

2024/25

## U.D.3. USO DE ESTILOS.

### 3.5.1 BOOTSTRAP: *introducción y diseño responsive.*



```
343 .widget-area-sidebar {top: 0; left: 0; width: 100%; height: 100%; position: absolute; z-index: 1000; background-color: #f0f0f0; transition: all 0.3s ease; border-right: 1px solid #ccc; padding: 10px; }
344 .widget-area-sidebar .wp-block-navigation {background-color: #fff; border-bottom: 1px solid #ccc; padding: 10px; margin-bottom: 10px; }
345 .widget-area-sidebar .wp-block-navigation ul {list-style-type: none; padding-left: 0; }
346 .widget-area-sidebar .wp-block-navigation li {margin-bottom: 10px; }
347 .widget-area-sidebar .wp-block-navigation li:last-child {margin-bottom: 0; }
348 .widget-area-sidebar .wp-block-navigation li a {color: #0070C0; font-size: 13px; }
349 }
350
351 /* =Menu */
352
353 #access {
354   display: inline-block;
355   height: 69px;
356   float: right;
357   margin: 11px 28px 0px 0px;
358   max-width: 880px;
359 }
360
361 #access ul {
362   font-size: 13px;
363   list-style: none;
364   margin: 0 0 0 -0.8125em;
365   padding-left: 0;
366   z-index: 9999;
367   text-align: right;
368 }
369
370 #access li {
371   display: inline-block;
372   float: left;
373 }
```



## ÍNDICE

<b>1. Introducción .....</b>	2
<b>2. Diferencias entre versiones .....</b>	4
<b>3. Descarga .....</b>	5
<b>4. Primeros pasos.....</b>	9
4.1   ¿Cómo integro Bootstrap a mi código? .....	9
<b>5. La Grid de Bootstrap.....</b>	11
5.1   Breakpoints .....	11
5.2   Reglas .....	12
5.2.1   Containers .....	14
5.3   Columnas y filas .....	15
5.3.1   Offset clases .....	17
5.3.2   Display Property .....	18
<b>6. Flexbox dentro de Bootstrap .....</b>	20
6.1   Flex-behaviour .....	20
6.2   Flex-direction.....	21
6.3   Justify-content.....	24
6.4   Align-items.....	26
6.5   Align-self .....	29
6.6   Margin y padding.....	31
6.7   Wrap .....	34
6.8   Order .....	37
6.9   Align-content .....	38
<b>7. Sizing: Width y Height .....</b>	43

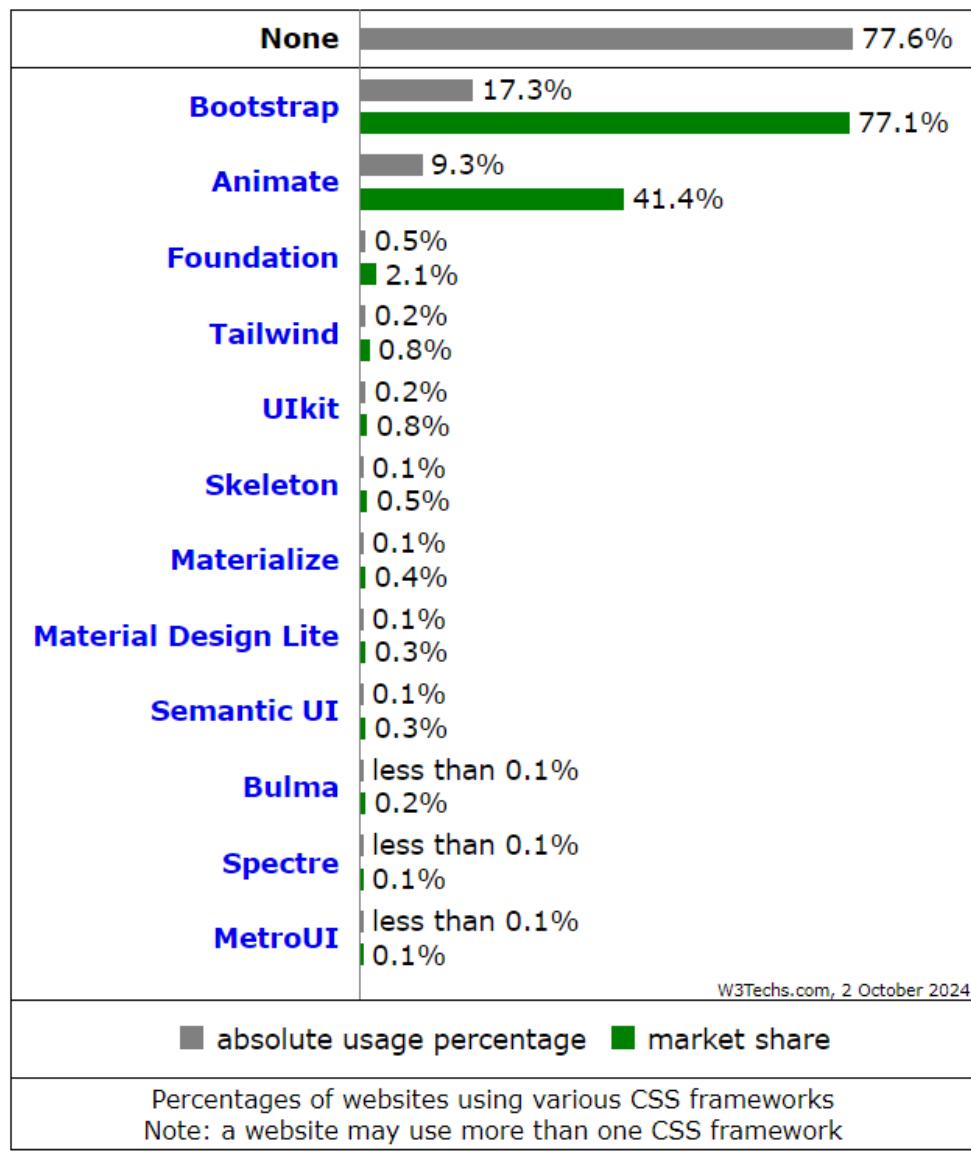


## 1. Introducción

**Bootstrap**, creado por Twitter, constituye uno de los **frameworks CSS** más populares en la actualidad; esto es, **plantillas CSS** con gran capacidad de adaptación a distintos tipos de plataformas web mediante el uso de librerías CSS, que incluyen un **amplio repositorio** de tipografías, botones y cuadros, entre otros.



Según w3techs.com, Bootstrap es el framework CSS más utilizado:



[https://w3techs.com/technologies/overview/css\\_framework](https://w3techs.com/technologies/overview/css_framework)

La versión actual es **Bootstrap 5.x**, aunque veremos también la **4.x** debido a que es una de las versiones más implementadas actualmente en el mundo empresarial.



Bootstrap está en código abierto y se encuentra disponible en el primer enlace. Puedes acceder a las plantillas a través del segundo enlace.

- <http://getbootstrap.com/getting-started/>
- <https://startbootstrap.com/templates>

Esta herramienta permite crear un **ilimitado conjunto de interfaces** combinando todos los recursos que este framework pone al alcance de los usuarios. Es decir, **Bootstrap nos da las clases para crear sitios con estilo y responsive** sin tener que preocuparnos por crear las media queries.

Los puntos fuertes de Bootstrap son:

- Es **fácil de instalar**, de **usar** y muy bien documentado.
- Es **compatible** con la mayor parte de los navegadores más extendidos hoy en día: Google Chrome, Safari, Mozilla Firefox, Internet Explorer y Opera.
- Es usado en **millones de sitios** y de los más **importantes**.
- Está mantenido con continuas **mejoras** y **correcciones** de errores.

Existen distintas **plantillas** sobre las que se pueden realizar los desarrollos y también se puede comenzar desde cero combinando todos los elementos que Bootstrap pone a disposición.



## 2. Diferencias entre versiones

A continuación, vamos a ver las **principales diferencias** entre las distintas versiones de Bootstrap, desde la 3.x a la 5.x.

	Bootstrap 3.x (2013)	Bootstrap 4.x (2018)	Bootstrap 5.x (2021)
<b>Flex</b>	No	Si	Si
<b>Preprocesador CSS</b>	Less	Sass	Sass
<b>Sistema Grid o de columnas.</b>	4 niveles (xs, sm, md, lg)	5 niveles (xs, sm, md, lg, xl)	6 niveles (xs, sm, md, lg, xl, xxl).
<b>Jquery</b>	Si	Si	No. Vanilla JS.
<b>Unidades de medida</b>	px o em (tamaño en relación al tamaño del contenedor)	rem (tamaño en relación al tamaño de letra que tiene el nodo raíz de nuestro documento). px son utilizados para los breakpoints y el width de los contenedores.	rem (tamaño en relación al tamaño de letra que tiene el nodo raíz de nuestro documento). px son utilizados para los breakpoints y el width de los contenedores.
<b>Tipografía por defecto.</b>	font-size de 14px, con un line-height de 1.428.	font-size de 1rem (16px) y un line-height de 1.5	font-size de 1rem (16px) y un line-height de 1.5
<b>Iconos</b>	Fuente de iconos Glyphicons	No tiene propios iconos. Se importan de Fontawesome.	Tiene sus propios iconos SVG.
<b>Colores</b>	Colores limitados.	Colores limitados, pero mejorados.	Se mejoran las paletas de colores.
<b>Black/Dark tables</b>	No soportada	con la clase .table-dark.	con la clase .table-dark.
<b>Imágenes responsive</b>	con la clase .img-responsive.	con la clase .img-fluid.	con la clase .img-fluid.

Puedes encontrar una guía completa de Bootstrap en:

- [https://www.w3schools.com/bootstrap/bootstrap\\_ver.asp](https://www.w3schools.com/bootstrap/bootstrap_ver.asp)
- <https://getbootstrap.com/docs/5.3/getting-started/introduction/>



### 3. Descarga

Bootstrap se puede descargar desde la página web oficial

<https://getbootstrap.com/docs/5.3/getting-started/download/>

Hay varias formas de descargar Bootstrap. Algunas de ellas son:

- 1) **Compiled CSS and JS.** Nos va a dar los archivos CSS y Javascript compilados.

Simplemente tenemos que importarlos en nuestro proyecto, como cualquier archivo HTML o CSS.

#### Compiled CSS and JS

Download ready-to-use compiled code for **Bootstrap v5.2.2** 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.** Nos va a permitir descargar Bootstrap, pero esta versión **no está compilada**. Esto significa que vamos a tener archivos de Sass que vamos a tener que compilar, y también vamos a tener que trabajar con otras herramientas como “Autoprefixer” y todas las que se mencionen en “build tools”.

#### 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](#)

Esta opción es más **avanzada** que la anterior, y suele ser utilizada por personas que quieren **editar Bootstrap**, obteniendo así una versión de Bootstrap más personalizada.



3) **Examples.** Podemos descargar algunos ejemplos de interfaces con su código CSS y HTML, los cuales podemos modificar.

## Examples

If you want to download and examine our [examples](#), you can grab the already built examples:

[Download Examples](#)

Algunos de ellos son:

Pricing example

Features Enterprise Support Pricing

## Pricing

Quickly build an effective pricing table for your potential customers with this Bootstrap example. It's built with default Bootstrap components and utilities with little customization.

Free	Pro	Enterprise
\$0/mo	\$15/mo	\$29/mo
10 users included	20 users included	30 users included
2 GB of storage	10 GB of storage	15 GB of storage
Email support	Priority email support	Phone and email support
Help center access	Help center access	Help center access

[Sign up for free](#) [Get started](#) [Contact us](#)

### Compare plans

	Free	Pro	Enterprise
Public	✓	✓	✓
Private		✓	✓
Permissions	✓	✓	✓
Sharing		✓	✓
Unlimited members	✓	✓	✓
Extra security			✓

© 2017-2023

**Features**  
Cool stuff  
Random feature  
Team feature  
Stuff for developers  
Another one  
Last time

**Resources**  
Resource  
Resource name  
Another resource  
Final resource

**About**  
Team  
Locations  
Privacy  
Terms

Punny headline  
And an even wider subheading  
Coming soon!

Another headline  
And an even wider subheading

**Sidebar**

[Home](#) [Dashboard](#) [Orders](#) [Products](#) [Customers](#)

**Sidebar**

[Home](#) [Dashboard](#) [Orders](#) [Products](#) [Customers](#)

**Collapsible**

Home  
Overview  
Updates  
Reports  
Dashboard  
Orders  
Account

**List group**

**List group item heading**  
Some placeholder content in a paragraph below the heading and date.

**List group item heading**  
Some placeholder content in a paragraph below the heading and date.

**List group item heading**  
Some placeholder content in a paragraph below the heading and date.

**List group item heading**  
Some placeholder content in a paragraph below the heading and date.

**List group item heading**  
Some placeholder content in a paragraph below the heading and date.



The image contains two side-by-side screenshots of Bootstrap-based web forms. The left screenshot shows a 'Checkout form' with three main sections: 'Billing address' (containing fields for First name, Last name, Username, Email, Address, and County), 'Your cart' (listing three items with descriptions and prices), and 'Payment' (with payment method selection, credit card number input, expiration date, and CVV). The right screenshot shows a 'Please sign in' form with fields for Email address and Password, a 'Remember me' checkbox, and a 'Sign in' button. Both forms include standard Bootstrap styling like buttons and input groups.

- 4) **CDN via jsDelivr.** Lo que nos va a permitir es que los archivos CSS y Javascript de Bootstrap estén alojados en una página web externa, que en este caso se llama jsDelivr. Lo único que tenemos que hacer en este caso, sería **copiar** las líneas de código que nos indican (estilos CSS y el archivo de Javascript), y pegarlos en nuestro código.

### 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@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js">
```

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@2.11.6/dist/umd/popper.min.js" integrity="sha384-hKSO背...>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.min.js" integrity="sha384-...>
```

Las **ventajas** de utilizar un CDN son:

- Reduce la carga de nuestros servidores.
- Facilita que Bootstrap quede en caché del navegador y se vuelva a cargar cada vez que se visita una web que lo utilice.
- Reduce la latencia.
- Reduce los costes de nuestros servidores por requerir menor ancho de banda y tráfico.



La **desventaja** de este método de descarga es que necesitamos una conexión a Internet para que funcione nuestra página o interface.

- 5) Mediante **gestores de paquetes**: npm, yarn, RubyGems, Composer, NuGet...



## 4. Primeros pasos

De los tipos de descarga vistos en el punto anterior, el más utilizado es el primero: **Compiled CSS and JS**. Por tanto, se va a describir el proceso de descarga e integración de Bootstrap en nuestro código, en base a este método de descarga.

### 4.1 ¿Cómo integro Bootstrap a mi código?

A continuación, se describen los pasos necesarios para integrar Bootstrap a nuestro proyecto web.

1) Descargamos Bootstrap por el método “Compiled CSS and JS”. Lo que se descarga es un fichero comprimido en formato .zip.

2) Descomprimimos el fichero .zip, el cual contendrá documentos css y js.



3) **Copiamos** las carpetas **css** y **js** a **nuestro proyecto**. En cuanto al **contenido** de cada una de las carpetas, tanto para la versión 4.x como 5.x, podemos destacar los siguientes documentos:

- **css:**

- **bootstrap.css**. Contiene todo el código CSS de Bootstrap.
- **bootstrap.min.css**. Es igual que el anterior, solo que está “minificado”, es decir, ocupa menos espacio. Por esta razón, **es el que utilizaremos**.
- **bootstrap-grid.css**. Contiene la parte de la Grid de Bootstrap. Este documento se usa solo, en aquellos casos en los que queramos usar sólo la parte responsive de Bootstrap.

- **js:**

- **bootstrap.js**. Contiene todo el código JS de Bootstrap.
- **bootstrap.min.js**. Es igual que el anterior, solo que está “minificado”, es decir, ocupa menos espacio. Por esta razón, **es el que utilizaremos**.

```
bootstrap/
└── css/
    ├── bootstrap-grid.css
    ├── bootstrap-grid.css.map
    ├── bootstrap-grid.min.css
    ├── bootstrap-grid.min.css.map
    ├── bootstrap-reboot.css
    ├── bootstrap-reboot.css.map
    ├── bootstrap-reboot.min.css
    └── bootstrap-reboot.min.css.map
    ├── bootstrap.css
    ├── bootstrap.css.map
    ├── bootstrap.min.css
    └── bootstrap.min.css.map
└── js/
    ├── bootstrap.bundle.js
    ├── bootstrap.bundle.js.map
    ├── bootstrap.bundle.min.js
    ├── bootstrap.bundle.min.js.map
    ├── bootstrap.js
    ├── bootstrap.js.map
    ├── bootstrap.min.js
    └── bootstrap.min.js.map
```



- 4) Creamos un `index.html` con la estructura base HTML de nuestro proyecto web. Es importante introducir el `meta viewport` en nuestro `<head>`, ya que si no lo hacemos, la parte adaptable o responsive de Bootstrap no funcionará.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

- 5) Enlazamos o vinculamos el documento `min.css` de Bootstrap en nuestro `<head>`:

```
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Bootstrap</title>
  <link rel="stylesheet" href="css/bootstrap.min.css">

</head>
```

- 6) Para trabajar con javascript, enlazamos o vinculamos el archivo `min.js` de Bootstrap en nuestro `<body>`:

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

- 7) Ahora sólo tenemos que **añadir clases** en nuestro `<body>`. Bootstrap funciona mediante clases. Esto significa que, al implementar sus clases, nosotros **no tendremos que preocuparnos por crear código CSS**.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap</title>
    <link rel="stylesheet" href="css/bootstrap.min.css">
  </head>
  <body>
    <h1>Hola Mundo</h1>
    <button class="btn btn-primary">Empezar</button>
    <script type="js/bootstrap.bundle.min.js"></script>
  </body>
</html>
```

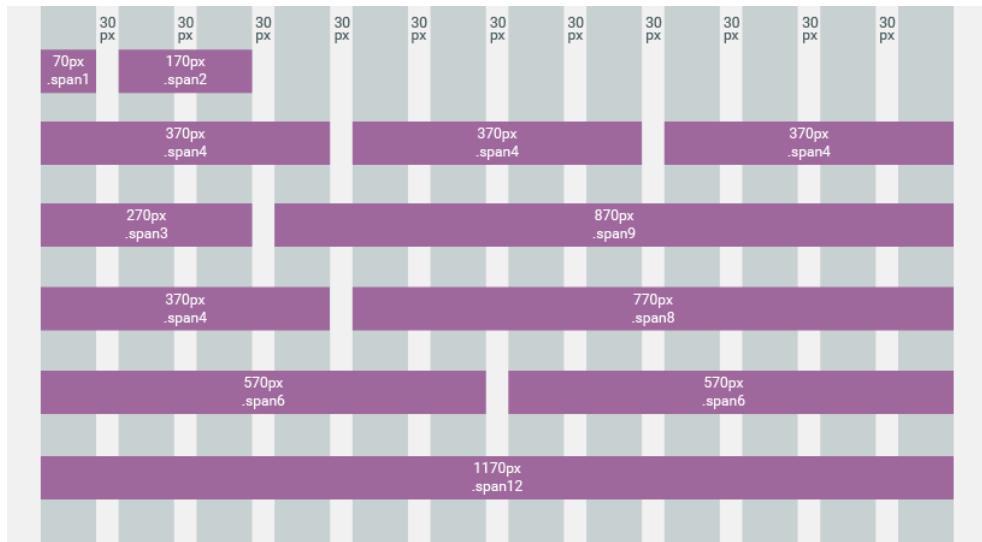
Hola Mundo

Empezar



## 5. La Grid de Bootstrap

Bootstrap trabaja con un sistema de **12 columnas** sobre las que se dispone los diferentes elementos que conforman la interfaz. Esto significa que tenemos 12 columnas disponibles para asignarle a cualquier elemento. Por ejemplo:



La Grid de Bootstrap trabaja con **Flexbox**, por lo que gracias a eso podemos hacer cosas como decirle que el espacio sobrante lo divida en medio de los elementos, que alinee elementos como el aside de forma vertical de una forma sencilla usando una simple clase, etc.

### 5.1 Breakpoints

Tenemos también la capacidad de **asignar tamaños de columna dependiendo del tamaño del dispositivo** que el usuario usa. Para ello, Bootstrap define los **breakpoints**.

En el mundo del diseño responsive, un **breakpoint es un tamaño de pantalla (en píxeles) donde se produce un cambio en la disposición de los elementos** que conforman nuestra página web.

Existen multitud de tipos de dispositivos, multitud de pantallas, pantallas que giran, etc. Por lo tanto, la elección de los distintos breakpoints no es algo trivial. Sin embargo, los profesionales de Twitter, usando su experiencia y los datos que sus millones de usuarios les proporcionan, han decidido que la elección más adecuada son **5 breakpoints** (v.4.x) o **6 breakpoints** (v.5.x):



## Bootstrap 4.x:

	sm	md	lg	xl
Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-
# of columns	12			
Gutter width	30px (15px on each side of a column)			
Nestable	Yes			
Column ordering	Yes			

## Bootstrap 5.x:

Breakpoint	Class infix	Dimensions
Extra small	None	<576px
Small	sm	≥576px
Medium	md	≥768px
Large	lg	≥992px
Extra large	xl	≥1200px
Extra extra large	xxl	≥1400px

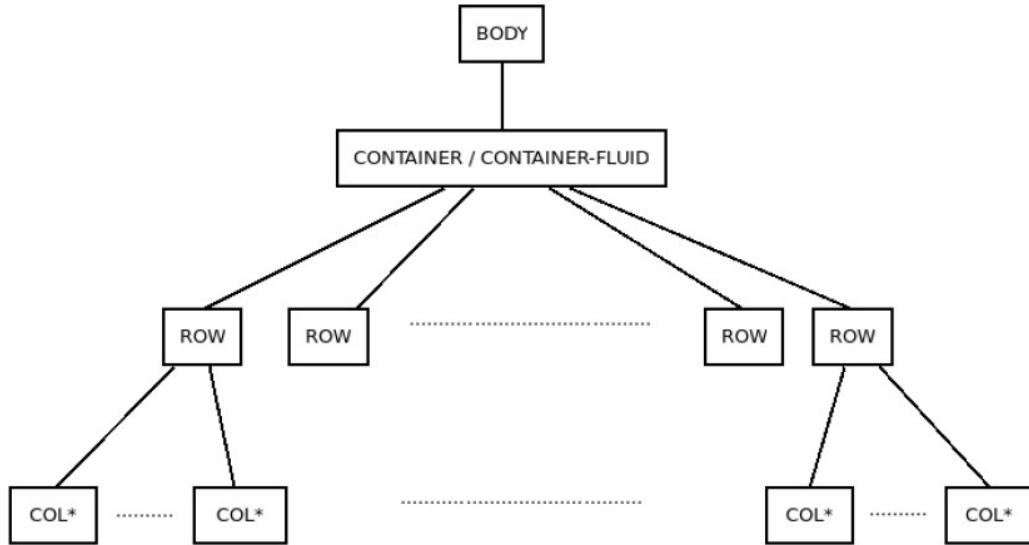
## 5.2 Reglas

Para utilizar la Grid de Bootstrap y obtener así un sitio responsive, debemos seguir las siguientes **reglas**:

- 1) Todo nuestro contenido lo tenemos que “encerrar” dentro de **clases contenedores**, normalmente asignadas a elementos **div**.
- 2) Un **contenedor o container principal** debe encerrar todo el contenido.



3) Dentro del **contenedor principal** tenemos que poner **filas**. Dentro de las filas tenemos que poner **columnas**. Por tanto, cada fila va a “encerrar” las columnas, permitiendo así separarlas de otras columnas.



A continuación, se presenta un ejemplo:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta http-equiv="X-UA-Compatible" content="IE-edge">
    <title>Grid</title>
    <link rel="stylesheet" href="../css/normalize.css">
    <link rel="stylesheet" type="text/css" href="../css/bootstrap.min.css">
    <style>
        .row > div {
            background: #f2f2f2;
            margin: 10px 0;
            padding: 40px;
        }
    </style>
</head>

<body>
    <div class="container">
        <div class="row">
            <div class="col-12 col-sm-6">
                <h1>Hola Mundo</h1>
                <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum rutrum neque ac dui facilisis, a sagittis justo tincidunt. Praesent tellus lacus, auctor at erat ac, malesuada accumsan lectus. Duis convallis ornare dui facilisis luctus. Donec non turpis id diam aliquam bibendum. </p>
                <button class="btn btn-primary">Ver más</button>
            </div>
            <div class="col-12">
                <h1>Hola Mundo</h1>
                <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum rutrum neque ac dui facilisis, a sagittis justo tincidunt. Praesent tellus lacus, auctor at erat ac, malesuada accumsan lectus. Duis convallis ornare dui facilisis luctus. Donec non turpis id diam aliquam bibendum. </p>
                <button class="btn btn-primary">Ver más</button>
            </div>
        </div>
        <script src="../js/bootstrap.bundle.min.js"></script>
    </div>
</body>
</html>
```

definimos la clase “container” (otra clase container podría ser “container-fluid”).

dentro del contendor hemos definido una fila, y dentro de esta, columnas.

div con tamaño de 12 columnas, desde el tamaño extra pequeño en adelante.

div con tamaño de 12 columnas desde el tamaño extra pequeño en adelante; y de 6 columnas desde el tamaño small en adelante.



El resultado será de este código ejemplo es el siguiente:

## 5.2.1 Containers

Un contenedor es el padre de todos los elementos de nuestra página web. Es una etiqueta que, como regla general, va a contener todas las otras etiquetas del contenido de nuestra página. Normalmente es un **div**, ya que es la etiqueta que se utiliza para maquetar en HTML.

Bootstrap nos da **dos tipos** de contenedores:

- **class="container"**. Ocupará diferentes anchos dependiendo del tamaño de la pantalla. Puede ocupar toda la pantalla o dejar unos márgenes a izquierda y a derecha aunque, en este caso, siempre estará centrado.

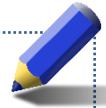
```
<body>
  <div class="container">
    <!-- -->
  </div>
</body>
```

- **class="container-fluid"**. Ocupará todo el ancho (100%) de lo que podemos ver en el navegador **<body>**. Por tanto, nos permite eliminar los espacios a la izquierda y a la derecha. Esto es útil para cuando queramos que **nuestra página no tenga espaciados** y abarque el 100% de la pantalla.

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



### Actividad propuesta 1.



Descarga Bootstrap (versión 5.x) y reescribe el código del apartado 5.2. Cambia la clase contenedor a `class="container-fluid"` y fíjate en su salida. ¿Qué **diferencias** ves entre utilizar `container` y `container-fluid`?

## 5.3 Columnas y filas

Tal y como se ha comentado, el container es el contenedor principal. En él irán todas las filas y columnas. Por tanto, con el contenedor podemos centrar nuestra página.

Las **filas** (row) son **grupos horizontales de columnas**, y se declaran mediante la clase `"row"` (`class="row"`).

Los tamaños de columna en Bootstrap 4.x y 5.x son:

- `.col`: extra pequeño (extra small). Pantallas con tamaño **inferior a 576px**, como **smartphones** en orientación vertical.
- `.col-sm`: pequeño (small o **sm**). Pantallas iguales o mayores que **576px** e inferiores a **768px**, como **smartphones** en orientación horizontal.
- `.col-md`: mediano (medium o **md**). Pantallas iguales o mayores que **768px** e inferiores a **992px**, como **tablets**.
- `.col-lg`: largo (large o **lg**). Pantallas iguales o mayores que **992px** e inferiores a **1200px**, como las de **desktop**.
- `.col-xl`: extra largo (extra large o **xl**). Pantallas iguales o superiores a **1200px**, como las de **desktop**.
- `.col-xxl`: extra extra largo (extra extra large o **xxl**). Pantallas iguales o superiores a **1400px**, como las de **TV**. **Solo para Bootstrap 5.x.**



Hay que tener en cuenta es que, si establecemos un tamaño para un cierto tamaño de pantallas, ese tamaño se va a mantener para pantallas de mayor tamaño. Para pantallas menores ocupará toda la pantalla (12 col).

Podemos también **fijar a la vez anchos diferentes** para distintos tipos de tamaños de pantallas. Para ello añadiremos varias clases a un mismo elemento.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="X-UA-Compatible" content="IE-edge">
<title>Grid</title>
<link rel="stylesheet" href="../css/normalize.css">
<link rel="stylesheet" type="text/css" href="../css/bootstrap.min.css">
<style>
    .row > div {
        background: #f2f2f2;
        margin: 10px 0;
        padding: 40px;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="row">
            <div class="col-12 col-sm-5 col-md-4 col-xl-12">
                <h1>Hola Mundo</h1>
                <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum rutrum neque ac dui facilisis, a sagittis justo tincidunt. Praesent tellus lacus, auctor at erat ac, malesuada accumsan lectus. Duis convallis ornare dui facilisis luctus. Donec non turpis id diam aliquam bibendum. </p>
                <button class="btn btn-primary">Ver más</button>
            </div>
            <div class="col-12">
                <h1>Hola Mundo</h1>
                <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum rutrum neque ac dui facilisis, a sagittis justo tincidunt. Praesent tellus lacus, auctor at erat ac, malesuada accumsan lectus. Duis convallis ornare dui facilisis luctus. Donec non turpis id diam aliquam bibendum. </p>
                <button class="btn btn-primary">Ver más</button>
            </div>
        </div>
        <script src="../js/bootstrap.bundle.min.js"></script>
    </div>
</body>
</html>
```

div con tamaño de 12 columnas, desde el tamaño extra pequeño en adelante.

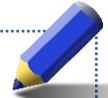
div con tamaño de 6 columnas, desde el tamaño pequeño en adelante.

div con tamaño de 4 columnas, desde el tamaño medium en adelante.

div con tamaño de 12 columnas, desde el tamaño xl en adelante.

Si en un **row** pasamos de 12 columnas, ese elemento que provoca que nos pasemos de las 12 columnas “pasará” debajo de los elementos de la fila actual ocupando el tamaño establecido.

### Actividad propuesta 2.



Escribe un documento html con Bootstrap que contenga dos elementos `<div>` con un color de fondo distinto. Estos `<div>` deben estar divididos de la siguiente manera:

- El primer `<div>` debe ocupar 12 columnas en pantallas extra pequeñas y 8 en medianas, 6 en pantallas xl.
- El segundo `<div>` debe ocupar 12 columnas siempre excepto cuando nos encontramos una pantalla de desktop o más grandes, que tendrá un tamaño de 4 columnas.



### 5.3.1 Offset clases

Las columnas se pueden mover a la derecha utilizando las clases `.offset-X`. Estas clases aumentan el margen izquierdo de una columna en X columnas para todo tipo de pantallas.

Este desplazamiento puede ser también establecido de manera diferente para cada tamaño de pantalla: `.offset-*X`, donde `X` es el número de **columnas** a la izquierda que queremos desplazar, y `*` es el tamaño de pantalla a partir de la cual queremos que se aplique (`sm` | `md` | `lg` | `xl` | `xxl`).

A continuación, se presenta un ejemplo:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
    <title>Offset</title>

    <style>
        .row > div {
            height: 50px;
        }
    </style>
</head>

<body>
    <div class="container">
        <div class="row">
            <div class="col-md-4 mt-4 bg-info".col-md-4</div>
            <div class="col-md-4 offset-md-4 mt-4 bg-warning".col-md-4 .offset-md-4</div>
        </div>

        <div class="row">
            <div class="col-md-3 offset-md-3 mt-4 bg-info".col-md-3 .offset-md-3</div>
            <div class="col-md-3 offset-md-3 mt-4 bg-warning".col-md-3 .offset-md-3</div>
        </div>

        <div class="row">
            <div class="col-md-6 offset-md-3 mt-4 bg-danger text-white".col-md-6 .offset-md-3</div>
        </div>
    </div>
    <script src="/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```



.col-md-4

.col-md-4 .offset-md-4

.col-md-3 .offset-  
md-3

.col-md-3 .offset-  
md-3

.col-md-6 .offset-md-3



### 5.3.2 Display Property

La propiedad **display** especifica el comportamiento de visualización de un elemento. Algunos de los **valores** que puede tomar son los siguientes:

- **none**. El elemento no se muestra. Es **eliminado** por completo.
- **inline**. Muestra un elemento como un elemento en **línea** (como `<span>`). Las propiedades de height y width no tendrán ningún efecto.
- **inline-block**. Muestra un elemento como un contenedor de **bloque a nivel inline**. El elemento en sí se formatea como un elemento en línea, pero puede aplicar valores de height y width.
- **block**. Muestra un elemento como un elemento de bloque (como `<p>`). Comienza en una nueva línea y **ocupa todo el ancho**.
- **flex**. Muestra un elemento como **contenedor flex** a nivel de bloque.
- **inline-flex**. Muestra un elemento como **contenedor flex** a nivel de línea.

Su **sintaxis** es la siguiente:

```
.d-{value} for xs  
.d-{breakpoint}-{value} for sm, md, lg, and xl.
```

A continuación, se presenta un pequeño ejemplo:

```
<!DOCTYPE html>  
<html>  
<head>  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1">  
    <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">  
    <title>Offset</title>  
  
    <style>  
        p {  
            height: 80px;  
            padding: 10px;  
        }  
    </style>  
</head>  
  
<body>  
    <div class="container">  
        <div class="row">  
            <div class="col">  
                <p class="d-block bg-primary text-white">d-block</p>  
                <p class="d-block d-md-inline d-lg-none bg-dark text-white">d-block d-md-inline d-lg-none</p>  
                <p class="d-inline bg-primary text-white">d-inline</p>  
                <p class="d-inline bg-secondary text-white">d-inline</p>  
                <p class="d-inline-block bg-success text-white">d-inline-block</p>  
                <p class="d-md-inline-block bg-warning text-white">d-md-inline-block</p>  
            </div>  
        </div>  
    </div>  
  
<script src="/js/bootstrap.bundle.min.js"></script>  
</body>  
</html>
```



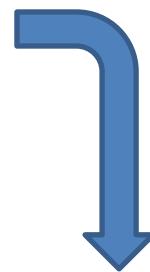
tamaño sm

d-block

d-block d-md-inline d-lg-none

d-inline d-inline d-inline-block

d-md-inline-block



tamaño md

d-block

d-block d-md-inline d-lg-none d-inline d-inline d-inline-block d-md-inline-block



tamaño lg

d-block

d-inline d-inline d-inline-block d-md-inline-block



## 6. Flexbox dentro de Bootstrap

A continuación, se presentan las **clases** que se emplean en Bootstrap para trabajar con **flexbox**.

### 6.1 Flex-behaviour

Cuando trabajamos con contenedores flex, debemos indicar con qué **tipo de contenedor**, **block o inline**, vamos a trabajar (al igual que hacíamos con la propiedad `display`).

- **.d-flex**. Le indicamos a un contenedor que queremos trabajar con flexbox. Los elementos hijo abarcarán el espacio que necesitan, poniéndose uno al lado del otro. El contenedor abarcará el **100% de su espacio**, ya que es de tipo **bloque**.
- **.d-inline-flex**. El contenedor no abarcará el 100%, ya que será de tipo **inline**. Es decir, solo abarcará el espacio que sus contenedores hijo necesiten. Por tanto, el contenedor usa el **espacio que necesita**.

Con estas clases podemos establecer también **desde qué tamaño de pantalla el contenedor será de tipo bloque o inline**. Para ello debemos introducir el prefijo correspondiente:

- **.d-sm-flex**
- **.d-sm-inline-flex**
- **.d-md-flex**
- **.d-md-inline-flex**
- **.d-lg-flex**
- **.d-lg-inline-flex**
- **.d-xl-flex**
- **.d-xl-inline-flex**

A continuación, se muestra un [ejemplo](#):



```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Flexbox 1</title>
    <link rel="stylesheet" type="text/css" href="../css/bootstrap.min.css">
    <style>
        .contenedor {
            margin-bottom: 20px;
            background: #8d6efd;
            color: #fff;
        }
        .contenedor > div {
            background: rgba(255, 255, 255, .2);
            padding: 15px;
            border: 1px solid #8d6efd;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="row my-5"> <!--row con margin de 5 en el eje y -->
            <div class="col">
                <h4>Tipos de Display Flex</h4>
                <hr>
                <!--background del color primario, un texto blanco, y un padding de 3 -->
                <div class="bg-primary text-white p-3 d-flex">
                    <!-- La clase mb es margin-bottom. Le pondemos un margin-bottom de 0 al parrafo -->
                    <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>
            </div>
        </div>
    </div>
</body>
</html>
```



## Tipos de Display Flex

Display Flex.d-flex

Display Inline Flex .d-inline-flex

## 6.2 Flex-direction

Las clases **flex-direction** nos permiten trabajar con la dirección de los elementos, es decir, nos permiten controlar la dirección en la que nuestros elementos hijos se van a posicionar. Tenemos las siguientes:



- **.flex-row**: nos permite poner los elementos hijo que están dentro de nuestro contenedor, en formato **fila de izquierda a derecha**. Por defecto se aplica cuando usamos la clase **.d-flex** en el contenedor padre.
- **.flex-row-reverse**: es igual a **flex-row**, solo que ordena los elementos hijos de **derecha a izquierda**. Esto es, cambia el flujo de elementos, posicionándolos de derecha a izquierda.
- **.flex-column**: nos permite colocar los elementos hijo en formato columna o **vertical de arriba a abajo**.
- **.flex-column-reverse**: es igual a **.flex-column**, es decir, ordena los elementos hijo en columna o **vertical** pero de **abajo a arriba**.

Con estas clases podemos establecer también **desde qué tamaño de pantalla queremos que se apliquen**. Para ello debemos introducir el prefijo correspondiente:

- **.flex-\* -row**
- **.flex-\* -row-reverse**
- **.flex-\* -column**
- **.flex-\* -column-reverse**

\* = sm | md | lg | xl | xxl

A continuación, se muestra un [ejemplo](#):

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Flexbox 1</title>
    <link rel="stylesheet" type="text/css" href="../css/bootstrap.min.css">
    <style>
        .contenedor {
            margin-bottom: 20px;
            background: #8d6efd;
            color: #fff;
        }
        .contenedor > div {
            background: rgba(255, 255, 255, .2);
            padding: 15px;
            border: 1px solid #8d6efd;
        }
    </style>
</head>
```



```
<body>
  <div class="container">
    <div class="row my-5">
      <div class="col">
        <h4>Direction</h4>
        <hr>
        <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>
      </div>
    </div>
  </body>
</html>
```



## Direction

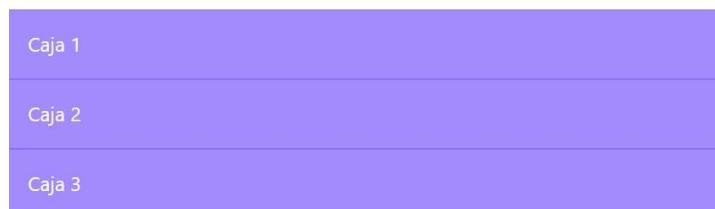
.flex-row



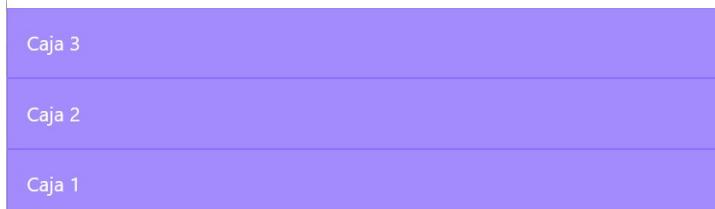
.flex-row-reverse



.flex-column



.flex-column-reverse





## 6.3 Justify-content

Las clases **justify-content** nos permiten trabajar con el contenido y poder alinearlo de forma **horizontal**. Tenemos las siguientes:

- **.justify-content-start**: toma los elementos hijo y los posiciona en el lado izquierdo del contenedor padre.
- **.justify-content-center**: posiciona los elementos hijo en el centro del contenedor padre.
- **.justify-content-end**: toma los elementos hijo y los posiciona en el lado derecho del contenedor padre.
- **.justify-content-between**: distribuye el espacio “sobrante” del contenedor padre, y lo distribuye entre los elementos hijo.
- **.justify-content-around**: distribuye el espacio “sobrante” del contenedor padre, y lo distribuye entre los elementos hijo, antes del primer elemento hijo, y después del último elemento hijo.
- **.justify-content-evenly**: distribuye el espacio “sobrante” del contenedor padre de forma proporcional entre cada uno de los elementos hijo. Solo para **Bootstrap 5.x**.

Con estas clases podemos establecer también **desde qué tamaño de pantalla queremos que se apliquen**. Para ello debemos introducir el prefijo correspondiente:

- **.justify-content-\* -start**
- **.justify-content-\* -end**
- **.justify-content-\* -center**
- **.justify-content-\* -between**
- **.justify-content-\* -around**
- **.justify-content-\* -evenly**

\* = sm | md | lg | xl | xxl

A continuación, se muestra un [ejemplo](#):



```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Flexbox 1</title>
    <link rel="stylesheet" type="text/css" href="../css/bootstrap.min.css">
    <style>
        .contenedor {
            margin-bottom: 20px;
            background: #8d6efd;
            color: #fff;
        }
        .contenedor > div {
            background: rgba(255, 255, 255, .2);
            padding: 15px;
            border: 1px solid #8d6efd;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="row my-5"> <!--row con margin de 5 en el eje y --&gt;
            &lt;div class="col"&gt;
                &lt;h4&gt;Justify Content&lt;/h4&gt;
                &lt;hr&gt;
                &lt;p&gt;.justify-content-start&lt;/p&gt;
                &lt;div class="contenedor d-flex justify-content-start"&gt;
                    &lt;div&gt;Caja 1&lt;/div&gt;
                    &lt;div&gt;Caja 2&lt;/div&gt;
                    &lt;div&gt;Caja 3&lt;/div&gt;
                &lt;/div&gt;

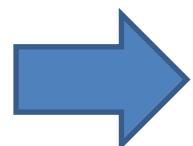
                &lt;p&gt;.justify-content-center&lt;/p&gt;
                &lt;div class="contenedor d-flex justify-content-center"&gt;
                    &lt;div&gt;Caja 1&lt;/div&gt;
                    &lt;div&gt;Caja 2&lt;/div&gt;
                    &lt;div&gt;Caja 3&lt;/div&gt;
                &lt;/div&gt;

                &lt;p&gt;.justify-content-end&lt;/p&gt;
                &lt;div class="contenedor d-flex justify-content-end"&gt;
                    &lt;div&gt;Caja 1&lt;/div&gt;
                    &lt;div&gt;Caja 2&lt;/div&gt;
                    &lt;div&gt;Caja 3&lt;/div&gt;
                &lt;/div&gt;

                &lt;p&gt;.justify-content-between&lt;/p&gt;
                &lt;div class="contenedor d-flex justify-content-between"&gt;
                    &lt;div&gt;Caja 1&lt;/div&gt;
                    &lt;div&gt;Caja 2&lt;/div&gt;
                    &lt;div&gt;Caja 3&lt;/div&gt;
                    &lt;div&gt;Caja 4&lt;/div&gt;
                &lt;/div&gt;

                &lt;p&gt;.justify-content-around&lt;/p&gt;
                &lt;div class="contenedor d-flex justify-content-around"&gt;
                    &lt;div&gt;Caja 1&lt;/div&gt;
                    &lt;div&gt;Caja 2&lt;/div&gt;
                    &lt;div&gt;Caja 3&lt;/div&gt;
                    &lt;div&gt;Caja 4&lt;/div&gt;
                &lt;/div&gt;

                &lt;!-- solo para bootstrap 5 en adelante --&gt;
                &lt;p&gt;.justify-content-evenly&lt;/p&gt;
                &lt;div class="contenedor d-flex justify-content-evenly"&gt;
                    &lt;div&gt;Caja 1&lt;/div&gt;
                    &lt;div&gt;Caja 2&lt;/div&gt;
                    &lt;div&gt;Caja 3&lt;/div&gt;
                    &lt;div&gt;Caja 4&lt;/div&gt;
                &lt;/div&gt;
            &lt;/div&gt;
        &lt;/div&gt;
        &lt;script src="../js/bootstrap.bundle.min.js"&gt;&lt;/script&gt;
    &lt;/body&gt;
&lt;/html&gt;</pre>
```



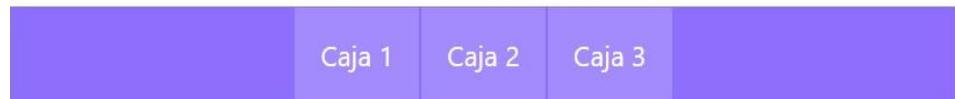


## Justify Content

.justify-content-start



.justify-content-center



.justify-content-end



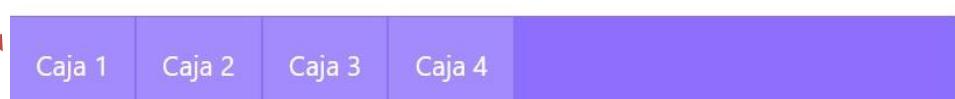
.justify-content-between



.justify-content-around



.justify-content-evenly



No funcionó  
porque estamos  
trabajando sobre  
Bootstrap 4.x.



## 6.4 Align-items

Las clases `align-items` nos permiten alinear elementos hijo de forma **vertical**, dentro del contenedor padre. Estas son:

- `.align-items-start`: nos permite posicionar los elementos hijo en la parte superior del contenedor padre.
- `.align-items-center`: nos permite posicionar los elementos hijo en el centro (vertical) del contenedor padre.



- **.align-items-end**: nos permite posicionar los elementos hijo en la parte inferior vertical del contenedor padre.
- **.align-items-baseline**: nos permite posicionar o alinear (verticalmente) los elementos hijo en base al texto (o baseline).
- **.align-items-stretch**: permite que los elementos hijo abarquen el 100% del espacio vertical disponible.
- **.flex-fill**. Esta clase solo es compatible con **Bootstrap 5.x**. Toma el **espacio sobrante y se lo asigna al elemento hijo** sobre el cual se le ha asignado esta clase. Por tanto, esta clase **se define/asigna sobre el elemento hijo** y no en el contenedor padre.

Con estas clases podemos establecer también **desde qué tamaño de pantalla queremos que se apliquen**. Para ello debemos introducir el prefijo correspondiente:

- **.align-items-\* -start**
- **.align-items-\* -end**
- **.align-items-\* -center**
- **.align-items-\* -baseline**
- **.align-items-\* -stretch**
- **.flex-\* -fill**

\* = sm | md | lg | xl | xxl

A continuación, se muestra un ejemplo:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Flexbox 2</title>
    <link rel="stylesheet" type="text/css" href="../css/bootstrap.min.css">

    <style>
        .contenedor {
            margin-bottom: 20px;
            background: #8d6efd;
            color: #fff;
            height: 100px;
        }
        .contenedor > div {
            background: rgba(255, 255, 255, .2);
            padding: 15px;
            border: 1px solid #8d6efd;
        }
    </style>
</head>
```



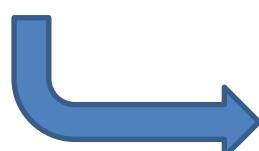
```
<body>
  <div class="container">
    <div class="row my-5"> <!--row con margin de 5 en el eje y -->
      <div class="col">
        <h4>Fill - solo bootstrap 5</h4>
        <hr>
        <div class="contenedor d-flex align-items-start">
          <div>Lorem ipsum dolor sit amet.</div>
          <div class="flex-fill">.flex-fill</div>
          <div class="flex-fill">.flex-fill</div>
        </div>
      </div>
    </div>
    <div class="row my-5"> <!--row con margin de 5 en el eje y -->
      <div class="col">
        <h4>Align items</h4>
        <hr>
        <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>
    </div>
  </div>
  <script src="../js/bootstrap.bundle.min.js"></script>
</body>
</html>
```





### Fill - solo bootstrap 5


### Align items

.align-items-start

Caja 1	Caja 2	Caja 3

.align-items-center

Caja 1	Caja 2	Caja 3

.align-items-end

Caja 1	Caja 2	Caja 3

.align-items-baseline

Caja 1	Caja 2	Caja 3

.align-items-stretch

Caja 1	Caja 2	Caja 3

## 6.5 Align-self

Las clases **align-self** se definen y aplican a los **elementos hijo**. Permiten alinear **verticalmente** a los elementos hijo de forma **individual**. Estas son:

- **.align-self-start**: nos permite posicionar el elemento hijo en la parte superior del contenedor padre.
- **.align-self-center**: nos permite posicionar el elemento hijo en el centro (vertical) del contenedor padre.
- **.align-self-end**: nos permite posicionar el elemento hijo en la parte inferior vertical del contenedor padre.
- **.align-self-baseline**: nos permite posicionar o alinear (verticalmente) el elemento hijo en base al texto (o baseline).



- **.align-self-stretch**: permite que el elemento hijo abarque el 100% del espacio vertical disponible.

Con estas clases podemos establecer también **desde qué tamaño de pantalla queremos que se apliquen**. Para ello debemos introducir el prefijo correspondiente:

- **.align-self-\*-start**
- **.align-self-\*-end**
- **.align-self-\*-center**
- **.align-self-\*-baseline**
- **.align-self-\*-stretch**

\* = sm | md | lg | xl | xxl

A continuación, se muestra un ejemplo:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Flexbox 2</title>
    <link rel="stylesheet" type="text/css" href="../css/bootstrap.min.css">

    <style>
        .contenedor {
            margin-bottom: 20px;
            background: #8d6efd;
            color: #fff;
            height: 100px;
        }
        .contenedor > div {
            background: rgba(255, 255, 255, .2);
            padding: 15px;
            border: 1px solid #8d6efd;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="row my-5"> <!--row con margin de 5 en el eje y -->
            <div class="col">
                <h4>Align Self</h4>
                <hr>
                <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>
            </div>
        </div>
    </div>
</body>
</html>
```



## Align Self

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



## 6.6 Margin y padding

Las clases `margin` nos permiten poner **márgenes horizontales y/o verticales** de forma **individual** a cada elemento hijo. Por tanto, se define y aplica a cada elemento hijo de forma individual. Tenemos las siguientes clases, según estemos trabajando con Bootstrap 4.x o Bootstrap 5.x. No son exclusivos de flex.

- `.mr-auto | 1 a 5.` **Bootstrap 4.x.** Margen derecho o “margin right”. Permite los valores: auto o números del 1 al 5.
- `.me-auto | 1 a 5.` **Bootstrap 5.x.** Margen derecho o “margin end”. Permite los valores: auto o números del 1 al 5.
- `.ml-auto | 1 a 5.` **Bootstrap 4.x.** Margen izquierdo o “margin left”. Permite los valores: auto o números del 1 al 5.
- `.ms-auto | 1 a 5.` **Bootstrap 5.x.** Margen izquierdo o “margin start”. Permite los valores: auto o números del 1 al 5.
- `.mb-auto | 1 a 5.` Margen inferior o “margin bottom”. Permite los valores: auto o números del 1 al 5.
- `.mt-auto | 1 a 5.` Margen superior o “margin top”. Permite los valores: auto o números del 1 al 5.
- `.mx-auto | 1 a 5.` Margen derecho e izquierdo. Permite los valores: auto o números del 1 al 5.
- `.my-auto | 1 a 5.` Margen superior e inferior. Permite los valores: auto o números del 1 al 5.

Como ya se ha visto, además de auto, podemos asignar valores numéricos del 1 al 5, los cuales equivalen a las siguientes medidas:

	Size	Visual representation
1	0.25rem	
2	0.5rem	
3	1rem	
4	1.5rem	
5	3rem	



Además, si sustituimos la letra m de todas estas clases por p, estaremos controlando el **padding**.

- **.pr-auto | 1 a 5.** **Bootstrap 4.x.** Margen derecho o “padding right”. Permite los valores: auto o números del 1 al 5.
- **.pe-auto | 1 a 5.** **Bootstrap 5.x.** Padding derecho o “padding end”. Permite los valores: auto o números del 1 al 5.
- **.pl-auto | 1 a 5.** **Bootstrap 4.x.** Padding izquierdo o “padding left”. Permite los valores: auto o números del 1 al 5.
- **.ps-auto | 1 a 5.** **Bootstrap 5.x.** Padding izquierdo o “padding start”. Permite los valores: auto o números del 1 al 5.
- **.pb-auto | 1 a 5.** Padding inferior o “padding bottom”. Permite los valores: auto o números del 1 al 5.
- **.pt-auto | 1 a 5.** Padding superior o “padding top”. Permite los valores: auto o números del 1 al 5.
- **.px-auto | 1 a 5.** Padding derecho e izquierdo. Permite los valores: auto o números del 1 al 5.
- **.py-auto | 1 a 5.** Padding superior e inferior. Permite los valores: auto o números del 1 al 5.

Al igual que las anteriores clases, cada una de estas clases se puede establecer según el tamaño de pantalla. Para ello debemos introducir el prefijo correspondiente.

- **.mr-\* -auto | 1 a 5.** (Bootstrap 4.x)
- **.me-\* -auto | 1 a 5.** (Bootstrap 5.x.)
- **.ml-\* -auto | 1 a 5.** (Bootstrap 4.x.)
- **.ms-\* -auto | 1 a 5.** (Bootstrap 5.x.)
- **.mb-\* -auto | 1 a 5.**
- **.mt-\* -auto | 1 a 5.**
- **.mx-\* -auto | 1 a 5.**
- **.my-\* -auto | 1 a 5.**
- **.pr-\* -auto | 1 a 5.** (Bootstrap 4.x.)
- **.pe-\* -auto | 1 a 5.** (Bootstrap 5.x.)
- **.pl-\* -auto | 1 a 5.** (Bootstrap 4.x.)

\* = sm | md | lg | xl | xxl



- `.ps-* -auto` | 1 a 5. (Bootstrap 5.x.)
- `.pb-* -auto` | 1 a 5.
- `.pt-* -auto` | 1 a 5.
- `.px-* -auto` | 1 a 5.
- `.py-* -auto` | 1 a 5.

A continuación, se muestra un ejemplo sencillo para Bootstrap 4.x:

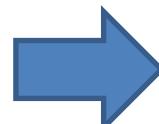
```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Flexbox 2</title>
<link rel="stylesheet" type="text/css" href="../css/bootstrap.min.css">

<style>
    .contenedor {
        margin-bottom: 20px;
        background: #8d6efd;
        color: #fff;
        height: 100px;
    }
    .contenedor > div {
        background: rgba(255, 255, 255, .2);
        padding: 15px;
        border: 1px solid #8d6efd;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="row my-5"> <!--row con margin de 5 en el eje y -->
            <div class="col">
                <h4>Auto Margins</h4>
                <hr>
                <div class="contenedor d-flex justify-content-end">
                    <div class="mr-auto">.mr-auto</div> <!--me-auto para bootstrap5 -->
                    <div>Caja 2</div>
                    <div>Caja 3</div>
                </div>

                <div class="contenedor d-flex justify-content-start">
                    <div>Caja 1</div>
                    <div>Caja 2</div>
                    <div class="ml-auto">.ml-auto</div>
                    <!--ms-auto para bootstrap5 -->
                </div>

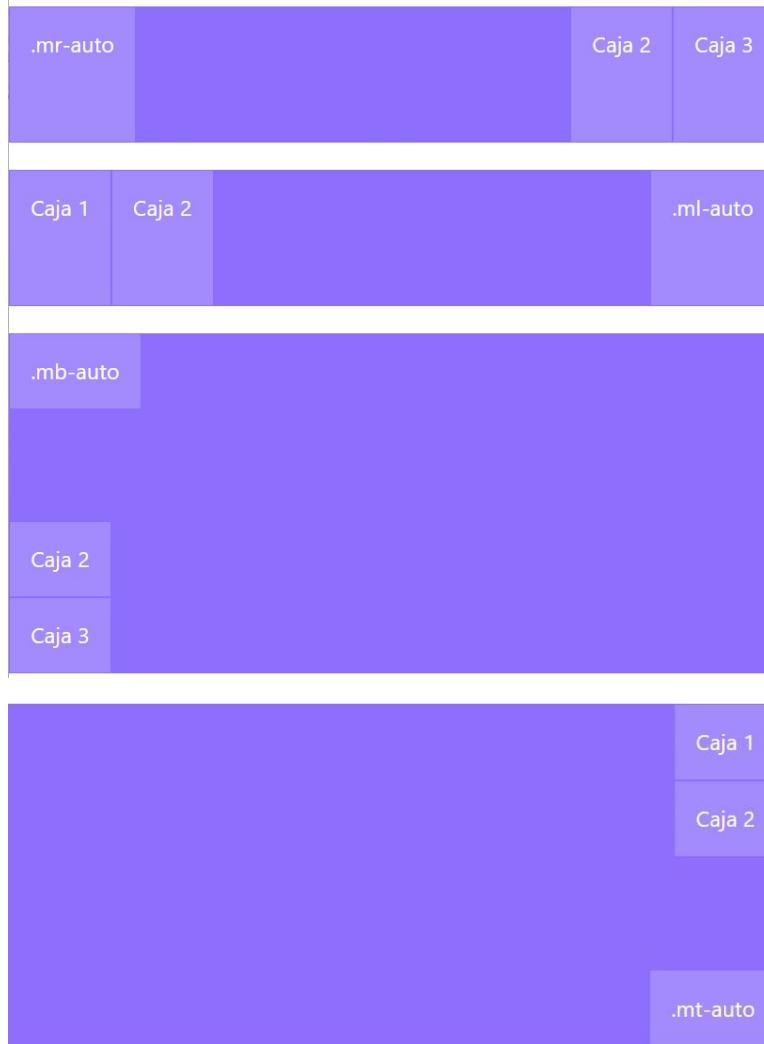
                <div class="contenedor d-flex flex-column align-items-start" style="height: 250px">
                    <div class="mb-auto">.mb-auto</div>
                    <div>Caja 2</div>
                    <div>Caja 3</div>
                </div>

                <div class="contenedor d-flex flex-column align-items-end" style="height: 250px">
                    <div>Caja 1</div>
                    <div>Caja 2</div>
                    <div class="mt-auto">.mt-auto</div>
                </div>
            </div>
        </div>
    </div>
<script src="../js/bootstrap.bundle.min.js"></script>
</body>
</html>
```





## Auto Margins



## 6.7 Wrap

Con las clases `wrap` podemos distribuir o no, los elementos hijo en diferentes líneas del contenedor padre. Podemos definir las siguientes:

- **.flexnowrap**. Es la clase por defecto. Con esta le indicamos al contenedor padre que no queremos que los elementos hijo ocupen varias filas, si no que queremos que dichos elementos hijo intenten estar **en una sola fila**.
- **.flex-wrap**. Con esta clase damos permiso para que los elementos hijo ocupen el espacio que sea necesario y, por tanto, se distribuyan en **distintas filas** dentro del contenedor padre si así fuese necesario.
- **.flex-wrap-reverse**. Es igual que la anterior clase, pero los elementos hijo se distribuirán de izquierda inferior a derecha.



Cada una de estas clases se puede establecer según el tamaño de pantalla. Para ello debemos introducir el prefijo correspondiente.

- *.flex-\* nowrap.*
- *.flex-\* wrap.*
- *.flex-\* wrap-reverse.*

\* = sm | md | lg | xl | xxl

A continuación, se muestra un ejemplo sencillo:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Flexbox 3</title>
    <link rel="stylesheet" type="text/css" href="../css/bootstrap.min.css">

    <style>
        .contenedor {
            margin-bottom: 20px;
            background: #8d6efd;
            color: #fff;
            min-height: 100px;
        }
        .contenedor > div {
            background: rgba(255, 255, 255, .2);
            padding: 15px 20px;
            border: 1px solid #8d6efd;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="row my-5" > <!--row con margin de 5 en el eje y -->
            <div class="col">
                <h4>Wrap</h4>
                <hr>
                <p>.flex-nowrap</p>
                <div class="contenedor d-flex flex-nowrap">
                    <div>Caja 1</div>
                    <div>Caja 2</div>
                    <div>Caja 3</div>
                    <div>Caja 4</div>
                    <div>Caja 5</div>
                    <div>Caja 6</div>
                    <div>Caja 7</div>
                    <div>Caja 8</div>
                    <div>Caja 9</div>
                    <div>Caja 10</div>
                    <div>Caja 11</div>
                    <div>Caja 12</div>
                    <div>Caja 13</div>
                    <div>Caja 14</div>
                    <div>Caja 15</div>
                </div>
            </div>
        </div>
    </div>
</body>
```



```
<p>.flex-wrap</p>
<div class="contenedor d-flex flex-wrap">
  <div>Caja 1</div>
  <div>Caja 2</div>
  <div>Caja 3</div>
  <div>Caja 4</div>
  <div>Caja 5</div>
  <div>Caja 6</div>
  <div>Caja 7</div>
  <div>Caja 8</div>
  <div>Caja 9</div>
  <div>Caja 10</div>
  <div>Caja 11</div>
  <div>Caja 12</div>
  <div>Caja 13</div>
  <div>Caja 14</div>
  <div>Caja 15</div>
</div>

<p>.flex-wrap-reverse</p>
<div class="contenedor d-flex flex-wrap-reverse">
  <div>Caja 1</div>
  <div>Caja 2</div>
  <div>Caja 3</div>
  <div>Caja 4</div>
  <div>Caja 5</div>
  <div>Caja 6</div>
  <div>Caja 7</div>
  <div>Caja 8</div>
  <div>Caja 9</div>
  <div>Caja 10</div>
  <div>Caja 11</div>
  <div>Caja 12</div>
  <div>Caja 13</div>
  <div>Caja 14</div>
  <div>Caja 15</div>
</div>
</div>
</div>
<script src="../js/bootstrap.bundle.min.js"></script>
</body>
</html>
```



## Wrap

.flexnowrap

Caja 1	Caja 2	Caja 3	Caja 4	Caja 5	Caja 6	Caja 7
--------	--------	--------	--------	--------	--------	--------

.flex-wrap

Caja 1	Caja 2	Caja 3	Caja 4	Caja 5	
Caja 6	Caja 7	Caja 8	Caja 9	Caja 10	
Caja 11	Caja 12	Caja 13	Caja 14	Caja 15	

.flex-wrap-reverse

Caja 11	Caja 12	Caja 13	Caja 14	Caja 15	
Caja 6	Caja 7	Caja 8	Caja 9	Caja 10	
Caja 1	Caja 2	Caja 3	Caja 4	Caja 5	



## 6.8 Order

Con las clases `order` podemos controlar el orden de visualización del contenido de nuestra interfaz web.

- `.order-x`, siendo x un número de **1 a 12** para **Bootstrap 4.x** y de **1 a 5** para **Bootstrap 5.x**. Esta clase se aplica de forma individual a los elementos hijo, de manera que va a distribuir el elemento hijo en la posición que le indiquemos.

La clase `.order-x` se puede establecer según el tamaño de pantalla. Para ello debemos introducir el prefijo correspondiente.

- `.order-*-1 a 5` (**Bootstrap 5.x**)
- `.order-*-1 a 12` (**Bootstrap 4.x**)

`*` = `sm` | `md` | `Lg` | `XL` | `xxL`

A continuación, se muestra un ejemplo sencillo:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Flexbox 3</title>
<link rel="stylesheet" type="text/css" href="../css/bootstrap.min.css">

<style>
    .contenedor {
        margin-bottom: 20px;
        background: #8d6efd;
        color: #fff;
        min-height: 100px;
    }
    .contenedor > div {
        background: rgba(255, 255, 255, .2);
        padding: 15px 20px;
        border: 1px solid #8d6efd;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="row my-5"> <!--row con margin de 5 en el eje y -->
            <div class="col">
                <h4>Order</h4>
                <hr>
                <div class="contenedor d-flex flex-wrap">
                    <div class="order-2">Caja 1</div>
                    <div class="order-3">Caja 2</div>
                    <div class="order-1">Caja 3</div>
                </div>
            </div>
        </div>
    </div>
    <script src="../js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

### Order





## 6.9 Align-content

Las clases ***align-content*** nos permite **alinear verticalmente** y de forma **grupal** (a nivel de contenedor padre), los elementos hijo. Sólo tiene efecto en los contenedores padre de varias líneas o filas, en los que ***flex-wrap*** o ***flexnowrap*** está configurado.

Las clases ***align-content*** son similares a ***align-items***, por lo que solo podemos usar una de las dos.

- ***.align-content-start***. Alinea el conjunto de elementos hijo **verticalmente**, en la **parte superior** del contenedor padre.
- ***.align-content-center***. Alinea el conjunto de elementos hijo **verticalmente**, en la **parte central** del contenedor padre.
- ***.align-content-end***. Alinea el conjunto de elementos hijo **verticalmente**, en la **parte inferior** del contenedor padre.
- ***.align-content-between***. El espacio sobrante vertical se distribuye entre los elementos hijo, **exceptuando el espacio inicial y el final**.
- ***.align-content-around***. El espacio sobrante vertical se distribuye entre los elementos hijo, **incluyendo el espacio inicial y el final**.
- ***.align-content-stretch***. Los elementos hijo se ajustan para **llenar el contenedor padre**. Respeta el padding. Si utilizamos esta clase, no podemos definir la altura de los elementos hijo.

Las clases ***.align-content*** se pueden establecer según el tamaño de pantalla. Para ello debemos introducir el prefijo correspondiente.

- ***.align-content-\* -start***
- ***.align-content-\* -center***
- ***.align-content-\* -end***
- ***.align-content-\* -between***
- ***.align-content-\* -around***
- ***.align-content-\* -stretch***

\* = sm | md | lg | xl | xxl

A continuación, se muestra un ejemplo sencillo:



```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Flexbox 3</title>
    <link rel="stylesheet" type="text/css" href="../css/bootstrap.min.css">

    <style>
        .contenedor {
            margin-bottom: 20px;
            background: #8d6efd;
            color: #fff;
            min-height: 100px;
        }
        .contenedor > div {
            background: rgba(255, 255, 255, .2);
            padding: 15px 20px;
            border: 1px solid #8d6efd;
        }
    </style>
</head>

<body>
    <div class="container">
        <div class="row my-5"> <!--row con margin de 5 en el eje y --&gt;
            &lt;div class="col"&gt;
                &lt;h4&gt;Align Content&lt;/h4&gt;
                &lt;hr&gt;

                &lt;p&gt;.align-content-start&lt;/p&gt;
                &lt;div class="contenedor d-flex flex-wrap align-content-start" style="height: 200px"&gt;
                    &lt;div&gt;Caja 1&lt;/div&gt;
                    &lt;div&gt;Caja 2&lt;/div&gt;
                    &lt;div&gt;Caja 3&lt;/div&gt;
                    &lt;div&gt;Caja 4&lt;/div&gt;
                    &lt;div&gt;Caja 5&lt;/div&gt;
                    &lt;div&gt;Caja 6&lt;/div&gt;
                    &lt;div&gt;Caja 7&lt;/div&gt;
                    &lt;div&gt;Caja 8&lt;/div&gt;
                    &lt;div&gt;Caja 9&lt;/div&gt;
                    &lt;div&gt;Caja 10&lt;/div&gt;
                    &lt;div&gt;Caja 11&lt;/div&gt;
                    &lt;div&gt;Caja 12&lt;/div&gt;
                    &lt;div&gt;Caja 13&lt;/div&gt;
                    &lt;div&gt;Caja 14&lt;/div&gt;
                    &lt;div&gt;Caja 15&lt;/div&gt;
                &lt;/div&gt;
                &lt;p&gt;.align-content-center&lt;/p&gt;
                &lt;div class="contenedor d-flex flex-wrap align-content-center" style="height: 200px"&gt;
                    &lt;div&gt;Caja 1&lt;/div&gt;
                    &lt;div&gt;Caja 2&lt;/div&gt;
                    &lt;div&gt;Caja 3&lt;/div&gt;
                    &lt;div&gt;Caja 4&lt;/div&gt;
                    &lt;div&gt;Caja 5&lt;/div&gt;
                    &lt;div&gt;Caja 6&lt;/div&gt;
                    &lt;div&gt;Caja 7&lt;/div&gt;
                    &lt;div&gt;Caja 8&lt;/div&gt;
                    &lt;div&gt;Caja 9&lt;/div&gt;
                    &lt;div&gt;Caja 10&lt;/div&gt;
                    &lt;div&gt;Caja 11&lt;/div&gt;
                    &lt;div&gt;Caja 12&lt;/div&gt;
                    &lt;div&gt;Caja 13&lt;/div&gt;
                    &lt;div&gt;Caja 14&lt;/div&gt;
                    &lt;div&gt;Caja 15&lt;/div&gt;
                &lt;/div&gt;
            &lt;/div&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/body&gt;</pre>
```



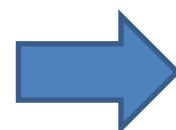
```
<p>.align-content-end</p>
<div class="contenedor d-flex flex-wrap align-content-end" style="height: 200px">
    <div>Caja 1</div>
    <div>Caja 2</div>
    <div>Caja 3</div>
    <div>Caja 4</div>
    <div>Caja 5</div>
    <div>Caja 6</div>
    <div>Caja 7</div>
    <div>Caja 8</div>
    <div>Caja 9</div>
    <div>Caja 10</div>
    <div>Caja 11</div>
    <div>Caja 12</div>
    <div>Caja 13</div>
    <div>Caja 14</div>
    <div>Caja 15</div>
</div>

<p>.align-content-between</p>
<div class="contenedor d-flex flex-wrap align-content-between" style="height: 200px">
    <div>Caja 1</div>
    <div>Caja 2</div>
    <div>Caja 3</div>
    <div>Caja 4</div>
    <div>Caja 5</div>
    <div>Caja 6</div>
    <div>Caja 7</div>
    <div>Caja 8</div>
    <div>Caja 9</div>
    <div>Caja 10</div>
    <div>Caja 11</div>
    <div>Caja 12</div>
    <div>Caja 13</div>
    <div>Caja 14</div>
    <div>Caja 15</div>
</div>

<p>.align-content-around</p>
<div class="contenedor d-flex flex-wrap align-content-around" style="height: 200px">
    <div>Caja 1</div>
    <div>Caja 2</div>
    <div>Caja 3</div>
    <div>Caja 4</div>
    <div>Caja 5</div>
    <div>Caja 6</div>
    <div>Caja 7</div>
    <div>Caja 8</div>
    <div>Caja 9</div>
    <div>Caja 10</div>
    <div>Caja 11</div>
    <div>Caja 12</div>
    <div>Caja 13</div>
    <div>Caja 14</div>
    <div>Caja 15</div>
</div>

<p>.align-content-stretch</p>
<div class="contenedor d-flex flex-wrap align-content-stretch" style="height: 200px">
    <div>Caja 1</div>
    <div>Caja 2</div>
    <div>Caja 3</div>
    <div>Caja 4</div>
    <div>Caja 5</div>
    <div>Caja 6</div>
    <div>Caja 7</div>
    <div>Caja 8</div>
    <div>Caja 9</div>
    <div>Caja 10</div>
    <div>Caja 11</div>
    <div>Caja 12</div>
    <div>Caja 13</div>
    <div>Caja 14</div>
    <div>Caja 15</div>
</div>
</div>
</div>

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





## Align Content

.align-content-start

Caja 1	Caja 2	Caja 3	Caja 4	Caja 5	Caja 6	Caja 7	Caja 8
Caja 9	Caja 10	Caja 11	Caja 12	Caja 13	Caja 14	Caja 15	

.align-content-center

Caja 1	Caja 2	Caja 3	Caja 4	Caja 5	Caja 6	Caja 7	Caja 8
Caja 9	Caja 10	Caja 11	Caja 12	Caja 13	Caja 14	Caja 15	

.align-content-end

Caja 1	Caja 2	Caja 3	Caja 4	Caja 5	Caja 6	Caja 7	Caja 8
Caja 9	Caja 10	Caja 11	Caja 12	Caja 13	Caja 14	Caja 15	

.align-content-between

Caja 1	Caja 2	Caja 3	Caja 4	Caja 5	Caja 6	Caja 7	Caja 8
Caja 9	Caja 10	Caja 11	Caja 12	Caja 13	Caja 14	Caja 15	



.align-content-around

Caja 1	Caja 2	Caja 3	Caja 4	Caja 5	Caja 6	Caja 7	Caja 8
Caja 9	Caja 10	Caja 11	Caja 12	Caja 13	Caja 14	Caja 15	

.align-content-stretch

Caja 1	Caja 2	Caja 3	Caja 4	Caja 5	Caja 6	Caja 7	Caja 8
Caja 9	Caja 10	Caja 11	Caja 12	Caja 13	Caja 14	Caja 15	



## 7. Sizing: Width y Height

Con las propiedades *width* y *height* podemos modificar la anchura y la altura de los elementos, tal y como ya sabemos.

En Bootstrap, además, podemos definir estas propiedades, *width* y *height*, en relación al **padre** o a la **ventana de visualización**.

⇒ En **relación al parente**, los **valores** que acepta son: **25%, 50%, 75%, 100%**, y **auto** (por defecto). De manera que su sintaxis es:

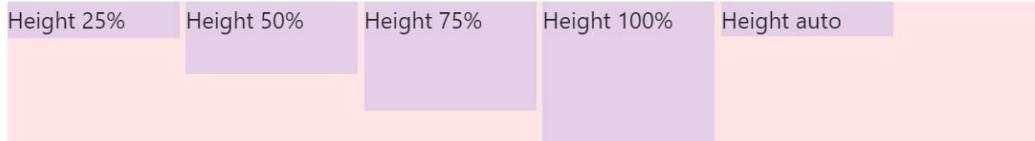
- Ancho: *w-valor*
- alto: *h-valor*

A continuación, se muestran unos ejemplos sencillos:

```
<div class="w-25 p-3" style="background-color: #eee;">Width 25%</div>
<div class="w-50 p-3" style="background-color: #eee;">Width 50%</div>
<div class="w-75 p-3" style="background-color: #eee;">Width 75%</div>
<div class="w-100 p-3" style="background-color: #eee;">Width 100%</div>
<div class="w-auto p-3" style="background-color: #eee;">Width auto</div>
```



```
<div style="height: 100px; background-color: rgba(255,0,0,.1);">
  <div class="h-25 d-inline-block" style="width: 120px; background-color: rgba(0,0,255,.1)">Height 25%</div>
  <div class="h-50 d-inline-block" style="width: 120px; background-color: rgba(0,0,255,.1)">Height 50%</div>
  <div class="h-75 d-inline-block" style="width: 120px; background-color: rgba(0,0,255,.1)">Height 75%</div>
  <div class="h-100 d-inline-block" style="width: 120px; background-color: rgba(0,0,255,.1)">Height 100%</div>
  <div class="h-auto d-inline-block" style="width: 120px; background-color: rgba(0,0,255,.1)">Height auto</div>
</div>
```



Más información en:

- *Bootstrap 4.x*: <https://getbootstrap.com/docs/4.6/utilities/sizing/>
- *Bootstrap 5.x*: <https://getbootstrap.com/docs/5.3/utilities/sizing/>