

Guia Bootstrap 4

Bootstrap es un framework open source de css desarrollado por Twitter utilizado para facilitar el desarrollo de páginas web adaptables y estilizadas.

Adaptables porque reconoce el ancho en píxeles de la pantalla en que se está mostrando y acomoda el tamaño de los componentes, y estilizadas porque define estilos por defecto para gran cantidad de elementos html, por lo que es muy utilizado por desarrolladores backend para hacer maquetas mas vistosas que en html simple.

*Por comodidad de sintaxis y por necesidad de práctica todos los ejemplos están escritos en el motor de plantillas pug.

La guia fue hecha como apunte de las siguientes fuentes:

<https://www.tutorialesprogramacionya.com/bootstrap4ya/index.php?inicio=0>

<https://www.youtube.com/playlist?list=PLL0TiOXBeDagsYUYFO9WMwDtuDuoGZVB9>

Sumario

1 – Instalación.....	2
2 – Primera página.....	3
3 – Sistema de grilla.....	6
3.1 – Columnas de regla múltiple.....	8
3.2 – Anidamiento de columnas.....	8
3.3 – Ordenamiento de columnas.....	9
3.4 – Columnas de ancho automático.....	9
4 – Alineación de columnas.....	10
4.1 – Alineación vertical.....	11
4.2 – Alineación horizontal.....	13
4.3 – Desplazamiento de columnas.....	15
5 – Tablas.....	17
5.1 – Estilos de tablas.....	17
5.2 – Estilos aplicados a filas y columnas.....	19
6 – Formularios.....	21
6.1 – Text, password.....	21
6.2 – textarea.....	22
6.3 – Checkbox y radio.....	23
6.4 – Select.....	25
7 – tipografías, posiciones y espaciados.....	26
7.1 – Tipografía.....	26
7.2 – Alineaciones.....	27
7.3 – Elementos en línea y bloque.....	29
7.4 – márgenes y rellenos.....	30
7.5 – Alto y ancho.....	30
8 – Breakpoints.....	31
9 – Botones.....	32
9.1 – Colores.....	33
9.2 – Tamaños.....	34
9.3 – Agrupaciones.....	35
10 – Listas.....	36
11 – navegación.....	38

1 – Instalación

Para utilizar bootstrap en un proyecto hay principalmente 3 métodos.

- **Instalación.** Descargar los estilos en un documento css y agregarlo localmente al proyecto.
- **CDN.** Content Delivery Network. Consiste en pegar en el encabezado del documento una dirección url que permite usarlo desde internet.
- **Administrador de paquetes.** Agregar la dependencia a un proyecto usando un administrador de paquetes, como por ejemplo NPM en un proyecto NodeJS.

El CDN de bootstrap se compone de dos enlaces, uno al archivo css que contiene los estilos para mejorar la visualización de la página, y otro al script en JavaScript para realizar animaciones y otras funciones. Estos pueden copiarse de la siguiente página:

<https://getbootstrap.com/docs/4.3/getting-started/introduction/>

Para usar componentes de Bootstrap que requieran JavaScript es necesario agregarle las referencias (ya sea por cdn o por descarga) de las funciones de bootstrap, jquery y ajax.

Las ventajas de utilizar esta forma de cargar el framework Bootstrap 4 es:

- Reduce la carga de nuestros servidores (importante si utilizamos un hosting compartido con recursos limitado)
- Facilita que Bootstrap 4 quede en caché del navegador y se vuelva a cargar cada vez que se visita una web que lo utilice.
- Reduce la latencia.
- Reducción de costos de nuestros servidores por requerir menor ancho de banda y tráfico.

Como desventaja podríamos decir:

- Está la posibilidad que el servidor donde se aloja Bootstrap 4 se caiga, si bien es mucho más probable que nuestro servidor sobre todo si es un hosting compartido es el que tenga problemas.
- No podemos probar en forma local nuestro sitio web si no se encuentra conectado a internet.

Para trabajar con los archivos locales, ya sea la versión compilada y comprimida, o con los archivos en código fuente es necesario descargarlos de la página de bootstrap, donde también se encuentran los enlaces actualizados del cdn, agregarlos al proyecto y referenciar su ubicación en el index.html.

<https://getbootstrap.com/docs/4.1/getting-started/download/>

2 - Primera página

Para asegurarse que la página se va a adaptar a todo tipo de navegador, ya sea de computadora, celular o tablet, es necesario agregar al head la etiqueta.

```
meta(name="viewport", content="width=device-width, initial-scale=1.0")
```

Para ver la diferencia entre html básico y bootstrap creamos una página con un título y un párrafo.

Página en html plano

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Eaque voluptatem suscipit obcaecati totam enim molestiae? Rerum minima veritatis natus fugiat facere blanditiis odit dolor, tempore explicabo commodi voluptatibus!

Agregando bootstrap y poniendo todo el documento en un contenedor se ve como cambia la familia y color de fuente, y agrega algunos márgenes al contenido. Para usar el cdn de bootstrap se agrega en el head la referencia

```
1 - Introduccion > index.pug
1  doctype
2  html(lang="en")
3    head
4      meta(charset="UTF-8")
5      meta(name="viewport", content="width=device-width, initial-scale=1.0")
6      link(rel="stylesheet", href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css")
7      title Document
8    body
9      .container
10        h1 página con bootstrap
11        br
12        p Lorem, ipsum dolor sit amet consectetur adipisicing elit. Temporibus, maxime. Incidunt necessitatibus illum nisi magni consectetur qui expedita distinctio mollitia?
13
```

La página ahora tiene este aspecto



Un contenedor es una clase aplicada al elemento div, los dos tipos mas usados son

- **Container.** Muestra todo el contenido de un div de manera centrada.

- **Container-fluid.** El contenedor sencillo muestra todo el contenido centrado y con un ancho fijo, en cambio el fluido se adapta al ancho de la pantalla. Así, comparando uno con el otro las vistas son las siguientes.

* En el directorio de los ejemplos creamos el archivo `css/estilos.css` donde vamos agregando algunos estilos de colores de fondo para diferenciar los contenedores, sus posiciones y tamaños. Agregamos entonces en el head también la referencia al css propio.

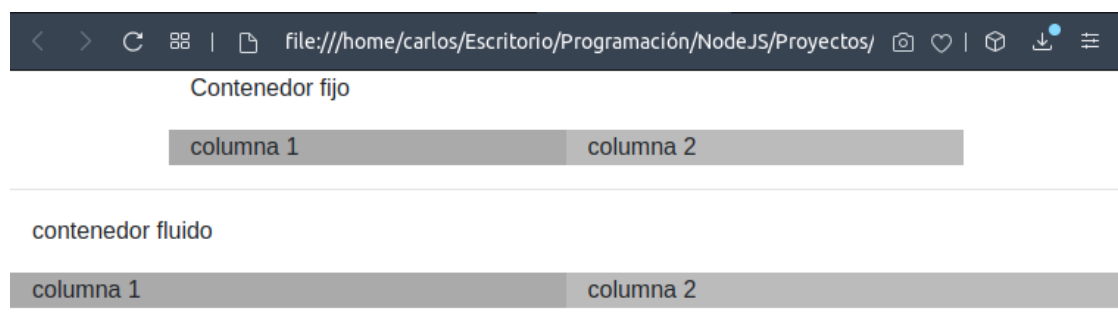
```
link(rel="stylesheet", href="../css/estilos.css")
```

estilos.css

```
css > # estilos.css > ...
1
2 .aaa{
3   background-color: #aaa;
4 }
5 .bbb{
6   background-color: #bbb;
7 }
```

Container.pug

```
1 - Introduccion > container.pug
11 body
12   .container
13     p Contenedor fijo
14     .row
15       .col.aaa columna 1
16       .col.bbb columna 2
17   hr
18   .container-fluid
19     p contenedor fluido
20     .row
21       .col.aaa columna 1
22       .col.bbb columna 2
```



3 - Sistema de grilla

Debemos pensar para armar nuestro esquema de la página que tenemos la posibilidad de definir **filas** y en cada fila definir de **1 hasta 12 columnas**. Cada columna con un ancho relativo a ese número 12.

Luego Bootstrap se encarga de colapsar las columnas cuando se accede al sitio desde un dispositivo con una capacidad limitada en cuanto al ancho en píxeles (esto permite una experiencia más placentera al visitante que accede a nuestra página desde un celular con capacidades limitadas).

Cada fila se declara con la clase **row**, y cada columna con la clase **col-*-***.

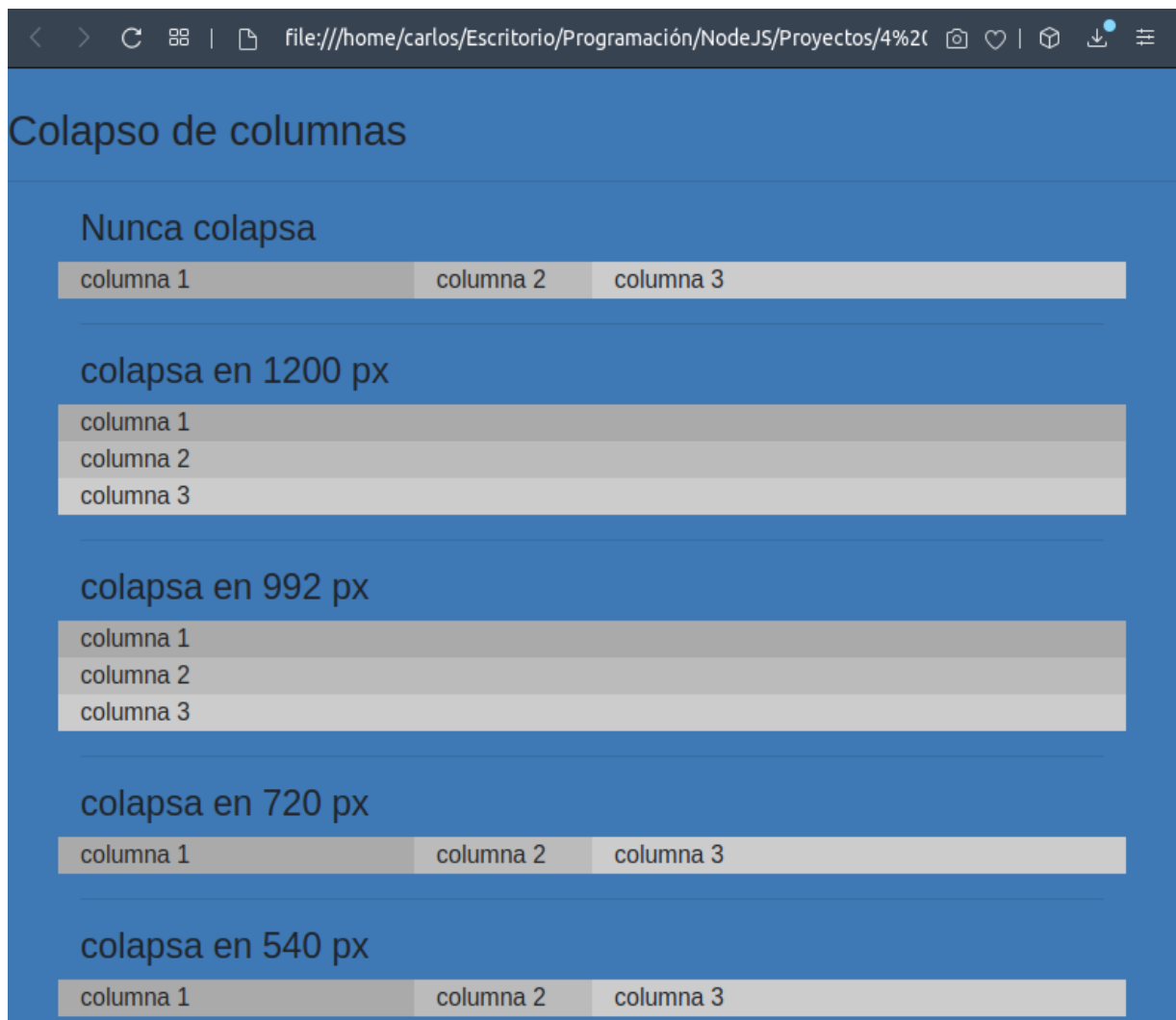
Los asteriscos de la clase col representan el tamaño según el que se ajustan las columnas, y la proporción de tamaño de cada columna en la fila (este número debe ser un entero entre 1 y 12).

Los valores que puede tomar el primer atributo indican **cuando** deben **colapsar** las columnas, es decir, separarse en dos filas cuando ya no entran en el ancho de la pantalla. Si este valor no se indica las columnas nunca colapsan.

Atributo	significado	Ancho de pantalla en que colapsa
xl	Extra large	< 1200px
lg	Large	< 992px
md	Medium	< 720px
<u>sm</u>	Small	< 540px

```
2 - grilla > 1_basico.pug
1  extends ../template/template.pug
2
3  block content
4    body
5      br
6      h3 Colapso de columnas
7      hr
8      .container
9        h4 Nunca colapsa
10       .row
11         .col-4.aaa columna 1
12         .col-2.bbb columna 2
13         .col-6.ccc columna 3
14       hr
15       h4 colapsa en 1200 px
16       .row
17         .col-xl-4.aaa columna 1
18         .col-xl-2.bbb columna 2
19         .col-xl-6.ccc columna 3
20       hr
```

```
2- grilla > 1_basico.pug
20 |         hr
21 |         h4 colapsa en 992 px
22 |         .row
23 |             .col-lg-4.aaa columna 1
24 |             .col-lg-2.bbb columna 2
25 |             .col-lg-6.ccc columna 3
26 |         hr
27 |         h4 colapsa en 720 px
28 |         .row
29 |             .col-md-4.aaa columna 1
30 |             .col-md-2.bbb columna 2
31 |             .col-md-6.ccc columna 3
32 |         hr
33 |         h4 colapsa en 540 px
34 |         .row
35 |             .col-sm-4.aaa columna 1
36 |             .col-sm-2.bbb columna 2
37 |             .col-sm-6.ccc columna 3
38 |
```



3.1 – Columnas de regla múltiple

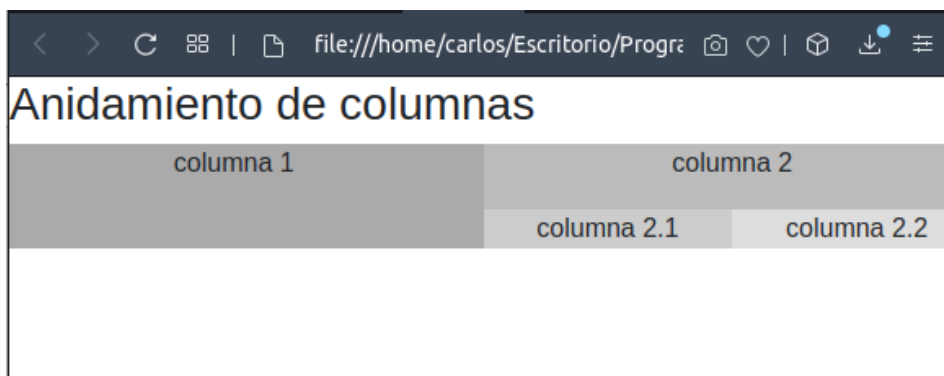
También se pueden definir reglas para cada tamaño, así se **ajusta el tamaño de las columnas** de distinta manera **según el tamaño de la pantalla**. Por ejemplo, definimos dos columnas y queremos que la primera al estar en una pantalla grande ocupara 9 de los 12 espacios de la fila, en una mediana 8, en una chica 7 , en una muy chica 6, y 5 cuando ya no colapsa mas, mientras que la segunda columna adapte su tamaño a los cambios de la primera debe definirse de la siguiente manera.

```
2-grilla > 2_anchoVariable.pug
1 extends ../template/template.pug
2
3 block content
4   body
5     h3 Columnas de ancho variable
6     .row
7       .col-xl-9.col-lg-8.col-md-7.col-sm-6.col-5.aaa columna 1
8       .col.bbb columna 2
```

3.2 – Anidamiento de columnas

También se puede anidar columnas dentro de otras, para eso se declara un **row** dentro de una columna y dentro de este otras columnas. Además al anidar se generan otras 12 unidades dentro de la columna particionada.

```
2-grilla > 3_anidamiento.pug
1 extends ../template/template.pug
2
3 block content
4   body
5     h3 Anidamiento de columnas
6     .row.blue.text-center
7       .col.aaa columna 1
8       .col.bbb
9         p columna 2
10      .row
11        .col.ccc columna 2.1
12        .col.ddd columna 2.2
13
14
```

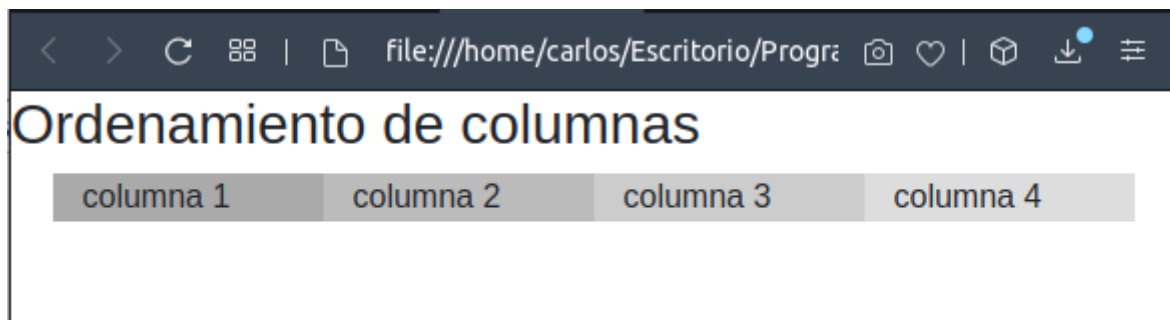


3.3 – Ordenamiento de columnas

Por defecto el navegador muestra las columnas en el orden que fueron declaradas, si uno quiere que se muestren en un orden específico se debe usar el modificador de clase **order-***. El reordenamiento se puede aplicar para los distintos tamaños de pantalla, luego tenemos las siguientes clases:

order-* **order-sm-*** **order-md-*** **order-lg-***
order-xl-*

```
2 - grilla > 4_ordenamiento.pug
1 extends ../template/template.pug
2
3 block content
4   body
5     h3 Ordenamiento de columnas
6     .container
7       .row.blue
8         .col.order-3.ccc columna 3
9         .col.order-1.aaa columna 1
10        .col.order-4.ddd columna 4
11        .col.order-2.bbb columna 2
```



3.4 – Columnas de ancho automático

Al hablar de ancho automático podemos estar refiriéndonos a dos cosas.

- Que el ancho se asigne automáticamente según el espacio disponible. El navegador considera que el ancho es automático al no asignarle ningún valor numérico.

```

2 - grilla > 5_anchoAuto.pug
5      h3.text-center caso 1
6      .container.text-center
7          .row
8              .col.aaa columna 1
9              .col-7.bbb columna 2
10             .col.ccc columna 3
11      hr

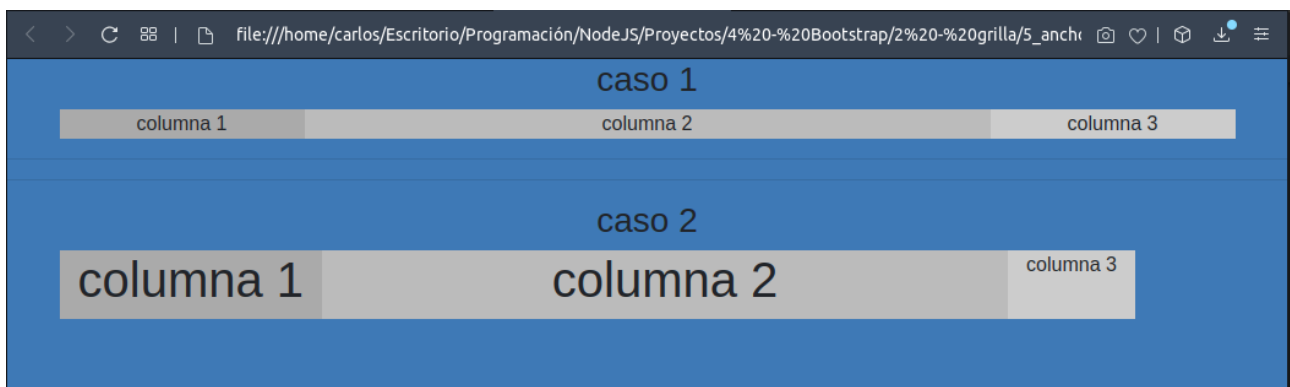
```

- para el segundo caso, si en vez de un número indicamos la palabra **auto**, el ancho se va a adaptar al contenido de la columna.

```

2 - grilla > 5_anchoAuto.pug
13     h3.text-center caso 2
14     .container.text-center
15         .row
16             .col-auto.aaa: h1 columna 1
17             .col-7.bbb: h1 columna 2
18             .col-auto.ccc: p columna 3

```



4 – Alineación de columnas

Bootstrap 4 trae como ventaja con respecto a versiones anteriores (3.x, 2.x etc.) la incorporación de los estilos de **flexbox**. Gracias a este se pueden alinear las columnas, tanto vertical como horizontalmente dentro de una fila con las siguientes clases. Si se quiere

*Todas estas alineaciones pueden practicarse con la página [flexbox froggy](#)

<https://flexboxfroggy.com/#es>

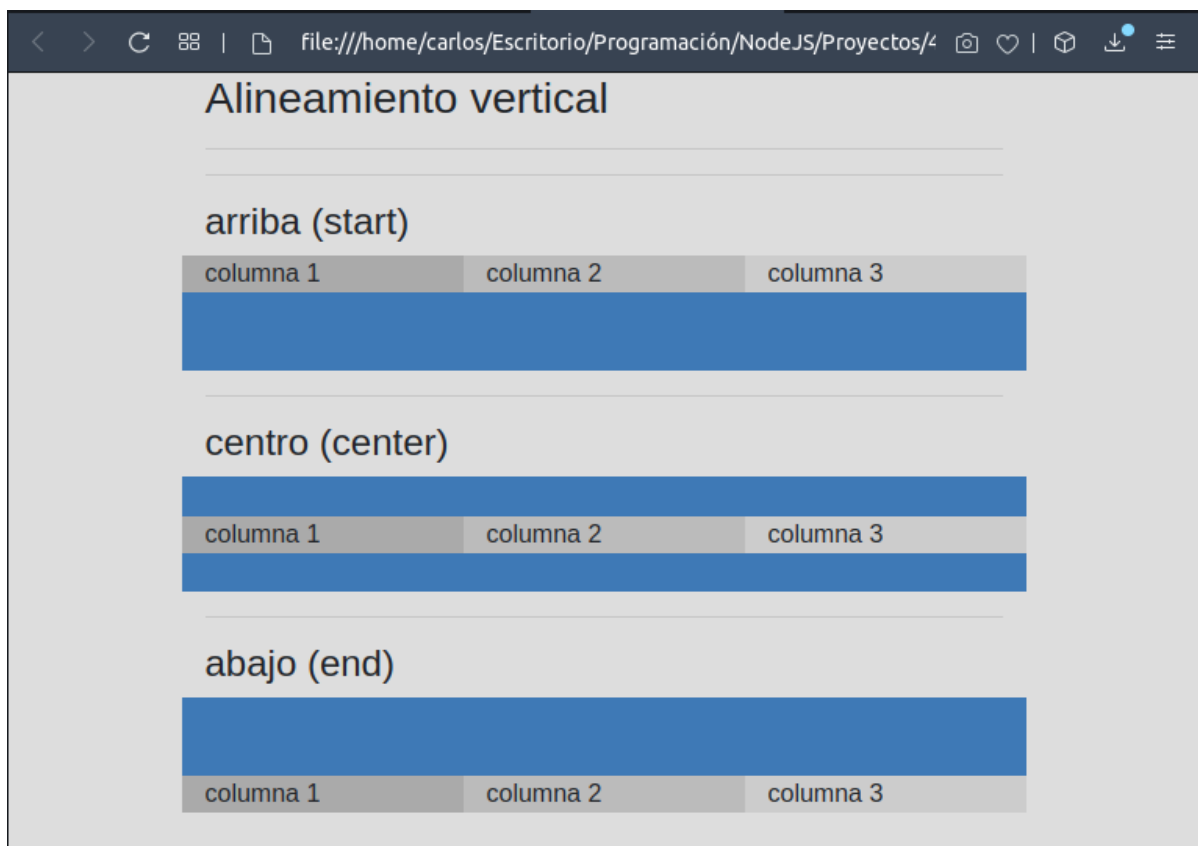
4.1 - Alineación vertical.

- **align-items-start.** Alinea las columnas en la parte superior de la fila.
- **align-items-center.** Centra verticalmente las columnas.
- **align-items-end.** Alinea las columnas en la parte inferior de la fila.

```

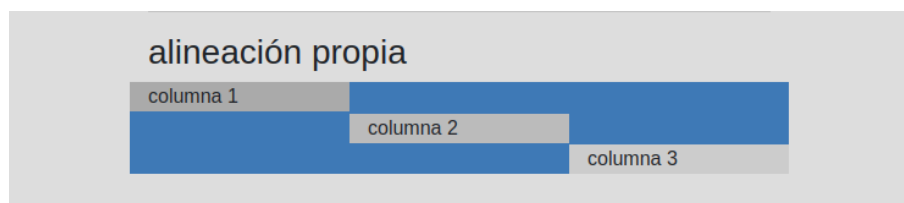
2 - grilla > 6_alinVertical.pug
1  extends ../template/template
2
3  block content
4      body.ddd
5          .container
6              h3 Alineamiento vertical
7              hr
8              hr
9              h4 arriba (start)
10             .row.blue.align-items-start(style="height: 75px;")
11                 .col.aaa columna 1
12                 .col.bbb columna 2
13                 .col.ccc columna 3
14
15             hr
16             h4 centro (center)
17             .row.blue.align-items-center(style="height: 75px;")
18                 .col.aaa columna 1
19                 .col.bbb columna 2
20                 .col.ccc columna 3
21
22             hr
23             h4 abajo (end)
24             .row.blue.align-items-end(style="height: 75px;")
25                 .col.aaa columna 1
26                 .col.bbb columna 2
27                 .col.ccc columna 3

```



También se puede modificar la clase col con **align-self-*** para que cada columna quede en una posición particular.

```
2-grilla > 6_alinVertical.pug
31      h3 alineación propia
32      .row.blue(style="height: 75px;")
33          .col.aaa.align-self-start columna 1
34          .col.bbb.align-self-center columna 2
35          .col.ccc.align-self-end columna 3
```



4.2 - Alineación horizontal

Para alinear las columnas horizontalmente se agregan a la clase row las siguientes.

- **justify-content-end.** Agrupa todos los elementos a la izquierda de la fila.
- **justify-content-start.** Agrupa todos los elementos a la derecha de la fila.
- **justify-content-center.** Centra todos los elementos juntos en la fila.
- **justify-content-between.** Establece una separación entre elementos sin considerar márgenes a izquierda y derecha de la fila.
- **justify-content-around.** Separa todos los elementos distribuyendo equitativamente el espacio sobrante.

alineación horizontal

izquierda (start)



centro (center)



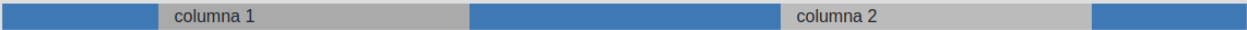
derecha (end)



separado (between)



distribuido (around)



```
2 - grilla > 7_alinHorizontal.pug
3  block content
4      body.ddd
5          .container
6              h3 alineación horizontal
7              hr
8              hr
9              h4 izquierda (start)
10             .row.blue.justify-content-start
11                 .col-3.aaa columna 1
12                 .col-3.bbb columna 2
13             hr
14             h4 centro (center)
15             .row.blue.justify-content-center
16                 .col-3.aaa columna 1
17                 .col-3.bbb columna 2
18             hr
19             h4 derecha (end)
20             .row.blue.justify-content-end
21                 .col-3.aaa columna 1
22                 .col-3.bbb columna 2
23             hr
24             h4 separado (between)
25             .row.blue.justify-content-between
26                 .col-3.aaa columna 1
27                 .col-3.bbb columna 2
28             hr
29             h4 distribuido (around)
30             .row.blue.justify-content-around
31                 .col-3.aaa columna 1
32                 .col-3.bbb columna 2
```

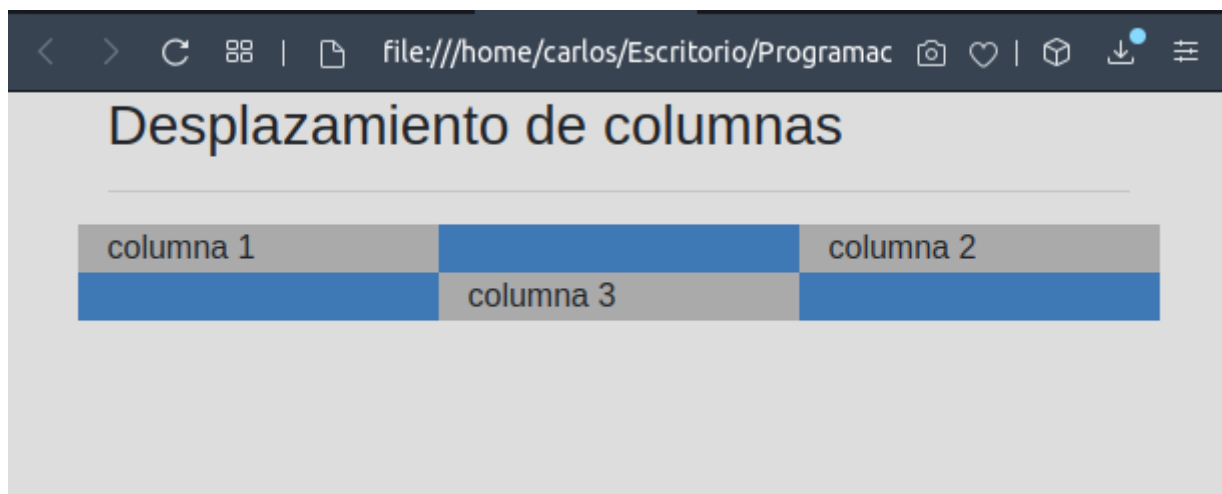
NOTA: todas estas alineaciones pueden usarse no solo en columnas sino en cualquier elemento **div** agregandole la clase **d-flex**.

4.3 – Desplazamiento de columnas

En caso de necesitar desplazar una columna una cantidad fija de unidades se debe utilizar alguna de las siguientes clases.

offset-*	offset-sm-*	offset-md-*	offset-lg-*	offset-xl-*
----------	-------------	-------------	-------------	-------------

```
2 - grilla > 8_desplCol.pug
1 extends ../template/template
2
3 block content
4   body.ddd
5     .container
6       h3 Desplazamiento de columnas
7       hr
8       .row.blue
9         .col.aaa columna 1
10        .col.offset-4.aaa columna 2
11      .row.blue
12        .col-4.offset-4.aaa columna 3
13
```



5 – Tablas

Cuando incluimos Bootstrap 4 also introducen una serie de reglas a los elementos relacionados a la etiqueta html "**table**". El estilo mas básico de tabla en bootstrap se logra añadiendo la clase **table** al elemento table de html.

5.1 – Estilos de tablas

Algunos de los estilos mas comunes de tabla son.

- Tabla básica html

```
3 - tablas > 💖 tablas.pug
8      table
9        thead
10         th Titulo 1
11         th Titulo 2
12         th Titulo 3
13       tbody
14         tr
15           td casilla 1-1
16           td casilla 1-2
17           td casilla 1-3
18         tr
19           td casilla 2-1
20           td casilla 2-2
21           td casilla 2-3
22         tr
23           td casilla 3-1
24           td casilla 3-2
25           td casilla 3-3
```

tabla básica html

Titulo 1	Titulo 2	Titulo 3
casilla 1-1	casilla 1-2	casilla 1-3
casilla 2-1	casilla 2-2	casilla 2-3
casilla 3-1	casilla 3-2	casilla 3-3

- Tabla básica bootstrap (**.table**)

titulo 1	titulo 2	titulo 3
casilla 1-1	casilla 1-2	casilla 1-3
casilla 2-1	casilla 2-2	casilla 2-3
casilla 3-1	casilla 3-2	casilla 3-3

- Tabla de colores alternados (**.table.table-striped**)

Titulo 1	Titulo 2	Titulo 3
Elemento 1,1	Elemento 1,2	Elemento 1,3
Elemento 2,1	Elemento 2,2	Elemento 2,3
Elemento 3,1	Elemento 3,2	Elemento 3,3
Elemento 4,1	Elemento 4,2	Elemento 4,3
Elemento 5,1	Elemento 5,2	Elemento 5,3

titulo 1	titulo 2	titulo 3
casilla 1-1	casilla 1-2	casilla 1-3
casilla 2-1	casilla 2-2	casilla 2-3
casilla 3-1	casilla 3-2	casilla 3-3

- Tabla oscura (**.table.table-dark**)

titulo 1	titulo 2	titulo 3
casilla 1-1	casilla 1-2	casilla 1-3
casilla 2-1	casilla 2-2	casilla 2-3
casilla 3-1	casilla 3-2	casilla 3-3

- Tabla con bordes (**.table.table-bordered**)

titulo 1	titulo 2	titulo 3
casilla 1-1	casilla 1-2	casilla 1-3
casilla 2-1	casilla 2-2	casilla 2-3
casilla 3-1	casilla 3-2	casilla 3-3

- Tabla condensada (**.table.table-sm**)

Útil a la hora de mostrar tablas con mucho contenido

titulo 1	titulo 2	titulo 3
casilla 1-1	casilla 1-2	casilla 1-3
casilla 2-1	casilla 2-2	casilla 2-3
casilla 3-1	casilla 3-2	casilla 3-3

5.2 – Estilos aplicados a filas y columnas

- Efecto **hover**. Este efecto ilumina la fila por donde está pasando el puntero. Para aplicar este estilo es necesario agregar la referencia al script con las funciones de javascript. (**.table.tabe-hover**)

- Estilos de cabecera. Disponemos las clases **thead-light** y **thead-dark** para aplicar a la etiqueta **thead** de una tabla.

thead-light

titulo 1	titulo 2	titulo 3
casilla 1-1	casilla 1-2	casilla 1-3
casilla 2-1	casilla 2-2	casilla 2-3

thead-dark

titulo 1	titulo 2	titulo 3
casilla 1-1	casilla 1-2	casilla 1-3
casilla 2-1	casilla 2-2	casilla 2-3
casilla 3-1	casilla 3-2	casilla 3-3

- Clases contextuales. Son clases que se pueden aplicar tanto a las etiquetas **<tr>** como **<td>**. Sirven para indicar estados relevantes de manera consistente dentro del sitio y con otros sitios que utilicen bootstrap, lo que mejora la experiencia de usuario.

Las 10 clases son:

clase	apariencia
table-danger"	peligro
table-warning"	advertencia
table-info"	información
table-success"	operación exitosa
table-primary"	Elemento importante
table-secondary"	Elemento secundario
table-dark"	estilo oscuro
table-light"	estilo claro
table-active"	elemento activo

NOTA: Los colores de las clases contestuales aplican en muchas características de bootstrap, como tipografía, colores de fondo y botones.

6 – Formularios

Bootstrap 4 requiere una serie de clases para la implementación de un formulario vertical. A continuación veremos los principales elementos de un formulario en html y su comparativa, tanto en código como en la vista, al utilizar bootstrap.

*Es importante aclarar que en imágenes no se nota, pero una diferencia importante es que mientras en html el tamaño de los elementos es fijo, con bootstrap este se adapta al tamaño de la pantalla.

6.1 – Text, password

Implementamos una página de login como ejemplo.

Los controles de tipo input deben encerrarse en un `<div class= "form-group">` el cual encierra tanto el **input** como el **label**, y el control debe ser de clase **"form-control"**, en caso de ser un input de tipo **file** la clase debe ser **"form-control-file"**.

El botón de momento tendrá solo las clases **"btn btn-primary"**. Mas adelante se ven mas en detalle las propiedades de botones que maneja bootstrap.

Html

```
h2 Formulario html
form(action="#")
  h4 inputs
  p
  | nombre:
  input#nombre(type="text", name="nombre")
  p
  | contraseña:
  input#pass(type="password", name="pass")
  br
  br
  button ingresar
```

Bootstrap

```
h4 formulario bootstrap
form(action="#")
  .form-group
  | label(for="nombre") nombre
  | input#nombre.form-control(type="text" name="nombre")
  .form-group
  | label(for="pass") contraseña
  | input#pass.form-control(type="password", name="pass")
  button.btn.btn-primary(type="submit") enviar
```

Html

inputs

nombre:

contraseña:

bootstrap

nombre

contraseña

Como podemos ver los dos controles input ocupan el 100% del contenedor, recordemos que están pensado primero que se vean bien en un dispositivo móvil, luego podemos disponerlos en una columna de una fila y darle distintos tamaños según el ancho del dispositivo. Por todo el código anterior se puede poner en una columna y que esta ocupe solo la mitad del tamaño de la fila.

```
30 .row.justify-content-center  
31 .col-6  
32 > form(action="#") ...
```

formulario centrado en un row

nombre

contraseña

6.2 – textarea

Html

```
p mensaje  
textarea#msg(name="msg", cols="50", rows="5")
```

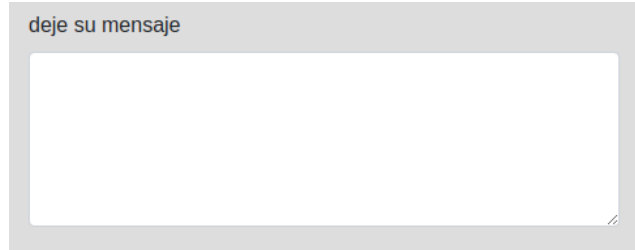
bootstrap

```
.form-group  
  label(for="msg") deje su mensaje  
  textarea#msg.form-control(name="", cols="50", rows="5")
```

Html

A simple HTML form consisting of a light gray rectangular box. Inside the box, at the top left, is the text "mensaje". Below the text is a large, empty white rectangular input area.

Bootstrap

A Bootstrap-style form. It features a light gray border with a rounded top-left corner. Inside, at the top left, is the text "deje su mensaje". Below this text is a large, empty white rectangular input area with a rounded bottom-right corner.

6.3 – Checkbox y radio

Los controles checkbox y radio pueden ordenarse en una columna, poniendo cada control en un `<div class="form-check">`, o en una fila con la clase `<div class="form-check form-check-inline">`, y se declaran dentro de un elemento `<input class="form-check-input"...>` con el atributo **type** que corresponda (checkbox o radio).

Recordemos que los controles checkbox pueden seleccionarse varios, mientras que los radio deben tener todos el mismo atributo **name** dentro de un mismo formulario para que solo se pueda seleccionar uno de ellos.

Html

```
h4 ceckbox y radio en html
p checkbox
  br
  input#ch1(type="checkbox" name="ch1")
  | opcion 1
  br
  input#ch2(type="checkbox" name="ch2")
  | opcion 2
p radio
  br
  input#rad1(type="radio" name="g1")
  | opcion 1
  br
  input#rad2(type="radio" name="g1")
  | opcion 2
```

La sintaxis de los elementos checkbox y radio en **bootstrap** es la siguiente.

```

h4 checkbox y radio en bootstrap
p checkbox
.form-check
  input#ch1.form-check-input(type="checkbox" name="ch1")
  label.form-check-label(for="ch1") opcion1
.form-check
  input#ch2.form-check-input(type="checkbox" name="ch2")
  label.form-check-label(for="ch1") opcion2
br
p radio
.form-check
  input#ch1.form-check-input(type="radio" name="ch1")
  label.form-check-label(for="ch1") opcion1
.form-check
  input#ch2.form-check-input(type="radio" name="ch1")
  label.form-check-label(for="ch1") opcion2

```

```

p checkbox alineados
.form-check.form-check-inline
  input#ch1.form-check-input(type="checkbox" name="ch1")
  label.form-check-label(for="ch1") opcion1
.form-check.form-check-inline
  input#ch2.form-check-input(type="checkbox" name="ch2")
  label.form-check-label(for="ch1") opcion2
.form-check.form-check-inline
  input#ch3.form-check-input(type="checkbox" name="ch3")
  label.form-check-label(for="ch1") opcion3

```

Html

checkbox y radio en html

checkbox

- ☐ opcion 1
☐ opcion 2

radio

- ☐ opcion 1
☐ opcion 2

Bootstrap

checkbox y radio en bootstrap

checkbox

- ☐ opcion1
☐ opcion2
☐ opcion3

radio

- ☐ opcion1
☐ opcion2
☐ opcion3

checkbox alineados

- ☐ opcion1 ☐ opcion2 ☐ opcion3

6.4 – Select

Para aplicar Bootstrap 4 a un control select debemos fijar la clase **"form-control"** al select respectivo y también debemos encerrarlo con un div con la clase **"form-group"**.

La sintaxis de clases aplicadas es similar a las etiquetas "input" que vimos anteriormente.

Html

```
h4 select
p seleccion simple
select#sel(name="sel")
  option(value="op1") opcion 1
  option(value="op2") opcion 2

p seleccion multiple
select#sel(multiple name="sel")
  option(value="op1") opcion 1
  option(value="op2") opcion 2
  option(value="op3") opcion 3
  option(value="op4") opcion 4
```

Bootstrap

```
// select
p select simple
.form-group
  label(for="sel1") Seleccionar un elemento
  select.form-control(name="sel1")
    option(value="op1") opcion1
    option(value="op2") opcion2
br
p select multiple
.form-group
  label(for="sel1") Seleccionar varios elementos
  select.form-control(multiple name="sel1")
    option(value="op1") opcion1
    option(value="op2") opcion2
    option(value="op3") opcion3
    option(value="op4") opcion4
```

Html

seleccion simple

opcion 1 ▼

seleccion multiple

opcion 1 ▲
opcion 2
opcion 3
opcion 4 ▼

Bootstrap

Seleccionar un elemento

opcion1 ▼

Seleccionar varios elementos

opcion1
opcion2
opcion3
opcion4 ▼

7 – tipografías, posiciones y espaciados

Para el control del tamaño y posicionamiento de todo lo que es texto dentro del documento bootstrap posee una gran cantidad de clases predefinidas que evitan tener que agregar estilos css.

7.1 – Tipografía

Las siguientes clases modifican el tamaño de las etiquetas de tipo texto, ya sea la etiqueta **p** o todos los encabezados **h1, h2, h3, h4, h5, h6**.

- Agregando las clases **h...** (**del 1 al 6**) se puede modificar cualquier etiqueta para que se vea como el encabezado indicado.
- Para encabezados aun mas grandes se puede usar las clases **display-<n>**, con n entre 1 y 4.
- Para resaltar un texto con una tipografía mas grande se agrega a la etiqueta **p** la clase **lead**.

```
h6.h1 h6 class h1
h5.h2 h5 class h2
h4.h3 h4 class h3
h3.h4 h3 class h4
h2.h5 h2 class h5
h1.h6 h1 class h6
p.h1 p class h1
```

h6 class h1

h5 class h2

h4 class h3

h3 class h4

h2 class h5

h1 class h6

p class h1

```
h1 h1 común
h1.display-1 display 1
h1.display-2 display 2
h1.display-3 display 3
h1.display-4 display 4
```

h1 común

display 1

display 2

display 3

display 4

Para modificaciones de párrafos o partes del mismo (dentro de un span) existen las siguientes clases

- **text-muted**: Cambia el color de fuente a gris
- **font-weight-bold**: negrita
- **font-italic**: cursiva
- **font-weight-normal**: texto normal
- **text-uppercase**: Pasa todo el texto a mayúsculas
- **text-lowercase**: Pasa todo el texto a minúsculas
- **text-capitalize**: Pone mayúsculas en la primer letra de cada palabra.

```
p texto sencillo
p.lead texto con clase lead
p texto con parte del mismo
|   span.text-muted mutedado
p.font-weight-bold texto en negrita
p.font-italic texto en cursiva
p.text-uppercase mayúsculas
p.text-lowercase MINUSCULAS
p.text-capitalize primeras letras en mayúsculas
```

texto sencillo

texto con clase lead

texto con parte del mismo mutedado

texto en negrita

texto en cursiva

MAYÚSCULAS

minusculas

Primeras Letras En Mayúsculas

7.2 – Alineaciones

El texto en pantalla se puede alinear horizontalmente a izquierda, centro, derecha o justificado con las clases

- **text-justify**
- **text-left**
- **text-center**
- **text-right**

```
h4 texto justificado
p.text-justify Lorem, ipsum dolor sit amet consectetur adipisicing elit. Accusantium
ratione vero illo esse a amet. Aut repellendus necessitatibus minima dolor, distinctio
laboriosam nihil amet tenetur minus at earum placeat quod! Lorem ipsum dolor sit amet
consectetur adipisicing elit. Quidem dignissimos eveniet reprehenderit qui quasi enim
nostrum cum porro earum veritatis vel, commodi eum, odit minima ab inventore ex voluptas
Eaque.
br

h4 alineaciones horizontales fijas
p.text-left texto a izquierda
p.text-center texto centrado
p.text-right texto a la derecha
```

texto justificado

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Accusantium ratione vero illo esse a amet. Aut repellendus necessitatibus minima dolor, distinctio laboriosam nihil amet tenetur minus at earum placeat quod! Lorem ipsum dolor sit amet consectetur adipisicing elit. Quidem dignissimos eveniet reprehenderit qui quasi enim nostrum cum porro earum veritatis vel, commodi eum, odit minima ab inventore ex voluptas. Eaque.

alineaciones horizontales fijas

texto a izquierda

texto centrado

texto a la derecha

Las alineaciones izquierda derecha y centro también se pueden ajustar al tamaño de la pantalla agregando el indicador de tamaño. La clase genería es **text-<tamaño>-<alineacion>**, donde tamaño puede ser **sm, md, lg y xl**, y la alineación **left, center** o **right**.

El texto también puede alinearse de manera vertical dentro de un mismo bloque para esto las clases son

- **align-baseline**
- **align-top**
- **aling-bottom**
- **align-middle**

- align-text-top
- align-text-bottom

```
span.align-baseline texto
span.align-top texto
span.align-bottom texto
span.align-middle texto
span.align-text-top texto
span.align-text-bottom texto
```

alineaciones verticales

texto texto texto texto texto texto

7.3 – Elementos en línea y bloque

Con bootstrap se puede alterar el comportamiento de los elementos en línea (p, span, etc) y de tipo bloque (h1, div, etc) con las clases.

- **d-inline.** Hace que un elemento de tipo bloque se comporte como elemento en línea
- **d-block.** Hace que un elemento en línea se comporte como bloque
- **d-inline-block.** UN elemento de tipo bloque conserva sus propiedades pero ya no en todo el ancho de la pantalla sino que se ajusta según su contenido

```
h3.bbb encabezado tipo bloque
h3.ccc.d-inline encabezado en línea 1
h4.ddd.d-inline encabezado en línea 2
br
span.bbb texto sencillo en línea
span.ccc otro texto en línea
span.ddd.d-block texto en tipo bloque
br
.d-inline-block.ccc
  h4 div en línea
  p con propiedades de bloque
```

encabezado tipo bloque

encabezado en línea 1 encabezado en línea 2

texto sencillo en línea otro texto en línea

texto en tipo bloque

div en línea

con propiedades de bloque

7.4 – márgenes y rellenos

Bootstrap ofrece unas clases por defecto para agregar márgenes y rellenos sin necesidad de crear estilos css. Para esto se usa una clase genérica **<tipo><lado>-<medida>**

El **tipo** puede ser

- **m**: márgenes (margin)
- **p**: relleno (padding)

El **lado**

- **t**: arriba
- **b**: abajo
- **l**: a la izquierda
- **r**: a la derecha
- **x**: a izquierda y derecha
- **y**: arriba y abajo
- si no se agrega nada es a los cuatro costados

La **medida** se encuentra predefinida en la implementación de bootstrap. Pueden ser valores entre **0** y **5**, o el valor **auto** (No son medidas en px o em). Estos valores si bien pueden modificarse en los archivos de origen, suelen ser los que quedan mas estéticos para cada tamaño de página así que son los mas usados.

* para elementos de tipo bloque los únicos que tienen efecto notorio son los valores 5, 4 y 3. Los demás valores aplican generalmente a palabras o elementos dentro de una línea.

REPASO. La **diferencia entre margin y padding** es que el primero establece una separación con otros elementos por fuera del borde del contenedor, mientras que el segundo agrega relleno dentro de los bordes del mismo.

7.5 – Alto y ancho

Por último vemos cómo hacer que los elementos de tipo bloque ocupen un ancho o alto definido dentro de la pantalla, para eso la clase a utilizar es **<dimension>-<porcentaje>**

La **dimensión** puede ser

- **w**: ancho
- **h**: alto

El porcentaje indica cuánto del contenedor va a ocupar.

```
h4 ancho
div
  .p-3.w-25.aaa ancho 25%
  .p-3.w-50.bbb ancho 50%
  .p-3.w-75.ccc ancho 75%
  .p-3.w-100.ddd ancho 100%
```

ancho

ancho 25%

ancho 50%

ancho 75%

ancho 100%

```
h4 alto
.row.justify-content-around.bg-light(style="height: 200px")
  .d-inline-block.h-25.aaa.p-4 alto 25%
  .d-inline-block.h-50.bbb.p-4 alto 50%
  .d-inline-block.h-75.ccc.p-4 alto 75%
  .d-inline-block.h-100.ddd.p-4 alto 100%
```

alto

alto 25%

alto 50%

alto 75%

alto 100%

8 – Breakpoints

Teniendo en cuenta que bootstrap define el comportamiento responsive a partir de la medida del ancho de la pantalla, usando **media queries** de css se puede agregar mas comportamiento mediante breakpoints usando estas medidas.

El ejemplo mas usual es cambiar el color de fondo según un ancho determinado. Vamos a hacerlo usando el media query de tipo **screen**, con la característica **max-width**. Este indica que hasta cierto ancho determinado de pantalla deben aplicarse los estilos declarados dentro del mismo.

Recordemos que los anchos de pantalla con los que trabaja bootstrap son

- xl: 1200 px
- lg: 992px
- md: 720px
- sm: 540px

Entonces estos son los que vamos a usar como breakpoints. En estilos.css agregamos los siguientes media queries.

* Es importante el orden en que se declaran. Si vamos a usar max-width hay que declarar los anchos de mayor a menor.

```
body{
  background-color: forestgreen;
}

@media screen and (max-width: 1200px){
  body{
    background-color: darkslategrey;
  }
}

@media screen and (max-width: 992px){
  body{
    background-color: darkolivegreen;
  }
}

@media screen and (max-width: 720px){
  body{
    background-color: darkcyan;
  }
}

@media screen and (max-width: 540px){
  body{
    background-color: darkgreen;
  }
}
```

Con estos anchos el fondo de pantalla va cambiando de color según se modifica el tamaño.

9 – Botones

Las clases de botones en bootstrap pueden clasificarse según su funcionalidad en

- colores
- Tamaños
- Agrupaciones

La clase base para un botón de bootstrap es **btn**, y tanto esta como todas las que le modifican los estilos pueden ser agregadas a las etiquetas **button**, **input** y **a**.

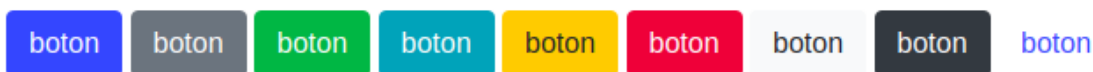
NOTA: En este punto es necesario agregar los script de jquery y bootstrap al documento para aplicar los efectos de los botones según su estado.

9.1 – Colores

Los colores básicos de los botones son los mismos que las clases contextuales

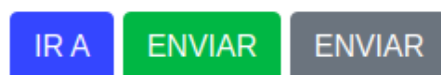
```
button.btn boton  
br  
button.btn.btn-primary boton  
button.btn.btn-secondary boton  
button.btn.btn-success boton  
button.btn.btn-info boton  
button.btn.btn-warning boton  
button.btn.btn-danger boton  
button.btn.btn-light boton  
button.btn.btn-dark boton  
button.btn.btn-link boton
```

boton



Los casos particulares son los enlaces **a** y los **inputs** dentro de un formulario

```
a.btn.btn-primary(href="#") IR A  
button.btn.btn-success(type="submit") ENVIAR  
input.btn.btn-secondary(type="button" value="ENVIAR")
```



Con las clases contextuales también se puede agregar botones en negativo que solo se colorean con el efecto hover agregando la clase **btn-outline-<color>**

```
button.btn.btn-outline-primary boton  
button.btn.btn-outline-secondary boton  
button.btn.btn-outline-success boton  
button.btn.btn-outline-info boton  
button.btn.btn-outline-warning boton  
button.btn.btn-outline-danger boton  
button.btn.btn-outline-light boton  
button.btn.btn-outline-dark boton  
button.btn.btn-outline-link boton
```



9.2 – Tamaños

Algunas clases para modificar el tamaño de los botones son

- `btn-lg`. Boton grande
- `btn-sm`. Boton chico
- `btn-block`. El boton ocupa todo el ancho de pantalla

```
button.btn.btn-primary.btn-lg grande  
button.btn.btn-primary normal  
button.btn.btn-primary.btn-sm chico  
br  
br  
button.btn.btn-primary.btn-block boton en bloque
```

grande

normal

chico

boton en bloque

Font awesome

Una librería muy usada por la amplia variedad de íconos que posee es font awesome. Esta puede agregarse usando la cdn disponible en esta página.


<https://www.bootstrapcdn.com/fontawesome/>

Y en esta otra ver el listado de íconos disponibles de manera gratuita.

<https://fontawesome.com/icons?d=gallery&c=business&m=free>

Estos íconos pueden usarse en diversas partes de la página, como bloques, navegaciones y botones. Si por ejemplo queremos agregar a un boton de login un icono de usuario se agrega la etiqueta `i` con las clases `.fa.fa-user`.

```
button.btn.btn-primary.btn-lg  
| i.fa.fa-user  
| login
```

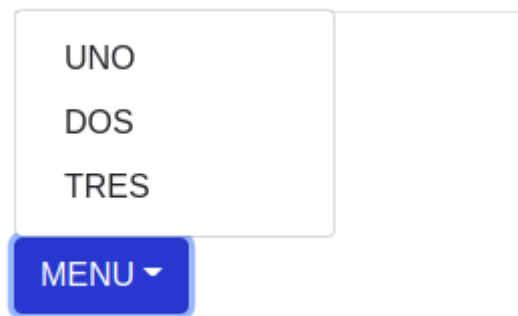
 login

9.3 – Agrupaciones

Las agrupaciones suelen hacerse mediante dos métodos, los dropdowns y los grupos de botones

Para hacer un menú desplegable o **dropdown** se define un div y dentro del mismo un boton con el atributo data-toggle= "dropdown" y otro div que contiene por ejemplo una lista de enlaces.

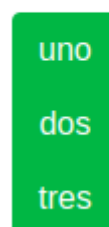
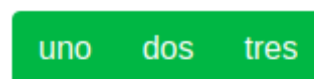
```
.dropdown
  button.btn.btn-primary.dropdown-toggle(data-toggle="dropdown") MENU
  .dropdown-menu
    a.dropdown-item(href="#") UNO
    a.dropdown-item(href="#") DOS
    a.dropdown-item(href="#") TRES
```



Los grupos de botones pueden ser de dos tipo, btn-group, que los muestra como si fueran un solo boton, y btn-toolbar, que los muestra juntos pegados pero se los puede separar agregando margenes.

El grupo de botones puede ser tanto horizontal como vertical.

```
hr
.btn-group
  button.btn.btn-success uno
  button.btn.btn-success dos
  button.btn.btn-success tres
hr
.btn-group-vertical
  button.btn.btn-success uno
  button.btn.btn-success dos
  button.btn.btn-success tres
```

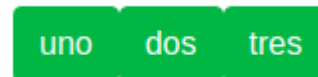


También puede combinarse esta agrupación con los dropdowns definiendo cual es el botón que despliega el menú.

```
.btn-group
  button.btn.btn-success boton
  button.btn.btn-success.dropdown-toggle.dropdown-toggle-split(data-toggle="dropdown") menu
.dropdown-menu
  a.dropdown-item(href="#") UNO
  a.dropdown-item(href="#") DOS
  a.dropdown-item(href="#") TRES
```

Por ultimo, otro tipo de agrupación es el **toolbar**, vemos que si bien la función es casi la misma, la estática es algo distinta.

```
.btn-toolbar(role="toolbar")
  button.btn.btn-success uno
  button.btn.btn-success dos
  button.btn.btn-success tres
```



Dentro del toolbar también se pueden definir grupos de botones y separarlos mediante márgenes. En este caso el toolbar tiene dos grupos de botones, y uno aislado.

```
.btn-toolbar(role="toolbar")
  .btn-group.mr-2
    button.btn.btn-primary 1
    button.btn.btn-primary 2
    button.btn.btn-primary 3

  .btn-group.mr-2
    button.btn.btn-primary 4
    button.btn.btn-primary 5

  button.btn.btn-primary 6
```



10 – Listas

Para aplicar estilos a las listas bootstrap dispone de la clase `list-group`, que se puede aplicar a los elementos **ul**, **ol** y **div**.

Dentro de la lista los items pueden ser elementos **li** o **a** con la clase **list-group-item**.

```
ul.list-group
  li.list-group-item elemento 1
  li.list-group-item elemento 2
```

elemento 1

elemento 2

Ademas también se puede agregar la clase **list-group-item-<color>**, donde color puede ser cualquiera de las clases contextuales de bootstrap (primary, succes, danger, etc).

```
ul.list-group
  a.list-group-item.list-group-item-primary(href="#") enlace 1
  a.list-group-item.list-group-item-secondary(href="#") enlace 2
  a.list-group-item.list-group-item-success(href="#") enlace 3
  a.list-group-item.list-group-item-danger(href="#") enlace 4
  a.list-group-item.list-group-item-warning(href="#") enlace 5
  a.list-group-item.list-group-item-info(href="#") enlace 6
  a.list-group-item.list-group-item-light(href="#") enlace 7
  a.list-group-item.list-group-item-dark(href="#") enlace 8
```

enlace 1
enlace 2
enlace 3
enlace 4
enlace 5
enlace 6
enlace 7
enlace 8

Breadcrumbs

Los breadcrumbs (migas de pan) son un tipo de lista ordenada que se suelen utilizar para seguir un trazado de las rutas por las que se navega en una página. Por ejemplo al entrar a una tienda online sería.

```
ol.breadcrumb
  li.breadcrumb-item inicio
  li.breadcrumb-item categoria
  li.breadcrumb-item subcategoría
  li.breadcrumb-item.active Artículo
```

inicio / categoria / subcategoría / Artículo

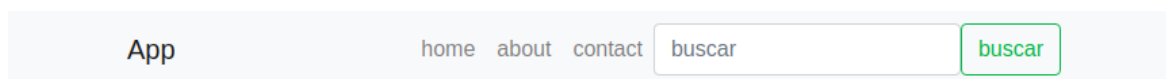
11 – Navegación

Un elemento asociado a las listas y muy utilizado en las páginas web es la navegación. En esta página están explicados los ejemplos mas comunes

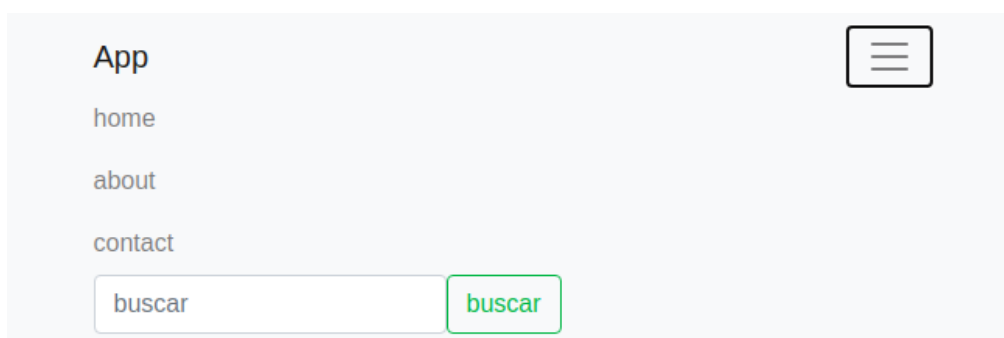
<https://getbootstrap.com/docs/4.0/components/navbar/>

Vamos a analizar el código de la siguiente navegación, cuyo menu colapsa al estar en una pantalla de tamaño menor al tamaño md(720px).

En pantalla md



En pantalla sm



El código para esta navegación es el siguiente

```
nav.navbar.navbar-expand-md.navbar-light.bg-light
.container
  a.navbar-brand(href="#") App
  button.navbar-toggler(type="button" data-toggle="collapse"
data-target="#nav1")
    span.navbar-toggler-icon

  #nav1.navbar-collapse.collapse
    ul.navbar-nav.ml-auto
      li.nav-item
        a.nav-link(href="#") home
      li.nav-item
        a.nav-link(href="#") about
      li.nav-item
        a.nav-link(href="#") contact
    form.form-inline
      input.form-control(type="text" placeholder="buscar")
      button.btn.btn-outline-success(type="button") buscar
```

Una navegación se declara con la etiqueta **nav**, y la clase **navbar** para preparar el contenido, luego se agrega la clase **navbar-expand-<tamaño>** para indicar cuándo colapsa y las clases **navbar-<estilo>** y **bg-<estilo>** para los estilos de tipografía y de fondo.

Todo el contenido de la navegación se ubica en un div class **container** para tener algo de márgenes con el borde de la pantalla, y dentro del mismo se declara

- La marca de la aplicación
- un botón para cuando el menú colapsa
- y un div que contiene el menú

La **marca** es usualmente un enlace con el nombre de la aplicación y lleva la clase **navbar-brand** para distinguirse del resto de los elementos.

El **botón** lleva la clase **navbar-toggler** y el atributo **data-toggle="collapse"** para indicar que aparece o se muestra según el tamaño de pantalla indicado en la etiqueta nav. También tiene una referencia al id del div (**data-target= "#nav1"**) que debe mostrar al ser presionado. Por último se agrega un icono para el botón con el elemento span de clase **navbar-toggler-icon**.

El **menú** requiere de un div con **id** y las clases **navbar-collapse** (para que colapse) y **collapse** (para que se oculte automáticamente al colapsar). Dentro del mismo por ahora ponemos una lista con los enlaces a las distintas partes de una página (home, about, contact, etc) y un formulario de búsqueda.

La lista (ul) tiene las clases **navbar-nav** (indica que contiene elementos de navegación) y **ml-auto** (margin left: coloca todos los elementos hacia la derecha), mientras los elementos de la misma (li) tienen la clase **nav-item** y encierran un enlace (a) de clase **nav-link**.

Esta es una de las navegaciones más comunes. Una variante que se suele encontrar es en lugar de una barra de búsqueda que tenga un menú desplegable, para hacer esto usamos la clase dropdown vista anteriormente. Para esto agregamos a uno de los elementos de la lista (li) la clase **dropdown**, y esta contiene además de un enlace (a) padre que ahora además tiene la clase **dropdown-toggle** y el atributo **data-toggle="dropdown"**, y un div de clase **dropdown-menu** con los enlaces correspondientes al menú desplegable, cada uno de estos con la clase **dropdown-item**.

```
li.nav-item.dropdown
  a.nav-link.dropdown-toggle(href="#" data-toggle="dropdown") Item
  padre
  .dropdown-menu
    a.dropdown-item(href="#") link1
    a.dropdown-item(href="#") link2
    a.dropdown-item(href="#") link3
```

12 – Ventanas modales

Bootstrap también facilita la creación de modales. Estos son carteles que se ponen en primer plano de la pantalla para destacar algún mensaje, mostrar formularios, etc.

Es importante tener en cuenta que para que los modales funcionen necesitan la funcionalidad de jQuery.

Ahora, una ventana modal en bootstrap tienen un conjunto de etiquetas que pre definen su estructura. Las mas básicas son divs con las siguientes clases

```
#modal1.modal.fade(tabindex="-1")
  .modal-dialog
    .modal-content
      .modal-header Titulo del modal
      .modal-body Cuerpo del modal
      .modal-footer Pie del modal
```

La etiqueta modal requiere un id, ya que es al que se va a hacer referencia desde el botón que ejecuta la ventana.

```
button.btn.btn-primary(type="button" data-toggle="modal" data-target="#modal1") Mostrar modal
```

También requiere el atributo **tabindex** en **-1** para que esta se cierre al hacer click fuera de la ventana, y por último agregamos la clase **fade** para que tenga una animación al aparecer.

Ahora pasamos a completar los atributos del resto de las etiquetas, una buena práctica de programación es agregar atributos **role** y **aria**, que mejoran la accesibilidad de la página.

```
#modal1.modal.fade(tabindex="-1" role="dialog" aria-labelledby="ejemplo")
  .modal-dialog(role="document")
    .modal-content
      .modal-header
        h5.modal-title#ejemploModalLabel Titulo del modal
        button.close(type="button" data-dismiss="modal" aria-label="Close") &times;
      .modal-body
        h6 contenido del cuerpo del modal
      .modal-footer
        button.btn.btn-primary guardar cambios
        button.btn.btn-danger(data-dismiss="modal") cancelar
```


Como podemos ver, en el encabezado agregamos un título y un botón con una cruz (elemento `×`) con el atributo **data-dismiss= "modal"**, este se encarga de cerrar la ventana al hacer click en ella.

En el pie del modal también agregamos dos botones, uno para una acción de **aceptar**, y otro para **cancelar**.

Dentro del cuerpo del modal por ejemplo puede haber un formulario de contacto.

```
#modal1.modal.fade(tabindex="-1" role="dialog" aria-labelledby="ejemplo")
  .modal-dialog(role="document")
    .modal-content
      .modal-header
        h5.modal-title#ejemploModalLabel Contactanos
        button.close(type="button" data-dismiss="modal" aria-label="Close") &times;

      .modal-body
        form(action="#")
          .form-group
            input.form-control(type="text" placeholder="E-mail")
          .form-group
            textarea#msg.form-control(cols="30" rows="10" placeholder="Mensaje")

      .modal-footer
        button.btn.btn-primary Enviar
        button.btn.btn-danger(data-dismiss="modal") cancelar
```

Con esta estructura, al convertir el archivo a html y abrirlo en el navegador aparece un boton, y al hacer click muestra la ventana modal.

Mostrar modal



Contactanos

E-mail

Mensaje

Enviar

cancelar