

FRAMEWORK BOOTSTRAP 5.1.3

INDICE

1. ¿Qué es Bootstrap?	2
2. Principales características de Bootstrap	2
3. Requisitos previos para usar Bootstrap	3
4. Cómo instalar Bootstrap en tu proyecto	3
5. Cómo funciona Bootstrap	5
5.1 Sistema de rejillas (Grid)	5
5.2 Contenedores	7
5.3 MARGIN Y PADDING	8
5.4 ANCHURA DE UN ELEMENTO	9
6.1 Configuraciones tipográficas por defecto	9
6.2 Iconos Bootstrap	9

1. ¿Qué es Bootstrap?



Bootstrap es un **framework basado en HTML, CSS y JavaScript** para crear **webs “responsive”**, es decir, 100% adaptables a todo tipo de dispositivos móviles.

En el ámbito del diseño y desarrollo web un **framework** es un conjunto de herramientas basadas en lenguajes web como PHP, JavaScript y CSS, cuyo principal objetivo es el de facilitar al máximo el proceso de diseño de páginas webs.

Bootstrap fue desarrollado por **Twitter** en el año 2011 y debido a su simplicidad de uso, utilidad y coste nulo se ha posicionado como uno de los frameworks de referencia.

2. Principales características de Bootstrap

Bootstrap posee muchas **funcionalidades** distintas que hacen de él una herramienta muy valiosa para cualquier persona que quiera diseñar una web totalmente responsive y que a su vez esté optimizada para conservar así buenos tiempos de carga.

Sus principales características son:

- **Open Source:** es un framework de código abierto y por tanto accesible para todo el mundo, sin que esto implique ningún coste o limitación.
- **Sistema de Rejillas o Grid:** gracias a su sistema de rejilla basado en un diseño de 12 columnas, conseguimos adaptar nuestro contenido a los distintos tamaños de pantalla.
- **HTML5 y CSS3:** soporta las últimas versiones de HTML y CSS.
- **Optimización:** posee un código limpio y optimizado para que la web cargue lo más rápido posible.

- **Uso:** tiene una curva de aprendizaje moderada y cuenta con documentación suficiente para conseguir resultados desde el primer día.
- **Componentes:** además del Grid, posee elementos de fácil utilización como por ejemplo iconos, botones, menús desplegables, etc...
- **Compatibilidad:** el framework es compatible en los navegadores:
 - ✓ Google Chrome
 - ✓ Mozilla Firefox
 - ✓ Internet Explorer
 - ✓ Safari
 - ✓ Opera
- **LESS:** funciona a la perfección con [LESS](#), un preprocesador que permite escribir CSS de forma mucho más rápida y con muchas más opciones (como variables, etc), compilando (traduciendo) después el código a CSS puro, listo para que sea interpretado por el navegador.
- **Garantías:** ha sido creado por Twitter, una empresa de renombre mundial que asegura un correcto funcionamiento, así como continuas actualizaciones.

3. Requisitos previos para usar Bootstrap

Antes de comenzar a utilizar la herramienta, necesitarás **conocimientos previos en HTML y CSS**, ya que el framework está basado en su totalidad en estos lenguajes y, por tanto, no olvides incluir `<!DOCTYPE html>` en todas tus páginas.

Sin embargo, aunque **JavaScript** está también involucrado para activar y desactivar diferentes elementos, como es el caso de los menús desplegables de Bootstrap, no es necesario que tengas ningún conocimiento previo.

4. Cómo instalar Bootstrap en tu proyecto

Abre la página oficial <http://getbootstrap.com/>

Arriba a la derecha nos aparece la última versión, que a fecha de enero de 2024 es la **versión 5.3**.

Por compatibilidad con los ejercicios de clase, vamos a usar la **versión 5.1.3**. No hay diferencias importantes respecto a la 5.3

Hay dos opciones principales para poder **utilizar Bootstrap**: instalar los paquetes o usarlos directamente **enlazando con la librería CDN**.

Una **CDN** (Content Delivery Network) es **un conjunto de ubicaciones en el mundo**, que redistribuyen localmente el contenido de los servidores y guardan en caché los archivos que no necesitan actualización permanente, según unas reglas personalizables. La principal ventaja de una CDN es que reduce la latencia acercando los sitios web y las aplicaciones a sus usuarios y liberando los servidores y el ancho de banda que los conecta a internet.

Bootstrap utiliza la **CDN jsDelivr** para evitar que tengas que descargar el framework previamente y transferirlo a tu servidor.

Bootstrap te ofrece los enlaces necesarios para implementarlo, que corresponden a los ficheros **“bootstrap.min.css”** y **“bootstrap.bundle.min.js”**:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>
```

Cópalos y pégalos justo debajo de la etiqueta <title> de tu documento.

Y listo. Con eso ya **tendríamos el framework listo para ser usado**.

Nota: el código de Bootstrap suele ser **“minified”**, que significa código minificado. **Minificar** es la técnica que consiste en reducir el peso de los archivos de código fuente a través de la eliminación de bytes innecesarios (espacios adicionales, saltos de línea, sangrías y comentarios).

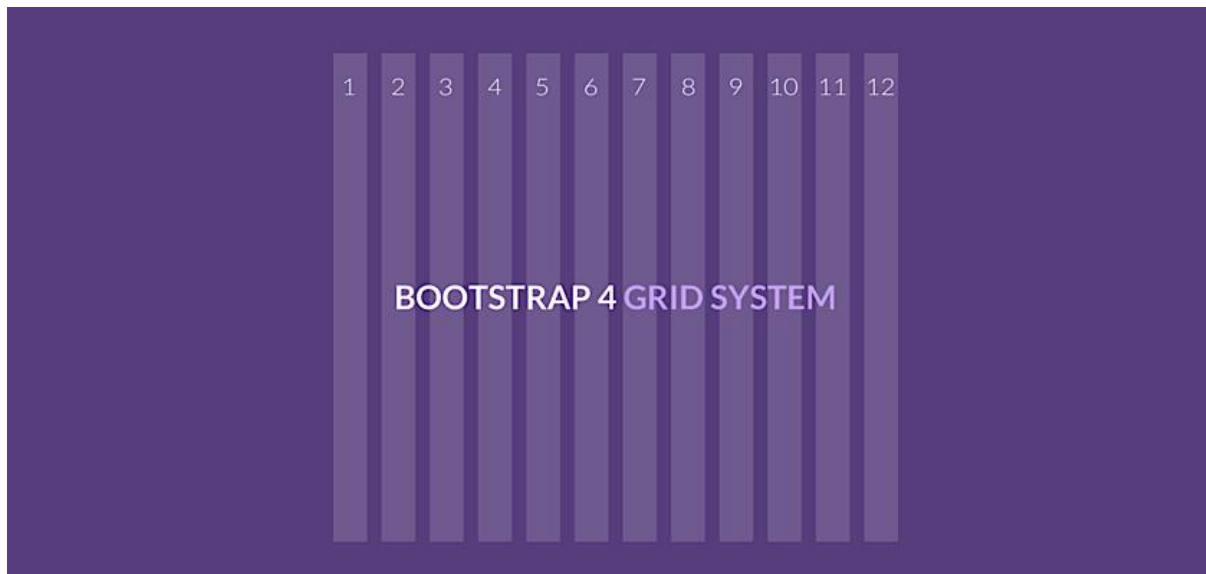
¿Es mejor descargar el código y tenerlo en tu servidor o es mejor usar la CDN?

A grandes rasgos es mejor usar una CDN, ya que su **instalación** es mucho más rápida y al estar alojada en una CDN consumirá mucho menos recursos, ancho de banda y, en definitiva, hará que tu sitio web vaya mucho **más rápido**.

Por el contrario, descargar todos los ficheros en tu servidor y enlazarlo a tu documento HTML nunca está de más para darte **mayor conocimiento** sobre el funcionamiento de un servidor y aprender más de estas magníficas herramientas. Incluso podrías trabajar offline con Bootstrap tras transferirlo a un servidor local.

5. Cómo funciona Bootstrap

5.1 Sistema de rejillas (Grid)



Lo más importante de Bootstrap es su [Sistema de Rejillas o Grid](#), por el que mediante una serie de clases de CSS se posiciona el contenido en base a un **sistema de 12 columnas**, adaptándose a la perfección en diferentes **resoluciones**.

Como podrás comprobar en la documentación, **cada resolución tiene asociada una clase CSS**:

Clase CSS	Dispositivos	Resolución
.col-	extra small devices	screen < 576px
.col-sm-	small devices	screen >= 576px
.col-md-	medium devices	screen >= 768px
.col-lg-	large devices	screen >= 992px
.col-xl-	xlarge devices	screen >=1200px
.col-xxl-	xxlarge devices	screen >=1400px

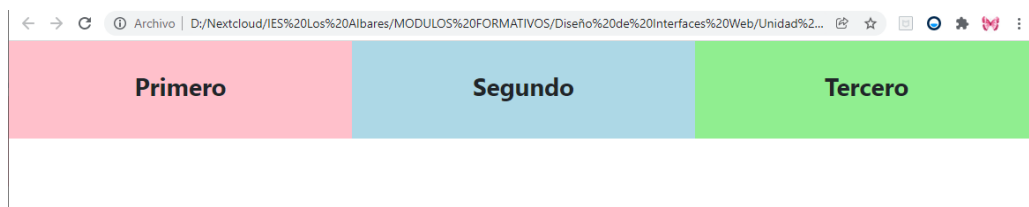
Para determinar el número de columnas que cada elemento tomará, le **añadiremos a las clases** un número que irá del **1 al 12**, siendo 1 el espacio que tomará para una columna y 12 el correspondiente a las 12 columnas, es decir, todo el ancho del grid.

Por ejemplo, en el siguiente caso tenemos un div con 3 clases:

`<div class="col-12 col-sm-8 col-md-6">Contenido del div</div>`

- **.col-12**: indica que para resolución < 576px el div tomará todo el ancho del Grid.
- **.col-sm-8**: para resolución >= 576px, el div ocupará solo 8 columnas.
- **.col-md-6**: a partir de 768px, el div ocupará la mitad del grid, 6 columnas.

Otro ejemplo: **Ejemplo1.Grid.html**



Código html:

```
<div class="row"> /* Indica a Bootstrap que el contenido irá en columnas */
  <div class="primero col-12 col-sm-4 col-md-6 col-lg-4">Primero</div>
  <div class="segundo col-12 col-sm-4 col-md-4 col-lg-4">Segundo</div>
  <div class="tercero col-12 col-sm-4 col-md-2 col-lg-4">Tercero</div>
</div>
```

Estilos:

```
div {font-weight: bold;
      font-size: 1.2em;
      text-align:center;
    }
.row {height:110px;}
.primero {background-color:pink;padding: 30px 0;} /*Padding superior e
inferior de 30px para centrar el texto*/
.segundo {background-color:lightblue;padding: 30px 0;}
.tercero {background-color:lightgreen;padding: 30px 0;}
```

Se puede comprobar que el tamaño de los divs va cambiando según el tamaño de la pantalla. Por ejemplo: para pantallas largas y dispositivos pequeños, todas las columnas tienen el mismo tamaño. Sin embargo, para pantallas medianas (entre 768px y 992px), la primera columna ocupa 6, la segunda 4 y la tercera 2. Y, por último, para dispositivos muy pequeños (móviles), cada columna ocupa todo el ancho del grid, por lo que quedan uno debajo del otro.

5.2 Contenedores

Los contenedores, llamados **container**, son “cajas” donde introducir el contenido de una página web.

Suelen ser elementos `<div>` de html.

Bootstrap requiere un elemento **contenedor** para situar todo el contenido web.

TIPOS DE CONTENEDORES:

Hay dos clases especiales para los contenedores: **.container** y **.container-fluid**.

- **.container:**

Crea un contenedor de anchura fija, pero responsive en función del dispositivo. Se coloca en el **centro de la página web**.

Fixed Container

Use the `.container` class to create a responsive, fixed-width container.

Note that its width (`max-width`) will change on different screen sizes:

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra Large ≥1200px	XXL ≥1400px
max-width	100%	540px	720px	960px	1140px	1320px

Es muy útil para el contenido principal de una web, porque normalmente no queremos que ese contenido ocupe el 100% de la página, ya que las líneas de texto allí incluidas, serían demasiado largas para leerlas cómodamente.

Con este elemento se consigue reducir el ancho del contenido, además de colocarlo en el centro de la página web.

En el ejemplo de abajo (**Ejemplo2.Contenedor.html**) he creado un contenedor con un párrafo simplemente. Le he colocado un color de fondo para que se vea lo que ocupa el contenedor, aunque el contenedor por defecto es transparente.

```
<div class="container">
  <p>
    Este párrafo va dentro de un div container.
  </p>
</div>
```

.container-fluid:

Crea un contenedor que **siempre ocupa todo el ancho de la pantalla**.

En el **Ejemplo2.Contenedor.html** hay un contenedor de clase .container-fluid para poder ver la diferencia con el contenedor normal.

5.3 MARGIN Y PADDING

Los contenedores vienen con un **padding izquierdo y derecho ya establecido**, pero no llevan padding superior e inferior.

Podemos indicarlo cómodamente con clases especiales como, por ejemplo, **.pt-5** que añade un padding top grande.

```
<div class="container pt-5"></div>
```

A esas clases se les llama **utilidades de espacio**, y las más utilizadas son:

- **mt:** margin-top
- **mb:** margin-bottom
- **ms:** margin-left (start)
- **me:** margin-right (end)
- **pt:** padding-top
- **pb:** padding-bottom
- **ps:** padding-left (start)
- **pe:** padding-right (end)

El número indica la magnitud del margen o el padding (del 0 al 5).

Ejemplo: `<div class="mt-0 me-1 mb-2 ms-3 pt-0 pe-1 pb-2 ps-3"></div>`

Si quieres más información sobre las **utilidades de espaciado** visita:

<https://getbootstrap.com/docs/5.0/utilities/spacing/>

Se deja para practicar el **Ejercicio 1**, subido al aula virtual y que consiste en el diseño de un Layout, adaptado a dispositivos pequeños, medianos y largos.

5.4 ANCHURA DE UN ELEMENTO

Tenemos 4 clases para especificar diferentes anchuras de un elemento:

w-25 | w-50 | w-75 | w-100

Hay que tener en cuenta que como indican porcentajes, el elemento padre debe tener especificada una anchura.

Ejemplos:

`<div class="w-25 bg-warning">Width 25%</div>` → width: 25%

`<div class="mw-100 bg-warning">Max Width 100%</div>` → max-width: 100%

`<div class="min-vw-100">Min-width 100vw</div>` → min-width: 100vw

6.1 Configuraciones tipográficas por defecto

https://www.w3schools.com/bootstrap5/bootstrap_typography.php

A partir de aquí seguir w3schools.

6.2 Iconos Bootstrap

<https://icons.getbootstrap.com/>

Lo más cómodo es usar el código `<svg...>` que viene con cada icono, no es necesario descargar la imagen.

<https://primer.style/octicons/>