

# ¿Qué es Bootstrap?

- Es un **framework** de front-end **gratuito**, desarrollado por Mark Otto y Jacob Thornton (Twitter) basado en **html**, **CSS** y **javaScript**, para crear **sitios web responsives**.
- Es **fácil de implementar**, solo se necesita **conocimientos básicos** de html y CSS... **acá festejan!!!** 😊

- Se adapta a **cualquier dispositivo** y es **compatible** con **todos** los **navegadores modernos**.
- Está desarrollado utilizando la técnica **Mobile First**.

## ¿Qué es Bootstrap?

- Es un **framework** de front-end **gratuito**, desarrollado por Mark Otto y Jacob Thornton (Twitter) basado en **html**, **CSS** y **javaScript**, para crear **sitios web responsives**.
- Es fácil de

implementar, solo se necesita **conocimientos básicos** de

html y CSS... **acá festejan!!!** 😊

- Se adapta a **cualquier dispositivo** y es **compatible** con **todos** los **navegadores modernos**.

- Está desarrollado utilizando la técnica **Mobile First**.

## ¿Qué es Bootstrap?

- Utiliza tecnología **SASS** (una especie de CSS programable) por lo que

podemos optar por dos formas de trabajo:

- **Archivos Fuente Programables**. Necesitan ser compilados por un compilador SASS (lo pasa a CSS “entendible” para el browser) ■

- **Archivos Compilados CSS y JS**. Se linkean al html como cualquier otro archivo de este tipo para ser usados directamente.

- Para utilizar los **archivos compilados** tenemos dos opciones: ■ **Descargar los archivos** desde **getbootstrap.com**

- **Linkearlos desde el CDN** (Content Delivery Network) proporcionado por **jsDelivr**.

## ¿Qué elegimos?

- La última versión de Bootstrap es la 5 que es la que utilizaremos.
- **CDN:**
  - Al estar cacheados por otro sitio que el usuario navegó previamente, el mío carga más rápido.
  - Se envían los archivos desde el servidor más cercano al usuario que lo solicita.
  - No agrega peso al sitio.
- **Locales:**
  - No fallan nunca.
  - Óptimo para navegación offline y desarrollos instalables.

# ¿Cómo empezamos?

- Si trabajamos con los **archivos descargados** los linkeamos al html.
- Se puede optar por las **versiones .min** (minificadas: menor peso) o por **las comunes** (podemos chusmear el código y entenderlo).

- **En el HEAD:** linkeamos **bootstrap.css** o

```
bootstrap.min.css <link rel="stylesheet"  
href="bootstrap.css">
```

- **Justo antes del cierre del BODY:**

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

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

```
</body>
```

## ¿Cómo empezamos?

- Si trabajamos con los **archivos descargados** los linkeamos al html.
- Se puede optar por las **versiones .min** (minificadas: menor peso) o por **las comunes** (podemos chusmear el código y entenderlo).

- **En el HEAD:** linkeamos **bootstrap.css** o **bootstrap.min.css**  

```
<link rel="stylesheet" href="bootstrap.css">
```

- **Justo antes del cierre del BODY:**

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

## Empecemos...

- Creamos un archivo html con **DOCTYPE html5**.
- Para asegurarnos la correcta implementación de las media queries debemos incluir la etiqueta **meta viewport**:

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

- Utiliza **reglas de clase** para aplicarse (grilla, utilitarias y



componentes).

- **Requiere de un contenedor** de elementos para acceder a la **grilla fluida**, en principio, hay dos tipos de contenedores generados por las clases: •
  - **.container** → contenedor fluido de un ancho menor a la ventana.
  - **.container-fluid** → contenedor fluido al 100% del ancho de la ventana.
- Se puede incluir un único container por página (no habitual) **o varios contenedores apilados uno debajo del otro** (bloques de elementos).

## Empecemos...

**xs, sm, md, lg, xl, xxl**

**xs** < 576px

- Creamos un archivo html con **DOCTYPE html5**.

- Para asegurarnos la correcta implementación de las media queries

**sm** >= 576px

debemos incluir la etiqueta **meta**

**viewport:** **md** >= 768px

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

**lg** >= 992px

- Utiliza **reglas de clase** para aplicarse (grilla, utilitarias y componentes).

**xl** >= 1200px

- **Requiere de un contenedor** de elementos para acceder a la **grilla fluida**, en

principio, hay dos tipos de contenedores generados por las clases:

xxl >= 1400px

- **.container** → contenedor fluido de un ancho menor a la ventana.
- **.container-fluid** → contenedor fluido al 100% del ancho de la ventana.
- Se puede incluir un único container por página (no habitual) o **varios contenedores apilados uno debajo del otro** (bloques de elementos).

## La clave: GRILLA BOOTSTRAP

- Dentro de cada container vamos a incluir **una o más franjas** o **filas** de contenido, generadas por la clase **row**.

- Cada row se divide en **celdas o columnas** generando una **grilla fluida de 12 columnas** que cubren todo su ancho.
- En principio, siempre debemos **sumar 12 columnas de contenido**.
- Puede ser que **en cada celda** (columna) haya **un elemento** o que éste **ocupe más de una celda** o, incluso, **todas**.
- Pero siempre la **suma horizontal de celdas** que ocupan los elementos **DEBEN sumar 12 columnas** (aunque hay excepciones), y **nunca más de 12 columnas** (la que no entra... se cae de línea).

[illegible]



contenido, generadas		row.4	4
• Cada row se divide en ce		column	generando una grilla fluida 6
de 12 columnas que cubren todo su ancho.12			
1	principio, siempre debemos.11		
• Puede ser	en cada celda (columna) haya un elemento o que éste 10		

		o, incluso, .6
• Pero siempre la 4	horizontal de celdas que ocupan los elementos	
DEBEN sumar 12 columnas 5	7	(aunque hay excepciones), y nunca más
de 12 columnas 3	(la que no entra... se cae de	3

	línea). 6	
--	-----------	--

## La clave: GRILLA BOOTSTRAP

- Las **columnas** se generan con **clases** para los **5 tamaños** de viewport:
  - **.col-\*** → **max-width: 575px** (xs o extra pequeño).
  - **.col-sm-\*** → **min-width: 576px** (sm o pequeño).
  - **.col-md-\*** → **min-width: 768px** (md o medio).
  - **.col-lg-\*** → **min-width: 992px** (lg o grande).
  - **.col-xl-\*** → **min-width: 1200px** (xl o extra grande).
  - **.col-xxl-\*** → **min-width: 1400px** (xxl o extra extra grande).
- Un elemento puede ocupar una **distinta cantidad de columnas** en **cada**



**tamaño** de viewport de acuerdo a las clases que tenga aplicadas `<div class="col-6 col-sm-4 col-md-8 col-lg-2 col-xl-5 col-xxl-8">` • Si se aplica una sola clase, por ejemplo, `.col-md-4` ocupa esa cantidad de columnas también en tamaños mayores y **todas** (las 12) en menores.

## La clave: GRILLA BOOTSTRAP

- Las **columnas** se generan con **clases** para los **5 tamaños** de viewport:
  - `.col-*` → `max-width: 575px` (xs o extra pequeño) `min-width: 1200px` (xl o extra grande).
  - `.col-sm-*` → `min-width: 576px` (sm o pequeño).
  - `.col-md-*` → `min-width: 768px` (md o medio).
  - `.col-lg-*` → `min-width: 992px` (lg o grande).
  - `.col-xl-*` →

- **.col-xxl-\*** **min-width: 1400px** (xxl o extra extra grande). • Un elemento puede ocupar una **distinta cantidad del columnas** en **cada tamaño** de viewport de acuerdo a las clases que tenga aplicadas `<div class="col-6 col-sm-4 col-md-8 col-lg-2 col-xl-5 col-xxl-8">` • Si se aplica una sola clase, por ejemplo, **.col-md-4** **ocupa esa cantidad** de columnas también en tamaños mayores y **todas** (las 12) en menores.