

UD5. Diseño responsivo. Bootstrap

PROF: ALICIA FERNÁNDEZ CATALAN

CURSO 24-25

DAW - DISEÑO INTERFACES WEB



¿Qué es Bootstrap?

- ▶ **Bootstrap es un framework CSS front-end** (es decir, un conjunto de herramientas para el desarrollo de la parte visual de una página web) que facilita la creación de páginas web con un diseño atractivo y funcional de forma rápida y sencilla.
- ▶ Fue desarrollado por Mark Otto y Jacob Thornton en 2010 en Twitter, con el objetivo de estandarizar y mejorar el proceso de desarrollo de interfaces web. Su popularidad creció rápidamente debido a su fácil integración, gran documentación y la capacidad de crear sitios web responsivos (es decir, que se adaptan a diferentes tamaños de pantalla, como móviles, tabletas y escritorios).
- ▶ **Bootstrap incluye una colección de:**
 - ▶ Componentes predefinidos como botones, formularios, menús de navegación, etc.
 - ▶ Herramientas CSS para manejar la tipografía, los colores, las rejillas (grids), márgenes y espaciados.
 - ▶ JavaScript y jQuery para efectos interactivos y elementos dinámicos (como modales, carruseles, alertas, etc.).

¿Quién desarrolla Bootstrap?

- ▶ Bootstrap fue creado por Mark Otto y Jacob Thornton mientras trabajaban en Twitter.
- ▶ Con el tiempo, el proyecto fue abierto al público y se convirtió en un proyecto de código abierto.
- ▶ Actualmente, es mantenido por la comunidad de desarrolladores de Bootstrap, aunque el equipo principal sigue siendo responsable de las actualizaciones y mejoras.

¿Cómo usar Bootstrap?

- ▶ **Para comenzar a usar Bootstrap, tienes dos opciones principales:**
 1. **Usar la versión alojada en línea (CDN):** Puedes incluir Bootstrap en tu proyecto sin necesidad de descargarlo. Solo tienes que agregar el enlace a los archivos CSS y JavaScript de Bootstrap en el archivo HTML.
 2. **Descargar los archivos de la plantilla:** Puedes descargar el paquete completo de Bootstrap y alojarlo en tu propio servidor.
- ▶ Veamos ejemplos de uso de cada caso

¿Cómo usar Bootstrap?

► Opción 1: Usar CDN

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Mi página con Bootstrap</title>
  <!-- CSS de Bootstrap desde CDN -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>

  <!-- Tu contenido aquí -->

  <!-- JS de Bootstrap y dependencias desde CDN -->
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

¿Cómo usar Bootstrap?

► Opción 2: Descargar Bootstrap

- Ve a <https://getbootstrap.com> y descarga la versión más reciente de Bootstrap.
- Incluye los archivos .css y .js en tu proyecto.
- Cuando descargas una plantilla de Bootstrap, generalmente obtienes un conjunto de archivos y carpetas organizadas de manera que puedas usar rápidamente los recursos de Bootstrap en tu proyecto.
- A continuación, vamos a ver qué suele contener una plantilla de Bootstrap descargada y cómo está estructurada.

Estructura típica de los archivos en una plantilla de Bootstrap

- ▶ En la transparencia anterior, se ha comentado que cuando descargamos una plantilla de Bootstrap, ésta está compuesta de un conjunto de archivos y carpetas organizadas de manera que puedas usar rápidamente los recursos de Bootstrap en tu proyecto.
- ▶ Al descargar una plantilla, **los archivos que obtenemos pueden variar dependiendo de la plantilla específica**, pero la estructura comúnmente incluye algo como lo que te muestra la imagen

```
/mi-plantilla-bootstrap
├── /assets
│   ├── /css
│   │   └── bootstrap.min.css
│   ├── /js
│   │   └── bootstrap.bundle.min.js
│   └── /images
│       └── logo.png
├── /fonts
│   └── fontawesome.ttf
├── /index.html
├── /about.html
├── /contact.html
└── /README.md
```

Descripción de los archivos y carpetas

- ▶ **/assets:** Esta carpeta contiene todos los archivos estáticos, como los archivos de estilo (CSS), scripts (JavaScript), imágenes y fuentes.
- ▶ Es común organizar los archivos dentro de subcarpetas como css, js, images, y fonts para mantener el proyecto bien estructurado.
 - ▶ **/css:** Aquí se encuentran los archivos CSS que contienen los estilos de la plantilla, incluidos los estilos de Bootstrap y los personalizados. **El archivo más común que verás es bootstrap.min.css**, que es la versión comprimida de Bootstrap con todos los estilos listos para ser utilizados.
 - ▶ **/js:** Contiene los archivos JavaScript necesarios para hacer funcionar las interacciones dinámicas de Bootstrap (como los modales, menús desplegados, etc.). **El archivo principal suele ser bootstrap.bundle.min.js**, que incluye tanto los scripts de Bootstrap como las dependencias necesarias (como Popper.js para los tooltips y popovers).
 - ▶ **/images:** Guarda las imágenes utilizadas en la plantilla, como logotipos, fondos, iconos, etc. Por ejemplo, logo.png podría ser el logotipo de la plantilla.
 - ▶ **/fonts:** Algunas plantillas incluyen fuentes personalizadas o iconos web como FontAwesome. Las fuentes que se usan en la plantilla se guardan aquí. Ejemplo: fontawesome.ttf.

Descripción de los archivos y carpetas

- ▶ **/index.html:** Este es el archivo principal de la plantilla. Es la página de inicio de tu sitio web. Generalmente, incluye la estructura básica de la plantilla con cabeceras, navegación, contenido principal y pie de página, usando las clases de Bootstrap para el diseño.
- ▶ **/about.html, /contact.html, etc.:**
 - ▶ Son otros archivos HTML que forman parte del sitio web. Por ejemplo, about.html es una página de "Acerca de" y contact.html es una página de contacto.
 - ▶ Estas páginas seguirán la misma estructura que index.html, pero con contenido diferente.
- ▶ **/README.md:**
 - ▶ Este archivo es un archivo de texto donde generalmente los desarrolladores de la plantilla explican cómo usarla. Contiene información sobre cómo configurar la plantilla, cómo personalizarla, cómo incluir Bootstrap, y cualquier otra recomendación o detalle importante.

Ejemplo de un archivo HTML con la estructura de una plantilla de Bootstrap

- ▶ Veamos un ejemplo.
- ▶ Supongamos que descargaste una plantilla básica de Bootstrap.
- ▶ Veamos que contiene el archivo **mi pagina bootstrap.html** (Archivos descargable en Aules)

Ejemplo de un archivo HTML con la estructura de una plantilla de Bootstrap

► Detalles importantes del ejemplo anterior

- **Enlace al archivo CSS de Bootstrap:** En el encabezado (<head>) se vincula el archivo **bootstrap.min.css** que se encuentra en la carpeta assets/css.
- **Estructura de la página:** La página está dividida en una barra de navegación (<nav>), un contenedor principal para el contenido (<div class="container">) y un pie de página (<footer>).
- **Enlace al archivo JavaScript:** Al final del cuerpo del documento (<body>), se incluye el archivo **bootstrap.bundle.min.js** para que las funciones interactivas de Bootstrap (como los menús desplegables) funcionen correctamente.

Ejemplo de un archivo HTML con la estructura de una plantilla de Bootstrap

- ▶ Resumen de los archivos descargados: Cuando descargas una plantilla de Bootstrap, los archivos esenciales que contiene son:
 - ▶ Archivos CSS (para los estilos visuales),
 - ▶ Archivos JavaScript (para la interactividad),
 - ▶ Archivos de imágenes y fuentes (para logotipos, iconos y tipografía personalizada),
 - ▶ Páginas HTML (que contienen la estructura de tu sitio web),
 - ▶ README.md (con información sobre cómo usar la plantilla).
- ▶ **Cada plantilla puede variar en términos de la cantidad de archivos o carpetas que incluye, pero en términos generales, siempre verás estos componentes clave.**
 - ▶ La estructura está diseñada para ser fácil de entender y personalizar según tus necesidades.

Estructura para crear una plantilla básica para Bootstrap

- ▶ **Esta estructura básica incluye:**
- ▶ Una etiqueta meta para la configuración del viewport (para hacer el diseño responsivo).
- ▶ Un `<link>` para incluir el CSS de Bootstrap.
- ▶ Un `<script>` para incluir los archivos JavaScript que habilitan las interacciones como el desplegable en el menú.
- ▶ Un container que es el contenedor principal donde se pone el contenido de la página.
- ▶ Un `<nav>` para la barra de navegación, un `<section>` para el contenido principal y un `<footer>` para el pie de página.
- ▶ Ejemplo de estructura de las plantillas en Bootstrap
 - ▶ Veamos una **plantilla básica en Bootstrap** (Archivos descargable en Aules)

Fundamentos de Bootstrap

1. Container

- ▶ El container es el **elemento principal de la estructura de Bootstrap**. Sirve para alinear el contenido de manera centrada y con un ancho máximo.
- ▶ Bootstrap ofrece dos tipos de contenedores:
 - ▶ **container**: Tiene un ancho fijo que varía según el tamaño de la pantalla.
 - ▶ **container-fluid**: Ocupa todo el ancho disponible en cualquier tamaño de pantalla.

Fundamentos de Bootstrap

► Ejemplo

```
<div class="container"> <!-- Contenedor con ancho fijo -->
  <p>Contenido centrado con un ancho máximo fijo.</p>
</div>
```

```
<div class="container-fluid"> <!-- Contenedor con ancho completo -->
  <p>Contenido que ocupa todo el ancho disponible.</p>
</div>
```

Fundamentos de Bootstrap

2. Grid System (Sistema de columnas en Bootstrap)

- ▶ **Bootstrap tiene un sistema de rejilla basado en 12 columnas.** Esto permite dividir el contenido en hasta 12 partes iguales y organizarlo de manera flexible.
- ▶ **En primer lugar, es necesario crear un elemento fila con `class=row` y a continuación con base en el sistema de 12 columnas , se indicarán las columnas y el número de porciones sobre 12 que ocupa cada una de ellas.**
- ▶ El sistema de columnas también es responsivo: puedes especificar cuántas columnas debe ocupar cada elemento dependiendo del tamaño de la pantalla (por ejemplo, en pantallas grandes puede ser de 4 columnas y en pantallas pequeñas puede ser de 12).
- ▶ **Para crear las columnas, se utiliza el elemento `col` que permite especificar el tamaño de las columnas creadas**

Fundamentos de Bootstrap

2. Grid System (Sistema de columnas en Bootstrap)

- Explicación del ejemplo:
- **<div class="container">**: Este es el contenedor principal que centra el contenido en la página y le da un ancho máximo. Hay dos tipos de contenedores: container (de ancho fijo) y container-fluid (de ancho completo).
- **<div class="row">**: Define una fila. Dentro de una fila, puedes colocar las columnas. Las filas deben contener solo columnas (o elementos de diseño específicos).
- **<div class="col-4">**: Define una columna. Aquí hemos usado col-4 que significa que la columna ocupará 4 de las 12 columnas disponibles en la fila. Puedes usar clases como col-1, col-2, ..., col-12 para especificar el tamaño de las columnas.

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

Clases de columnas básicas

- ▶ Bootstrap especifica el tamaño de las columnas en su sistema de rejilla.
- ▶ Como se ha mencionado anteriormente, Bootstrap organiza el contenido en una estructura de 12 columnas.
- ▶ Esto significa que, dentro de una fila (.row), el espacio total se divide en 12 partes iguales, y las columnas se pueden ajustar en función de esas partes.
- ▶ El sistema de columnas en Bootstrap se basa en clases que definen cuántas columnas debe ocupar un elemento dentro de una fila.
- ▶ **La forma básica de hacerlo es utilizando las clases como col-, col-1, col-2, ..., col-12,** que definen el tamaño de la columna en función de las 12 partes disponibles.

Combinación de columnas

- ▶ En Bootstrap, puedes **combinar columnas para que sumen hasta un total de 12** para que se ajusten a la fila de manera correcta.
- ▶ Sin embargo, no es obligatorio usar todas las 12 columnas.
- ▶ Puedes usar menos columnas y el espacio restante se distribuirá automáticamente.
- ▶ Ejemplo con columnas de diferentes tamaños:
 - ▶ col-4: La primera columna ocupa 4 de las 12 partes (un tercio del espacio).
 - ▶ col-8: La segunda columna ocupa las 8 partes restantes (dos tercios del espacio).

```
<div class="container">  
  <div class="row">  
    <div class="col-4">Columna 1 (3/12)</div>  
    <div class="col-8">Columna 2 (9/12)</div>  
  </div>  
</div>
```

Combinación de columnas

- ▶ Si quieres dividir el contenido en varias filas, puedes hacerlo fácilmente añadiendo más filas dentro del contenedor.
- ▶ Otro ejemplo:
 - ▶ **En Fila 1** tienes 2 columnas (col-6), lo que significa que cada columna ocupa la mitad del ancho disponible (6 de las 12 columnas). Es decir, cada columna ocupa la mitad del espacio disponible (6 de las 12 partes).
 - ▶ **En Fila 2** tienes 3 columnas (col-4), lo que significa que cada columna ocupa un tercio del ancho disponible (4 de las 12 columnas). Es decir, cada columna ocupa 4 de las 12 partes disponibles (un tercio del total), ya que $4 + 4 + 4 = 12$.

```
<div class="container">
  <!-- Fila 1 -->
  <div class="row">
    <div class="col-6">Columna 1 (50%)</div>
    <div class="col-6">Columna 2 (50%)</div>
  </div>
  <!-- Fila 2 -->
  <div class="row">
    <div class="col-4">Columna 1 (33.33%)</div>
    <div class="col-4">Columna 2 (33.33%)</div>
    <div class="col-4">Columna 3 (33.33%)</div>
  </div>
</div>
```

Combinación de columnas

- ▶ Si alguna de las columnas no encaja en la fila debido a un exceso de tamaño, Bootstrap automáticamente las llevará a la siguiente fila, lo cual es ideal para diseños responsivos.
- ▶ Ejemplo:
 - ▶ Las tres primeras columnas en la primera fila ocupan el 100% del espacio disponible (12 de las 12 columnas).
 - ▶ Luego, la fila siguiente tiene dos columnas que ocupan 6 de las 12 partes, sumando 12 también.

```
<div class="container">  
  <div class="row">  
    <div class="col-4">Columna 1 (4/12)</div>  
    <div class="col-4">Columna 2 (4/12)</div>  
    <div class="col-4">Columna 3 (4/12)</div>  
  </div>  
  <div class="row">  
    <div class="col-6">Columna 4 (6/12)</div>  
    <div class="col-6">Columna 5 (6/12)</div>  
  </div>  
</div>
```

Alineación de columnas

- ▶ Puedes alinear las columnas dentro de una fila de varias maneras usando las clases de Bootstrap:
- 1. **Alinear a la izquierda (por defecto):** Las columnas se alinean a la izquierda por defecto, como hemos visto en los ejemplos anteriores.
- 2. **Alinear al centro:** Usa la clase `justify-content-center` en la fila para centrar las columnas.
- 3. **Alinear a la derecha:** Usa la clase `justify-content-end` en la fila para alinear las columnas a la derecha.

```
<div class="container">
  <div class="row justify-content-center">
    <div class="col-4">Columna centrada</div>
    <div class="col-4">Columna centrada</div>
  </div>
</div>
```

```
<div class="container">
  <div class="row justify-content-end">
    <div class="col-4">Columna alineada a la derecha</div>
    <div class="col-4">Columna alineada a la derecha</div>
  </div>
</div>
```

Clases responsivas para diferentes tamaños de pantalla

- ▶ La gran ventaja de Bootstrap es que puedes especificar cómo deben comportarse las columnas en diferentes tamaños de pantalla.
- ▶ Puedes usar las clases de columnas responsivas para hacer que las columnas cambien su tamaño según el ancho de la pantalla del dispositivo.
- ▶ **Las clases col-{size}-{number}** te permiten ajustar el tamaño de las columnas en función del tamaño de la pantalla.
- ▶ Bootstrap ofrece clases que permiten definir el tamaño de las columnas para varios rangos de dispositivos:
 - ▶ col-: Para pantallas extra pequeñas (por ejemplo, móviles).
 - ▶ col-sm-: Para pantallas pequeñas (tabletas).
 - ▶ col-md-: Para pantallas medianas (escritorios pequeños).
 - ▶ col-lg-: Para pantallas grandes (escritorios grandes).
 - ▶ col-xl-: Para pantallas extra grandes (monitores de alta resolución).

Ejemplo de columnas responsivas:

- ▶ Ejemplo:
- ▶ col-12: En pantallas muy pequeñas (móviles), cada columna ocupa el 100% del espacio disponible (12 de las 12 columnas). Esto significa que, en dispositivos pequeños, las columnas se apilan verticalmente.
- ▶ col-sm-6: En pantallas pequeñas (tabletas), cada columna ocupará el 50% del espacio (6 de las 12 columnas).
- ▶ col-md-4: En pantallas medianas (escritorios pequeños), cada columna ocupará el 33.33% del espacio disponible (4 de las 12 columnas).
- ▶ col-lg-3: En pantallas grandes (escritorios), cada columna ocupará el 25% del espacio (3 de las 12 columnas).

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


Columnas autoajustables

- ▶ Bootstrap también tiene una clase que permite a las columnas autoajustarse al contenido disponible.
- ▶ **La clase col-auto** hace que la columna ocupe solo el espacio necesario según el contenido.
- ▶ Ejemplo:
 - ▶ **col-auto**: Las dos columnas con esta clase ocuparán solo el espacio necesario para su contenido.
 - ▶ **col-6**: La columna de tamaño 6 ocupará la mitad del espacio disponible (6 de las 12 columnas), mientras que las otras dos columnas ajustarán su tamaño según su contenido.

```
<div class="container">  
  <div class="row">  
    <div class="col-auto">Columna 1 (auto)</div>  
    <div class="col-6">Columna 2 (6/12)</div>  
    <div class="col-auto">Columna 3 (auto)</div>  
  </div>  
</div>
```

Columnas con desplazamiento

- ▶ Si se necesita dejar espacio entre las columnas podemos hacerlo fácilmente con las clases de gutter (espaciado entre columnas) que ofrece Bootstrap.
 - ▶ Los **gutters** son el padding entre las columnas, que se utilizan para espaciar y alinear de manera responsive el contenido en el sistema de cuadrícula de Bootstrap.
 - ▶ Más información:
<https://getbootstrap.esdocu.com/docs/5.1/layout/gutters/>
- ▶ Estas clases se aplican a la fila (row) o a las columnas mismas.
- ▶ Ejemplo:
 - ▶ La **clase g-3** agrega un espaciado de 1 rem (16 píxeles) entre las columnas.

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

Uso de offsets (desplazamiento de columnas)

- ▶ Bootstrap permite desplazar columnas utilizando offsets.
- ▶ **Un offset es una manera de mover una columna a la derecha para crear espacio en blanco.**
- ▶ Por ejemplo, si quieres que una columna comience a partir de la columna número 4, puedes usar un offset.
- ▶ Ejemplo:
 - ▶ col-4: La primera columna ocupa 4 de las 12 partes.
 - ▶ col-4 offset-4: La segunda columna ocupa 4 de las 12 partes, pero se desplaza 4 columnas a la derecha.

```
<div class="container">  
  <div class="row">  
    <div class="col-4">Columna 1 (4/12)</div>  
    <div class="col-4 offset-4">Columna 2 (4/12 con offset)</div>  
  </div>  
</div>
```

Resumen uso de columnas en Bootstrap

- ▶ **Resumen:**
- ▶ **Clases de columnas:** col-1 a col-12 definen el tamaño de las columnas, donde el número indica cuántas partes del total de 12 ocupa la columna.
- ▶ **Clases responsivas:** Puedes usar clases como col-sm-, col-md-, col-lg-, col-xl- para especificar cómo deben comportarse las columnas en diferentes tamaños de pantalla.
- ▶ **Columnas autoajustables:** Usa col-auto para que las columnas ocupen solo el espacio necesario según su contenido.
- ▶ **Desplazamiento de columnas:** Usa offset- para mover las columnas hacia la derecha y crear espacio en blanco.

Orden de las columnas en Bootstrap

- ▶ En Bootstrap, el orden de las columnas en una fila puede modificarse utilizando clases específicas para reordenar los elementos sin tener que alterar el HTML.
- ▶ Esto es especialmente útil cuando quieres cambiar el orden de las columnas dependiendo del tamaño de la pantalla (por ejemplo, en dispositivos móviles o en escritorios grandes).
- ▶ Bootstrap ofrece clases para modificar el orden de las columnas tanto en pantallas grandes como pequeñas.
- ▶ Estas clases se basan en el sistema de rejilla (grid system) y permiten cambiar el orden visualmente sin afectar el orden del código fuente, lo cual es ideal para mantener la accesibilidad y la correcta estructura del HTML.

Clases para cambiar el orden de las columnas

- ▶ Bootstrap usa las siguientes clases para modificar el orden de las columnas:
 - ▶ **order-0, order-1, order-2, ..., order-12**: Estas clases te permiten especificar el orden de las columnas en cualquier tamaño de pantalla.
 - ▶ **order-first**: Coloca la columna en primer lugar.
 - ▶ **order-last**: Coloca la columna al final.
 - ▶ **order-*-first y order-*-last**: Para especificar el orden en diferentes tamaños de pantalla (donde * puede ser sm, md, lg, xl).

Cambiar el orden de las columnas en pantallas grandes

- ▶ Para cambiar el orden de las columnas en pantallas grandes, se usa **la clase order- seguida de un número que indica el orden deseado**.
- ▶ El número 0 significa que el elemento ocupará el primer lugar. El valor máximo es 12, que representa el último lugar en una fila de 12 columnas.
- ▶ Ejemplo:
 - ▶ order-3: La primera columna se coloca al final (es decir, ocupa la última posición).
 - ▶ order-1: La segunda columna se coloca en primer lugar.
 - ▶ order-2: La tercera columna ocupa el segundo lugar.
- ▶ Este ejemplo muestra cómo el orden de las columnas cambia visualmente, pero el HTML sigue manteniendo el orden original. Esto permite una mayor flexibilidad en el diseño.

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

Cambiar el orden en diferentes tamaños de pantalla

- ▶ Para cambiar el orden en función del tamaño de la pantalla, puedes usar las clases `order-` específicas para tamaños de pantallas, como:
 - ▶ `order-sm-`, `order-md-`, `order-lg-`, etc.
- ▶ Ejemplo: Este ejemplo ilustra cómo el orden de las columnas cambia de acuerdo con el tamaño de la pantalla. En pantallas pequeñas (móviles), las columnas se reordenan, pero en pantallas más grandes (por ejemplo, escritorios), el orden es diferente.
- ▶ **En pantallas pequeñas (móviles):**
 - ▶ `order-1`: La columna 1 aparecerá en primer lugar.
 - ▶ `order-3`: La columna 2 aparecerá en tercer lugar.
 - ▶ `order-2`: La columna 3 aparecerá en segundo lugar.

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

- ▶ **En pantallas medianas y grandes (escritorios):**
 - ▶ `order-md-2`: La columna 1 ocupará la segunda posición.
 - ▶ `order-md-1`: La columna 2 ocupará la primera posición.
 - ▶ `order-md-3`: La columna 3 ocupará la tercera posición.

Usando order-first y order-last

- ▶ **Las clases order-first y order-last** permiten mover una columna al primer lugar o al último lugar, respectivamente, sin tener que usar valores numéricos.
- ▶ Ejemplo:
 - ▶ order-first: La columna 1 se moverá al primer lugar de la fila.
 - ▶ order-last: La columna 3 se moverá al último lugar de la fila.

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

Ejemplo completo con orden y clases responsivas

- ▶ A continuación, un ejemplo donde se combina el reordenamiento con las clases responsivas para que las columnas cambien su orden en función del tamaño de la pantalla.
- ▶ Ejemplo:
 - ▶ **En pantallas pequeñas (móviles):**
 - ▶ Las columnas se apilan verticalmente y tienen el siguiente orden: Columna 1, Columna 2, Columna 3.
 - ▶ **En pantallas medianas y grandes (escritorios):**
 - ▶ order-md-1: La columna 1 se moverá al primer lugar.
 - ▶ order-md-2: La columna 2 se moverá al segundo lugar.
 - ▶ order-md-3: La columna 3 se moverá al tercer lugar.

```
<div class="container">  
  <div class="row">  
    <div class="col-12 col-md-4 order-2 order-md-1">Columna 1</div>  
    <div class="col-12 col-md-4 order-3 order-md-2">Columna 2</div>  
    <div class="col-12 col-md-4 order-1 order-md-3">Columna 3</div>  
  </div>  
</div>
```

Resumen de las clases de orden

- ▶ order-0 a order-12: Cambian el orden de las columnas en una fila, donde order-0 es la primera posición y order-12 es la última.
- ▶ order-first: Mueve la columna al primer lugar.
- ▶ order-last: Mueve la columna al último lugar.
- ▶ order-*-n (responsivo): Permite cambiar el orden de las columnas en diferentes tamaños de pantalla, donde * es el tamaño de pantalla (sm, md, lg, xl).
 - ▶ Ejemplo: order-md-1, order-lg-2.
- ▶ **Conclusión**
- ▶ El sistema de orden de columnas en Bootstrap es muy útil para crear diseños flexibles y responsivos. Permite cambiar el orden de las columnas visualmente sin afectar el código fuente, lo que es ideal para mejorar la experiencia del usuario y optimizar la accesibilidad.
- ▶ Con el uso de clases como order-, order-first, order-last, y las clases responsivas como order-sm- o order-md-, puedes adaptar el diseño a diferentes tamaños de pantalla de manera eficiente.

Tipografía

- ▶ **Bootstrap tiene clases predefinidas para la tipografía** que puedes usar para ajustar el estilo del texto.
- ▶ Algunas **clases comunes** son:
 - ▶ **h1, h2, h3, etc.:** Para los títulos de diferentes tamaños.
 - ▶ **lead:** Para resaltar párrafos.
 - ▶ **text-center, text-left, text-right:** Para alinear el texto.
 - ▶ **font-weight-bold, font-italic:** Para aplicar estilos de fuente.

```
<p class="lead">Este es un párrafo destacado.</p>  
<h1 class="text-center">Título centrado</h1>
```

Colores

- ▶ **Bootstrap ofrece clases para cambiar los colores del texto, el fondo y los bordes.**
- ▶ Algunos ejemplos:
 - ▶ bg-primary: Fondo de color azul.
 - ▶ text-white: Texto blanco.
 - ▶ bg-success: Fondo verde (usado para éxito).
 - ▶ text-danger: Texto rojo (usado para errores o alertas).

```
<div class="bg-primary text-white p-3">Este es un fondo azul con texto blanco.</div>
```

Conclusión

- ▶ Bootstrap es una herramienta poderosa para crear páginas web modernas y funcionales sin tener que empezar desde cero.
- ▶ A través de su sistema de rejillas, sus componentes listos para usar, y sus clases para manejar la tipografía y los colores, puedes desarrollar sitios web que se adapten a diferentes dispositivos y que tengan un diseño atractivo de manera rápida.
- ▶ **Es imposible recoger todas las clases que usa este framework de CSS. Por ello es recomendable revisar la documentación y conocer los estilos de las plantillas utilizadas**