

Introducción. ¿Qué es bootstrap?

Fuente: codingpotions.com / bootstrap.com

Un **framework** aporta una estructura completa en la que nosotros "encajamos" nuestro código, implementando la lógica concreta de la aplicación. Es mucho más que una librería. Impone unas condiciones a nuestra aplicación e incluso puede definir su arquitectura. Es un *marco (framework)* en el que nosotros vamos a definir piezas. El *marco* define las reglas del juego a las que nos tenemos que atener. (Fuente: notajs.blogspot.com)

Bootstrap es un **framework CSS**, es decir, es un archivo CSS que se añade en los proyectos para tener una serie de estilos ya preparados para utilizar. Este tipo de librerías CSS suelen incluir estilos para los elementos más comunes de una página web, como por ejemplo, botones, tarjetas, navbars, etc. Además tiene una serie de estilos para crear columnas fácilmente.



¿Por qué usar Bootstrap? Ventajas

- Facilidad de uso. Lo único que tienes que hacer es poner clases a elementos HTML.
- Responsive. Perfecto para crear páginas web adaptables a cualquier dispositivo.
- Personalizable. Puedes personalizar su descarga para usar los elementos que necesites.
- Gran comunidad. Este framework está muy extendido y si tenemos un problema podremos encontrar mucha información en Internet.

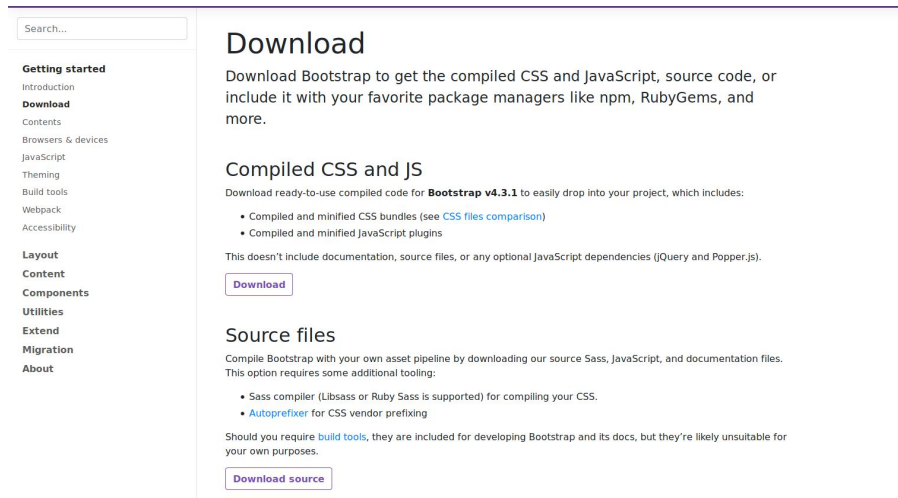
Descarga y instalación de Bootstrap

Para usar Bootstrap en un proyecto tenemos dos formas de hacerlo:

1) Instalación de forma manual

Para instalar Bootstrap de forma manual lo que tienes que hacer es descargar los archivos de Bootstrap desde su página web haciendo clic en Download dentro de Compiled CSS and JS:

<https://getbootstrap.com/docs/4.6/getting-started/introduction/>



Además, para que funcione correctamente hay que descargar también **jQuery** y **Popper.js**, un par de librerías que utiliza Bootstrap para que funcione correctamente.

jQuery: <https://jquery.com/>

Popper.js: <https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.15.0/popper.min.js> > Botón derecho > “Guardar cómo” y descargar.

Una vez descargados todos los archivos, se tienen que ubicar dentro de una carpeta en el proyecto. Evitar esta forma de importar si se usa algún framework como React, Vue o Angular.

Dentro de la carpeta que se ha descargado de Bootstrap hay una carpeta css y otra carpeta js. En la carpeta css es donde están los estilos y en la carpeta js el JavaScript necesario para que algunos elementos funcionen.

Dentro de la carpeta css tiene que estar el archivo que se llama **bootstrap.min.css** que es el que contiene todos los estilos y además está minificado sin espacios para que ocupe menos.

Dentro de la carpeta js tiene que estar el archivo **bootstrap.min.js**.

Estos dos archivos junto con el archivo js de **jQuery** y el archivo js de **Popper.js** tienes que tenerlos también dentro de tu proyecto.

Hecho esto ya puedes importar estos archivos dentro de la sección head de los archivos HTML:

```
<head>
  <link href="css/bootstrap.min.css" rel="stylesheet" media="screen">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <script src="js/jquery.min.js"></script>
  <script src="js/popper.min.js"></script>
  <script src="js/bootstrap.min.js"></script>
</head>
```

2) Mediante CDN, más sencillo

Este método es más sencillo porque solo tenemos que llamar desde dentro de la etiqueta head del HTML a cada una de los archivos en la nube y por tanto no tienes que descargar nada (Corroborar que sea la última versión disponible en la web de Bootstrap).

```
<head>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css"
integrity="sha384-B0vP5xmATw1+K9KRQJQERJvTumQW0nPEzvF6L/
Z6nronJ3oUOFUFpCjEUQouq2+I" crossorigin="anonymous">

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

<body>
...
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-9/reFTGAW83EW2RDu2S0VKalzap3H66lZH81PoYlFhbGU
+6BZp6G7niu735Sk7lN" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/js/bootstrap.min.js" integrity="sha384-
+YQ4JLhgyBLPDQt//I+STsc9iw4uQqACwlvpslubQzn4u2UU2UFM80nGisd026JF"
crossorigin="anonymous"></script>
</body>
```

Elementos que posee Bootstrap

Containers

Lo primero que vamos a ver son los contenedores. Los contenedores como su nombre indica, sirven para crear una “caja” o “contenedor” dentro de la que va el contenido de una página web.

Cuando le aplicas a un elemento HTML la clase **container** lo que ocurre es que a ese elemento se le aplica un **ancho** y un **padding** determinado y además se coloca en el **centro** de la página web.

Lo bueno de los containers es que van a ajustar su ancho automáticamente sin que tengas que hacer nada al ancho de los distintos elementos para que se vean correctamente.

Los contenedores son el elemento de diseño más básico en Bootstrap y son necesarios cuando se usa nuestro sistema de cuadrícula predeterminado. Los contenedores se utilizan para contener, rellenar y (a veces) centrar el contenido dentro de ellos. Si bien los

contenedores se pueden anidar, la mayoría de los diseños no requieren un contenedor anidado.

Bootstrap viene con tres contenedores diferentes:

- `.container`, que establece un **max-width** en todos los breakpoints responsive
- `.container-fluid`, que establece un **width: 100%** en todos los breakpoints
- `.container- {breakpoint}`, que tiene un **width: 100%** hasta el breakpoint especificado

La siguiente tabla ilustra cómo se compara el ancho máximo de cada contenedor con el `.container` y `.container-fluid` original en cada punto de interrupción.

Ejemplo: [Grid example](#).

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
<code>.container</code>	100%	540px	720px	960px	1140px
<code>.container-sm</code>	100%	540px	720px	960px	1140px
<code>.container-md</code>	100%	100%	720px	960px	1140px
<code>.container-lg</code>	100%	100%	100%	960px	1140px
<code>.container-xl</code>	100%	100%	100%	100%	1140px
<code>.container-fluid</code>	100%	100%	100%	100%	100%

All-in-one

La clase default `.container` es responsive y posee ancho fijo, lo cual quiere decir que su **max-width** cambia en cada breakpoint.

```
<div class="container">
  <!-- Content here -->
</div>
```

Fluid

La clase `.container-fluid` crea un contenedor con el 100% de ancho, ocupando todo el ancho del viewport.

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

Responsive

Los contenedores responsive fueron incorporados en la versión 4.4 de Bootstrap. Permiten especificar una clase que los muestra con ancho al 100% hasta que se alcance el breakpoint especificado. Por ejemplo: `.container-sm` tiene **width: 100%** hasta que el breakpoint `sm` es alcanzado, a partir de allí, se escalará con `md`, `lg`, y `xl`.

Copy

```
<div class="container-sm">100% ancho hasta el small breakpoint</div>
<div class="container-md">100% ancho hasta el medium breakpoint</div>
<div class="container-lg">100% ancho hasta el large breakpoint</div>
<div class="container-xl">100% ancho hasta el extra large breakpoint</div>
```

Sistema Grid

Utiliza una serie de contenedores, filas y columnas para maquetar y alinear contenido. Está construido con **flexbox** y es completamente responsive.

One of three columns	One of three columns	One of three columns
----------------------	----------------------	----------------------

```
<div class="container">
  <div class="row">
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
  </div>
</div>
```

Este ejemplo crea tres columnas iguales utilizando las clases del sistema grid predefinidas. Dichas columnas serán centradas en la página con el componente padre `.container`.

Los contenedores proveen una forma de centrar y desplazar horizontalmente los contenidos de la página. Se usa `.container` para un ancho responsive en píxeles o `.container-fluid` para un ancho de 100% en cualquier viewport y tamaño de dispositivo.

Los divs con las clases **row** son contenedores para las columnas. Cada columna tiene un padding horizontal, también llamado **gutter**, para controlar el espacio entre ellas. Este padding es contrarrestado en cada row con márgenes negativos.

Opciones

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Ancho máximo del contenedor	None (auto)	540px	720px	960px	1140px
Prefijo de clase	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>
# de columnas	12				
Tamaño del gutter	30px (15px a cada lado de la columna)				
Anidable	Si				
Ordenamiento de columnas	Si				

Ancho equitativo

1 of 2	2 of 2	
1 of 3	2 of 3	3 of 3

```
<div class="container">
```

```
<div class="row">
```

```
<div class="col">
```

```
1 of 2
```

```
</div>
```

```
<div class="col">
```

```
2 of 2
```

```
</div>
```

```
</div>
```

```
<div class="row">
```

```
<div class="col">
```

```
1 of 3
```

```
</div>
```

```
<div class="col">
```

```
2 of 3
```

```
</div>
```

```
<div class="col">
```

```
3 of 3
```

```
</div>
```

```
</div>
```

```
</div>
```

La clase **.w-100** hace que el contenido que le sigue al elemento que la posee se desplace hacia abajo.

col	col
col	col

```
<div class="container">
```

```
<div class="row">
```

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

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

```
<div class="w-100"></div>
```

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

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

```
</div>
```

```
</div>
```

Al tener un sistema de auto-layout, es posible setear el ancho de una columna y que las columnas “hermanas” se redimensionen automáticamente en base a dicho valor.

1 of 3	2 of 3 (wider)	3 of 3
1 of 3	2 of 3 (wider)	3 of 3

```
<div class="container">
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col-6">
      2 of 3 (wider)
    </div>
    <div class="col">
      3 of 3
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col-5">
      2 of 3 (wider)
    </div>
    <div class="col">
      3 of 3
    </div>
  </div>
</div>
```

Especificando el sufijo **-auto** las columnas se adaptarán en ancho a lo que ocupe su contenido.

1 of 3	Variable width content	3 of 3
1 of 3	Variable width content	3 of 3


```

<div class="container">
  <div class="row justify-content-md-center">
    <div class="col col-lg-2">
      1 of 3
    </div>
    <div class="col-md-auto">
      Variable width content
    </div>
    <div class="col col-lg-2">
      3 of 3
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col-md-auto">
      Variable width content
    </div>
    <div class="col col-lg-2">
      3 of 3
    </div>
  </div>
</div>

```

Columnas con espacio establecido numéricamente

El sistema de grillas reparte el espacio en 12 porciones. Se puede establecer cuántas de esas 12 porciones debe ocupar una columna introduciendo un número.

col	col	col	col
col-8			col-4

```

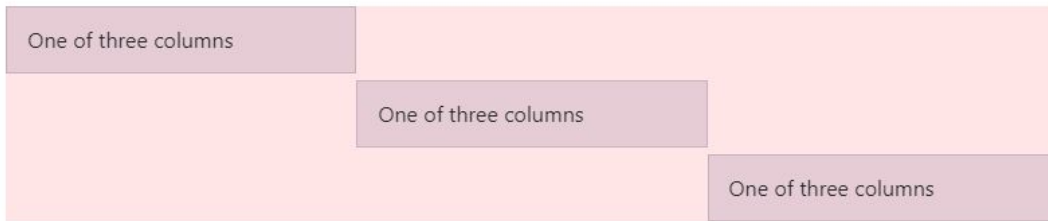
<div class="container">
  <div class="row">
    <div class="col">col</div>
    <div class="col">col</div>
    <div class="col">col</div>
    <div class="col">col</div>
  </div>
  <div class="row">
    <div class="col-8">col-8</div>
    <div class="col-4">col-4</div>
  </div>
</div>

```

Alineamiento vertical

One of three columns	One of three columns	One of three columns
One of three columns	One of three columns	One of three columns
One of three columns	One of three columns	One of three columns

```
<div class="container">
  <div class="row align-items-start">
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
  </div>
  <div class="row align-items-center">
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
  </div>
  <div class="row align-items-end">
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
  </div>
</div>
```

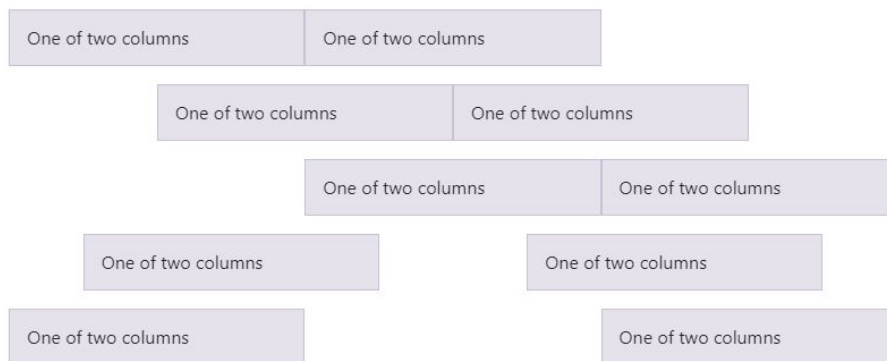


```

<div class="container">
  <div class="row">
    <div class="col align-self-start">
      One of three columns
    </div>
    <div class="col align-self-center">
      One of three columns
    </div>
    <div class="col align-self-end">
      One of three columns
    </div>
  </div>
</div>

```

Alineamiento horizontal



```

<div class="container">
  <div class="row justify-content-start">
    <div class="col-4">
      One of two columns
    </div>
    <div class="col-4">
      One of two columns
    </div>
  </div>
  <div class="row justify-content-center">
    <div class="col-4">
      One of two columns
    </div>
    <div class="col-4">
      One of two columns
    </div>
  </div>

```

```

</div>
<div class="row justify-content-end">
  <div class="col-4">
    One of two columns
  </div>
  <div class="col-4">
    One of two columns
  </div>
</div>
<div class="row justify-content-around">
  <div class="col-4">
    One of two columns
  </div>
  <div class="col-4">
    One of two columns
  </div>
</div>
<div class="row justify-content-between">
  <div class="col-4">
    One of two columns
  </div>
  <div class="col-4">
    One of two columns
  </div>
</div>
</div>

```

Sacar espacios

Los espacios (gutters) entre las columnas pueden ser eliminados incluyendo en el componente la clase **.no-gutters**.

```

<div class="row no-gutters">
  <div class="col-sm-6 col-md-8">.col-sm-6 .col-md-8</div>
  <div class="col-6 col-md-4">.col-6 .col-md-4</div>
</div>

```



Clases de ordenamiento

Se utiliza la clase **.order** para controlar el ordenamiento del contenido. Dicha clase es responsive.

Use **.order-** classes for controlling the visual order of your content. These classes are responsive, so you can set the **order** by breakpoint (e.g., **.order-1.order-md-2**). Includes support for **1** through **12** across all five grid tiers.

First in DOM, no order applied	Third in DOM, with an order of 1	Second in DOM, with a larger order
--------------------------------	----------------------------------	------------------------------------

```

<div class="container">
  <div class="row">
    <div class="col">
      First in DOM, no order applied
    </div>
    <div class="col order-12">
      Second in DOM, with a larger order
    </div>
    <div class="col order-1">
      Third in DOM, with an order of 1
    </div>
  </div>
</div>

```

También se puede usar **.order-first** u **.order-last** para cambiar el orden aplicando un orden de -1 y 13 respectivamente.

Third in DOM, ordered first	Second in DOM, unordered	First in DOM, ordered last
-----------------------------	--------------------------	----------------------------

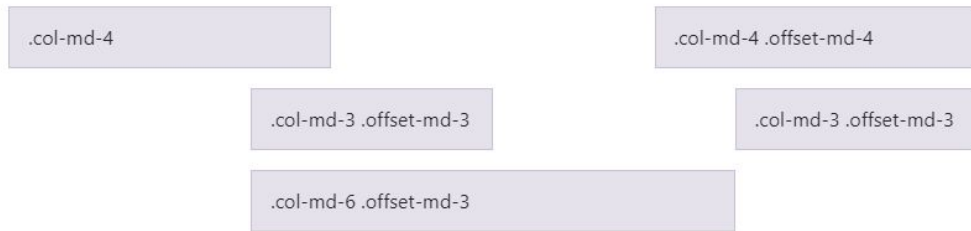
```

<div class="container">
  <div class="row">
    <div class="col order-last">
      First in DOM, ordered last
    </div>
    <div class="col">
      Second in DOM, unordered
    </div>
    <div class="col order-first">
      Third in DOM, ordered first
    </div>
  </div>
</div>

```

Aplicar offset a las columnas

Se puede desplazar a las columnas una cantidad de espacios con la clase responsive **.offset**. Por ejemplo: La clase **.offset-md-4** desplazará 4 columnas a la derecha al elemento que se le esté aplicando.



```
<div class="container">
  <div class="row">
    <div class="col-md-4">.col-md-4</div>
    <div class="col-md-4 offset-md-4">.col-md-4 .offset-md-4</div>
  </div>
  <div class="row">
    <div class="col-md-3 offset-md-3">.col-md-3 .offset-md-3</div>
    <div class="col-md-3 offset-md-3">.col-md-3 .offset-md-3</div>
  </div>
  <div class="row">
    <div class="col-md-6 offset-md-3">.col-md-6 .offset-md-3</div>
  </div>
</div>
```