

Bootstrap

AUTOR: ABREU, Pablo

Índice

Introducción	3
Objetivos	3
Historia	4
Versiones	4
Características	5
Instalación	5
1. CSS y Javascript compilados y/o minimizados	6
2. CDN a través de jsDelivr	9
3. Administradores de paquetes	9
4. Archivos	10
¿Qué nos provee Bootstrap?	11
Diseño	11
Contenedores	11
Sistema de Grillas	12
Columnas de Ancho Automático	13
Columnas de Ancho Fijo	14
Puntos de interrupción	15
Columnas Responsive	16
Columnas de Ancho Mixto	16
Contenido	17
Imágenes	17
Tablas	18
Figuras	21
Formularios	22
Introducción	22
Controles	22
Layout	23
Validación de Formularios	23
Componentes	24
Alertas	24
Botones	26
Barra de navegación	26
Ventanas Emergentes (Modal)	28

Utilidades (utilities)	28
Colores en Bootstrap	28
Colorear Texto	29
Colorear fondo	30
Display	32
Ocultar elementos	33
Referencias	34

Introducción

El siguiente curso que estás por ver, trata sobre un framework de interfaz de usuario o UI según sus siglas en inglés. Si ya has programado anteriormente sitios web seguramente lo conoces. Bootstrap se ha convertido en uno de los frameworks que utilizan HTML, CSS y JS más popular y a su vez más utilizado por los desarrolladores. Su potente sistema de grilla, su extensa y clara documentación, su gran capacidad de adaptación a los distintos tamaños de dispositivos, su variada colección de componentes y la extensa cantidad de templates desarrollados con bootstrap son algunas de las razones, por lo cual hoy, es tan elegido.

Objetivos

El objetivo de este tutorial es brindarles una guía de inicio al framework que les facilite su pronta incorporación a sus proyectos. En la misma trataremos los temas más importante del mismo tales como instalación, reglas para su uso, características principales entre otros. Al finalizar la lectura de este tutorial ya debemos estar en condiciones de hacer uso de bootstrap en nuestras aplicaciones y tener la capacidad para comenzar a profundizar sobre otros aspectos que se pueden encontrar en su documentación oficial.

Historia

Bootstrap, originalmente llamado Blueprint de Twitter, fue desarrollado por Mark Otto y Jacob Thornton de Twitter, como un marco de trabajo (framework) para fomentar la consistencia entre las herramientas internas.

Antes de Bootstrap, se usaron varias bibliotecas para el desarrollo de interfaces de usuario, lo que generó inconsistencias y una gran carga de trabajo en su mantenimiento.

El primer desarrollo en condiciones reales ocurrió durante la primera "Semana de Hackeo" (Hackweek) de Twitter. Mark Otto mostró a algunos colegas cómo acelerar el desarrollo de sus proyectos con la ayuda de la herramienta de trabajo. Como resultado, decenas de temas se han introducido en el marco de trabajo.

En agosto del 2011, Twitter liberó Bootstrap como código abierto. En febrero del 2012, se convirtió en el proyecto de desarrollo más popular de GitHub.¹

Versiones



Versión 3



Versión 4



Versión 5

Bootstrap utiliza un versionado semántico incremental "Principal.Menor.Parche". La versión "Principal" se utiliza para un cambio importante e incompatible con la versión anterior, actualmente se encuentra en la versión v5.0.1. La versión "Menor" agrega funcionalidad de una manera compatible con la misma versión principal. Por ejemplo la v3.4.2 significa que es compatible con la versión 3. Finalmente el cambio "Parche" se utiliza cuando realiza correcciones de errores dentro de la misma versión menor.

¹ [https://es.wikipedia.org/wiki/Bootstrap_\(framework\)](https://es.wikipedia.org/wiki/Bootstrap_(framework))

Características

Bootstrap permite la combinación de HTML, CSS y JS permitiendo que el desarrollo de aplicaciones web sea de forma sencilla, fácil y compatible con la mayoría de los navegadores web. Su diseño de grillas y adaptable (*responsive en inglés*) a los diferentes tamaños de pantallas de los dispositivos, permite que pueda ser visualizado de forma amigable a la experiencia del usuario en la mayoría de ellos. Su diseño está basado inicialmente en pantallas móviles. Esto significa que una página web se puede ajustar dinámicamente y tomar en cuenta las características del dispositivo usado, dígame celulares, tablets, computadoras, etc. El uso de javascript en el framework permite que sus componentes puedan ser usados de forma interactiva en el navegador.



Investiga. ¿Qué significa diseño web responsive? RWD (Responsive Web Design)
¿Por qué es tan importante en el desarrollo de aplicaciones web?

Instalación

En el siguiente apartado mostraremos cómo incorporar fácil y rápidamente el framework a un proyecto, describiendo tres de las principales formas de instalación. El funcionamiento de Bootstrap requiere tanto de su propia librería de javascript como la de Popper (<https://popper.js.org/>), esta característica hace que se pueda instalar tanto atado (*bundle en inglés*) o por separado.

Es importante recordar que la referencia a los archivos CSS se debe incluir dentro de la etiqueta **<head>...</head>** mediante la etiqueta **<link>**, un ejemplo sería `<head>... <link href="dist/bootstrap/css/bootstrap.min.css" rel="stylesheet"></head>`. Para el caso de los archivos JS (javascript) preferentemente se deben incluir al final de la etiqueta **<body>** mediante la etiqueta **<script>**, un ejemplo de que incluye la librería de bootstrap y popper sería `<body>... <script src="dist/bootstrap/js/bootstrap.bundle.min.js"></script> </body>`.

1. CSS y Javascript compilados y/o minimizados

Esta forma de instalación requiere que los archivos de Bootstrap sean descargados previamente de forma local. Por ejemplo, la última versión disponible al momento de la edición de este documento, se encuentra en el siguiente enlace <https://github.com/twbs/bootstrap/releases/download/v5.0.1/bootstrap-5.0.1-dist.zip>.

Luego se incorporan a la estructura del proyecto, para finalmente poder referenciar los archivos del framework (ver Figura 1.1). Cumpliendo algunos estándares del desarrollo web, para el ejemplo de la Figura 1.1 se copian los archivos dentro de la carpeta “dist” (distribución), pero esta puede tener cualquier otro nombre.

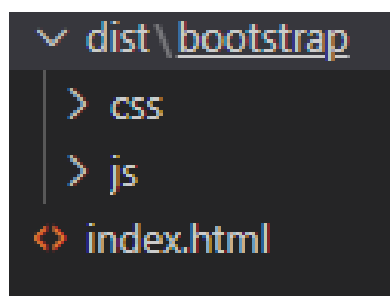


Figura 1.1

Ejemplo 1: Instalación atada, compilada y minimizada

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1">
```

```

<!-- Bootstrap CSS -->
<link href="dist/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<title>Instalación atada, compilada y minimizada</title>
</head>
<body>
  <h1>Atado, compilado y minimizado</h1>
  <!-- Bootstrap JS -->
  <script src="dist/bootstrap/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

Ejemplo 2: Instalación atada y compilada

```

<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <!-- Bootstrap CSS -->
    <link href="dist/bootstrap/css/bootstrap.css" rel="stylesheet">
    <title>Instalación atada y compilada</title>
  </head>
  <body>
    <h1>Atado y compilado</h1>
    <!-- Bootstrap JS -->
    <script src="dist/bootstrap/js/bootstrap.bundle.js"></script>
  </body>
</html>

```

Ejemplo 3: Instalación separada, compilada y minimizada

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <!-- Bootstrap CSS -->
    <link href="dist/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <title>Instalación separada, compilada y minimizada</title>
  </head>
  <body>
    <h1>Separado, compilado y minimizado</h1>
    <!-- Bootstrap JS -->
    <script src="dist/bootstrap/js/bootstrap.min.js"></script>
    <script src="dist/bootstrap/js/bootstrap.esm.min.js"></script>
  </body>
</html>
```

Figura Ejemplo 4: Instalación separada, compilada y minimizada

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <!-- Bootstrap CSS -->
    <link href="dist/bootstrap/css/bootstrap.css" rel="stylesheet">
    <title>Instalación separada y compilada</title>
  </head>
  <body>
    <h1>Separado y compilado</h1>
    <!-- Bootstrap JS -->
    <script src="dist/bootstrap/js/bootstrap.js"></script>
    <script src="dist/bootstrap/js/bootstrap.esm.js"></script>
  </body>
</html>
```


2. CDN a través de jsDelivr

La instalación a través de una Red de Distribución de Contenido (*content delivery network* o *CDN según siglas en inglés*) reemplaza la necesidad de descargar el contenido del framework y de referenciar sus archivos en forma local. La Red de Distribución de Contenido propuesta por Bootstrap de manera oficial para su versión actual (5.0.1) es jsDelivr. (<https://www.jsdelivr.com/package/npm/bootstrap?path=dist>)

```
<!-- Bootstrap CSS -->
<head>...<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOO17+AMvyTG2x" crossorigin="anonymous"></head>
<!-- Bootstrap JS -->
<body>...<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-gtEjrD/SeCtmISkJKNUaaKMoLD0//ElJ19smozuHV6z3Iehds+3U1b9Bn9Plx0x4" crossorigin="anonymous"></script></body>
```

3. Administradores de paquetes

Bootstrap también puede ser instalado o incorporado a nuestros proyectos mediante un administrador de paquetes (*package managers en inglés*). Esta forma nos permite continuar trabajando y utilizando los administradores de paquetes según el lenguaje de programación que estemos usando:

Node Js, Angular: npm (<https://www.npmjs.com/package/bootstrap>)

```
$ npm install bootstrap
```

PHP: Composer (<https://getcomposer.org/>)

```
$ composer require twbs/bootstrap:5.0.1
```

.NET: Nuget (<https://www.nuget.org/packages/bootstrap/>)

```
PM> Install-Package bootstrap
```

4. Archivos

Archivos CSS:

CSS files	Layout	Content	Components	Utilities
bootstrap.css bootstrap.rtl.css bootstrap.min.css bootstrap.rtl.min.css	Included	Included	Included	Included
bootstrap-grid.css bootstrap-grid.rtl.css bootstrap-grid.min.css bootstrap-grid.rtl.min.css	<u>Only grid system</u>	—	—	<u>Only flex utilities</u>
bootstrap-utilities.css bootstrap-utilities.rtl.css bootstrap-utilities.min.css bootstrap-utilities.rtl.min.css	—	—	—	Included
bootstrap-reboot.css bootstrap-reboot.rtl.css bootstrap-reboot.min.css bootstrap-reboot.rtl.min.css	—	<u>Only Reboot</u>	—	—

Figura 1.2 Archivos CSS

Archivos Javascript:

JS files	Popper
bootstrap.bundle.js bootstrap.bundle.min.js	Included
bootstrap.js bootstrap.min.js	—

Figura 1.3 Archivos Javascript <https://getbootstrap.com/docs/5.0/getting-started/contents/>

¿Qué nos provee Bootstrap?

Opciones que nos permite incluir en nuestros proyectos:

- Layout
- Content
- Components
- Formularios
- Utilities



Investiga. ¿A qué hace referencia cada opción? Enumera.

Diseño

El uso correcto del framework hace que los desarrolladores tengan que obedecer cierto estándar en el cumplimiento de algunas reglas, maquetando las páginas web de una forma muy rápida, permitiendo abstraerse de las configuraciones propias de css.

Todos los elementos deben estar contenidos dentro de un contenedor con la clase **.container** o **.container-fluid**. Ejemplo `<div class="container">...</div>`.



Figura: Contenedores

Fuente de la imagen: https://www.w3schools.com/bootstrap4/bootstrap_containers.asp

Contenedores

Tal como se indicó al inicio de esta sección, los contenedores son el espacio de construcción fundamental que incluye Bootstrap, y es donde se ajustan y alinean su contenido dentro de un dispositivo o ventana gráfica determinada.

	Extra pequeño <576px	Pequeño ≥576px	Medio ≥768px	Grande ≥992px	X-grande ≥1200px	XX-Large ≥1400px
.container	100%	540 px	720px	960 px	1140px	1320px
.container-sm	100%	540 px	720px	960 px	1140px	1320px
.container-md	100%	100%	720px	960 px	1140px	1320px
.container-lg	100%	100%	100%	960 px	1140px	1320px
.container-xl	100%	100%	100%	100%	1140px	1320px
.container-xxl	100%	100%	100%	100%	100%	1320px
.container-fluid	100%	100%	100%	100%	100%	100%

 Tabla 1.2 <https://getbootstrap.com/docs/5.0/layout/containers/>

Sistema de Grillas

El sistema Grid permite dividir la página web en 12 columnas permitiendo adaptar la disposición de los elementos como se muestra abajo fácilmente.

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

Figura: Sistema de Grillas

Fuente de la imagen: https://www.w3schools.com/bootstrap4/bootstrap_grid_basic.asp

Antes de continuar, es importante mencionar algunas reglas a la hora de trabajar con el sistema de grillas de Bootstrap:

- Las filas deben estar dentro de un contenedor (.container o .container-fluid).
- Usar filas para crear grupos de columnas. No a la inversa. Esto es posible agregando la clase .row al contenedor horizontal y la clase col al contenedor columna. Ejemplo `<div class="row"><div class="col">Contenido</div></div>`.
- El contenido debe estar dentro de las columnas.
- Los únicos elementos anidados dentro de una fila (row) deben ser columnas (col).
- Las columnas se deben crear especificando el número de columnas disponibles (12).

A continuación se muestra la estructura básica del sistema de grillas que propone Bootstrap:

```
<!-- Control the column width, and how they should appear on different devices -->
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>

<!-- Or let Bootstrap automatically handle the layout -->
<div class="row">
  <div class="col"></div>
  <div class="col"></div>
  <div class="col"></div>
</div>
```

Figura: Estructura básica

Fuente de la imagen: https://www.w3schools.com/bootstrap4/bootstrap_grid_basic.asp

Columnas de Ancho Automático

Para definir columnas de ancho automático simplemente debemos utilizar la clase "col" como se muestra en el siguiente ejemplo:

Column	Column	Column
--------	--------	--------

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

```

    </div>
  </div>

```

Columnas de Ancho Fijo

Las columnas de ancho automático distribuyen el ancho de la pantalla equitativamente para cada columna pero ¿y si deseamos establecer un ancho fijo?. En este caso, simplemente debemos especificar el número de columnas como sigue:

.col-4	.col-8
--------	--------

```

<div class="container">
  <div class="row">
    <div class="col-4"> Column </div>
    <div class="col-8"> Column </div>
  </div>
</div>

```

Las columnas se deben crear especificando de manera explícita teniendo en cuenta el número de columnas disponibles, cuya cantidad total es 12.

La combinación del sistema de grillas con los puntos de interrupción, es la unión perfecta para colocar nuestro contenido en la web según cada criterio de diseño y con la posibilidad de adaptarlo y ordenarlo al tamaño de pantalla donde se esté mostrando.

Nota: El sistema de grilla utiliza una cuadrícula flexbox para su diseño.

xs	sm	md	lg	xl	xxl
<576px	≥576px	≥768px	≥992px	≥1200px	≥1400px

Envase max-width	Ninguno (automático)	540 px	720px	960 px	1140px	1320px
Prefijo de clase	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-	.col-xxl-
# de columnas	12					

Puntos de interrupción

El framework para lograr un diseño ajustable o responsive incluye seis puntos de interrupción o de ruptura, predeterminados. El uso eficiente de estos puntos, como diseñadores o maquetadores web, nos habilita una herramienta para combinar en conjunto al sistema de grillas que propone bootstrap para generar contenidos que se ajusten correctamente a la pantalla del dispositivo donde se esté mostrando. Por esta razón debemos conocer en detalle cuáles son los infijos de clases y el rango de píxeles de pantallas en los que son identificados. (Ver Tabla 1.1)

Nota: Es importante recordar que el concepto responsive hace alusión a un diseño que sea accesible y adaptable a todos los dispositivos.

Breakpoint	Infijo de clase	Dimensiones
X-pequeño	<i>Ninguno</i>	<576 px
Pequeña	sm	≥576px
Medio	md	≥768px
Grande	lg	≥992px
Extra grande	xl	≥1200px
Extra extra grande	xxl	≥1400px

Tabla 1.1 <https://getbootstrap.com/docs/5.0/layout/breakpoints/>

Columnas Responsive

De acuerdo a lo expresado arriba, podemos utilizar los puntos de interrupción para lograr un diseño ajustable como sigue:

.col-sm-4	.col-sm-8
-----------	-----------

```
<div class="container">
  <div class="row">
    <div class="col-sm-4"> .col-sm-4 </div>
    <div class="col-sm-8"> .col-sm-8 </div>
  </div>
</div>
```

Columnas de Ancho Mixto

Podemos combinar las columnas de ancho automático y fijo como sigue:

.col	.col-6	.col
------	--------	------

```
<div class="container">
  <div class="row">
    <div class="col"> .col</div>
    <div class="col-6"> .col-6 </div>
    <div class="col"> .col</div>
  </div>
</div>
```



Desafío

1- Haciendo uso del sistema de grillas de Bootstrap, intenta llegar al siguiente resultado (incluyendo el responsive)

Tamaño de sm en adelante

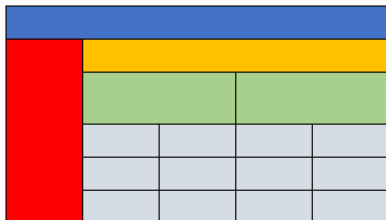


Tamaño xs



2- Haciendo uso del sistema de grillas de Bootstrap, intenta ahora llegar al siguiente resultado.

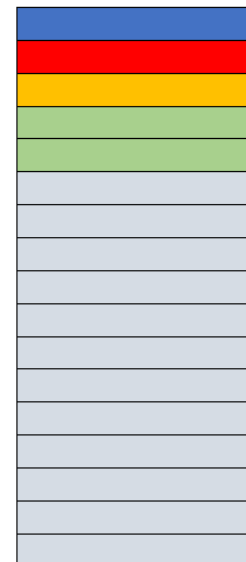
Tamaño md, lg y xl



Tamaño sm



Tamaño xs



Contenido

Imágenes

El framework, para el caso de las imágenes, nos propone una serie de clases con el objetivo de hacerlas adaptables y/o que nunca sean más grandes que las columnas que la contienen.

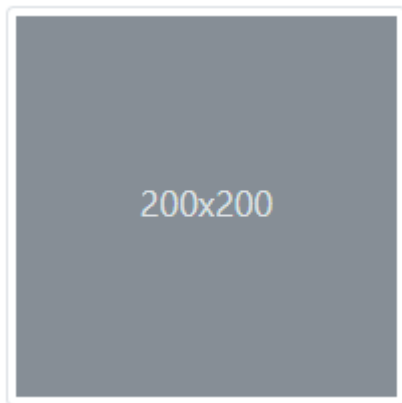
Ejemplo:

Imagen Responsive

```

```

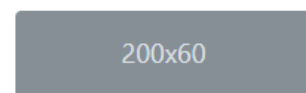
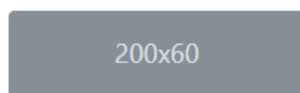
Ejemplo: Imágenes en miniaturas (200 x 200) con un borde de 1px



```

```

Ejemplo: Alinear imágenes (1 al inicio de la fila y 2 al final de la fila)



```


```

Tablas

El uso de tablas en bootstrap es opcional, ya que no es un elemento que se herede del framework. Dado a su uso popular por los programadores, se propone un grupo de clases para trabajar con los estilos de una tabla.

Ejemplo:

Nro	Nombre	Apellido
1	Maximiliano	Guerra
2	Erik	Thomas

```
<table class="table">
  <thead>
    <tr>
      <th scope="col">Nro</th>
      <th scope="col">Nombre</th>
      <th scope="col">Apellido</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>Maximiliano</td>
      <td>Guerra</td>
    </tr>
    <tr>
      <th scope="row">2</th>
      <td>Erik</td>
      <td>Thomas</td>
    </tr>
  </tbody>
</table>
```

También se puede hacer uso de las clases contextuales para dar color a las tablas, de la misma forma que a cualquier otro elemento. (ver Figura 1.3)

Class	Heading	Heading
Default	Cell	Cell
Primary	Cell	Cell
Secondary	Cell	Cell
Success	Cell	Cell
Danger	Cell	Cell
Warning	Cell	Cell
Info	Cell	Cell
Light	Cell	Cell
Dark	Cell	Cell

Figura 1.3 Colores en las Tablas

```

<!-- On tables -->
<table class="table-primary">...</table>
<table class="table-secondary">...</table>
<table class="table-success">...</table>
<table class="table-danger">...</table>
<table class="table-warning">...</table>
<table class="table-info">...</table>
<table class="table-light">...</table>
<table class="table-dark">...</table>
<!-- On rows -->
<tr class="table-primary">...</tr>
<tr class="table-secondary">...</tr>
<tr class="table-success">...</tr>
<tr class="table-danger">...</tr>
<tr class="table-warning">...</tr>
<tr class="table-info">...</tr>
<tr class="table-light">...</tr>
<tr class="table-dark">...</tr>
<!-- On cells (`td` or `th`) -->
<tr>

```

```
<td class="table-primary">...</td>
<td class="table-secondary">...</td>
<td class="table-success">...</td>
<td class="table-danger">...</td>
<td class="table-warning">...</td>
<td class="table-info">...</td>
<td class="table-light">...</td>
<td class="table-dark">...</td>
</tr>
```

Figuras

Al usar imágenes con texto relacionado se recomienda el uso de la etiqueta `<figure>`. La etiqueta `<figcaption>` contiene el texto de la imagen, pudiendo éste ser alineado de igual manera que los elementos de texto.

Ejemplo:




Figura 1.2

Figura 1.2

```
<figure class="figure">
  
  <figcaption class="figure-caption">Figura 1.2</figcaption>
</figure>
```

Formularios

Introducción

Al igual que las imágenes, bootstrap provee un grupo de clases para la presentación de los controles dentro de un formulario.

Ejemplo:

```
<form>
  <div class="mb-3">
    <label for="idE" class="form-label">Correo</label>
    <input type="email" class="form-control" id="idE"
aria-describedby="idEH">
    <div id="idEH" class="form-text">Debe ser un correo válido.</div>
  </div>
  <div class="mb-3">
    <label for="exampleInputPassword1"
class="form-label">Password</label>
    <input type="password" class="form-control"
id="exampleInputPassword1">
  </div>
  <div class="mb-3 form-check">
    <input type="checkbox" class="form-check-input"
id="exampleCheck1">
    <label class="form-check-label"
for="exampleCheck1">Recordarme</label>
  </div>
  <button type="submit" class="btn btn-primary">Enviar</button>
</form>
```

Controles

Los controles dentro de un formulario pueden ser personalizados e incluso utilizados de forma adaptativa, gracias a las clases provistas por bootstrap.

Dimensionamiento:

- .form-control-sm (tamaño pequeño)

- `.form-control` (tamaño normal)
- `.form-control-lg` (tamaño grande)

Habilitado:

- Atributo **disabled** (Control deshabilitado)
- Atributo **readonly** (Control sólo lectura)
- Atributos **disabled readonly** (Control deshabilitado, sólo lectura)

Layout

“Cada grupo de campos de formulario debe residir en un elemento `<form>`”. “Bootstrap no proporciona un estilo predeterminado para el `<form>` elemento, pero hay algunas funciones de navegador poderosas que se proporcionan de forma predeterminada.”

- “`<button>`s dentro de un `<form>` valor predeterminado de `type="submit"`, así que esfuércese por ser específico y siempre incluya un atributo `type`.”
- Puede deshabilitar todos los elementos del formulario dentro de un formulario con el `disabled` atributo en `<form>`.

Ya que Bootstrap se emplea `display: block` y `width: 100%` para casi todos los controles del formulario, los formularios se mostrarán por defecto de manera vertical. Se pueden usar clases adicionales para variar este diseño en función de cada formulario.”
<https://getbootstrap.com/docs/5.0/forms/layout/>

Validación de Formularios

La validación de formularios HTML se aplica a través de dos pseudoclases de CSS, `:invalid` y `:valid`. Se aplica a los elementos `<input>`, `<select>` y `<textarea>`.

- Bootstrap aplica los estilos `:invalid` y `:valid` a la clase padre `.was-validated`, generalmente aplicada a `<form>`. De lo contrario, cualquier campo obligatorio sin valor aparecerá como no válido al cargar la página. De esta forma, puedes elegir cuándo activarlos (por lo general, después de intentar enviar el formulario).
- Para restablecer la apariencia del formulario (por ejemplo, en el caso de envíos de formularios dinámicos usando AJAX), elimina la clase `.was-validated` del `<form>` nuevamente después del envío.
- Como alternativa, las clases `.is-invalid` y `.is-valid` se pueden usar en lugar de las pseudoclases para validación del lado del servidor. No requieren una clase padre `.was-validated`.
- Debido a restricciones en el funcionamiento de CSS, no podemos (en este momento) aplicar estilos a una `<label>` que se encuentra antes de un control de formulario en el DOM sin la ayuda de JavaScript personalizado.

- Todos los navegadores modernos admiten la API de validación de restricciones, una serie de métodos de JavaScript para validar controles de formulario.
- Los mensajes de feedback pueden utilizar los predeterminados del navegador (diferentes para cada navegador y sin estilo a través de CSS) o con los estilos personalizados de feedback con HTML y CSS adicionales de Bootstrap.
- Es posible proporcionar mensajes de validez personalizados con `setCustomValidity` en JavaScript.

Consulta el siguiente enlace para más información:
<https://getbootstrap.esdocu.com/docs/5.1/forms/validation/>



Desafío. Crear un formulario de *registro de usuario* tipo cuadrícula. El mismo debe incorporar los estilos semánticos propios de la web (ej. verde ok, rojo fallo, etc.) a fin de brindar una buena experiencia de usuario en cuanto al ingreso de los datos. Incluir validaciones de datos requeridos y de formatos (ej. email, cantidad de caracteres como mínimo y máximos según el caso, etc.)

Componentes

Alertas

Este componente es muy utilizado a la hora de mostrar mensajes al usuario de forma flexible y de una manera más amigable. Las alertas son consecuentes con los estilos de colores que propone el framework (ver Figura 1.4)

Ejemplo:



Figura 1.4 Colores en Alertas

```
<div class="alert alert-primary" role="alert">
  A simple primary alert—check it out!
</div>
<div class="alert alert-secondary" role="alert">
  A simple secondary alert—check it out!
</div>
<div class="alert alert-success" role="alert">
  A simple success alert—check it out!
</div>
<div class="alert alert-danger" role="alert">
  A simple danger alert—check it out!
</div>
<div class="alert alert-warning" role="alert">
  A simple warning alert—check it out!
</div>
<div class="alert alert-info" role="alert">
```

```

    A simple info alert-check it out!
  </div>
  <div class="alert alert-light" role="alert">
    A simple light alert-check it out!
  </div>
  <div class="alert alert-dark" role="alert">
    A simple dark alert-check it out!
  </div>

```

Botones

Al igual que el resto de los controles, los botones incluyen el estilo predefinido. (ver Figura 1 Ejemplo:

```

<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>
<button type="button" class="btn btn-link">Link</button>

```



Figura 1.5 Colores en Botones

Barra de navegación

Por último abordaremos el tema de la barra de navegación, elemento muy utilizado en el desarrollo de aplicaciones web. A continuación citamos las principales reglas para el uso de la barra de navegación. (ver Figura 1.6)

- Las barras de navegación requieren comenzar con la clase .navbar
- Las barras de navegación y su contenido son fluidos por defecto.
- Las barras de navegación responden de forma predeterminada, pero se pueden modificar fácilmente para cambiar su comportamiento.
- Garantice la accesibilidad mediante el uso de un elemento <nav> o, si usa un elemento más genérico como <div>, agregue a role="navigation" a cada barra de navegación.

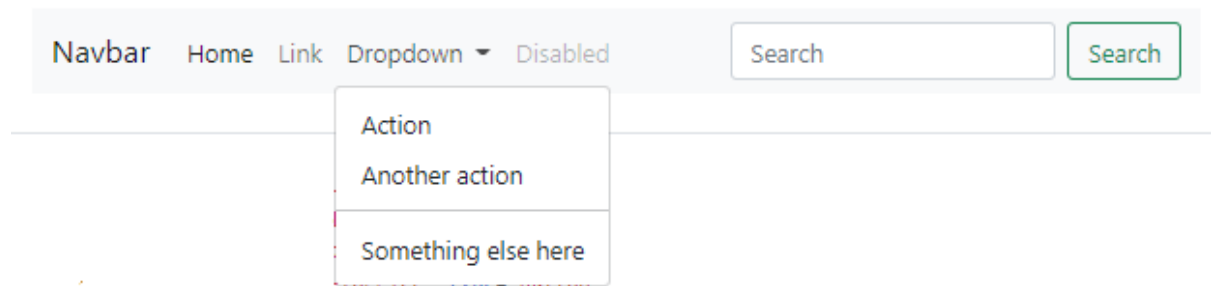


Figura 1.6 Colores en Alertas

Los mismos sirven para:

- Brindar una notificación.
- Emitir una alerta.
- Solicitar información.
- Hacer publicidad.

Características

Las ventanas emergentes en Bootstrap tienen las siguientes características:

- Se construyen con HTML, CSS y Javascript. Se colocan sobre el documento web y eliminan el acceso al mismo hasta que se cierre el modal.
- Al hacer click en fondo (o documento web), el modal se cierra automáticamente.
- Bootstrap sólo admite un formulario emergente a la vez. Los modales anidados no son compatibles porque Bootstrap considera que no permiten experiencias de usuario eficientes.

Para más información sobre ventanas modales consulta:

<https://getbootstrap.esdocu.com/docs/5.1/components/modal/>



Desafío. A continuación intenta crear una ventana emergente para visualizar una imagen en su tamaño real. Es decir, el documento web de base muestra la imagen en tamaño pequeño y, al hacer click en el botón "Ver imagen", se mostrará en una ventana emergente la imagen en tamaño real.

Nota: Bootstrap brinda muchos más componentes para agregar a nuestras páginas web. Son ejemplos: carruseles, tarjetas, paginación, barras de progreso, acordeones, etc. Para conocer más acerca de estos componentes, hacer click en el siguiente enlace:

<https://getbootstrap.esdocu.com/docs/5.1/components/accordion/>

Ventanas Emergentes (Modal)

Las ventanas emergentes (o modales) no son más que cuadros de diálogo.

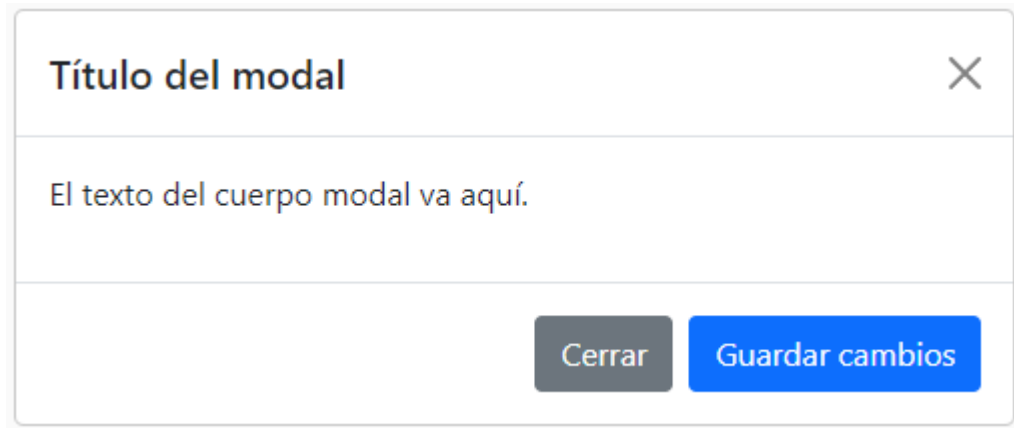


Figura 1.7: Ventana Modal

Fuente de la imagen: <https://getbootstrap.esdocu.com/docs/5.1/components/modal/>

Utilidades (utilities)

En utilidades podremos encontrar cómo trabajar los colores, bordes, el posicionamiento de los elementos, sombras, alineación vertical, entre otros.

Colores en Bootstrap

Bootstrap nos provee una serie de colores semánticos. Es decir que los colores transmiten un significado.

Ejemplo:

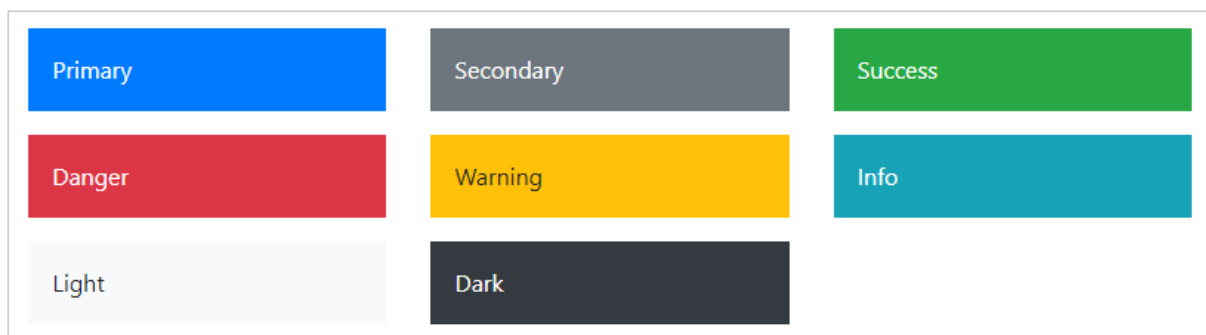


Figura 1.8: Semántica de los colores en bootstrap

Fuente de la imagen:

<https://juancarlosarteaga.wordpress.com/2018/11/12/como-usar-css-bootstrap-contenido-estilos-y-componentes/>

Colorear Texto

Para colorear textos, utilizar las clases text-* dónde (*) es uno de los colores semánticos definidos por bootstrap (definidos arriba).

`.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`

Figura 1.9: Color de texto

En HTML:

```
<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 bg-dark">.text-warning</p>
<p class="text-info bg-dark">.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>
```

Nota: En desuso la utilidades .text-opacity-*, .text-black-50 y .text-white-50 a partir de la v5.1.0 . Se eliminarán en v6.0.0.



Investiga. ¿Qué es la opacidad? ¿Cómo trabajamos la opacidad en la nueva versión de Bootstrap?

Colorear fondo

Al igual que para colorear texto, podemos colorear los fondos. Para ello, utilizar las clases bg-* dónde (*) es uno de los colores semánticos definidos por bootstrap.

Ejemplo:

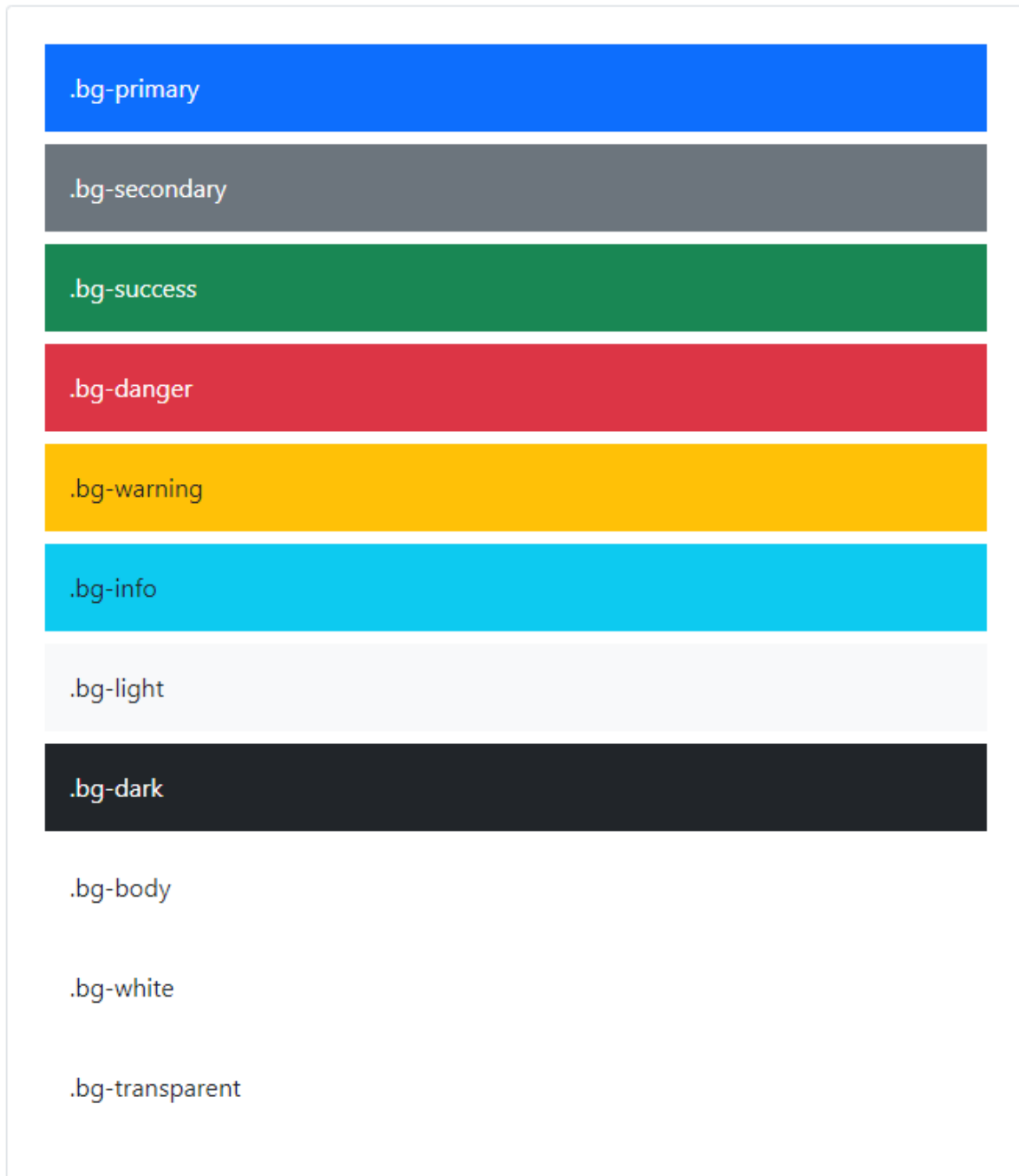


Figura 1.20: Colores de fondo

Fuente de la imagen: <https://getbootstrap.esdocu.com/docs/5.1/utilities/background/>

En HTML:

```
<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-dark">.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-body text-dark">.bg-body</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>
```



Desafío: ¿Es posible agregar un degradado a los fondos? Si la respuesta es afirmativa, investiga cómo e intenta hacer una práctica.

Display

La utilidad Display alterna de manera rápida y responsive el valor de visualización de los componentes incluyendo también soporte para algunos de los valores más comunes, así como algunos extras para controlar la visualización al imprimir.

La propiedad Display establece si un elemento se trata como un bloque o un elemento de línea y, el layout (diseño) para los elementos secundarios tales como: grid o flex

Sintaxis

Las clases se nombran usando el formato:

.d-{value} para xs

.d-{breakpoint}-{value} para sm, md, lg, xl, y xxl.

Donde value puede ser:

- none
- inline
- inline-block
- block
- grid
- table
- table-cell

- table-row
- flex
- inline-flex

Ejemplo:

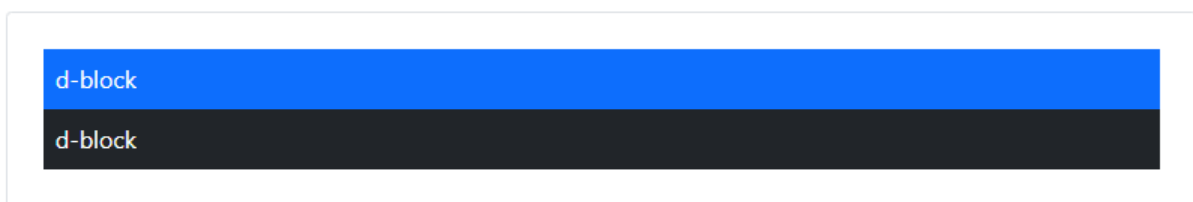


Figura 1.21: Display

Fuente de la imagen: <https://getbootstrap.esdocu.com/docs/5.1/utilities/display/>

Código HTML:

```
<div class="d-inline p-2 bg-primary text-white">d-inline</div>
<div class="d-inline p-2 bg-dark text-white">d-inline</div>
```



Fuente de la imagen: <https://getbootstrap.esdocu.com/docs/5.1/utilities/display/>

Código HTML:

```
<span class="d-block p-2 bg-primary text-white">d-block</span>
<span class="d-block p-2 bg-dark text-white">d-block</span>
```

Ocultar elementos

Para un desarrollo optimizado de aplicaciones web para dispositivos móviles más rápido, muchas veces debemos usar clases de visualización responsive para mostrar y ocultar elementos por dispositivo evitando crear versiones completamente diferentes de la aplicación para diferentes dispositivos.

Para ocultar elementos, simplemente debemos usar la clase `.d-none` o una de las clases `.d-{sm,md,lg,xl,xxl}-none` para cualquier variación de pantalla responsive.

Para mostrar un elemento solo en un intervalo dado de tamaños de pantalla, puedes combinar una clase `.d-*-none` con una clase `.d-*-*`, por ejemplo `.d-none .d-md-block`

.d-xl-none .d-xxl-none ocultará el elemento para todos los tamaños de pantalla excepto en dispositivos medianos y grandes (<https://getbootstrap.esdocu.com/docs/5.1/utilities/display/>).

Tamaño de pantalla	Clase
Oculto en todos	.d-none
Oculto solo en xs	.d-none .d-sm-block
Oculto solo en sm	.d-sm-none .d-md-block
Oculto solo en md	.d-md-none .d-lg-block
Oculto solo en lg	.d-lg-none .d-xl-block
Oculto solo en xl	.d-xl-none .d-xxl-block
Oculto solo en xxl	.d-xxl-none
Visible en todos	.d-block
Visible solo en xs	.d-block .d-sm-none
Visible solo en sm	.d-none .d-sm-block .d-md-none
Visible solo en md	.d-none .d-md-block .d-lg-none
Visible solo en lg	.d-none .d-lg-block .d-xl-none
Visible solo en xl	.d-none .d-xl-block .d-xxl-none
Visible solo en xxl	.d-none .d-xxl-block

Figura 1.22: Ocultar elementos

Fuente de la imagen: <https://getbootstrap.esdocu.com/docs/5.1/utilities/display/>

Consulta el siguiente enlace para más información:

<https://getbootstrap.esdocu.com/docs/5.1/utilities/api/>

Referencias

Semver. Semantic Versioning. <https://semver.org/>

Bootstrap (framework). [https://es.wikipedia.org/wiki/Bootstrap_\(framework\)](https://es.wikipedia.org/wiki/Bootstrap_(framework))

Getbootstrap. Bootstrap v5.0, <https://getbootstrap.com/>