

# Bootstrap 5

**DigitalHouse** >  
Coding School



**Certified Tech  
Developer**  
The Ultimate Degree

# Índice

1. [¿Qué es Bootstrap?](#)
2. [Implementación](#)
3. [Contenedores y breakpoints](#)
4. [Sistema de grillas](#)
5. [Elementos y componentes](#)

# 1 | ¿Qué es Bootstrap?



**Bootstrap** es un **framework** open source. Trae consigo una combinación de **reglas CSS** y **funcionalidades JavaScript** que facilitan el desarrollo de una estructura web.



# 2 | Implementación

# Vinculación **externa**

Tal como hicimos con fonts e iconos, también existen dos maneras de vincular Bootstrap a nuestros proyectos.

En esta presentación vamos a ver la vinculación externa, pero compartimos el [enlace](#) de referencia que se encuentra en la web oficial, para poder revisar ambas maneras de forma detallada.

Para vincular Bootstrap de forma externa en nuestro proyecto tenemos que ingresar en la [web oficial](#), copiar el link que está debajo del título **CSS** y pegarlo entre las etiquetas de apertura y cierre del `head` de nuestro HTML.

```
html  <head>
      <!-- Aquí va el link a la hoja de estilo CSS -->
      </head>
```

# Vinculación externa

Muchas de las funcionalidades que ofrece Bootstrap dependen directamente de algunas librerías de Javascript.

Para vincularlas en nuestro proyecto tenemos que entrar al mismo link anterior, copiar los links que están debajo del título JS, y pegarlos antes de la etiqueta de cierre del `body` de nuestro documento HTML.

```
html      <!-- Acá van los links a las librerías de JS -->
          </body>
```

Con la vinculación externa estamos consumiendo, vía Internet, todos los recursos que ofrece Bootstrap. Por eso, es necesario tener siempre conexión, de lo contrario, no funcionarán las herramientas que implementemos.

# HTML 5 y viewport

Bootstrap requiere el uso del tipo de documento **HTML5** y se desarrolla **mobile first**, una estrategia en la que optimizamos el código para dispositivos móviles primero y luego ampliamos los componentes según sea necesario a otros dispositivos.

Para asegurar la representación adecuada y una función de zoom táctil para todos los dispositivos, debemos agregar la etiqueta **viewport** al `head` de nuestro HTML.

html

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



**3**

# **Contenedores y breakpoints**

“

Los contenedores y breakpoints representan el **núcleo** de Bootstrap. A través de estos **elementos** crearemos la estructura que **contendrá** todo el flujo de nuestro sitio web.



”

# Qué es un **contenedor**

Un **contenedor** es un elemento HTML cuya función principal es anidar otros elementos y **contenerlos** para que se acomoden dentro del ancho y alto del mismo.

Bootstrap 5 cuenta con tres clases definidas para generar un elemento contenedor: `container` , `container-fluid` y `container-{breakpoint}` .

Podemos asignarle estas clases a cualquier etiqueta HTML, aunque por lo general usaremos elementos de bloque (div, section, article, entre otros). En función del contexto, decidiremos a cual de ellas transformar en un contenedor.

# Contenedor predeterminado: **container**

Una etiqueta establecida como **container** es un contenedor receptivo de ancho fijo, es decir que establece un **max-width** en cada punto de interrupción de respuesta.

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

# Contenedor fluido: **container-fluid**

Una etiqueta establecida como **container-fluid** es un contenedor de ancho completo, que abarque todo el ancho de la ventana gráfica. En este caso **width: 100%** aplica a todos los puntos de interrupción.

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

# Contenedor receptivo: **container-{breakpoint}**

Una etiqueta establecida como **container-{breakpoint}** es un contenedor receptivo que tiene un ancho del 100% hasta que se alcanza el punto de interrupción especificado, después de lo cual aplicamos **max-width** para cada uno de los puntos de interrupción más altos. Por ejemplo, **.container-sm** es 100% de ancho para iniciar hasta que el sm alcanza punto de interrupción, donde se ampliará con md, lg, xl, y xxl.

```
html <div class="container-sm">100% hasta sm</div>
      <div class="container-md">100% hasta md</div>
      <div class="container-lg">100% hasta lg</div>
      <div class="container-xl">100% hasta xl</div>
      <div class="container-xxl">100% hasta xxl </div>
```

# Ahora... ¿Qué es un **breakpoint**?

Un **breakpoint** es un **punto de quiebre** a partir del cual nuestro sitio cambia. Sucede cuando alguna de las propiedades de la pantalla, normalmente el ancho, alcanza un determinado valor.

En ese momento se aplican las reglas asociadas a ese breakpoint, contenidas en los **media-queries**, y el aspecto del sitio cambia.

Bootstrap ya tiene predefinidos su conjunto de breakpoints.



# Breakpoint de Bootstrap

Ancho viewport	Ancho container
< 576px ( <i>extra small</i> )	100%
>= 576px ( <i>small</i> )	540px
>= 768px ( <i>medium</i> )	720px
>= 992px ( <i>large</i> )	960px
>= 1200px ( <i>extra large</i> )	1140px
>= 1400px ( <i>extra extra large</i> )	1320px





# Mobile first / Desktop first



**Mobile first** es una forma de encarar el diseño y el armado del sitio, **declarando** primero las reglas de estilo para los **dispositivos** más **pequeños** hasta llegar a los más grandes. Es lo más usado hoy en día.



**Desktop first** es una forma de encarar el diseño y el armado del sitio, **declarando** primero las reglas de estilo para los **dispositivos** más **grandes** hasta llegar a los más pequeños. Era lo habitual en el pasado, hoy en día ya no se usa mucho.

# 4 | Sistema de grillas

“

Es el **modelo de maquetado** que propone Bootstrap, basado en una estructura de **12 columnas**, sobre la cual diseñaremos y definiremos el flujo de todo nuestro sitio.



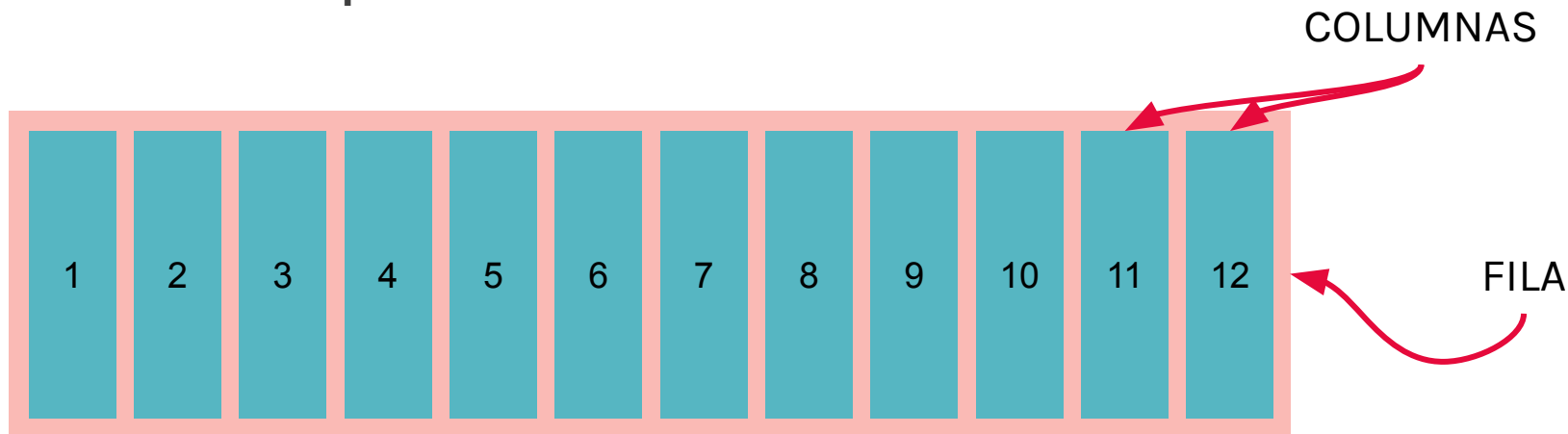
”

# Modelo de **grilla**

Una grilla es una **estructura invisible** que sirve para organizar el contenido, aportando orden y claridad visual.

Para poder implementar este sistema, primero tenemos que crear una fila.

Las **filas** son los **padres contenedores** de las **columnas**.



# Filas y columnas

## Row

Nos indica que creamos una fila de ilimitadas que podemos colocar

HTML

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

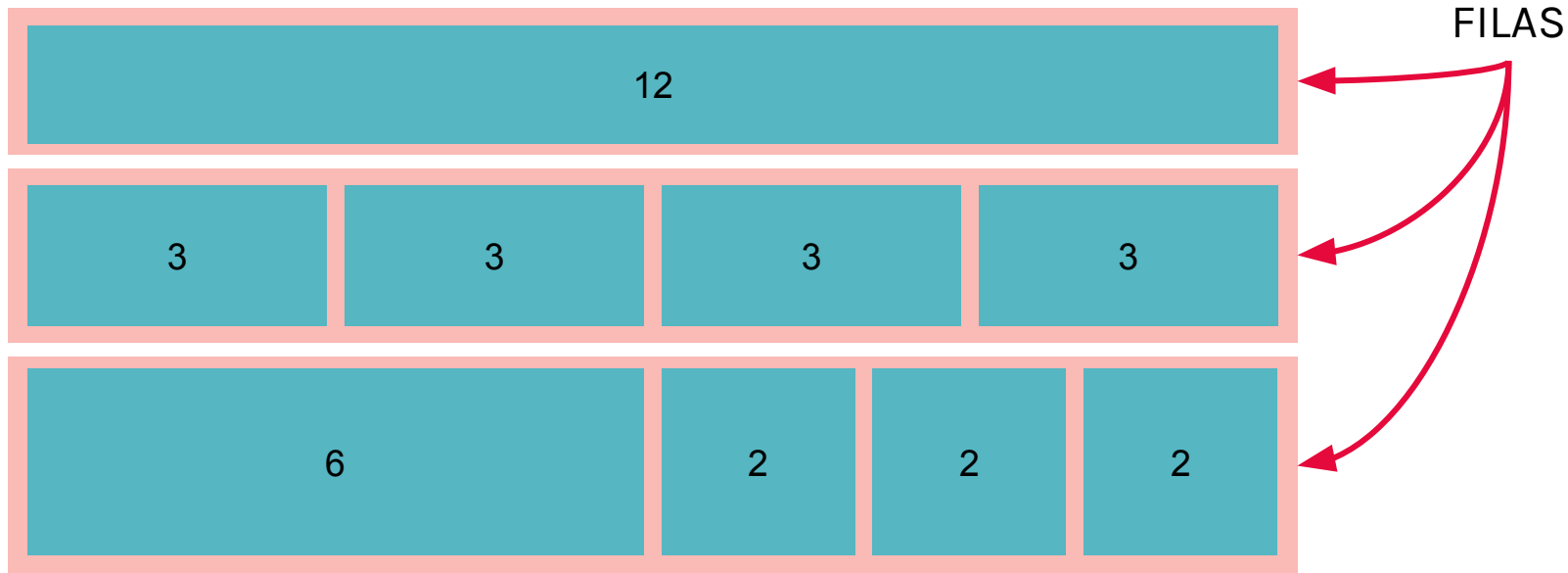
Las filas y las columnas se pueden anidar, creando casi cualquier estructura que necesitemos.

## Col

Nos indica que vamos a ocupar una columna de hasta 12 columnas ("col") dentro de una fila ("row").

# Múltiples **filas**

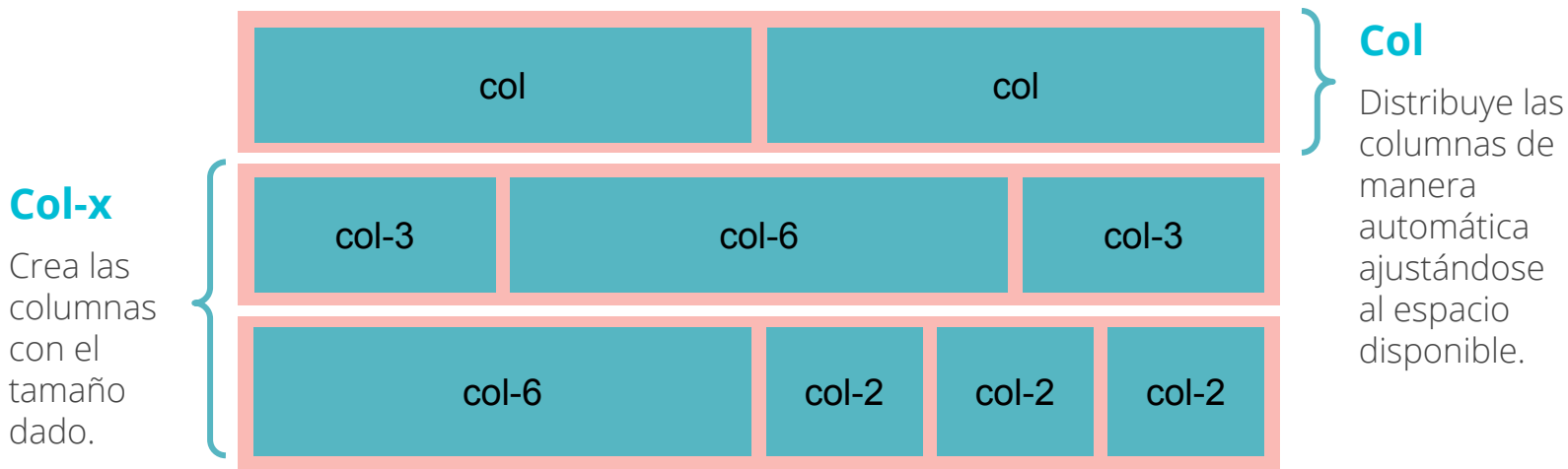
Cada **fila** permite un nuevo conjunto de columnas y eso nos da mucha flexibilidad a la hora de diseñar la estructura del sitio.



# Las columnas

Como **Bootstrap** está basado en **Flexbox** y es totalmente responsive, las columnas se distribuirán automáticamente el ancho disponible.

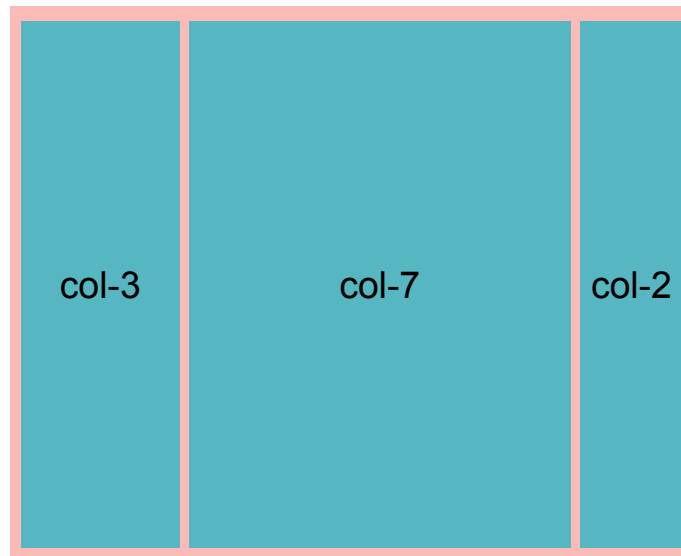
También podemos asignarle un tamaño específico a cada columna utilizando la clase **col-x**, donde x será el número de columnas que queremos ocupar.



# Veamos un ejemplo

H  
T  
M  
L

```
<div class="container">
  <div class="row">
    <div class="col-3">col-3</div>
    <div class="col-7">col-7</div>
    <div class="col-2">col-2</div>
  </div>
</div>
```





# Opciones de cuadrícula

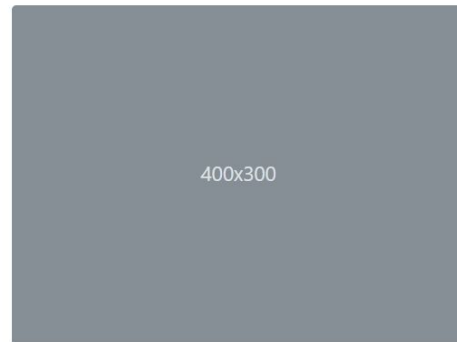
Ancho viewport	Ancho container	Prefijo de clase
< 576px ( <i>extra small</i> )	100%	.col-num
>= 576px ( <i>small</i> )	540px	.col-sm-num
>= 768px ( <i>medium</i> )	720px	.col-md-num
>= 992px ( <i>large</i> )	960px	.col-lg-num
>= 1200px ( <i>extra large</i> )	1140px	.col-xl-num
>= 1400px ( <i>extra extra large</i> )	1320px	.col-xxl-num

# 5 | Elementos y componentes

# Elementos y componentes

Una vez implementado Bootstrap en nuestro proyecto, podemos agregar componentes HTML5 y darle formato con las clases que nos proporciona el framework.

En este ejemplo incluimos una **figura**, que contiene una **imagen** y su **epígrafe**.



Descripción de la imagen.

H  
T  
M  
L

```
<figure class="figure">
  
  <figcaption class="figure-caption">Descripción de la imagen.</figcaption>
</figure>
```

# Elementos y componentes

Bootstrap nos proporciona elementos y componentes, tales como [navbar](#), [carrusel](#) y [modal](#) entre muchos otros. Para conocerlo y poder aprovechar todo su potencial solo tenemos que revisar la documentación oficial.

A pesar de que podemos ponerlos a crear nuestro diseño desde cero, es muy posible que si nuestro sitio cae en alguna de las categorías habituales ya haya algún diseño hecho a nuestra medida.

¡Los **temas** y los **templates** son justamente eso, **estilos para Bootstrap**!

Los **temas** solo modifican los estilos de Bootstrap y los **templates** ya traen páginas armadas para que modifiquemos.

Con toda esta info, ya podemos empezar a probar y ver todas las opciones disponibles.

DigitalHouse>  
Coding School