

Bootstrap 4

Fermín Minetto, Dirección de Gestión Digital, Buenos Aires Provincia, 2018

Antes de empezar...

Los objetivos de esta guía son:

- 1. Aprender a instalar Bootstrap 4 en un proyecto*
- 2. Aprender a utilizar el sistema de grids del framework*
- 3. Tener una noción básica de los contenidos CSS de Bootstrap*
- 4. Tener una noción básica de los componentes que ofrece Bootstrap*
- 5. Aprender las opciones de customización*

Si se está leyendo esta guía sin los archivos de ejemplo *es recomendable clonar el proyecto de GitHub [desde esta URL](#)* para tener acceso a los archivos de ejemplo para acompañar visualmente y de forma práctica los conceptos explicados y ejercicios para afianzar los conocimientos aprendidos

1) Fundamentos de Bootstrap

1.1) ¿Qué es Bootstrap?

Bootstrap es un framework frontend de código abierto que provee un conjunto de clases CSS y Javascript que ayudan al momento de maquetar un sitio web; utilizando clases CSS y código Javascript que al usarlas en elementos HTML generan componentes agradables visualmente y responsive. Bootstrap fue creado teniendo como objetivo la construcción de sitios responsive, accesibles y que utilicen de forma semánticamente correcta las etiquetas HTML.

Bootstrap goza de gran popularidad entre los desarrolladores frontend, siendo el más utilizado (entre el 1.5 y 3% de todos los sitios web del mundo).

1.2) ¿Por qué nos interesa usarlo?

Bootstrap es un framework, lo cual implica que nos va a facilitar las cosas para realizar algunas tareas, sin embargo como todo framework esto implica cierto

esfuerzo de nuestra parte por eso hay que conocer las ventajas de usar Bootstrap para determinar si este esfuerzo va a valer la pena a largo plazo. A nosotros nos interesa aprender Bootstrap por las siguientes razones:

1. **Velocidad de desarrollo:** Gracias al sistema de grillas va a ser mucho más rapido maquetar la estructura del sitio; a los componentes de Bootstrap ahorrarnos en andar poniendo muchas reglas a las tablas, botones y otros componentes típicos de una interfaz web para animarlas un poco.
2. **Desarrollo responsive:** Al usar el sistema de grillas y los componentes de Bootstrap nos va a resultar mucho más sencillo (y en algunos casos sin tener que escribir una línea más de código) hacer un sitio responsive puesto que Bootstrap está pensado para eso.
3. **Consistencia:** Al utilizar Bootstrap nuestro sitio se va a ver de manera consistente independientemente en todos los navegadores. Esto no siempre pasa cuando maquetamos el frontend con reglas CSS desde cero.
4. **Customizable:** Una creencia común es que Bootstrap no permite a los usuarios del framework customizarlo. Bootstrap es customizable al 100% para poder adaptarse a cualquier tipo de diseño. Incluso se ofrecen varios tipos de tema para Bootstrap, así se puede iniciar a customizarlo con un diseño de base lo más similar posible a nuestras maquetas; ejemplo: [Bootswatch](#).

2) Utilizando Bootstrap 4

2.1) Instalación

Utilizar Bootstrap es sencillo. Históricamente se requirieron dos archivos: un CSS minificado, un Javascript minificado y jQuery. Desde Bootstrap 4 a estos dos archivos se le sumó un archivo Javascript de una librería JS llamada *popper*. Todos estos archivos menos jQuery y popper vienen incluidos cuando se descarga Bootstrap.

Como muchos framework frontend, existen dos maneras de incluir Bootstrap 4 en nuestro proyecto: la primera es vía CDN que no nos interesa puesto que nos vamos a tener que fiar de un servidor externo, y la segunda es descargándolo, poniéndolo en nuestra carpeta static e incluyendo los archivos

(*bootstrap.min.css* y *bootstrap.min.js*). Luego descargarnos jQuery y popper e incluínos de la misma manera.

Una forma fácil de arrancar (puesto que incluye los tags que no pueden faltar en Bootstrap) es [copiar la plantilla para arrancar que provee Bootstrap](#) y reemplazar los links de CDN por nuestros archivos que descargamos.

2.2) Diseño principal y sistema de grillas

Bootstrap utiliza dos tipos de container para poner el contenido de una pagina: las clases *container*, que poseen un margen amplio entre los bordes izquierdos y derechos de una pantalla, y los *container-fluid*, que representan contenido casi sin margen entre los bordes izquierdos y derechos de una pantalla. Para verlo de manera visual abrir la página de ejemplo "1-containers".

El sistema de grillas provee una estructura a la página para que sea más facil ubicar el contenido en la página y hacer que este contenido se reubique de manera responsive.

Se recomienda ver el código de ejemplo del sistema de grillas en la página de ejemplo "2-grid".

Las reglas del sistema de grillas no son complejas y se basan siguiendo el concepto de *filas y columnas*. Bootstrap nos ofrece una clase llamada *row* que representa una fila de contenido en la página. Dentro de ella podemos ubicar en columnas contenido. Bootstrap divide el ancho de la pantalla en 12 columnas, así que para ocupar el máximo ancho debemos hacer que la suma de estas columnas dentro de una fila sea un total de doce.

Nota: cuando se utiliza el sistema de grillas de Bootstrap se sugiere fuertemente que el elemento hijo de un div de clase *row* sea siempre *de un tipo col*.

Si se desea dejar columnas en blanco (lo que sería "un margen a la izquierda") entonces hay que definir la columna que deseemos que este espaciada y agregarle la clase *offset-[tipo de columna que puede ser: md, xs, etc]-[número de columna]*, ejemplo:

```
<div class="col-md-3 offset-md-9"><h1>Algo</h1></div>
```

Lo que haría esto sería crear contenido en la parte derecha de la pantalla, es decir, como la fila se divide en doce columnas, va a dejar un espacio en blanco

de nueve columnas y dibujar el título "Algo" en el espacio imaginario de las últimas tres columnas del extremo derecho.

Nota: Bootstrap 4 tuvo cambios estructurales en los cuales fue construido mayormente con Flexbox, cambiando su estructura de floats. Esto hace que sea aún más fácil que en las versiones anteriores de Bootstrap utilizar Flexbox. Incluso Bootstrap 4 ofrece [un conjunto de reglas típicas de Flexbox en forma de clases](#).

[+ Más información de diseño principal y sistema de grillas.](#)

2.3) Contenido CSS

Bootstrap incluye clases CSS que sirven de base para construir sitios. Estas clases son un conjunto de reglas que sirven para tener un diseño elegante y consistente "de entrada" en un proyecto. Son geniales cuando no se tiene un diseño específico en mente y sirven de ayuda cuando ya se tiene una maqueta predefinida puesto que nos va a ahorrar algunas reglas o definir clases por nuestra cuenta.

Hay algunas clases de CSS de ejemplo en el archivo de ejemplo *3-css-content*. Sin embargo, es recomendable leer la documentación de Bootstrap para ver que clases nos provee de ayuda.

[+ Más información de contenido CSS.](#)

2.4) Componentes CSS

Bootstrap incluye diversos componentes que nos resultan de bastante utilidad al momento de crear elementos para una interfaz de usuario. La ventaja que tienen estos elementos aparte de ahorrarnos muchas líneas de código y Javascript es que son altamente customizables y responden de manera responsive (siempre y cuando se utilice el sistema de grillas). Los componentes que provee Bootstrap son muchos y se van a mencionar los más importantes.

1. Alerts
2. Badges
3. Buttons
4. Cards
5. Carousel
6. Navbar
7. Modal

8. Tooltip

Es importante y recomendable ver de manera visual estos componentes abriendo la página de ejemplo "4-components" y [leer la documentación](#) para saber como usarlos.

3) Personalizando Bootstrap

3.1) Sobreescribiendo clases CSS

Cuando queremos personalizar Bootstrap para que sea más similar a nuestros diseños podemos sobrecribir las clases CSS de nuestros componentes o añadir unas nuevas (solución mas elegante) que sobreescriban los selectores que modifica Bootstrap en sus clases.

Si se necesitan cambios más generales que afecten a todo nuestro proyecto en lugar de a algunos componentes específicos podemos redefinir las clases de Bootstrap, pero aunque funciona y vamos a notar los cambios, no es la solución más elegante. Existen otros métodos de personalizar la base de Bootstrap para que por defecto se adapte mucho mejor a nuestros proyectos, estos métodos los veremos en las siguientes secciones.

3.2) Recompilando los CSS y JS

Cuando utilizamos Bootstrap en versiones anteriores a la 4 podemos recompilar el código fuente desde la página para generar una versión personalizada para nosotros. Así si solamente nos interesa el sistema de grillas y componentes específicos podemos dejar de lado cosas que sobran y así hacer más liviano el archivo final. Las opciones para personalizar Bootstrap son muchísimas, desde toda la paleta de colores hasta el nivel más alto que son los diversos componentes principales del framework

Esto puede ser útil si ya tenemos un proyecto funcionando y solo queremos importar algunos componentes específicos sin que Bootstrap nos cambie la tipografía nuestra por la que utiliza el framework, usualmente con algunos márgenes añadidos a algunos tags HTML.

Desde [esta página](#) se puede personalizar las opciones que queremos generar para nuestra distribución personalizada de Bootstrap.

3.3) Temas de Bootstrap

A pesar de que el tema por defecto de Bootstrap (colores semánticos para signos de éxito, signos de "peligro", colores principales y secundarios, animaciones, etc) usa colores neutros y visualmente agradables, es posible que nos guste más alguna otra paleta de colores. Si tampoco queremos gastarnos en redefinir todas estas variables Sass en la página de Bootstrap [existen páginas que ofrecen temas](#) con distintas paletas de colores y animaciones para que nosotros usemos al comenzar un proyecto.

Utilizar estos temas es sencillo, solo hay que bajarse el bootstrap.min.css del tema y reemplazarlo por el archivo con el mismo nombre que se encuentra en nuestra instalación de Bootstrap actual