

# Bootstrap 5

---

**Bootstrap 5** es un popular framework front-end que facilita el diseño de interfaces web responsivas y móviles. Con una amplia colección de componentes listos para usar y un sistema de rejillas (grid) flexible, Bootstrap permite a los desarrolladores construir rápidamente interfaces atractivas y funcionales sin necesidad de escribir mucho código CSS desde cero.

## Índice

- **Bootstrap 5**
  - Índice
  - **Conceptos básicos de Bootstrap 5**
    - Ejemplo de contenedores
    - Ejemplo de CSS
  - **Sistema de rejillas de Bootstrap, clases y puntos de corte**
    - Ejemplo de un sistema de rejillas
  - 1. Clase `row`
    - Ejemplo de uso de `row`
    - Características
  - 2. Clase `col`
    - Ejemplo de uso de `col`
    - Comportamiento
  - 3. Puntos de quiebre (Breakpoints)
    - Ejemplo de uso de puntos de quiebre
    - Cómo funcionan
  - 4. Combinaciones de clases
  - 5. Uso de Flexbox en Bootstrap
    - Ventajas de Flexbox en Bootstrap
    - Clases de Flexbox en Bootstrap
      - 1. Clases de contenedor flex
      - 2. Clases de alineación
      - 3. Clases de orden
      - 4. Clases de tamaño
    - Ejemplo completo
    - Descripción del ejemplo

## Conceptos básicos de Bootstrap 5

Bootstrap 5 se basa en dos conceptos clave:

1. **Contenedor (Container):** Es el elemento que actúa como el contenedor principal para el contenido. Se puede declarar como `container`, `container-fluid` (para ancho completo), o `container-{breakpoint}` (para diferentes puntos de quiebre).
2. **Filas y Columnas:** El sistema de rejillas de Bootstrap se compone de filas (`row`) y columnas (`col`). Las filas contienen columnas que se ajustan automáticamente en función del tamaño de la pantalla.

## Ejemplo de contenedores

```
<body>
  <div class="container">.container</div>
  <div class="container-fluid">.container-fluid</div>
  <div class="container-lg">.container-lg</div>
</body>
```

## Ejemplo de CSS

```
div {
  height: 250px;
  background-color: purple;
  color: white;
  font-size: 50px;
  margin: 20px;
}
```

## Sistema de rejillas de Bootstrap, clases y puntos de corte

Bootstrap utiliza un sistema de rejillas de 12 columnas que permite dividir el espacio en varias partes. Las propiedades principales incluyen:

- **row**: Crea una fila que contiene las columnas.
- **col**: Crea columnas que ocupan el mismo ancho dentro de una fila.
- **col-{breakpoint}-{n}**: Define el tamaño de las columnas según el punto de quiebre. Por ejemplo, **col-md-4** significa que la columna ocupará 4 de las 12 columnas disponibles en dispositivos medianos o más grandes.

## Ejemplo de un sistema de rejillas

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

Las clases **row**, **col**, y los diferentes puntos de corte (**xs**, **sm**, **md**, **lg**, **xl**, **xxl**) son fundamentales en el sistema de rejillas (grid) de Bootstrap 5. A continuación, te explico cada uno de estos conceptos en detalle:

### 1. Clase **row**

La clase `row` se utiliza para crear filas dentro del sistema de rejillas de Bootstrap. Agrupa columnas y se asegura de que el contenido dentro de ellas se alinee correctamente.

### Ejemplo de uso de `row`

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

### Características

- Las filas (`row`) utilizan márgenes negativos para compensar el padding aplicado a las columnas, lo que asegura que las columnas se alineen correctamente con el borde del contenedor.
- Las filas son flexibles y permiten que las columnas dentro de ellas se distribuyan de manera horizontal.

## 2. Clase `col`

La clase `col` se usa para definir columnas dentro de una fila. Puedes usarla de varias maneras para controlar el ancho de las columnas en función de las necesidades de tu diseño. Al usar simplemente `col`, Bootstrap asigna automáticamente el mismo ancho a todas las columnas dentro de la misma fila.

### Ejemplo de uso de `col`

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

### Comportamiento

- Si utilizas `col`, las columnas ocuparán el mismo ancho, distribuyéndose equitativamente dentro de la fila.
- Puedes especificar el ancho de las columnas utilizando clases específicas que indican el número de columnas que ocupará cada una.

## 3. Puntos de quiebre (Breakpoints)

Bootstrap 5 utiliza seis puntos de quiebre que permiten personalizar la disposición de las columnas según el tamaño de la pantalla. Estos puntos de quiebre son:

Clase	Tamaño de pantalla	Descripción
-------	--------------------	-------------

Clase	Tamaño de pantalla	Descripción
<code>xs</code>	< 576px	No se necesita especificar
<code>sm</code>	>= 576px	Pantallas pequeñas
<code>md</code>	>= 768px	Pantallas medianas
<code>lg</code>	>= 992px	Pantallas grandes
<code>xl</code>	>= 1200px	Pantallas extra grandes
<code>xxl</code>	>= 1400px	Pantallas extra extra grandes

### Ejemplo de uso de puntos de quiebre

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

### Cómo funcionan

- **`col-{breakpoint}-{n}`**: Permite definir cuántas columnas ocupará un elemento en función del tamaño de la pantalla. Por ejemplo, `col-md-4` significa que la columna ocupará 4 de las 12 columnas disponibles en dispositivos medianos o más grandes.
- **Sin prefijo**: `col-6` significa que la columna ocupará 6 de las 12 columnas disponibles en todos los tamaños de pantalla. Si quieres que una columna ocupe el mismo espacio en todos los dispositivos, simplemente usa `col-6`.

## 4. Combinaciones de clases

Puedes combinar diferentes clases para lograr un diseño flexible y adaptativo. Por ejemplo:

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

En este ejemplo:

- En pantallas pequeñas (menos de 768px), cada columna ocupará el 100% del ancho (12 columnas).
- En pantallas medianas y más grandes, cada columna ocupará el 50% del ancho (6 columnas).

## 5. Uso de Flexbox en Bootstrap

Bootstrap 5 utiliza Flexbox de manera extensiva para facilitar la alineación y distribución de elementos en el diseño. Flexbox es un modelo de diseño que proporciona una forma más eficiente de organizar y alinear elementos en un contenedor, tanto en dirección horizontal como vertical.

### Ventajas de Flexbox en Bootstrap

- **Alineación fácil:** Permite alinear elementos en el centro, a la izquierda, a la derecha, o distribuir el espacio entre ellos de manera sencilla.
- **Orden flexible:** Permite cambiar el orden de los elementos visualmente sin afectar el orden en el código HTML.
- **Tamaño adaptable:** Los elementos dentro de un contenedor Flexbox pueden crecer o reducirse para ocupar el espacio disponible.

### Clases de Flexbox en Bootstrap

Bootstrap proporciona varias clases de utilidad para trabajar con Flexbox, permitiendo aplicar estilos de manera rápida y eficiente. Estas clases incluyen:

#### 1. Clases de contenedor flex

- `d-flex`

`d-flex`: Convierte un contenedor en un flex container.

- `flex-row`: Define la dirección de los elementos en fila (horizontal).
- `flex-column`: Define la dirección de los elementos en columna (vertical).

#### 2. Clases de alineación

- `align-items-start`: Alinea los elementos al inicio del contenedor.
- `align-items-center`: Alinea los elementos en el centro del contenedor.
- `align-items-end`: Alinea los elementos al final del contenedor.

#### 3. Clases de orden

- `order-1`: Asigna un orden a los elementos, donde `1` es el primer elemento y `12` el último.

#### 4. Clases de tamaño

- `flex-grow-1`: Permite que un elemento crezca para ocupar el espacio disponible.
- `flex-shrink-0`: Evita que un elemento se reduzca.

### Ejemplo completo

```
<div class="container">
  <div class="row">
    <div class="col d-flex align-items-center justify-content-between">
```

```
<div class="order-2">Elemento 1</div>
<div class="order-1">Elemento 2</div>
<div class="order-3">Elemento 3</div>
</div>
</div>
</div>
```

## Descripción del ejemplo

- **d-flex** convierte el contenedor en un flex container.
- **align-items-center** alinea verticalmente los elementos en el centro.
- **justify-content-between** distribuye el espacio entre los elementos.
- **order-1, order-2, order-3** permiten cambiar el orden visual de los elementos.