* + 1. Guia Bootstrap 4

1. Bootstrap es un framework open source de css desarrollado por Twitter utilizado para facilitar el desarrollo de páginas web adaptables y estilizadas.
   * 1. 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.
2. \*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](#__RefHeading___Toc1693_4153976175)

[2 – Primera página 3](#__RefHeading___Toc1697_4153976175)

[3 - Sistema de grilla 6](#__RefHeading___Toc1699_4153976175)

[3.1 – Columnas de regla múltiple 8](#__RefHeading___Toc1701_4153976175)

[3.2 – Anidamiento de columnas 8](#__RefHeading___Toc1703_4153976175)

[3.3 – Ordenamiento de columnas 9](#__RefHeading___Toc1705_4153976175)

[3.4 – Columnas de ancho automático 9](#__RefHeading___Toc1709_4153976175)

[4 – Alineación de columnas 10](#__RefHeading___Toc1711_4153976175)

[4.1 - Alineación vertical. 11](#__RefHeading___Toc1713_4153976175)

[4.2 - Alineación horizontal 13](#__RefHeading___Toc1715_4153976175)

[4.3 – Desplazamiento de columnas 15](#__RefHeading___Toc1717_4153976175)

[5 – Tablas 17](#__RefHeading___Toc1719_4153976175)

[5.1 – Estilos de tablas 17](#__RefHeading___Toc1721_4153976175)

[5.2 – Estilos aplicados a filas y columnas 19](#__RefHeading___Toc1723_4153976175)

[6 – Formularios 21](#__RefHeading___Toc1725_4153976175)

[6.1 – Text, password 21](#__RefHeading___Toc1727_4153976175)

[6.2 – textarea 22](#__RefHeading___Toc1729_4153976175)

[6.3 – Checkbox y radio 23](#__RefHeading___Toc1731_4153976175)

[6.4 – Select 25](#__RefHeading___Toc1733_4153976175)

[7 – tipografías, posiciones y espaciados 26](#__RefHeading___Toc1735_4153976175)

[7.1 – Tipografía 26](#__RefHeading___Toc1737_4153976175)

[7.2 – Alineaciones 27](#__RefHeading___Toc1739_4153976175)

[7.3 – Elementos en linea y bloque 29](#__RefHeading___Toc1741_4153976175)

[7.4 – margenes y rellenos 30](#__RefHeading___Toc1743_4153976175)

[7.5 – Alto y ancho 30](#__RefHeading___Toc1745_4153976175)

[8 – Breakpoints 31](#__RefHeading___Toc1747_4153976175)

[9 – Botones 32](#__RefHeading___Toc1749_4153976175)

[9.1 – Colores 33](#__RefHeading___Toc1274_3373961720)

[9.2 – Tamaños 34](#__RefHeading___Toc1276_3373961720)

[9.3 – Agrupaciones 35](#__RefHeading___Toc1278_3373961720)

[10 – Listas 36](#__RefHeading___Toc1280_3373961720)

[11 – navegación 38](#__RefHeading___Toc1282_3373961720)

## 1 – Instalación

1. 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:

1. <https://getbootstrap.com/docs/4.3/getting-started/introduction/>
2. 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.
4. 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 Bootstap 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.

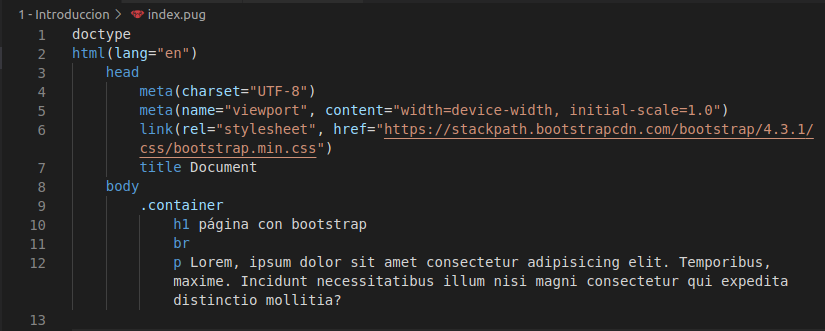
1. Como desventaja podríamos decir:

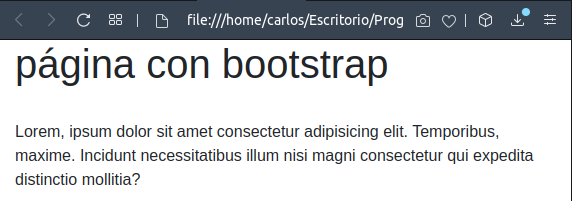
* Está la posibilidad que el servidor donde se aloja Bootstrap 4 se caiga, si bien es mucho más probable que nuestro servidor sobre todo si es un hosting compartido es el que tenga problemas.
* No podemos probar en forma local nuestro sitio web si no se encuentra conectado a internet.

1. 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.
2. <https://getbootstrap.com/docs/4.1/getting-started/download/>

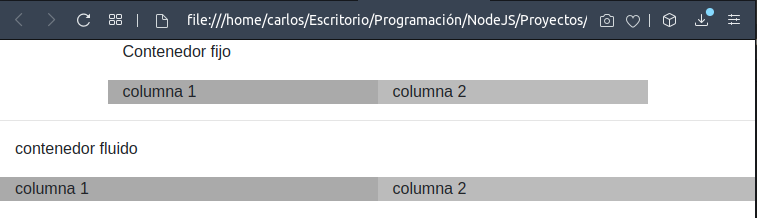
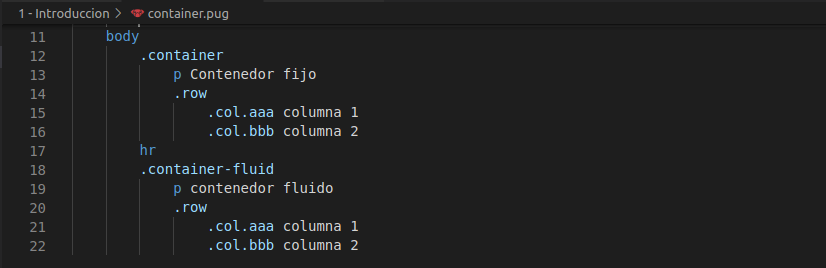
## 2 – Primera página

1. 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.
2. meta(name="viewport", content="width=device-width, initial-scale=1.0")
4. Para ver la diferencia entre html básico y bootstrap creamos una página con un titulo y un párrafo.
5. 
6. Agregando bootstrap y poniendo todo el documento en un contenedor se ve como cambia la familia y color de fuente, y agrega algunos margenes al contenido. Para usar el cdn de bootstrap se agrega en el head la referencia



1. La página ahora tiene este aspecto
2. 
3. 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.

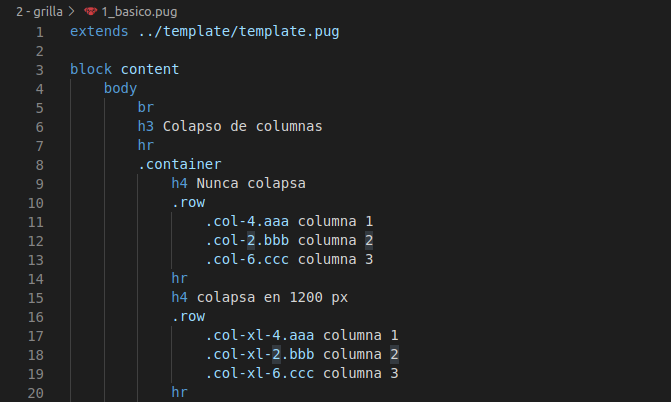
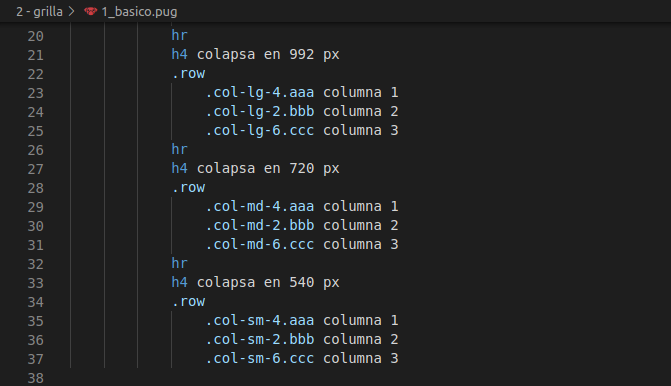
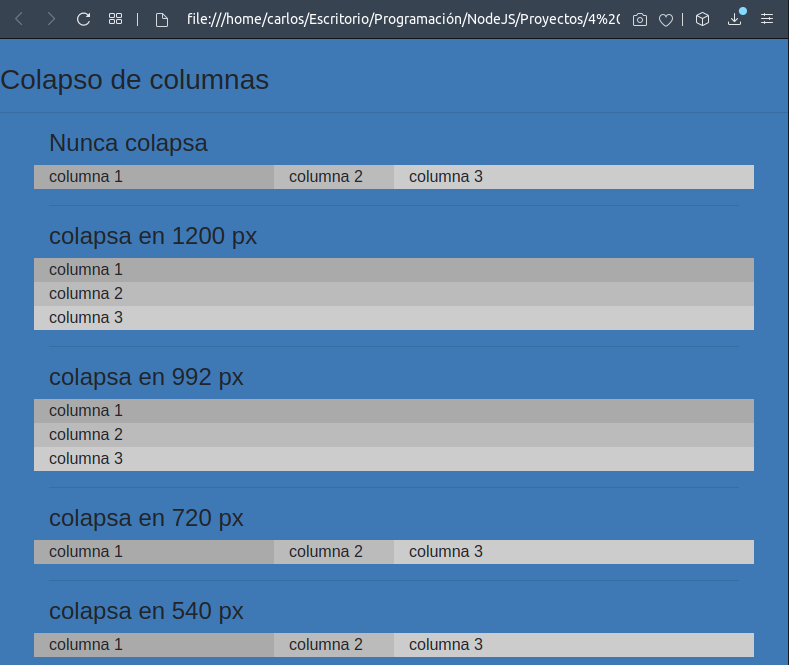
1. \* En el directorio de los ejemplos creamos el archivo css/estilos.css donde vamos agregando algunos estilos de colores de fondo para diferenciar los contenedores, sus posiciones y tamaños. Agregamos entonces en el head también la referencia al css propio.
2. 
3. **estilos.css**
4. 
5. **Container.pug**
6. 



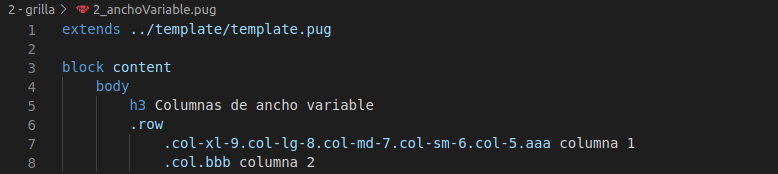
## 3 - Sistema de grilla

1. 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.
2. 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).
3. Cada fila se declara con la clase **row**, y cada columna con la clase **col-\*-\*.**
4. 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).
5. 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.

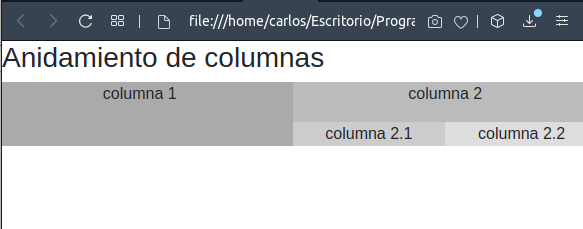
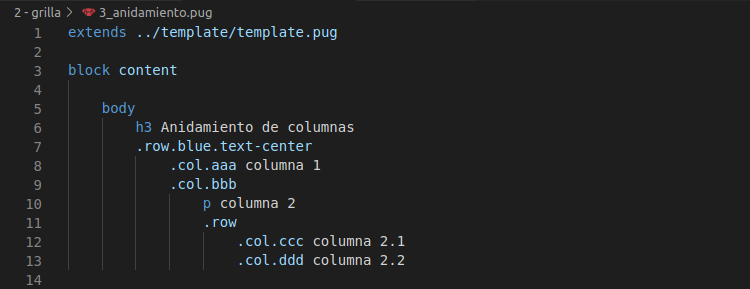
|  |  |  |
| --- | --- | --- |
| **Atributo** | **significado** | **Ancho de pantalla en que colapsa** |
| **xl** | Extra large | < 1200px |
| **lg** | Large | < 992px |
| **md** | Medium | < 720px |
| **sm** | Small | < 540px |

2. ****
3. ****
4. 

### 3.1 – Columnas de regla múltiple

1. 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. 

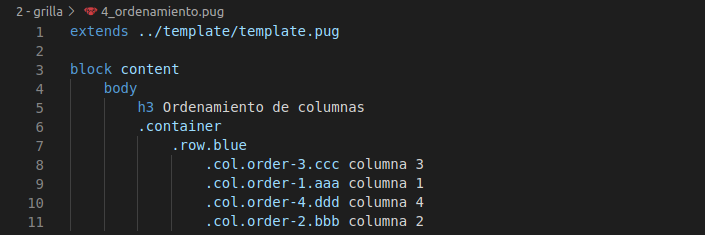
### 3.2 – Anidamiento de columnas

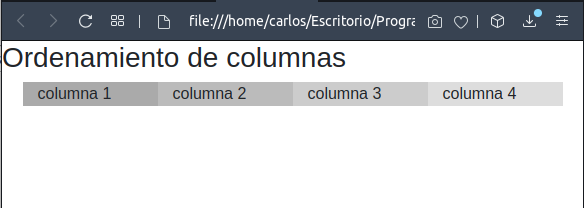
1. También se puede anidar columnas dentro de otras, para eso se declara un **row** dentro de una columna y dentro de este otras columnas. Además al anidar se generan otras 12 unidades dentro de la columna particionada.
2. 

### 3.3 – Ordenamiento de columnas

* + - 1. 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-\***

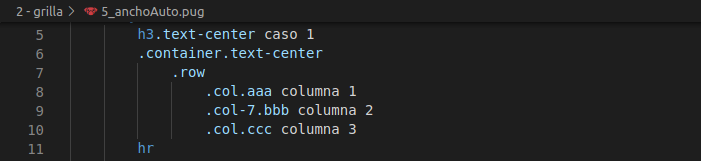




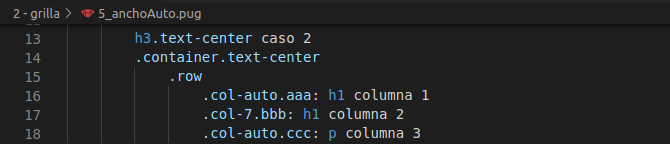
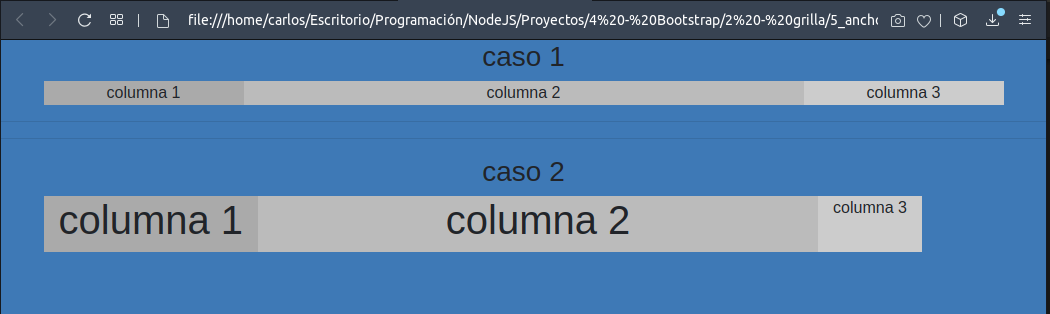
### 3.4 – Columnas de ancho automático

1. 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.



* 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.

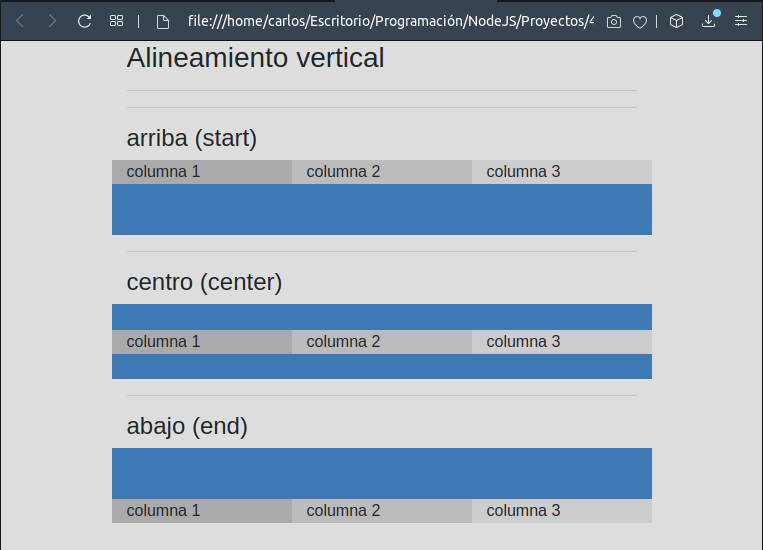
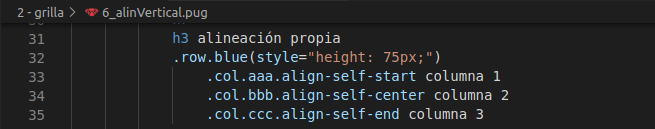
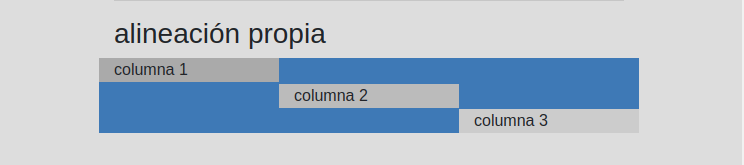
1. 
2. 

## 4 – Alineación de columnas

1. 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
2. \*Todas estas alineaciones pueden practicarse con la página flexbox froggy
3. [https://flexboxfroggy.com/#es](https://flexboxfroggy.com/" \l "es)

### 4.1 - Alineación vertical.

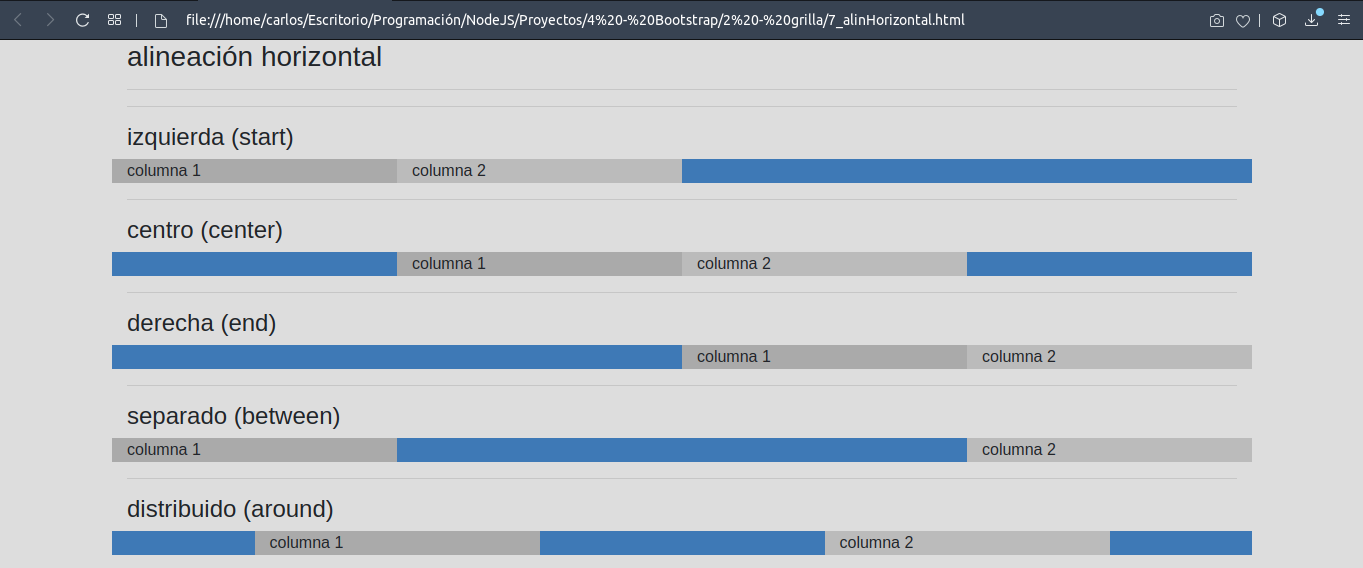
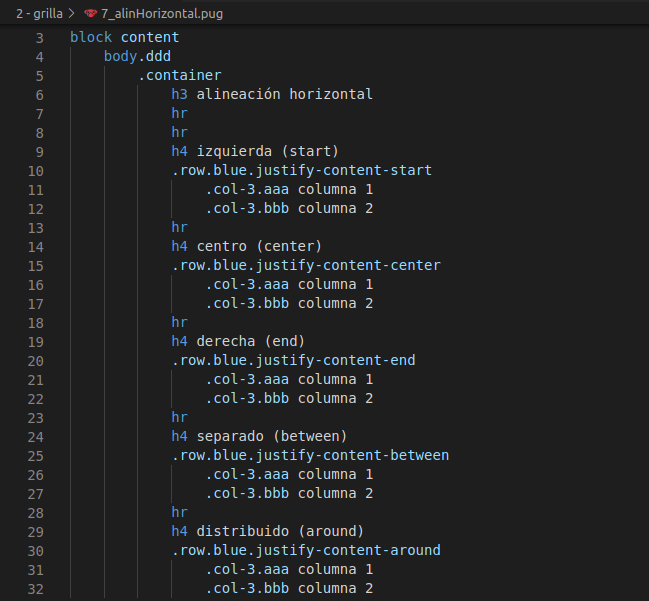
* **align-items-start**. Alinea las columnas en la parte superior de la fila.
* **align-items-center**. Centra verticalmente las columnas.
* **align-items-end**. Alinea las columnas en la parte inferior de la fila.

1. 
2. 
3. También se puede modificar la clase col con **align-self-\*** para que cada columna quede en una posición particular.
4. 

### 4.2 - Alineación horizontal

1. 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.

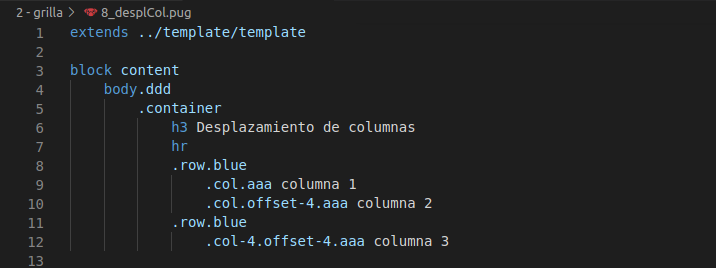
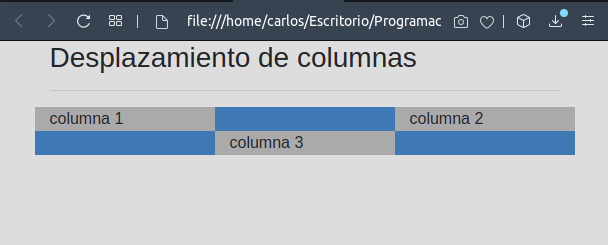
1. 
2. 

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

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. **offset-\*** | 1. **offset-sm-\*** | 1. **offset-md-\*** | 1. **offset-lg-\*** | 1. **offset-xl-\*** |

1. 
2. 

## 5 – Tablas

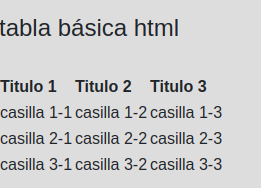
1. Cuando incluimos Bootstrap 4 alse introducen una serie de reglas a los elementos relacionados a la etiqueta html "**table**". El estilo mas básico de tabla en bootstrap se logra añadiendo la clase **table** al elemento table de html.

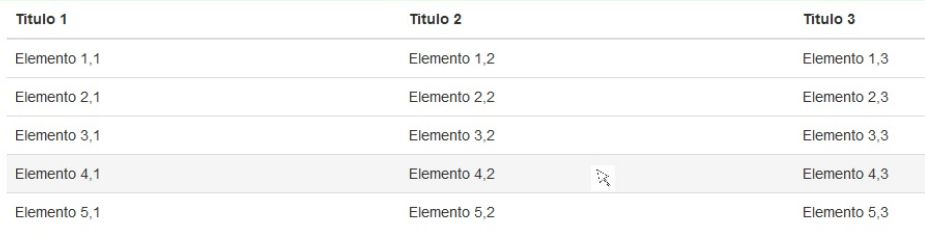
### 5.1 – Estilos de tablas

1. Algunos de los estilos mas comunes de tabla son.

* Tabla básica html

1. 



* Tabla básica bootstap (**.table**)



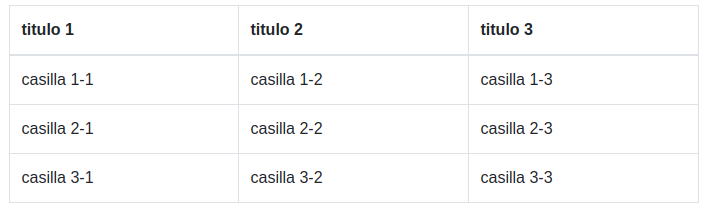
* Tabla de colores alternados (**.table.table-striped**)

1. 

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

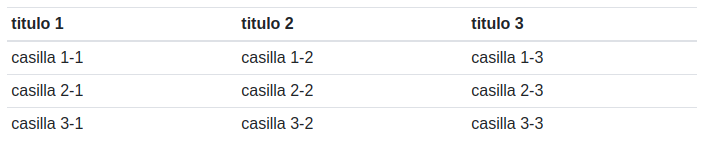
1. 

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

1. 

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

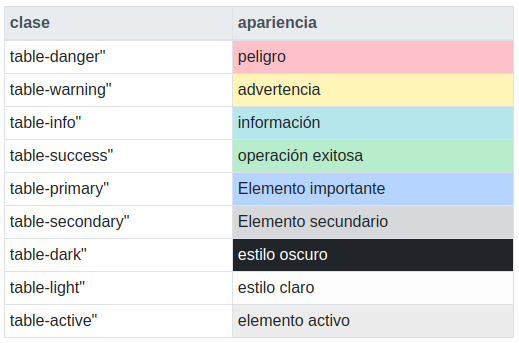
Útil a la hora de mostrar tablas con mucho contenido

1. 

### 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.
* 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:

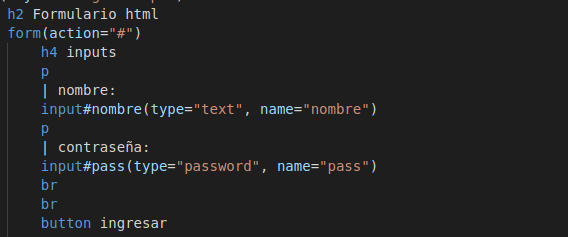
1. 
2. NOTA: Los colores de las clases contestuales aplican en muchas características de bootstrap, como tipografía, colores de fondo y botones.

## 6 – Formularios

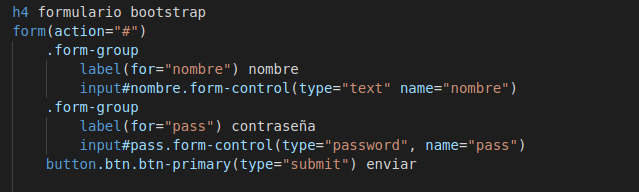
1. Bootstrap 4 requiere una serie de clases para la implementación de un formulario vertical. A continuación veremos los principales elementos de un formulario en html y su comparativa, tanto en código como en la vista, al utilizar bootstrap.
2. \*Es importante aclarar que en imágenes no se nota, pero una diferencia importante es que mientra en html el tamaño de los elementos es fijo, con bootstrap este se adapta al tamaño de la pantalla.

### 6.1 – Text, password

1. Implementamos una página de login como ejemplo.
2. Los controles de tipo input deben encerrarse en un **<div class= “form-group”>** el cuel 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”**.
3. 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.
4. **Html**

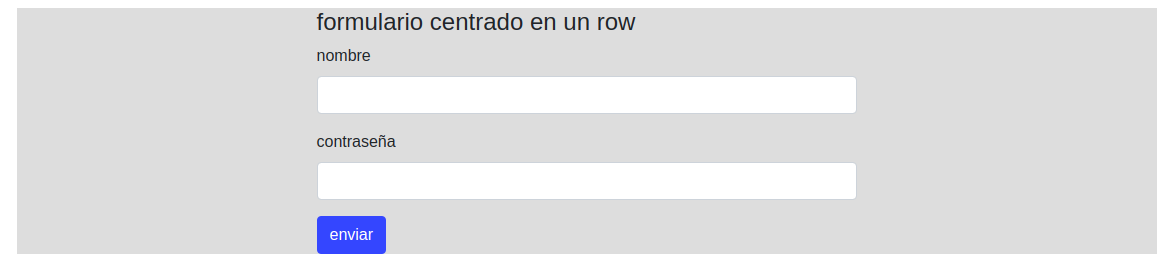
****

1. **Bootstrap**

****

|  |  |  |
| --- | --- | --- |
| **Html** |  | **bootstrap** |
|  |  |  |

1. 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.

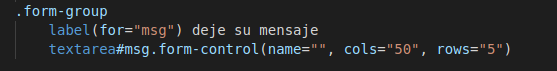


### 6.2 – textarea

1. **Html**



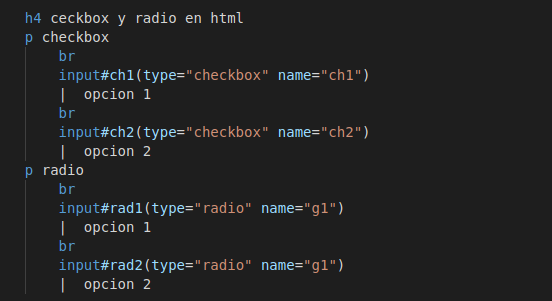
1. **bootstrap**

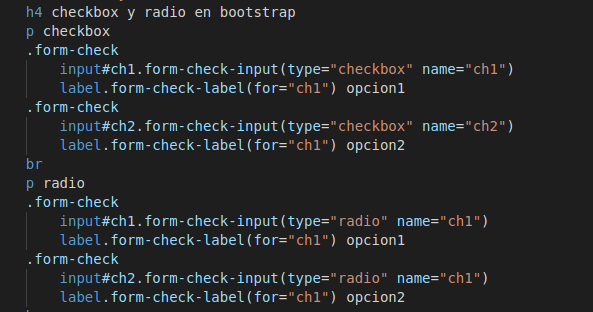


|  |  |
| --- | --- |
| **Html** | **Bootstrap** |
|  |  |

### 6.3 – Checkbox y radio

1. 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).
2. 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.
3. **Html**



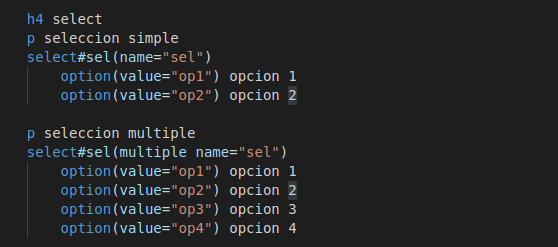
1. La sintaxis de los elementos checkbox y radio en **bootstrap** es la siguiente.
2. 

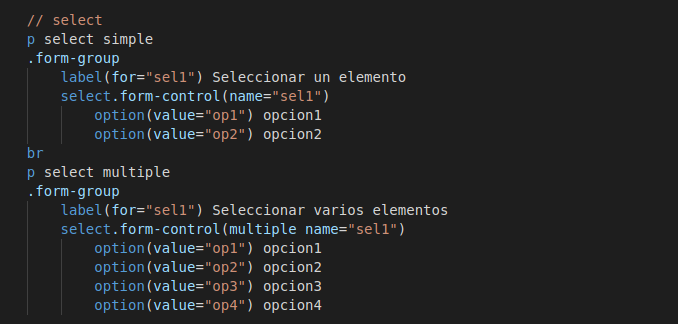
## 

|  |  |
| --- | --- |
| **Html** | **Bootstrap** |
|  |  |

### 6.4 – Select

1. 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"**.
2. La sintaxis de clases aplicadas es similar a las etiquetas "input" que vimos anteriormente.
3. **Html**

****

1. **Bootstrap**
2. ****

|  |  |
| --- | --- |
| **Html** | **Bootstrap** |
|  |  |

## 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.**

|  |  |
| --- | --- |
|  |  |
|  |  |

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.

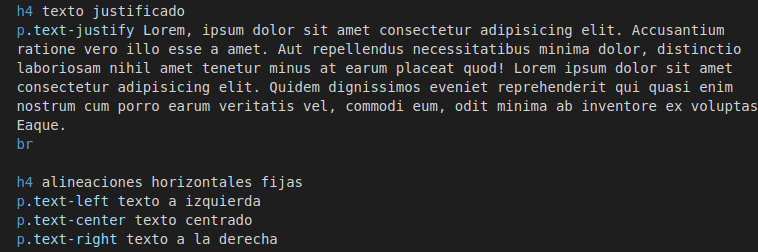
### 

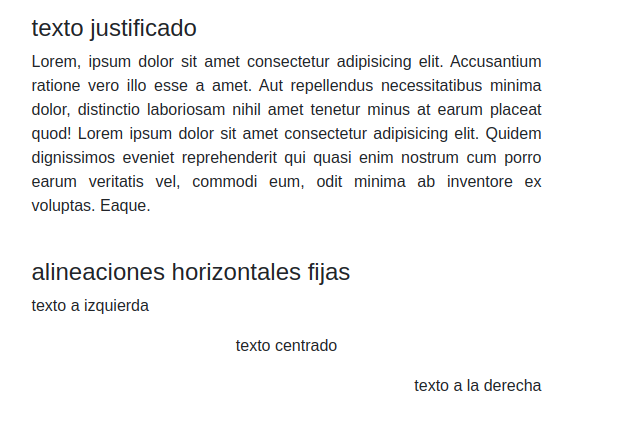
### 

### 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**





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

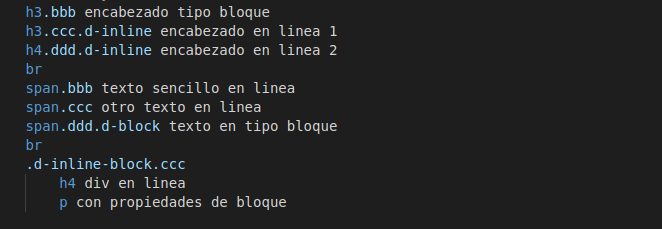
* **align-baseline**
* **align-top**
* **aling-bottom**
* **align-middle**
* **align-text-top**
* **align-text-bottom**

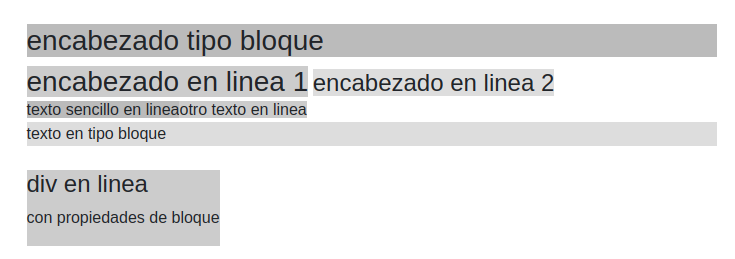
|  |  |
| --- | --- |
|  |  |

### 7.3 – Elementos en linea y bloque

Con bootstrap se puede alterar el comportamiento de los elementos en linea (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 linea
* **d-block**. Hace que un elemento en linea 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





### **7.4 – margenes 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 te 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 linea.

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.

|  |  |
| --- | --- |
|  |  |
|  |  |

## 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.

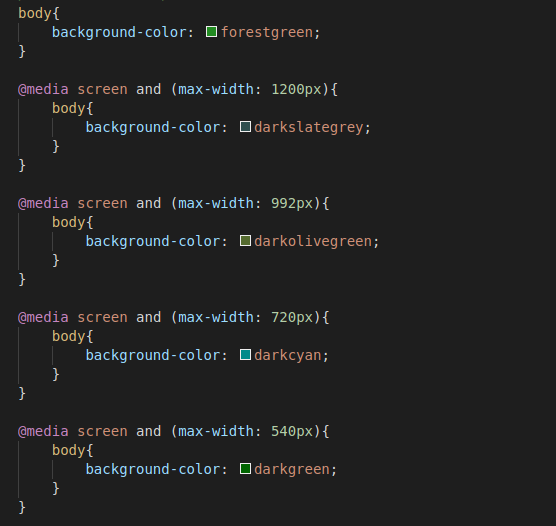
El ejemplo mas usual es cambiar el color de fondo según un ancho determinado. Vamos a hacerlo usando el media query de tipo **screen**, con la característica **max-widht**. Este indica que hasta cierto ancho determinado de pantalla deben aplicarse los estilos declarados dentro del mismo.

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.



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

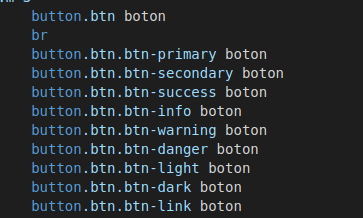
* colores
* Tamaños
* Agrupaciones

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.

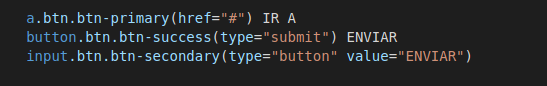
### 9.1 – Colores

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



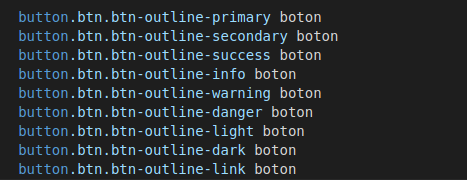


Los casos particulares son los enlaces **a** y los **inputs** dentro de un formulario





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>**

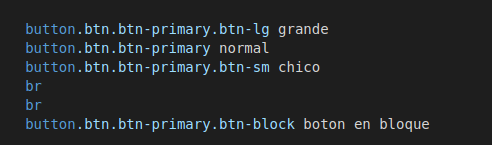
****

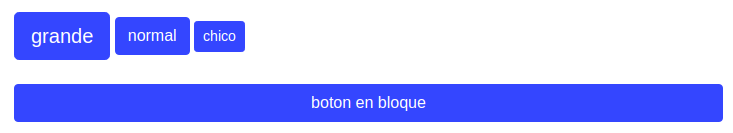
****

### 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



****

**Font awesome**

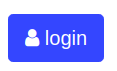
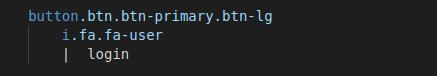
Una librería muy usada por la amplia variedad de íconos que posee es font awesome. Esta puede agregarse usando la cdn disponible en esta página.

<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>

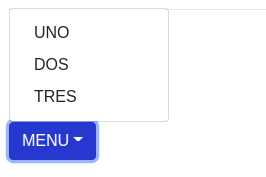
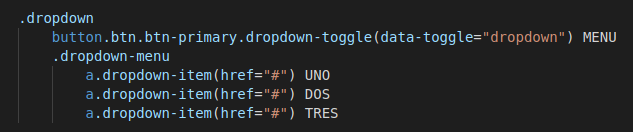
Estos íconos pueden usarse en diversas partes de la página, como bloques, navegaciones y botones. Si por ejemplo queremos agregar a un boton de login un icono de usuario se agrega la etiqueta **i** con las clases **.fa.fa-user**.



### 9.3 – Agrupaciones

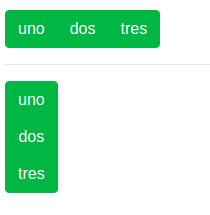
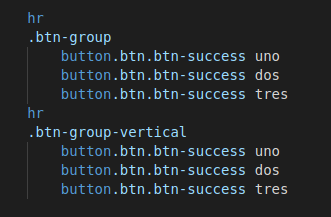
Las agrupaciones suelen hacerse mediante dos métodos, los dropdowns y los grupos de botones

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.

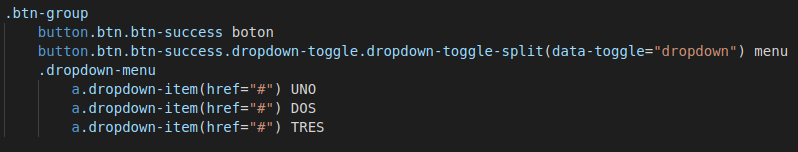


Los grupos de botones pueden ser de dos tipo, btn-group, que los muestra como si fueran un solo boton, y btn-toolbar, que los muestra juntos pegados pero se los puede separar agregando margenes.

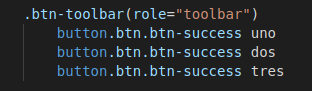
El grupo de botones puede ser tanto horizontal como vertical.



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



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



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.

|  |  |
| --- | --- |
|  |  |

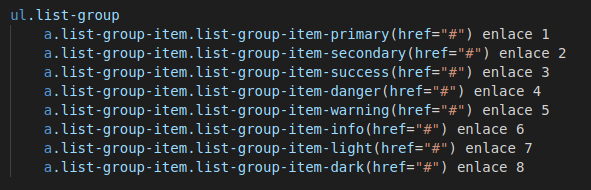
## 10 – Listas

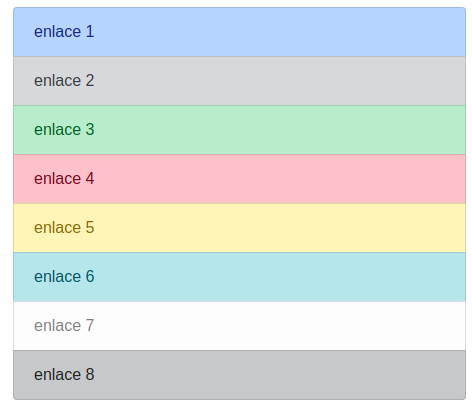
Para aplicar estilos a las listas bootstrap dispone de la clase list-group, que se puede aplicar a los elementos **ul**, **ol** y **div.**

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

|  |  |
| --- | --- |
|  |  |

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).





**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.

|  |  |
| --- | --- |
|  |  |

## 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

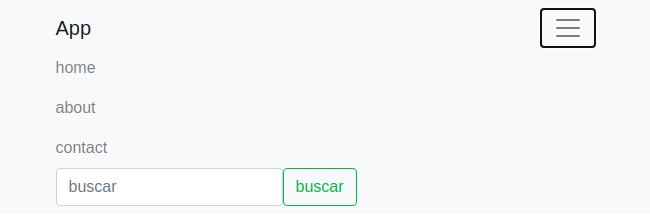
<https://getbootstrap.com/docs/4.0/components/navbar/>

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

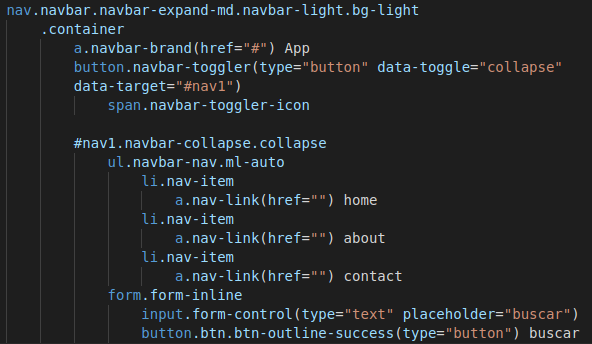
**En pantalla md**



**En pantalla sm**



El código para esta navegación es el siguiente



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 **nabvar-<estilo>** y **bg-<estilo>** para los estilos de tipografía y de fondo.

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

* La marca de la aplicación
* un boton 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 boton 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 busqueda.

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), mientra 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 mas comúnes. 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 ademas 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 menu desplegable, cada uno de estos con la clase **dropdown-item**.

