir una imagen que pase por la siguiente transición:



Hasta conseguir que la imagen sea la que se muestra a continuación cuando la pantalla sea para móviles y desaparezca la columna izquierda:



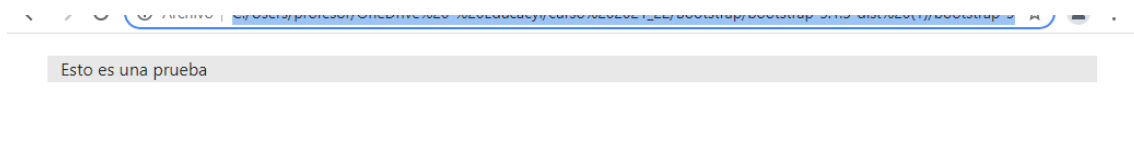
3.1.1. Ejemplo 2: columnas anidadas

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:

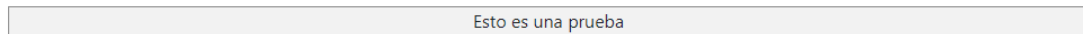


Ahora voy a crear una clase columna propia en la columna Bootstrap:

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="..."></script>
```

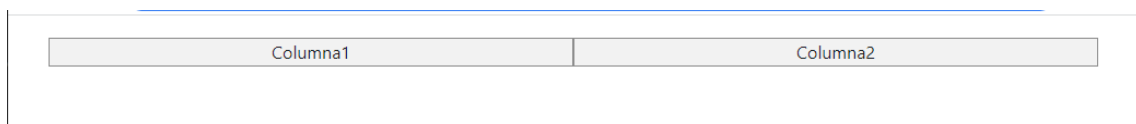
Y el resultado es:



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=".../js/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>
</body>
```

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	
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-12 col-md-6">Columna 21</div>
          <div class="columna-interior col-12 col-md-6">Columna 22</div>
        </div>
      </div>
    </div>
  </div>
</body>
```

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
mediante

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>
```

En cada fila 3 columnas

El 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>
```

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.

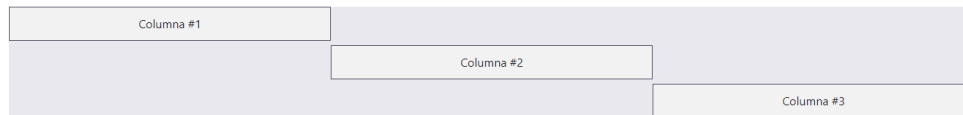
El código podría ser el siguiente:

```

<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.
- 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>
```

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>  
/
```

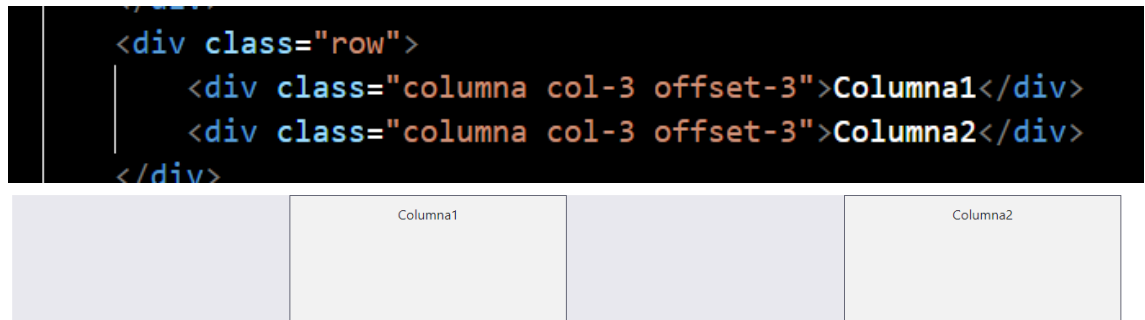
El resultado será el siguiente cuando la pantalla sea pequeña, mediana, etc.

	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:



Ejercicio:

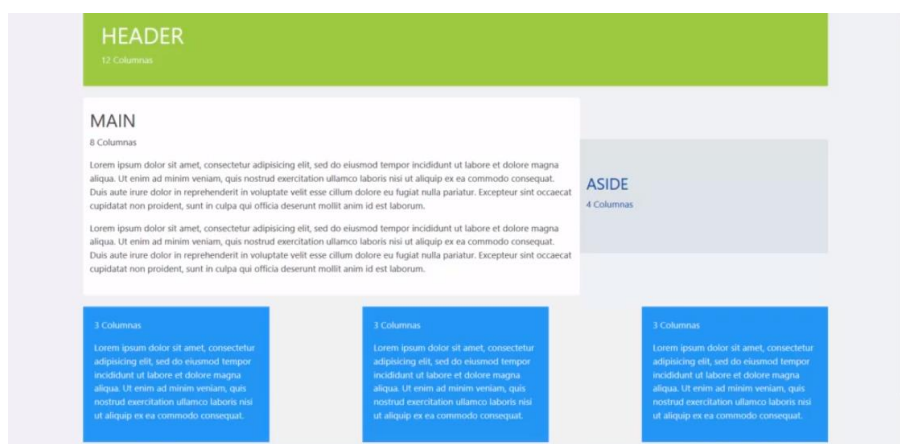
Diseñar la página web con Bootstrap. Consta de un header o un encabezado que tiene un tamaño de 12 columnas, tiene un main o un contenido principal que tiene un tamaño de 8 columnas junto con aside de cuatro columnas.

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.