



Tutorial Bootstrap en español. Guía completa 🏆

📅 Jan 20, 2017 | Última revisión en: **Nov de 2019** 🔥

~/Artículos/ 📁 ~/Alquimias/ 🧪 ~/Sortilegios/ ✨ ~/About/ 🧙

¿Qué es bootstrap?

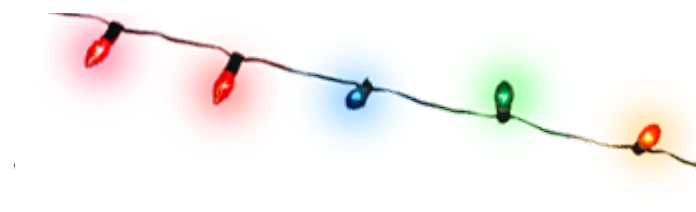
Bienvenido a la guía en español y con ejemplos de Bootstrap. Antes de empezar al lío voy a explicar qué es Bootstrap por si es la primera vez que utilizar esta librería.

Bootstrap es una librería de estilos CSS, es decir, es un archivo CSS que añades en tus 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.



Bootstrap



¿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.

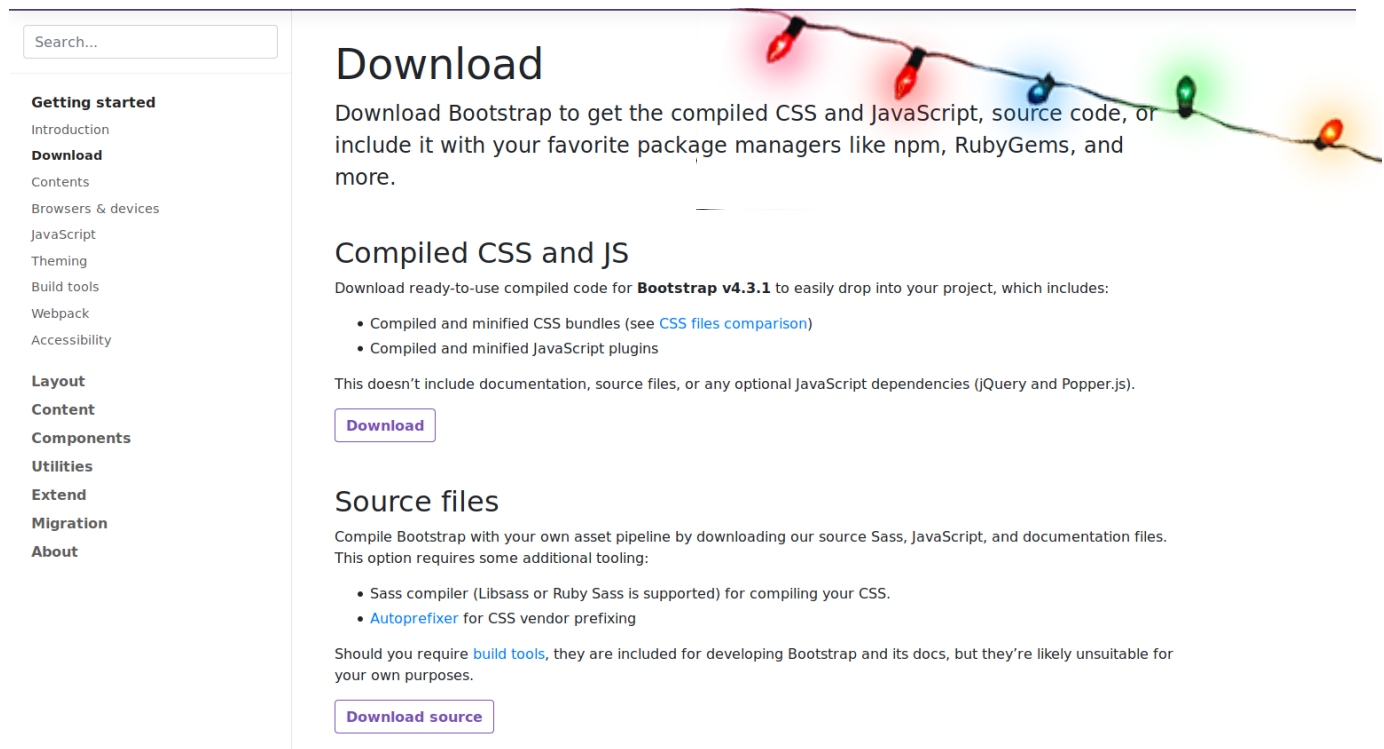
Descarga y instalación de Bootstrap

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

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/4.0/getting-started/download/>

A screenshot of the Bootstrap website's download page. The page has a dark header with a search bar and a navigation menu on the left. The main content area is white with a string of colorful Christmas lights at the top. The 'Download' section is prominent, followed by 'Compiled CSS and JS' and 'Source files' sections, each with a 'Download' button.

Search...

Getting started
Introduction
Download
Contents
Browsers & devices
JavaScript
Theming
Build tools
Webpack
Accessibility

Layout
Content
Components
Utilities
Extend
Migration
About

Download

Download Bootstrap to get the compiled CSS and JavaScript, source code, or include it with your favorite package managers like npm, RubyGems, and more.

Compiled CSS and JS

Download ready-to-use compiled code for **Bootstrap v4.3.1** to easily drop into your project, which includes:

- Compiled and minified CSS bundles (see [CSS files comparison](#))
- Compiled and minified JavaScript plugins

This doesn't include documentation, source files, or any optional JavaScript dependencies (jQuery and Popper.js).

[Download](#)

Source files

Compile Bootstrap with your own asset pipeline by downloading our source Sass, JavaScript, and documentation files. This option requires some additional tooling:

- Sass compiler (Libsass or Ruby Sass is supported) for compiling your CSS.
- [Autoprefixer](#) for CSS vendor prefixing

Should you require [build tools](#), they are included for developing Bootstrap and its docs, but they're likely unsuitable for your own purposes.

[Download source](#)

Además, para que funcione correctamente tienes 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 lo descargas.

Una vez descargados todos los archivos, los tienes que meter dentro de una carpeta en tu proyecto. **IMPORTANTE:** Si usas un framwework como React, Vue o Angular no te recomiendo que lo hagas así.

Dentro de la carpeta que te has 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** tienes que coger 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** tienes que coger el archivo **bootstrap.min.js**.

Estos dos archivos junto con el archivo **js de jquery** y el archivo **js de popper.js** tienes que meterlos en tu proyecto. Lo recomendable es meterlos separados para identificarlos mejor, es decir, los archivos de estilos **.css** en una carpeta y los **.js** en otra carpeta, eso ya como quieras.

Hecho esto ya puedes importar estos archivos dentro de la sección **head** de los 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>
```

Cuando abras la página, en la sección **Network** de las herramientas para los programadores del navegador (para abrirlas puedes pulsar **ALT + SHIFT + I** o puedes hacer **botón derecho > inspeccionar elemento** dentro del navegador) tienes que revisar que los archivos que has importado (recarga la página si no te sale nada) los encuentra bien.

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.

```
<head>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css,

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

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8,

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"

</head>
```

Tutorial bootstrap. Ejemplos y elementos

A continuación te voy a enseñar los elementos más interesantes y que más va a usar de Bootstrap. Aún así Bootstrap tiene más elementos de los que te voy a enseñar ya que no me daría un solo artículo para explicarlos todos.

Si quieres personalizar alguno de estos elementos para tu web, lo que yo suelo hacer es añadir, detrás de las clases de Bootstrap, mis propias clases para poder sobrescribir los

estilos de Bootstrap que no me gusten.



Sistema de grid y contenedores

Estos elementos de Bootstrap son los que más me gustan y los que más uso. Muchas veces acababa usando de Bootstrap solo estos elementos por la enorme utilidad que tienen.

Containers

Lo primero que vamos a ver son los contenedores. Los contenedores como su nombre indica, sirven para crear una "caja" o "contenedor" sobre 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 que ocurre es que normalmente en las páginas webs no quieres que el contenido principal de la página web ocupe el 100% de la página ya que es muy incómodo para el usuario a la hora de leer que las líneas sean tan largas.

Con este elemento consigues reducir el ancho del contenido además de colocarlo 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 vea correctamente,

En el ejemplo de abajo 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 container
  </p>
</div>
```

Este párrafo va dentro de un container

[Ver código de ejemplo en jsfiddle](#)



Cómo crear columnas en Bootstrap

Si has trabajado con HTML y CSS con anterioridad entenderás el sufrimiento de tener que crear columnas y que encima se adaptaran de forma responsive. Aunque en los últimos tiempos tenemos **flexbox** y **css grid** de forma oficial en CSS sin tener que descargar nada, lo cierto es que crear columnas con Bootstrap es todavía más sencillo.

Para crear columnas lo primero que tienes que hacer es crear un elemento HTML padre con la etiqueta **row**. Esta etiqueta le dice a Bootstrap que el contenido van a ser columnas.

Otra cosa que tienes que tener en cuenta es que Bootstrap se basa en un sistema de 12 columnas, es decir, teniendo 12 columnas en total, si quieres crear dos columnas de igual tamaño, tienes que crear 2 columnas de 6. Si quieres 3 columnas tienes que crear 3 columnas de 4 porque $3 * 4 = 12$ columnas en total.

Veamos un ejemplo creado dos columnas de igual tamaño:

```
<div class="container">
  <div class="row">
    <div class="col col-6">
      Primera columna
    </div>
    <div class="col col-6">
      Segunda columna
    </div>
  </div>
</div>
```

Primera columna

Segunda columna


[Ver código de ejemplo en jsfiddle](#)

Como antes, he puesto dos colores para diferenciar, pero las columnas por defecto no tienen background.

Si te fijas, para crear columnas tienes que usar la clase **col** seguido de otra clase con el ancho que quieras del 1 al 12, por ejemplo: col-6, col-3, col-2, etc.

Otro ejemplo pero ahora con una columna que ocupe más que las otras:

```
<div class="container">
  <div class="row">
    <div class="col col-2">
      Primera columna más pequeña
    </div>
    <div class="col col-6">
      Segunda columna mucho más grande que las otras
    </div>
    <div class="col col-2">
      Tercera columna más pequeña
    </div>
  </div>
</div>
```



Por cierto, si lo que quieres es que todas las columnas de un row tengan el mismo ancho, no hace falta que especifiques un tamaño, te vale con poner simplemente la clase **col** a cada columna.

```
<div class="container">
  <div class="row">
    <div class="col">
      1 of 2
    </div>
    <div class="col">
      1 of 2
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col">
      1 of 3
    </div>
    <div class="col">
      1 of 3
    </div>
  </div>
</div>
```

Las columnas que hemos creado hasta ahora, tienen el ancho que especifiquemos en la clase y siempre será el mismo para todos los dispositivos. Si quieres cambiar el ancho de una columna para cierto dispositivo tienes que usar estas clases:

- **xs** ancho de pantalla de más de 576px
- **sm** ancho de pantalla de más de 768px
- **md** ancho de pantalla de más de 992px
- **lg** ancho de pantalla de más de 1200px

Como los estilos se van a aplicar desde la medida en píxeles hacia arriba, tienes que tener

en cuenta que si aplicas un ancho a una columna para los dispositivos **sm**, también se van aplicar en los **md** y en los **lg**



Por ejemplo, imagina que quieres crear una fila que tenga tres columnas y quieres que en ordenador una de ellas ocupe más, en tablets ocupen lo mismo y en móvil se pongan una debajo de la otra ocupando toda la fila:

```
<div class="container">
  <div class="row">
    <div class="col col-md-2 col-sm-3 col-xs-12">
      Primera columna más pequeña
    </div>
    <div class="col col-md-6 col-sm-3 col-xs-12">
      Segunda columna mucho más grande que las otras
    </div>
    <div class="col col-md-2 col-sm-3 col-xs-12">
      Tercera columna más pequeña
    </div>
  </div>
</div>
```

Primera
columna más
pequeña

Segunda columna mucho más grande que las otras

Tercera
columna más
pequeña

Si las columnas dentro de un row suman más de 12 se apilarán unas encima de otras, es decir, si dentro de un row creas 4 columnas de 6 lo que ocurrirá es que se crearan 2 filas con 2 columnas cada una ocupando la mitad.

Elementos funcionales

Los elementos que voy a explicar ahora son elementos que Bootstrap ha decidido darle estilos por defecto para que se vean mejor, es decir, los contenedores y márgenes eran elementos layout que no añadían look and feel y los que hay a continuación tienen ya apariencia visual, colores, bordes, etc.

Botones de Bootstrap

Para crear un botón tienes que usar la clase **btn**. Dentro de los botones bootstrap ha creado una serie de tipos:

```
<!-- Botón standart -->
<button type="button" class="btn btn-default">Default</button>
```



```
<!-- Botón primario -->
<button type="button" class="btn btn-primary">Primary</button>

<!-- Botón de éxito -->
<button type="button" class="btn btn-success">Success</button>

<!-- Botón de información -->
<button type="button" class="btn btn-info">Info</button>

<!-- Botón de aviso -->
<button type="button" class="btn btn-warning">Warning</button>

<!-- Botón de peligro -->
<button type="button" class="btn btn-danger">Danger</button>

<!-- Botón con énfasis en un link -->
<button type="button" class="btn btn-link">Link</button>
```



Default

Primary

Success

Info

Warning

Danger

Link

[Ver código de ejemplo en jsfiddle](#)

Para aumentar o disminuir el tamaño del botón tenemos que añadir la clase **btn-lg**, **btn-sm**, o **btn-xs** dependiendo del tamaño que queramos.

Alertas

Las alertas sirven para mostrar un mensaje al usuario. Como pasa con los botones, las alertas tienen varios tipos:

```
<div class="alert alert-primary" role="alert">
  Contenido de la alerta
</div>
<div class="alert alert-secondary" role="alert">
  Contenido de la alerta
</div>
<div class="alert alert-success" role="alert">
  Contenido de la alerta
</div>
<div class="alert alert-danger" role="alert">
  Contenido de la alerta
</div>
<div class="alert alert-warning" role="alert">
  Contenido de la alerta
</div>
<div class="alert alert-info" role="alert">
  Contenido de la alerta
</div>
```

```
<div class="alert alert-light" role="alert">
  Contenido de la alerta
</div>
<div class="alert alert-dark" role="alert">
  Contenido de la alerta
</div>
```



Contenido de la alerta

Contenido de la alerta

Contenido de la alerta

Contenido de la alerta

Contenido de la alerta

Contenido de la alerta

Contenido de la alerta

Contenido de la alerta

[Ver código de ejemplo en jsfiddle](#)

Breadcrumbs

Los breadcrumbs o migas de pan, sirven para mostrar la situación del usuario dentro de una página. Indica al usuario dónde está y de dónde viene:

```
<nav aria-label="breadcrumb">
  <ol class="breadcrumb">
    <li class="breadcrumb-item"><a href="#">Home</a></li>
    <li class="breadcrumb-item"><a href="#">Sección</a></li>
    <li class="breadcrumb-item active" aria-current="page">Sección actual</li>
  </ol>
</nav>
```

[Home](#) / [Sección](#) / Sección actual

[Ver código de ejemplo en jsfiddle](#)



Dentro de cada elemento de la lista, puedes usar la clase **active** para indicar a Bootstrap la sección actual sobre la que está el usuario. Además, opcionalmente, puedes usar el enlace **href** para poner un link a la sección.

Cards

Las cards o tarjetas, sirven para agrupar el contenido. Se suelen utilizar para crear listas de elementos, por ejemplo, artículos de blog, tweets, colecciones de elementos, etc.

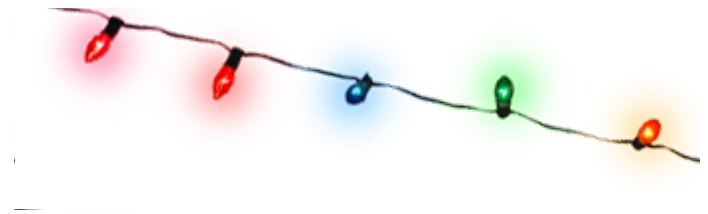
Esta sección en Bootstrap daría para un artículo entero, por lo que voy a ponerte un par de ejemplos, si quieres ver todas las posibilidades te dejo el enlace a la sección en la documentación de bootstrap:

<https://getbootstrap.com/docs/4.3/components/card/>

```
<div class="card" style="width: 18rem;">
  <div class="card-body">
    <h5 class="card-title">Título de la tarjeta</h5>
    <h6 class="card-subtitle mb-2 text-muted">Subtítulo</h6>
    <p class="card-text">Contenido de la tarjeta</p>
    <a href="#" class="card-link">Enlace 1</a>
    <a href="#" class="card-link">Enlace 2</a>
  </div>
</div>

<div class="card border-primary mb-3" style="max-width: 18rem;">
  <div class="card-header">Header</div>
  <div class="card-body text-primary">
    <h5 class="card-title">Tarjeta con borde</h5>
    <p class="card-text">Contenido de la tarjeta</p>
  </div>
</div>

<div class="card text-right" style="width: 18rem;">
  <div class="card-body">
    <h5 class="card-title">Tarjeta con alineación a la derecha</h5>
    <p class="card-text">Contenido alineado a la derecha</p>
    <a href="#" class="btn btn-primary">Botón dentro de la tarjeta</a>
  </div>
</div>
```



Título de la tarjeta

Subtítulo

Contenido de la tarjeta

[Enlace 1](#) [Enlace 2](#)

Header

Tarjeta con borde

Sontenido de la tarjeta

Tarjeta con alineación a la derecha

Contenido alineado a la derecha

Botón dentro de la tarjeta

[Ver código de ejemplo en jsfiddle](#)

Collapse

Este elemento sirve para poder añadir un botón cuya funcionalidad sea poder ocultar y mostrar cierto contenido, es decir, crear elementos colapsables.

```
<button class="btn btn-primary" type="button" data-toggle="collapse" data-target="#co
  Haz clic para mostrar y ocultar el contenido
</button>
<div class="collapse" id="collapseExample">
  <div class="card card-body">
    Dentro de esta etiqueta puedes añadir todo el contenido que quieres que se muestre
  </div>
</div>
```

[Ver código de ejemplo en jsfiddle](#)

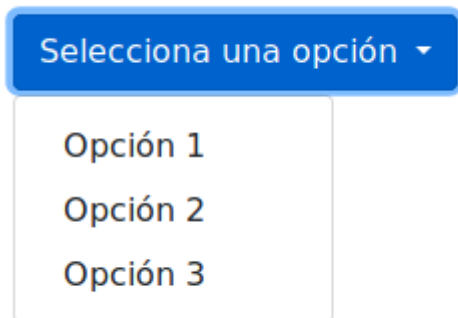
Dropdowns



Los dropdowns o multiselect, sirven para que el usuario pueda escoger una opción para un conjunto de posibilidades:

```
<div class="dropdown">
  <button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton">
    Selecciona una opción
  </button>
  <div class="dropdown-menu" aria-labelledby="dropdownMenuButton">
    <a class="dropdown-item" href="#">Opción 1</a>
    <a class="dropdown-item" href="#">Opción 2</a>
    <a class="dropdown-item" href="#">Opción 3</a>
  </div>
</div>
```

Lo bueno es que puedes usar cualquier botón de los que hemos visto antes.



[Ver código de ejemplo en jsfiddle](#)

Modal

Los popups son ventanas emergentes que se abren cuando el usuario interacciona con cierto elemento de una página como un botón o un enlace.

Para que el modal pueda funcionar, Bootstrap lo que hace es usar la etiqueta de **data-toggle** (en este caso modal) y de la etiqueta de **data-target** (en el ejemplo es el id del modal que se crea abajo). Para cerrar el modal se usa la etiqueta html de **data-dismiss="modal"**. Dentro del modal puedes colocar el contenido que quieras.

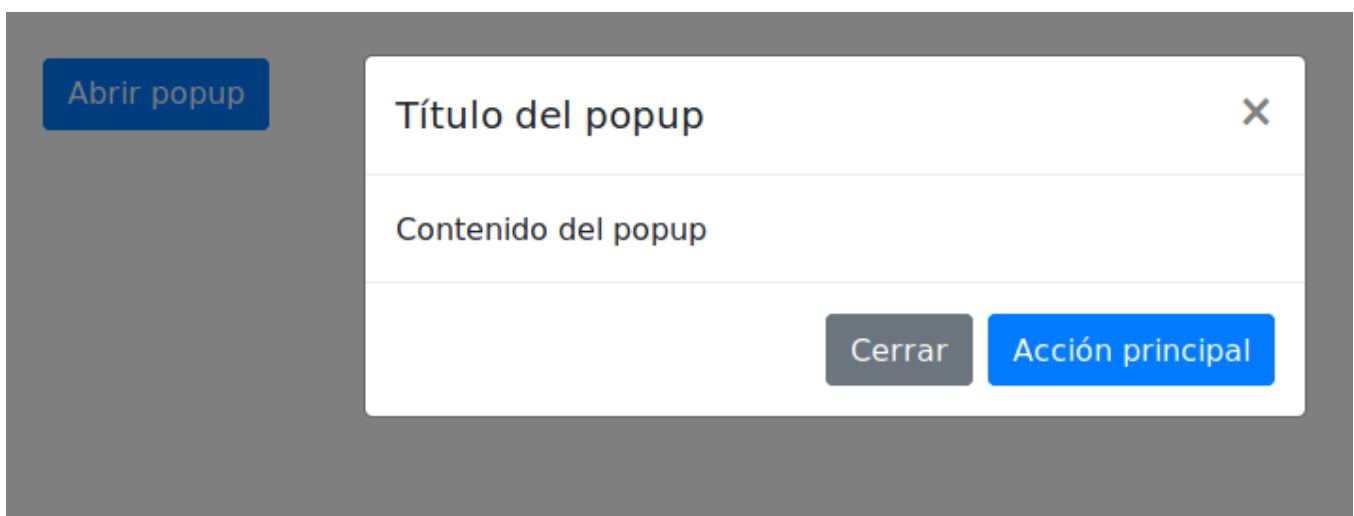
```
<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal">
  Abrir popup
</button>

<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel">
  <div class="modal-dialog" role="document">
```

```

<div class="modal-content">
  <div class="modal-header">
    <h5 class="modal-title" id="exampleModalLabel">Título del popup</h5>
    <button type="button" class="close" data-dismiss="modal" aria-label="Close">
      <span aria-hidden="true">&times;</span>
    </button>
  </div>
  <div class="modal-body">
    Contenido del popup
  </div>
  <div class="modal-footer">
    <button type="button" class="btn btn-secondary" data-dismiss="modal">Cerrar</button>
    <button type="button" class="btn btn-primary">Acción principal</button>
  </div>
</div>
</div>
</div>

```



[Ver código de ejemplo en jsfiddle](#)

Navbar de Bootstrap

Como pasa con los **cards** de Bootstrap, el navbar también tiene muchas opciones. Lo que voy a poner a continuación es el ejemplo más típico con todas las cosas. Si quieres encontrar más información mira este enlace:


<https://getbootstrap.com/docs/4.3/components/navbar/>

Lo bueno del navbar es que ya viene preparado con el típico icono de **burger** (tres líneas horizontales) que aparece en la versión móvil sin que tengas que hacer nada.

```

<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#na
    <span class="navbar-toggler-icon"></span>
  </button>

```



```

<div class="collapse navbar-collapse" id="navbarSupportedContent">
  <ul class="navbar-nav mr-auto">
    <li class="nav-item active">
      <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link</a>
    </li>
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button"
        Dropdown
      </a>
      <div class="dropdown-menu" aria-labelledby="navbarDropdown">
        <a class="dropdown-item" href="#">Action</a>
        <a class="dropdown-item" href="#">Another action</a>
        <div class="dropdown-divider"></div>
        <a class="dropdown-item" href="#">Something else here</a>
      </div>
    </li>
    <li class="nav-item">
      <a class="nav-link disabled" href="#" tabindex="-1" aria-disabled="true">Disal
    </li>
  </ul>
  <form class="form-inline my-2 my-lg-0">
    <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="
    <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
  </form>
</div>
</nav>

```

Navbar Home Link Dropdown ▾ Disabled

[Ver código de ejemplo en jsfiddle](#)

Otros elementos que no hemos visto

- **Badge:** Pequeñas etiquetas para meter texto o números
- **Button group:** Agrupación de botones
- **Carousel:** Agrupación de botones
- **Formularios:** Una de las secciones más extensas de Bootstrap. Tiene todo tipo de elementos para crear formularios. **Muy recomendable.**
- **Jumbotron:** Header con información
- **List group:** Agrupación de elementos en forma de lista
- **Media object:** Componente para la creación de elementos repetitivos como tweets,

artículos de blog, etc

- **Modal**: Popup
- **Navs**: Lista de enlaces para la navegación de una web
- **Pagination**: Para maquetar la paginación de una web
- **Popovers**: Pequeños popups que aparecen al lado del contenido
- **Progress**: Barra de progreso
- **Scrollspy**: Para maquetar áreas scrollables
- **Spinners**: Animaciones de carga
- **Toasts**: Para maquetar notificaciones
- **Tooltips**: Ayudas dentro del contenido



Otras utilidades

Además de los elementos funcionales y contenedores que hemos visto antes, Bootstrap también ofrece una serie de clases de apoyo que puedes utilizar donde quieras.

Colores que trae Bootstrap

Bootstrap tiene una serie de clases preparadas para poder poner colores directamente a los textos y colores de fondo. La única pega es que los colores son los que ya vienen con Bootstrap y si quieres colores nuevos te los tienes que hacer tu a mano:

Colores para textos:

```
<p class="text-primary">.text-primary</p>
<p class="text-secondary">.text-secondary</p>
<p class="text-success">.text-success</p>
<p class="text-danger">.text-danger</p>
<p class="text-warning">.text-warning</p>
<p class="text-info">.text-info</p>
<p class="text-light bg-dark">.text-light</p>
<p class="text-dark">.text-dark</p>
<p class="text-body">.text-body</p>
<p class="text-muted">.text-muted</p>
<p class="text-white bg-dark">.text-white</p>
<p class="text-black-50">.text-black-50</p>
<p class="text-white-50 bg-dark">.text-white-50</p>
```


`.text-primary`

`.text-secondary`

`.text-success`

`.text-danger`

`.text-warning`

`.text-info`

`.text-light`

`.text-dark`

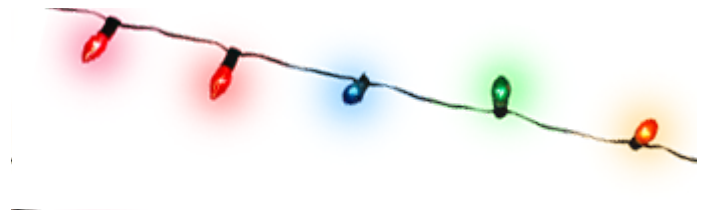
`.text-body`

`.text-muted`

`.text-white`

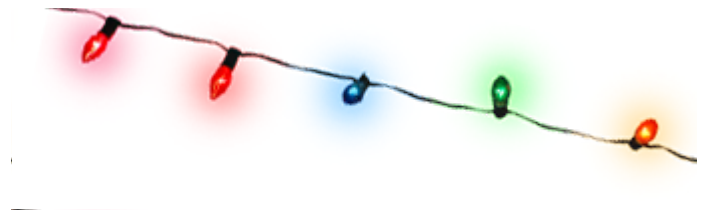
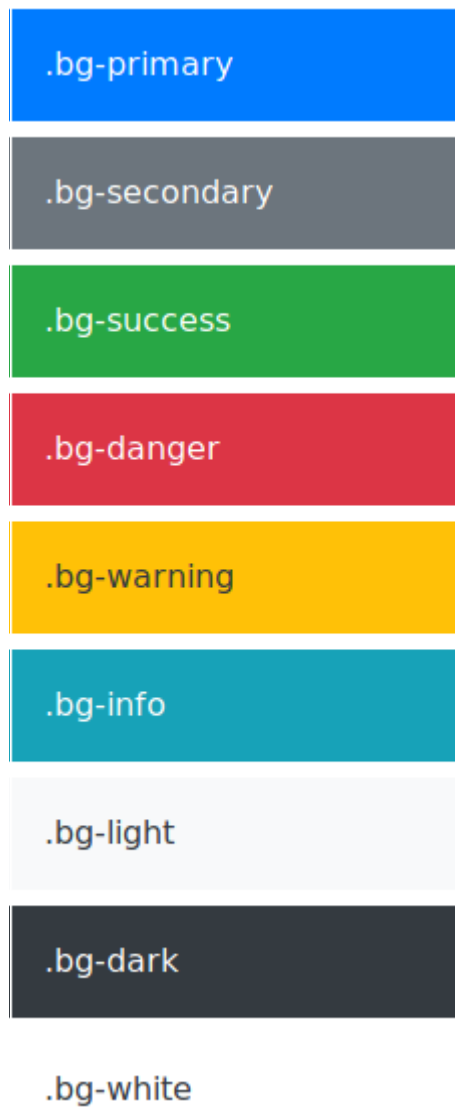
`.text-black-50`

`.text-white-50`



Colores para fondos:

```
<div class="p-3 mb-2 bg-primary text-white">.bg-primary</div>
<div class="p-3 mb-2 bg-secondary text-white">.bg-secondary</div>
<div class="p-3 mb-2 bg-success text-white">.bg-success</div>
<div class="p-3 mb-2 bg-danger text-white">.bg-danger</div>
<div class="p-3 mb-2 bg-warning text-dark">.bg-warning</div>
<div class="p-3 mb-2 bg-info text-white">.bg-info</div>
<div class="p-3 mb-2 bg-light text-dark">.bg-light</div>
<div class="p-3 mb-2 bg-dark text-white">.bg-dark</div>
<div class="p-3 mb-2 bg-white text-dark">.bg-white</div>
<div class="p-3 mb-2 bg-transparent text-dark">.bg-transparent</div>
```



Sombras

También viene con sombras ya hechas para ser aplicadas rápidamente sin esfuerzo:

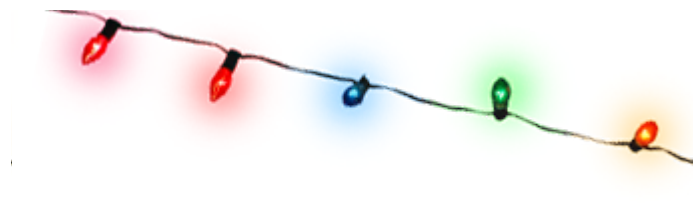
```
<div class="shadow-none p-2 mb-5 bg-light rounded">Sin sombra</div>  
<div class="shadow-sm p-2 mb-5 bg-white rounded">Sombra pequeña</div>  
<div class="shadow p-2 mb-5 bg-white rounded">Sombra normal</div>  
<div class="shadow-lg p-2 mb-5 bg-white rounded">Sombra grande</div>
```

Sin sombra

Sombra pequeña

Sombra normal

Sombra grande



Tamaños

Muy útil cuando necesitas poner un tamaño en porcentaje y no quieres crear una clase solo para eso:

```
<div class="w-25 p-3" style="background-color: #eee;">Width 25%</div>
<div class="w-50 p-3" style="background-color: #eee;">Width 50%</div>
<div class="w-75 p-3" style="background-color: #eee;">Width 75%</div>
<div class="w-100 p-3" style="background-color: #eee;">Width 100%</div>
<div class="w-auto p-3" style="background-color: #eee;">Width auto</div>
```

Width 25%

Width 50%

Width 75%

Width 100%

Width auto

Mostrar y ocultar elementos

Para mostrar y ocultar elementos de forma sencilla puedes utilizar estas dos clases:

```
<div class="visible">Contenido visible</div>
```

```
<div class="invisible">Contenido invisible</div>
```



Otros elementos que no hemos visto

Como en el apartado de los elementos visuales, me dejo una serie de elementos que también vienen con Bootstrap pero que no los he incluido porque personalmente no los uso mucho, aunque a ti te pueden venir bien:

- **Bordes**: Para aplicar bordes a elementos
- **Clearfix**: Limpiar los floats
- **Icono de cerrar**: Icono con la típica "X" para cerrar cosas
- **Display**: Para poder añadir propiedades de **display** de CSS poniendo solo su etiqueta html
- **Embed**: Para incrustar elementos como vídeos o webs dentro de una página
- **Flex**: Propiedades flex de CSS
- **Float**: Propiedades float de CSS
- **Image replacement**: Reemplazar contenido por una imagen de fondo
- **Overflow**: Propiedades overflow de CSS
- **Position**: Propiedades position de CSS
- **Screen readers**: Ocultar contenido a los lectores de pantalla
- **Spacing**: Dar espacios y padding entre elementos
- **Stretched link**: links que ocupen todo el contenido
- **Text**: Clases de ayuda para texto
- **Vertical align**: Alineación vertical de elementos

Conclusiones

Mi opinión sobre este tipo de librerías CSS es que vienen bien para aprender o prototipar, pero para construir webs profesionales pienso que lo mejor es contruir todo el CSS desde 0. Al final lo que haces es añadir demasiadas clases y estilos que solo están como etiquetas

HTML y cuando te toque mantener no te vas a acordar.

Como he dicho de este tipo de frameworks lo que he acabado usando siempre es el sistema de grid y columnas y poco más, el resto de elementos siempre he tenido que personalizar a mano y para eso no los uso.

Aún así si te ha gustado y quieres seguir aprendiendo acerca de este framework te invito a que busques más información en su página <http://getbootstrap.com/>

[Artículos](#)[Alquimias](#)[Sortilegios](#)[About](#)[Twitter](#)[Facebook](#)[Email](#)[R
F](#)

Alquimias

 Newsletter semanal y gratuita con recursos para desarrolladores web

[Más información](#)

¿CUÁNTO COBRA UN PROGRAMADOR EN ESPAÑA EN 2021?

He hecho una encuesta y he extraído estos resultados para saber cuál es la media de salario en el sector de la programación en España en el año 2021

[► Leer más](#)

CÓMO CREAR UN RELOJ ANALÓGICO CON JAVASCRIPT - SORTILEGIOS 02

Voy a crear un reloj analógico, pero el lugar de crearlo con HTML lo voy a hacer con SVG para explicar alguna pincelada. Además voy a añadir movimiento con Javascript a las manecillas para que se muevan con la hora actual.

[► Leer más](#)