

Documentación Bootstrap 4

Índice

Parte 1 . Maquetación Responsive y Layout	2
<i>Contenedor</i>	2
<i>Breakpoint</i>	3
<i>Glutter</i>	4
<i>Offset</i>	5
<i>Alineación</i>	6
<i>Márgenes</i>	7
<i>Imágenes</i>	8
<i>Tablas</i>	9
<i>Formulario</i>	11
<i>Objeto Media</i>	12
<i>Elemento embed</i>	15
Parte 2. Componentes	16
<i>Dropdown</i>	16
Nav	17
Navbar	19
BreadCrumbs	21
Pagination	22
Alert	23
Badge	24
PopOver	25
Tooltip	26
Modal	27
ScrollSpy	29
Progress	30
Collapse	31
Form	32
Button	34
Input-Group	36
List-Group	38
Carousel	40
Card	42
Jumbotron	43

Parte 1 . Maquetación Responsive y Layout

Contenedor

En BootStrap tenemos 2 tipos de contenedores:

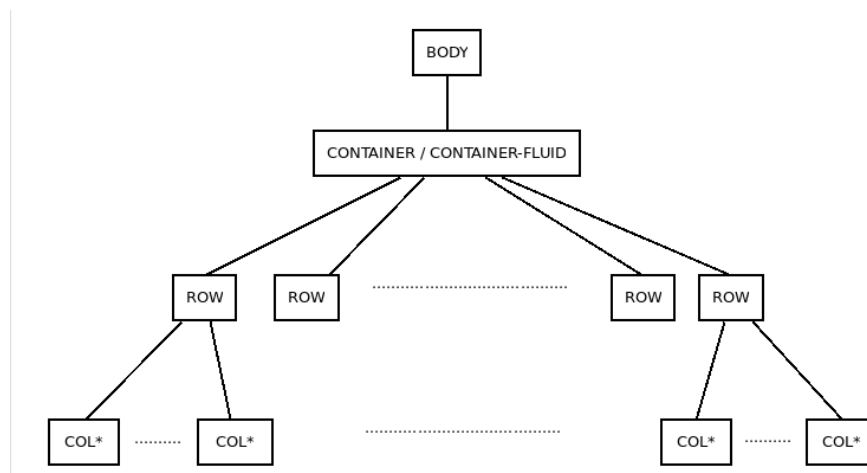
- **container** : Ocupará diferentes anchos dependiendo del tamaño de la pantalla. Puede ocupar toda la pantalla o dejar unos márgenes a izquierda a derecha aunque, en este caso, siempre estará centrado.

```
<body>  
  <div class="container"></div>  
</body>
```

- **container-fluid**: Ocupará todo el ancho (100%) de lo que podemos ver en el navegador.

```
<body>  
  <div class="container-fluid"></div>  
</body>
```

Estructura dentro de un contenedor



Dentro de cada row podremos tener hasta 12 columnas.

Breakpoint

En el mundo del diseño responsivo un **breakpoint** es un tamaño de pantalla (en pixels) donde se produce un cambio en la disposición de los elementos que conforman nuestra página web. Existen 4 breakpoints:

- En **576px**
- En **768px**
- En **992px**
- En **1200px**

Teniendo en cuenta estos 4 breakpoints podemos diferenciar entre 5 tipos de pantallas

- **Pantallas extra pequeñas:** Entre 0 y 576px (`.col`)
- **Pantallas pequeñas:** Entre 576px y 768px (`.col-sm-`)
- **Pantallas medianas:** Entre 768px y 992px (`.col-md-`)
- **Pantallas grandes:** Entre 993px y 1200px (`.col-lg-`)
- **Pantallas extra-grandes:** A partir de 1200px (`.col-xl-`)

El prefijo de la clase es → `.col-sm-XX` (donde **XX** es el tamaño de la columna, con valores entre **1** y **12**).

- **col-X:** hará que ese contendor X/12 columnas en pantallas extra pequeñas.

```
<!-- col-6 ocupará justo la mitad en pantallas extra pequeñas -->
<div class="col-6"></div>
```

- **col-sm-X:** hará que ese contendor X/12 columnas en pantallas pequeñas.

```
<!-- col-sm-4 ocupará un tercio de la pantalla en pantallas pequeñas-->
<div class="col-sm-4"></div>
```

Así por ejemplo si tenemos la siguiente estructura:

```
<div class="row">
  <div class="col-md-6">
  </div>
  <div class="col-md-4">
  </div>
  <div class="col-md-4">
  </div>
</div>
```

IMPORTANTE → El tercer div irá abajo ya que $6+4+4 = 14$ que es mayor que 12.

Columnas automáticas

Usando elementos en los que no especifiquemos el número de columnas, el espacio que haya en la fila se va a distribuir de manera uniforme.

- **col** : Para todo tipo de pantallas
- **col-sm** : De pantallas pequeñas en adelante ($\geq 576\text{px}$)
- **col-md** : De pantallas medianas en adelante ($\geq 768\text{px}$)
- **col-lg** : De pantallas grande en adelante ($\geq 992\text{px}$)
- **col-xl** : Para pantallas de 1200px en adelante

Ejemplo:

```
<div class="row">
  <div class="col-md yellow"></div>
  <div class="col-md pink"></div>
  <div class="col-md yellow"></div>
  <div class="col-md pink"></div>
  <div class="col-md yellow"></div>
</div>
```

- En este caso todo se divide en 5 partes para pantallas menores de 768px pero en cuanto la pantalla es más pequeña que eso, todos los elementos pasarán a ocupar todo el centro de la pantalla.

Glutter

Por defecto BootStrap 4 introduce separación entre el contenido de las columnas mediante *padding*s y márgenes a derecha e izquierda. Es lo que se denomina *gutter*.

Ejemplo

```
<div class="container">
  <div class="row">
    <div class="col-md-6"><h3>Elemento con Gutter</h3></div>
    <div class="col-md-6"><h3>Elemento con Gutter</h3></div>
  </div>
  <div class="row no-gutters"> //Para no aplicar gutters!!
    <div class="col-md-6"><h3>Elemento sin gutter</h3></div>
    <div class="col-md-6"><h3>Segundo Elemento</h3></div>
  </div>
</div>
```

Offset

Aplica un desplazamiento lateral a nuestra columna.

Este desplazamiento puede ser también establecido de manera diferente para cada tamaño de pantalla:

- **offset-sm-X** : Para desplazamientos para pantallas entre 576px y 768px.
- **offset-md-X** : Para desplazamientos para pantallas entre 768px y 992px.
- **offset-lg-X** : Para desplazamientos para pantallas entre 992 y 1200px.
- **offset-xl-X** : Para desplazamientos para pantallas mayores de 1200px.

```
<div class="container-fluid">
  <div class="row">
    <div class="col-sm-4 offset-md-1 red"></div>
    <div class="col-sm-4 offset-sm-4 offset-md-2 blue"></div>
  </div>
</div>
```

En este ejemplo para pantallas mayores de 768px el primer div está desplazado una columna a la izquierda y ambos bloques se separan dos columnas mediante el uso de un desplazamiento de dos columnas en el segundo bloque.

Además, para pantallas entre 576px y 768px no hay desplazamiento para el primer bloque y hay un desplazamiento de 4 columnas para el segundo bloque consiguiendo, de esta forma que ese bloque quede totalmente pegado a la derecha.

Alineación

Podremos establecer:

- La alineación vertical de los elementos de la fila.
- La alineación horizontal de los elementos de la fila.

Tanto para establecer la alineación horizontal como la vertical deberemos añadir una nueva clase al contenedor *row* o fila.

Alineación vertical

Si queremos alinear los distintos elementos que componen una fila de manera vertical podremos hacerlo añadiendo a la *row* fila una de estas tres clases:

- **align-items-start** : Los elementos de la fila se alinearán verticalmente con el borde superior de la fila.

```
<div class="row align-items-start"></div>
```

- **align-items-center** : Los elementos de la fila se alinearán centrados verticalmente.

```
<div class="row align-items-center"></div>
```

- **align-items-end** : Los elementos de la fila se alinearán verticalmente con el borde inferior de la fila.

```
<div class="row align-items-end"></div>
```

Alineación horizontal

Para fijar la alineación horizontal de los elementos en la *row* (fila) dispondremos de las siguientes clases:

- **justify-content-start** : Los elementos empezarán en la parte izquierda y ocuparán el tamaño horizontal establecido. Es la opción por defecto.

```
<div class="row justify-content-start"></div>
```

- **justify-content-center** : Los elementos de la fila se centrarán horizontalmente.
- **justify-content-end** : Los elementos de la fila se alinearán al final (la derecha normalmente) de la fila.
- **justify-content-around** : El espacio restante se distribuirá de manera equitativa alrededor de los elementos empezando y acabando con espacios libres.

- **justify-content-between** : El espacio restante se distribuirá de manera equitativa entre los distintos elementos estando uno de ellos totalmente a la derecha y otro totalmente a la izquierda.

- Para ordenar columnas utilizaremos la clase **order** →

`<div class="col-md4 order-3">` (Si queremos ponerlo en tercera posición)

Márgenes

Para establecer los márgenes entre las columnas usaremos la clas *m{lado}-{tamaño}* o si queremos distinguir según los distintos tamaños de pantalla:

- **m{lado}-sm-{tamaño}** Para pantallas entre 576px y 768px.
- **m{lado}-md-{tamaño}** Para pantallas entre 768px y 992px.
- **m{lado}-lg-{tamaño}** Para pantallas entre 992px y 1200px.
- **m{lado}-xl-{tamaño}** Para pantallas de más de 1200px.

Pudiendo ser **lado**:

- **t** para margen superior (top).
- **b** para margen inferior (bottom).
- **l** para margen izquierdo (left).
- **r** para margen derecho (right).
- **x** para los márgenes superior e inferior.
- **y** para los márgenes izquierdo y derecho.
- En blanco si es para todos los lados.

Y pudiendo ser tamaño de **tamaño**:

- **0** : No hay margen
- **1** : 0.25rem
- **2** : 0.25rem
- **3** : 1rem
- **4** : 1.25rem
- **5** : 3rem
- **auto** : Para clases que establecen una margen auto

Relacionado con el tema de los márgenes hay que destacar que BooStrap 4 introduce una nueva clase **mx-auto** que permite centrar horizontalmente un elemento dentro de su contenedor siempre que tenga una.

La notación de las clases para establecer los paddings en las columnas es muy similar a la notación de las clases para establecer márgenes entre las columnas. Simplemente tenemos que cambiar la **m** por la **p**.

Imágenes

```

```

En relación a las imágenes las clases de interés son:

- **img-fluid** : Que nos permite convertir una imagen en responsiva. Al añadir esa clase a una imagen ésta ocupará toda la anchura de su elemento padre manteniendo sus proporciones (aspect ratio). Si hemos utilizado la etiqueta *picture* debemos añadir esta clase a la *img* que contiene y no a *picture*.
- **img-thumbnail** : Para añadir un marco redondeado a la imagen. Para esto podríamos también usar las clases relativas a bordes que veremos posteriormente en el capítulo sobre utilidades.
- **rounded** . Si queremos que las esquinas sean redondeadas.
- Una **figura** (etiqueta *figure*) es un conjunto compuesto de una imagen (etiqueta *img*) y de un texto descriptivo sobre la imagen (etiqueta *caption*). Esta etiqueta es una de las novedades en HTML5. Tradicionalmente es lo que se usa en *libros* para hacer posteriormente un índice de figuras.

```
<figure>
  
  <figcaption>Texto descriptivo de la imagen</figcaption>
</figure>
```

En relación a las figuras las clases de interés para dar estilos con Bootstrap 4 son:

- **figure** : Clase a añadir a la etiqueta *figure*.
- **figure-img** : Clase a añadir a la etiqueta *img* que contiene la figura.
- **figure-caption** : Clase a añadir a la etiqueta *figcaption* que contiene la figura. El texto se podrá adicionalmente alinear de distintas maneras usando *text-justify*, *text-left*, *text-right* o *text-center*.

IMPORTANTE : Si queremos que esto siga siendo responsivo debemos añadir *.img-fluid* a la imagen de la figura.

Tablas

Simplemente debemos añadir la clase **table** y adicionalmente la clase **table-dark** si queremos que se inviertan los colores de fondo y de letra.

```
<table class="table">
    ....
</table>
```

Clases para la cabecera de las tablas

Para dar estilos a las cabeceras de las tablas tenemos que las clases **thead-light** o **thead-dark** a la etiquetas de las filas (*tr*) o a las etiquetas (*thead*).

```
<thead class="thead-light">
    <tr>
        <th>...</th>
        ....
    </tr>
</thead>
```

IMPORTANTE : Este estilo sólo se aplica a la etiqueta *th* (celdas de cabecera).

Alternado colores

Si queremos que las filas de las tablas tengan colores alternativos para poder distinguirlas mejor debemos añadir la clase **table-striped** a la etiqueta *table*. Esto funcionará también para tablas oscuras (clase *table-dark*).

```
<table class="table table-striped">
    ....
</table>
```

Destacar

Si queremos que conforme pase el puntero de ratón por encima de una fila (que no sea la cabecera) esa fila de destaque cambiando ligeramente de color debo añadir la clase **table-hover** a las etiqueta *table*.

```
<table class="table table-hover">
    ....
</table>
```

Bordes de las tablas

Por defecto en Bootstrap 4 las tablas únicamente muestran unos ligeros bordes inferiores en las filas para separar estas. Si, por el contrario, queremos que todas las celdas muestren los 4 bordes (inferior, superior, derecha e izquierda) debemos añadir la clase ***table-bordered*** a la etiqueta *tabla*.

```
<table class="table table-bordered">
  ....
</table>
```

Haciendo las tablas responsivas

Hasta ahora únicamente hemos visto como Bootstrap 4 da estilos a las tablas pero, como ya habrá quedado patente durante todo lo que llevamos de curso, la verdadera potencia de esta librería reside en la facilidad que nos da para hacer las páginas responsivas.

Existen distintas técnicas para hacer las tablas responsivas pero los desarrolladores de Bootstrap 4 han optado porque aparezca un scroll horizontal cuando sea necesario.

Para conseguir esto debemos insertar nuestra tabla dentro un div con la clase Bootstrap ***table-responsive***

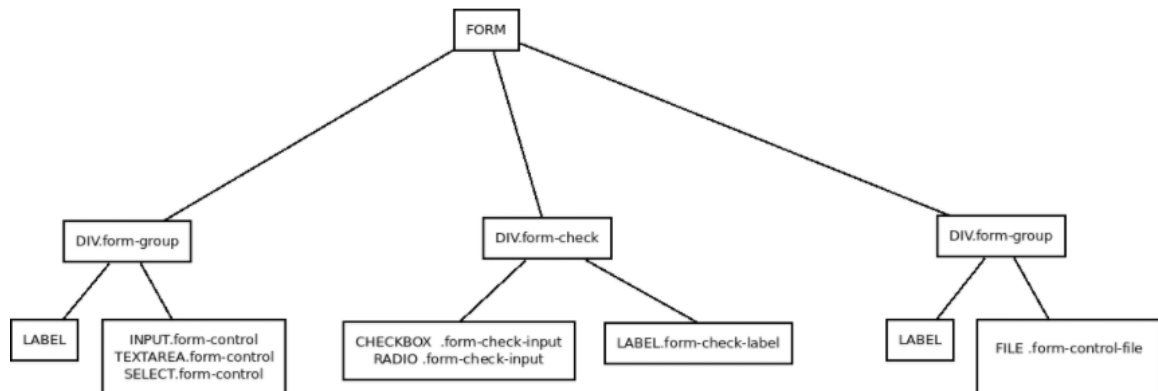
```
<div class="table-responsive">
  <table class="table table-bordered">
    ....
  </table>
</div>
```

También puedo fijar el breakpoint a partir de la cual quiero que la tabla sea responsiva. En esos casos usaré:

- ***table-responsive-sm***
- ***table-responsive-md***
- ***table-responsive-lg***
- ***table-responsive-xl***

Formulario

Estructura dentro de un formulario



Además, se le añadirá al final un *input* de tipo *submit* o *button* con las clases correspondientes a los botones cuyos ejemplos más comunes (hay muchos más) son:

- **btn btn-primary**
- **btn btn-secondary**
- **btn btn-success**
- **btn btn-danger**
- **btn btn-warning**

Podemos además modificar ciertos aspectos de estos componentes del formulario. Los más interesantes son los siguientes.

- **Modificar el tamaño en altura del control.** Añadiendo clases como **form-control-lg** (grande) o **form-control-sm** (pequeños) en los **form-control**.
- **Modificar el tamaño en altura de la etiqueta del control.** Usando clases como **col-label-lg** (grandes) o **col-label-sm** (pequeños).
- **Hacer que todos los elementos del formulario se vean en la misma línea** añadiendo la clase **form-inline** a la etiqueta **form**.
- **Hacer que las distintas opciones para elementos *radio* o *checkbox* se vean en la misma línea** añadiendo al **div** que tenía la clase **form-check** la clase **form-check-inline**
- **Añadir texto de ayuda a los diferentes elementos** usando una etiqueta **small** dentro del **form-group** o **form-check** y dando a esa etiqueta las clases **form-text** y **text-mute**.

En cuanto a su disposición, los formularios por defecto ocupan en anchura lo que ocupen el contenedor padre al que corresponden, pero podemos adaptar su tamaño jugando con el grid de Bootstrap 4 que hemos visto en capítulos anteriores añadiendo clases **col-X** (o atendiendo a distintos breakpoints) al elemento que contenga la clase **form-group** o **form-check**.

Para hacer los formularios más compactos hay una nueva clase que suprime el gutter, **form-row** que debe ser usada en vez de **row**.

Objeto Media

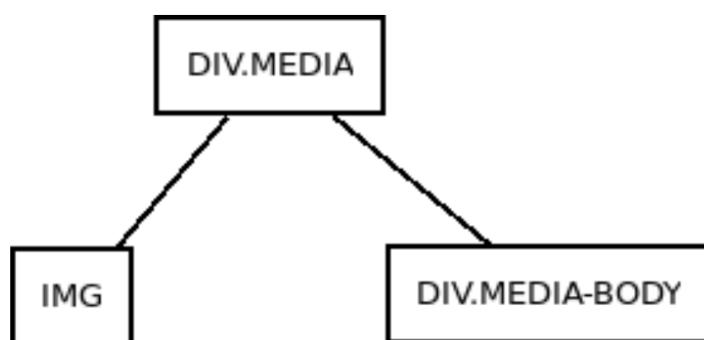
Para entender lo que es un objeto media lo mejor es irnos a la propia definición que nos da Bootstrap 4:

“son elementos complejos y repetitivos en los cuáles algún elemento multimedia (una imagen normalmente) se posiciona junto a otro elementos que “

¿ Y éso qué es?. Pues son cosas que vemos diariamente como anuncios en la web, comentarios en blogs que llevan fotos de perfil, listas de tweets etc...

Bootstrap 4 nos proporciona una manera fácil de construir este tipo de estructuras utilizando elementos flex, clases y etiquetas que compongan la siguiente jerarquía:

Estructura



```
<div class="media">
  
  <div class="media-body">
    <h3>Titulo</h3>
    <p>Texto relacionado</p> ...
  </div>
</div>
```

Este tipo de estructuras se pueden anidar (como lo comentarios y las respuestas a un post).

Podemos también decidir cómo se van a disponer el resto de los elementos alrededor de la imagen usando clases como las siguiente y los márgenes correctos:

- ***align-self-start*** : Que alinea el contenido verticalmente con el borde superior de la imagen.
- ***align-self-center*** : La imagen queda centrada verticalmente en relación al resto de los elementos.
- ***align-self-end*** : Que alinea el contenido verticalmente con el borde superior de la imagen.

Si queremos que la imagen salga a la derecha sólo tenemos que cambiar la etiqueta en el HTML.

```

<div class="media">
  <div class="media-body">
    <h3>Titulo</h3>
    <p>Texto relacionado</p>
    ...
  </div>
  
</div>

```

Si queremos construir listas (como si fuera el *timeline* de twitter debemos ponerle la clase *media* a los elementos las etiquetas *li* que sustituirán al *div.media* anterior)

```

<ul>
  <li class="media">
    
    <div class="media-body">
      <h3>Titulo 1</h3>
      <p>Texto relacionado</p>
      ...
    </div>
  </li>
  <li class="media">
    
    <div class="media-body">
      <h3>Titulo 2</h3>
      <p>Texto relacionado 2</p>
      ...
    </div>
  </li>
</ul>

```

Ejemplo

```
<div class="container">
  <div class="row">
    <div class="col m-3">
      <div class="media">
        
        <div class="media-body">
          <h5>Anuncio 1</h5>
          <p>Compra este objeto para poder ser feliz</p>
        </div>
      </div>
    </div>
    <div class="col m-3">
      <div class="media">
        
        <div class="media-body">
          <h5>Anuncio 2</h5>
          <p>Compra este objeto para poder ser guapo</p>
        </div>
      </div>
    </div>
    <div class="col m-3">=
  </div>
</div>
```

```
<div class="row justify-content-center mt-5">
  <div class="col-md-6">
    <div class="media">
      
      <div class="media-body">
        <h6>Curso BootStrap 4</h6>
        <p>Me encanta este curso</p>
      </div>
    </div>
    <div class="media">
      
      <div class="media-body">
        <h6>Curso BootStrap 4</h6>
        <p>No sabes lo que dices</p>
      </div>
    </div>
  </div>
  <div class="media">
    
    <div class="media-body">
      <h6>Curso BootStrap 4</h6>
      <p>No está mal</p>
    </div>
  </div>
</div>
```

Resultado



Elemento embed

Los elementos embebidos son algo común en todas las webs , vídeos de youtube, reproductores de canciones de spotify, presentaciones de slideshare están embebidas o incrustadas por todos los sitios.

Desde el punto de vista más técnico nos estamos refiriendo a las etiquetas *iframe*, *embed*, *video* y *object*

Para hacer estas etiquetas responsivas debemos añadir la clase ***embed-responsive*** a un elemento que la contenga y la clase ***embed-responsive-item*** a la etiqueta en cuestión. Algo así:

```
<div class="embed-responsive">
  <iframe src="..." class="embed-responsive-item"></iframe>
</div>
```

Adicionalmente podemos modificar las proporciones elementos (su *aspect ratio*) añadiendo al elemento padre clases que lo modificarán:

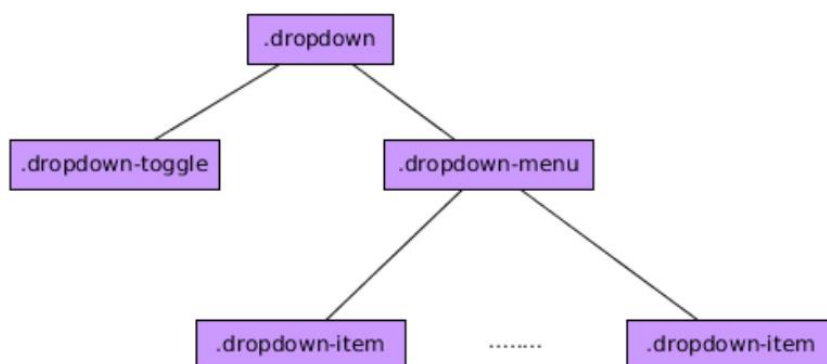
- ***embed-responsive-21by9*** : Aspect Ratio 21x9
- ***embed-responsive-16by9*** : Aspect Ratio 16x9
- ***embed-responsive-4by3*** : Aspect Ratio 4x3
- ***embed-responsive-1by1*** : Aspect Ratio 1x1

Parte 2. Componentes

Dropdown

El componente **dropdown** de Bootstrap 4 es lo que conocemos como un elemento desplegable con una lista de opciones (submenú) que se muestran al hacer click sobre el elemento padre.

Estructura dentro de un dropdown



- **dropdown** el elemento general del componente.
- **dropdown-toggle** el elemento que servirá para mostrar u ocultar el submenú.
- **dropdown-menu** el elemento que contiene las opciones.
- **dropdown-menu-item** cada una de las opciones.

Ejemplo

```
<div class="dropdown">
  <div class="btn btn-primary dropdown-toggle"
    data-toggle="dropdown">
    Muestra el submenú
  </div>
  <div class="dropdown-menu">
    <div class="dropdown-item"> Opción 1 </div>
    <div class="dropdown-item"> Opción 2 </div>
    <div class="dropdown-item"> Opción 3 </div>
    <div class="dropdown-item"> Opción 4 </div>
  </div>
</div>
```

Es importante resaltar que el elemento con la clase **dropdown-toggle** debe también poseer el atributo **data-toggle="dropdown"** para que el elemento funcione correctamente.

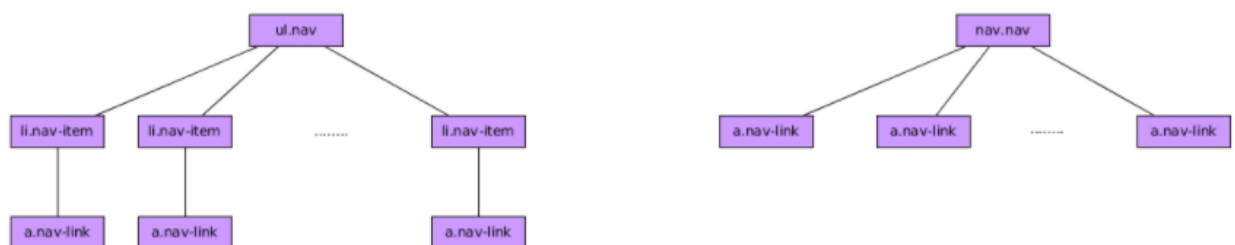
Dirección en la que aparece el menú.

Podemos mostrar el submenú en las cuatro direcciones sólo con añadir las clases **dropdown**, **dropright**, **dropleft** al elemento general. Si no ponemos nada la opción por defecto es hacia abajo. IMPORTANTE Tiene que haber...si no coge la opción por defecto.

Nav

El componente Bootstrap 4 **Nav** es el componente básico de Bootstrap 4 para formar menús y barras de navegación.

El componente Nav tiene 2 tipos de estructuras:



```
<!-- Menú de navegación con listas -->
<ul class="nav">
  <li class="nav-item"><a href="#" class="nav-link">Inicio</a></li>
  <li class="nav-item"><a href="#" class="nav-link">Novedades</a></li>
  <li class="nav-item"><a href="#" class="nav-link">Nosotros</a></li>
  <li class="nav-item"><a href="#" class="nav-link">Contacto</a></li>
</ul>

<!-- Menú de navegación con elementos semánticos -->
<nav class="nav">
  <a href="#" class="nav-link">Inicio</a>
  <a href="#" class="nav-link">Novedades</a>
  <a href="#" class="nav-link">Nosotros</a>
  <a href="#" class="nav-link">Contacto</a>
</nav>
```

Además de la forma tradicional de menú podemos optar por un menú de pestañas (**tabs**) o un menú de píldoras (**pills**). Para ello nos basta con añadir las clases **nav-tabs** o **nav-pills** al elemento que padres, el que posee la clase **nav**.

En estos dos casos para ver que lo estamos aplicando correctamente debemos añadir la clase **active** a alguno de los elementos del menú.

```

<!-- ELEMENTO DE NAVEGACIÓN -->
<nav class="nav nav-tabs">
  <a href="#inicio" class="nav-link active" data-toggle="tab">
    Inicio</a>
  <a href="#novedades" class="nav-link" data-toggle="tab">
    Novedades</a>
  <a href="#nosotros" class="nav-link" data-toggle="tab">
    Nosotros</a>
  <a href="#contacto" class="nav-link" data-toggle="tab">
    Contacto</a>
</nav>

<!-- PANELES DE CONTENIDOS -->
<div class="tab-content">
  <div class="tab-pane fade show active" id="inicio">
    Contenido del Panel Inicio</div>
  <div class="tab-pane fade" id="novedades">
    Contenido del Panel Novedades</div>
  <div class="tab-pane fade" id="nosotros">
    Contenido del Panel Nosotros</div>
  <div class="tab-pane fade" id="contacto">
    Contenido del Panel Contacto</div>
</div>

```

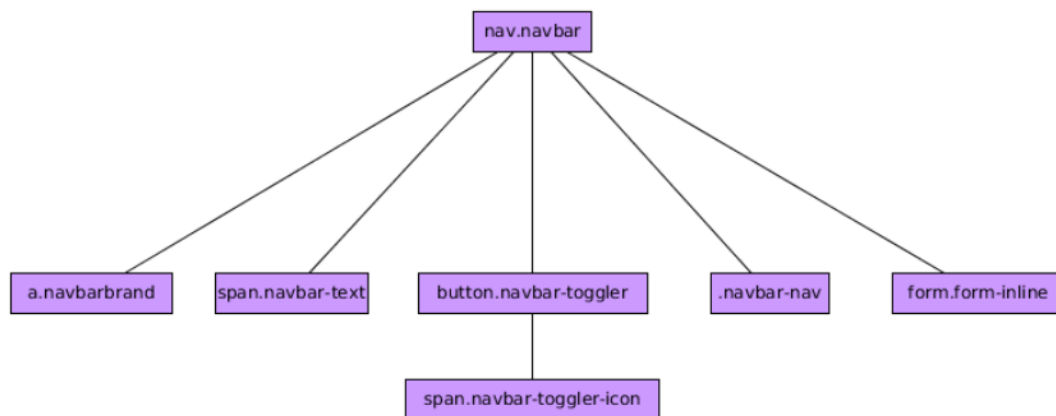
Debiendo:

- Añadir a los elementos del menú el atributo ***data-toggle="tab"*** o ***data-toggle="pill"***
- Añadiendo a los elementos del menú el ***href*** correcto indicando el id del elemento en cuestión que quiero mostrar (podríamos usar ***data-target*** también)
- Creando la jerarquía ***tab-content***, ***tab-pane*** con los ids correctos.
- Indicando cuál es el seleccionado inicialmente mediante las clases ***active*** y ***show*** y añadiendo la clase ***fade*** si queremos un efecto de desvanecimiento.

Navbar

El componente Bootstrap 4 **Navbar** es un componente que nos permite construir barras de navegación responsivas y más complejas que las que podemos construir con el componente *Nav*

Estructura



Puede estar formado (no todos son obligatorios) por lo siguiente elementos:

- Un elemento **navbar-brand** para mostrar el logo, nombre de tu empresa, web etc...
- Un elemento menú **navbar-nav** que es similar al componente *nav* que hemos visto anteriormente.
- Un formulario que forme parte del menú de navegación y que se mostrará en línea (**form-inline**)
- Textos que queramos incluir en el propio menú mediante un elemento en línea **navbar-text**.
- Un elemento **navbar-toggler** que se usará con el componente **collapse** para mostrar el menú cuando se haya ocultado porque ya no cabe en pantalla por su comportamiento responsivo.

```

<nav class="navbar navbar-expand-sm navbar-dark bg-dark">
  <a class="navbar-brand" href="#"></a>
  <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#menu">
    <span class="navbar-toggler-icon"></span>
  </button>
  <span class="navbar-text text-white">
    Asegura tu futuro con nosotros
  </span>
  <div class="collapse navbar-collapse" id="menu">
    <nav class="navbar-nav">
      <a class="nav-link" href="#">Inicio</a>
      <a class="nav-link active" href="#">Novedades</a>
      <a class="nav-link" href="#">Nosotros</a>
      <a class="nav-link" href="#">Contacto</a>
    </nav>
  </div>
  <form class="form-inline">
    <input class="form-control form-control-sm mr-md-2"
    type="search" placeholder="Buscar">
    <input class="btn btn-primary btn-sm" type="submit">
  </form>
</nav>

```

IMPORTANTE: Si queremos que el menú se muestre desplegado desde el principio no debemos olvidarnos de la clase **nav-expand-X** siendo X uno de los breakpoints de Bootstrap (sm,md,lg,xl). A partir de ese breakpoint se mostrará sin colapsar. Para más detalles ver el curso de maquetación.

Podemos cambiar los colores del menú usando las clases **navbar-X** y **bg-X** siendo X un color de la paleta de colores de Bootstrap 4.

Si queremos posicionar el menú de manera fija (no posiciones *static*) podemos usar las clases **fixed-top**, **fixed-bottom** y **sticky-top** (aún no soportado por todos los navegadores).

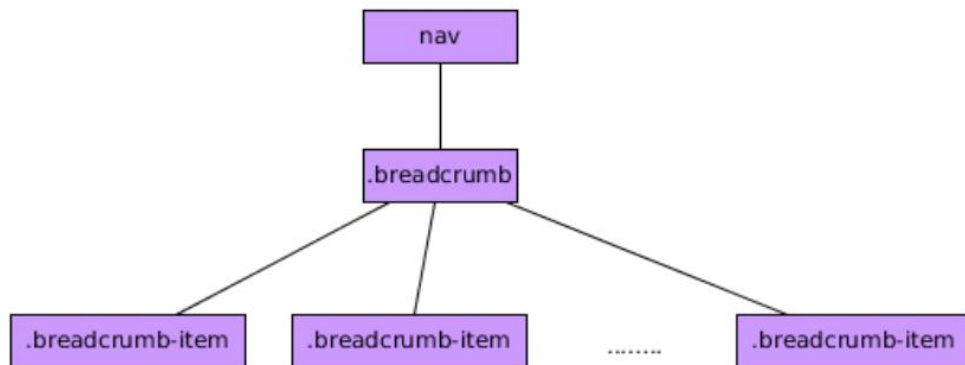
El ejemplo sugerido ya era una barra de menú responsiva. Para conseguir eso debemos hacer lo siguiente:

- En primer lugar decidir a partir de qué tamaño el menú se va a mostrar entero. Para ello usaremos las clases **navbar-expand-X** tal y como hemos explicado anteriormente.
- Tener un elemento con la clase **navbar-toggler** y los atributos **data-toggle="collapse"** y **data-target="#MenuID"** donde el menú **navbar-nav** estará dentro de un elemento con las clases **collapse navbar-collapse** y el id indicado en el **data-target**

BreadCrumbs

El componente Bootstrap 4 *BreadCrumbs* (miguitas de pan en castellano) es un componente sencillo de Bootstrap 4 que nos permite mostrar en qué lugar de la jerarquía de los contenidos de nuestra página nos encontramos.

Estructura dentro de un BreadCrumb



```
<nav>
  <div class="breadcrumb">
    <div class="breadcrumb-item"><a href="#">Inicio</a></div>
    <div class="breadcrumb-item"><a href="#">Noticias</a></div>
    <div class="breadcrumb-item active">Últimas noticias</div>
  </div>
</nav>
```

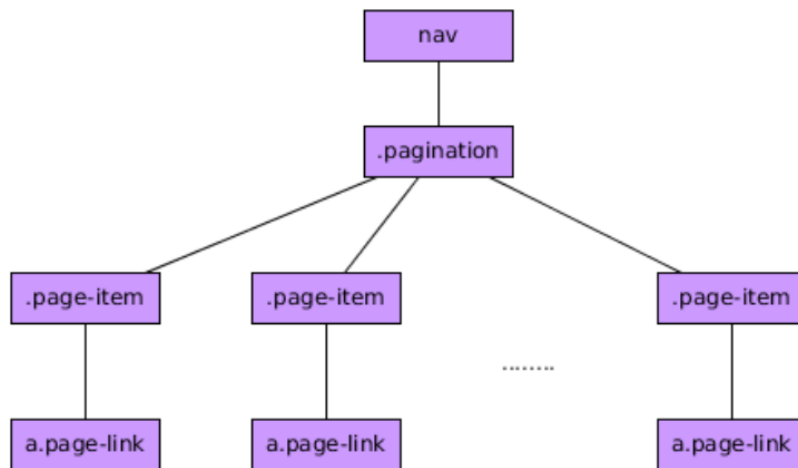
Siendo:

- ***breadcrumb*** el elemento general que debería dentro de una etiqueta *nav* ya que es un elemento de navegación.
- ***breadcrumb-item*** cada uno de los niveles de la jerarquía debiendo ser el último el que tenga, adicionalmente, la clase ***active***.

Pagination

El componente Bootstrap 4 *Pagination* es un componente sencillo de Bootstrap 4 que permite navegar entre grupos de contenidos relacionados a lo largo de distintas páginas.

Estructura



```
<nav>
  <div class="pagination pagination-sm">
    <div class="page-item">
      <a class="page-link" href="#">Anterior</a>
    </div>
    <div class="page-item">
      <a class="page-link" href="#">1</a>
    </div>
    <div class="page-item">
      <a class="page-link" href="#">2</a>
    </div>
    <div class="page-item">
      <a class="page-link" href="#">3</a>
    </div>
    <div class="page-item">
      <a class="page-link" href="#">4</a>
    </div>
    <div class="page-item">
      <a class="page-link" href="#">Siguiete</a>
    </div>
  </div>
</nav>
```

- Puedo hacer que un elemento sea destacado como el activo añadiendo la clase **active** al elemento que tiene la clase **page-item**.
- Puedo hacer que un elemento se muestre como desactivado añadiendo la clase **disable** al elemento que tiene la clase **page-item**.
- Puedo modificar el tamaño añadiendo las clases **pagination-lg** (grande) o **pagination-sm** (pequeño) al elemento que tiene la clase **pagination**.
- Para alinear horizontalmente el elemento paginación utilizaré las clases de Bootstrap 4 pertenecientes a las utilidades flexbox.

Alert

El componente Alert nos va a servir para mostrar mensaje destacados que queremos que el usuario vea.

```
<div class="alert alert-warning alert-
dismissible fade show" role="alert">
  Texto de la alerta
  <button type="button" class="close" data-dismiss="alert" aria-
label="Close">
    <span aria-hidden="true">X</span>
  </button>
</div>
```

- Añadir la clase **alert-dismissible** al elemento que tenía la clase **alert**. Esto además me servirá para posicionar el elemento que, al hacer click en él, hará que desaparezca la alerta.
- Añadir al elemento de cierre el atributo **data-dismiss="alert"** para asociar correctamente la acción javascript. La documentación recomienda que esto sea una etiqueta **button**
- Y para animar los elementos al desaparecer podremos añadir las clases **fade** y **show** al elemento principal.

Badge

El componente Badge es un componente que se utiliza para dar formato a pequeños elementos como etiquetas, contadores etc...

Para convertir un elemento en un componente *badge* deberemos añadir la clase **badge** y al hacer esto a nivel de estilo básicamente se produce lo siguiente

- Adapta su tamaño al elemento padre.
- Le podemos dar un fondo coloreado atendiendo a la paleta de colores de Bootstrap 4 **badge-X**. Siendo X uno de los colores (*secondary, primary*)
- Redondea los bordes de ese elemento.

```
<h3>Cesta<span class="badge badge-info">4 productos</h3>
```

Si queremos que sean aún más redondeados podemos cambiar la clase **badge** por la clase **badge-pill**.

```
<h3>Cesta<span class="badge-pill badge-info">4 productos</h3>
```

Si el elemento que contiene el badge es un enlace Bootstrap añade por defecto estilos diferentes para los estado *hover* y *focus*.

```
<a ref="#">Cesta<span class="badge badge-info">4 productos</a>
```


Popover

El componente Bootstrap 4 *popover* es lo que tradicionalmente hemos conocido en español como *bocadillo*. Es un elemento que aparecerá al hacer click sobre otro y que nos servirá para aportar explicaciones o aclaraciones sobre el elemento sobre el que hemos interactuado.

Para que funcionen requieren que hayamos añadido la librería popper.js a nuestro proyecto.

```
<div class="btn btn-danger mt-5"
  data-toggle="popover"
  title="Título del bocadillo"
  data-content="Explicación o aclaración sobre el el elemento">
  Muestra el bocadillo
</div>
```

Hay que añadir esto:

```
<script>
  $(function () {
    $('.clase_de_popover').popover({
      container: 'body'
    })
  })
</script>
```

Siendo *.clase_de_popover* el selector para escoger este tipo de elementos (lo habremos decidido nosotros) y con el parámetro *container:'body'* especificaremos que el bocadillo aparece en relación a dicho elemento.

Eso, o poner el atributo *data-container='body'* a los elementos afectados.

Con Bootstrap 4 podemos elegir entre 4 posibles posiciones en las que puede aparecer el componente *popover*:

- Arriba o *top*
- Abajo o *bottom*
- A la derecha o *right*
- A la izquierda o *left*

Para especificar esta posición deberemos añadir el atributo *data-placement* con alguno de los valores anteriormente mencionados.

Tooltip

El componente Bootstrap 4 *tooltip* es un componente muy similar al componente *popover* pero difiere de este en que el llamado *bocadillo* no aparece al hacer click en el elemento si no al mover el ratón por encima de éste.

Para que funcionen requieren que hayamos añadido la librería *popper.js* a nuestro proyecto.

```
<div class="btn btn-danger mt-5"
  data-toggle="tooltip"
  title="Título del bocadillo">
  Muestra el bocadillo
</div>
```

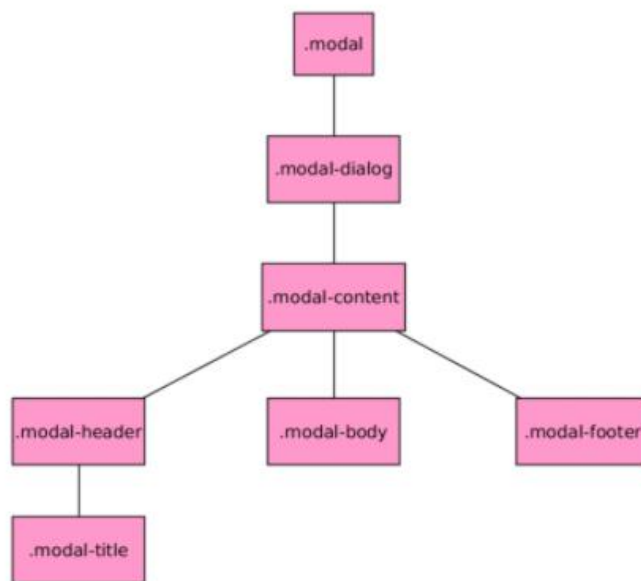
Hay que añadir esto:

```
<script>
  $(function () {
    $('[data-toggle="tooltip"]').tooltip()
  });
</script>
```

Modal

El componente *Modal* es la típica ventana flotante que se posiciona sobre el resto de los elementos oscureciendo y deshabilitando el resto de nuestra página web.

Estructura dentro de un modal



```
<div class="modal">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <div class="modal-title">
          Título
        </div>
      </div>
      <div class="modal-body">
        Contenido de la venta modal
      </div>
      <div class="modal-footer">
        Pie de la ventana modal
      </div>
    </div>
  </div>
</div>
```

Además de esta estructura necesitaremos un elemento que sirva para mostrar dicha ventana *Modal*.

```
<div class="btn btn-info" data-toggle="modal" data-target="#my_modal_ID">
  Mostrar ventana modal
</div>
```

- ***data-toggle***: Atributo Bootstrap 4 que indica que el elemento interaccionará mostrando las ventanas modales.
- ***data-target***: Que enlace con ID del elemento de nuestro árbol que se comportará como ventana modal.

Si queremos que la aparición/desaparición sea mediante un efecto de fundido debemos añadir la clase ***fade*** al elemento padre que tiene la clase ***modal***.

Es importante destacar que si el contenido de la ventana es muy grande hace que esta tenga un **scroll independiente** del resto de la página.

Si queremos centrar verticalmente la ventana modal debemos añadir la clase ***modal-dialog-centered*** al elemento que tiene la clase ***modal-dialog***.

Si queremos añadir elementos dentro de la ventana que fuercen el cierre de la misma debemos añadir a esos elementos el siguiente atributo-valor ***data-dismiss="modal"*** .

Puedo hacer más ancha o más estrecha la ventana modal añadiendo la clase ***modal-sm*** (más estrecha) o ***modal-lg*** al elemento que tiene la clase ***modal-dialog***

ScrollSpy

El componente Bootstrap 4 **ScrollSpy** nos va a permitir actualizar el elemento activo de componentes como un **ListGroup**, **Nav** o **NavBar** de tal manera que conforme vamos haciendo scroll sobre los contenidos se destacará el elemento sobre el que estamos.

Para conseguir tenemos que seguir los siguientes pasos:

- Añadir a los distintos elementos que conforman el elemento de navegación un atributo **data-target="#SeccionID"** o **href="#SeccionID"** en caso de que estos elementos sean enlaces. Además ponerle a ese elemento un ID.
- Meter los distintos contenidos sobre los que queremos hacer *scroll* dentro de un elemento que contenga el atributo **data-spy="scroll"** y ***data-target="#ID"** del componente de navegación. Además debemos dar a ese elemento una altura determinada y darle **overflow:auto** para que aparezca el scroll.
- Añadir a los contenidos de la zona de *scroll* los mismos IDs que habíamos referenciado en el punto 1.
- Podemos fijar la distancia con respecto al inicio de un elemento con **data-offset=X**. La distancia se mide en pixel.

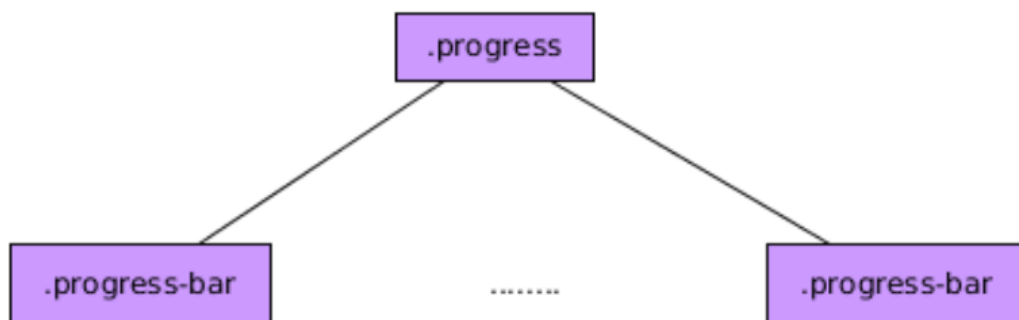
```
<ul class="list-group" id="navegacion">
  <li class="list-group-item" data-target="#s1">Sección 1</li>
  <li class="list-group-item" data-target="#s1">Sección 2</li>
  <li class="list-group-item" data-target="#s1">Sección 3</li>
  <li class="list-group-item" data-target="#s1">Sección 4</li>
</ul>

<div data-spy="scroll" data-target="#navegacion" data-
offset="0" class="ejemploscroll">
  <h1 id="s1">Sección 1</h1>
  <p>LoremOccaecat mollit enim non ullamco sunt exercitation et sit
labore. Cillum elit magna voluptate sunt deserunt est irure. Nostrud com
modo exercitation dolore enim voluptate sit laborum laboris aute.
  <h1 id="s2">Sección 2</h1>
  <p>LoremOccaecat mollit enim non ullamco sunt exercitation et sit
labore. Cillum elit magna voluptate sunt deserunt est irure. Nostrud com
modo exercitation dolore enim voluptate sit laborum laboris aute.
  <h1 id="s3">Sección 3</h1>
  <p>LoremOccaecat mollit enim non ullamco sunt exercitation et sit
labore. Cillum elit magna voluptate sunt deserunt est irure. Nostrud com
modo exercitation dolore enim voluptate sit laborum laboris aute.
  <h1 id="s4">Sección 4</h1>
  <p>LoremOccaecat mollit enim non ullamco sunt exercitation et sit
labore. Cillum elit magna voluptate sunt deserunt est irure. Nostrud com
modo exercitation dolore enim voluptate sit laborum laboris aute.
</div>
```

Progress

El componente Progreso es el componente BootStrap 4 para crear barras de progreso. La estructura general que deberemos seguir para utilizar dichos componentes es la siguiente:

Estructura dentro de un progress



En cada una de estos elementos **`.progress-bar`** podremos hacer cosas como:

- Establecer el color de los mismos usando las clases de Bootstrap **`bg-X`** siendo X uno de los colores de la paleta de Bootstrap 4.
- Establecer la altura de los mismos jugando con el atributo CSS **`height`** del elemento padre (**`.progress`**).
- Establecer el porcentaje de progreso jugando con el atributo CSS **`width`** de los posibles elementos **`.progress-bar`**.
- Hacer que tenga estilo rallado añadiendo la clase Bootstrap 4 **`progress-bar-striped`** a los elementos con las clase **`progress-bar`**.
- Repartir todo el ancho de un mismo **`progress`** entre varios **`progress-bar`**.
- Hacer que los elementos tengan una animación añadiendo la clase **`progress-bar-animated`** a los elementos **`progress-bar`**.
- Añadir contenido al elemento **`progress-bar`**. Normalmente para describir el avance del mismo.

```
<div class="container mt-5">
  <div class="progress mt-5" style="height: 100px">
    <div class="progress-bar progress-bar-striped progress-bar-
animated bg-warning" style="width:50%">
      50%
    </div>
    <div class="progress-bar bg-danger" style="width:25%">
      25%
    </div>
  </div>
</div>
```

Collapse

El componente Bootstrap 4 **Collapse** es un componente que nos ayudará a mostrar/ocultar otros elementos con el efecto de *cortinilla* (ya sea arriba o abajo).

Tenemos por lo tanto dos elementos principales dentro de este componente:

- El elemento que muestra/oculta lo demás al hacer *click* sobre él. Este normalmente será o bien un botón o un enlace con dos atributos especiales ***data-toggle="collapse"*** y o bien ***data-target="#MyCollapseID"*** si es un botón o bien ***href="#MyCollapseID"*** si es un enlace. ***#MyCollapseID*** será el id del elemento que quiero ocultar/mostrar.
- El elemento que se muestra/oculta que debe de tener un id y la clase ***collapse***.

```
<a class="btn btn-danger" data-  
toggle="collapse" href="#MyCollapseID">Ocultar/Mostrar</a>  
  
<button class="btn btn-warning" data-toggle="collapse"  
data-target="#MyCollapseID">Ocultar/Mostrar  
</button>  
  
<div class="collapse" id="MyCollapseID">  
  Lorem ipsum dolor sit amet, consectetur adipisicing elit  
</div>
```

Si queremos que el elemento en el que vamos a hacer click oculte/muestre varias cosas a la vez tenemos que modificar el selector que hemos añadido en *href* o en *data-target* de tal manera que con ese selector estemos abarcando todos los elementos que queremos ocultar/mostrar.

```
<button class="btn btn-primary" data-toggle="collapse"  
data-target=".varios">Ocultar/Mostrar</button>  
  
<div class="collapse varios border">  
  Primer elemento  
</div>  
  
<div class="collapse varios border">  
  Segundo elemento  
</div>
```

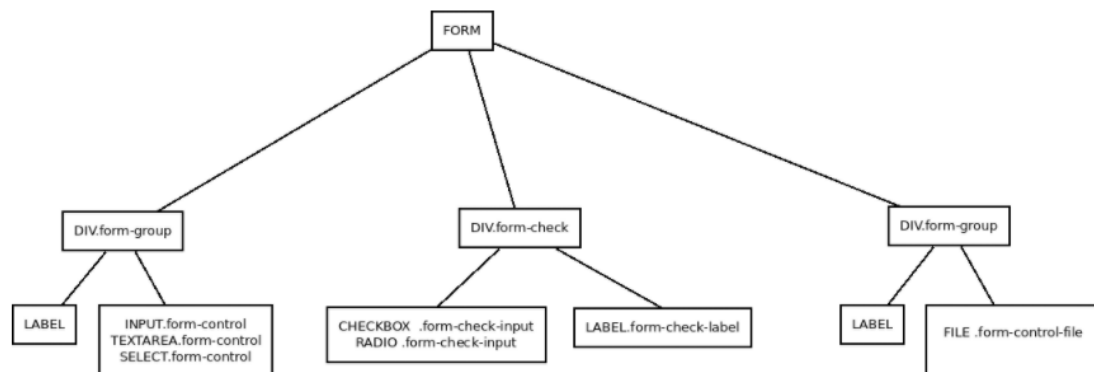
Como se puede apreciar el ***data-target=".varios"*** tiene un selector que está seleccionando, en este caso, dos elementos. Estos son los elementos que mostrará/ocultará cada vez.

Con este componente es muy fácil conseguir *Acordeones horizontales*.

Form

El componente Bootstrap 4 formulario es un elemento complejo que además tiene muchos más componentes relacionados. En este capítulo nos vamos a centrar en una estructura general de los mismos y en capítulos posteriores profundizaremos más en el tema.

Estructura dentro de un form



Además, se le añadirá al final un *input* de tipo *submit* o *button* con las clases correspondientes a los botones cuyos ejemplos más comunes (hay muchos más) son:

- **btn btn-primary**
- **btn btn-secondary**
- **btn btn-success**
- **btn btn-danger**
- **btn btn-warning**

- **Modificar el tamaño en altura del control.** Añadiendo clases como **form-control-lg** (grande) o **form-control-sm** (pequeños) en los **form-control**.
*
- **Modificar el tamaño en altura de la etiqueta del control.** Usando clases como **col-label-lg** (grandes) o **col-label-sm** (pequeños).
*
- **Hacer que todos los elementos del formulario se vean en la misma línea** añadiendo la clase **form-inline** a la etiqueta form.
- **Hacer que las distintas opciones para elementos *radio* o *checkbox* se vean en la misma línea** añadiendo al div que tenía la clase **form-check** la clase **form-check-inline**
- **Añadir texto de ayuda a los diferentes elementos** usando un etiqueta small dentro del **form-group** o **form-check** y dando a esa etiqueta las clases **form-text** y **text-mute**.
- **Deshabilitar los elementos** añadiendo el atributo **disabled** que dará estilos Bootstrap 4 al elemento deshabilitado (no se podrá interactuar con él).

En cuanto a su disposición, los formularios por defecto ocupan en anchura lo que ocupen el contenedor padre al que corresponden, pero podemos adaptar su tamaño jugando con el grid de Bootstrap 4 añadiendo clases **col-X** (o atendiendo a distintos breakpoints) al elemento que contenga la clase *form-group* o *form-check*.

Para hacer los formularios más compactos hay una nueva clase que suprime el gutter, **form-row** que debe ser usada en vez de *row*.

```
<div class="row justify-content-center mt-5">
  <div class="col-md-6 border">
    <h1 class="text-center">Formulario de Registro</h1>
    <form>
      <div class="form-group">
        <label class="col-label-sm">Usuario:</label>
        <input type="text" class="form-control form-control-sm" placeholder="Username" required />
      </div>
      <div class="form-group">
        <label class="col-label-sm">Email:</label>
        <input type="email" class="form-control form-control-sm" placeholder="Username" required />
      </div>
      <div class="form-group">
        <label class="col-label-sm">Contraseña:</label>
        <input type="password" class="form-control form-control-sm" placeholder="Username" required />
      </div>
      <div class="form-check form-check-inline ml-4">
        <input type="radio" name="sexo" class="form-check-input" value="h" />
        <label class="form-check-label">Hombre</label>
      </div>
      <div class="form-check form-check-inline ml-4">
        <input type="radio" name="sexo" class="form-check-input" value="m" />
        <label class="form-check-label">Mujer</label>
      </div>
      <div class="form-group">
        <label class="col-label-sm">Seleccionar foto de perfil:</label>
        <input type="file" class="form-control-file" />
      </div>
      <div class="text-right mb-3">
        <input type="submit" value="Registrar" class="btn btn-primary" />
      </div>
    </form>
  </div>
</div>
```

Button

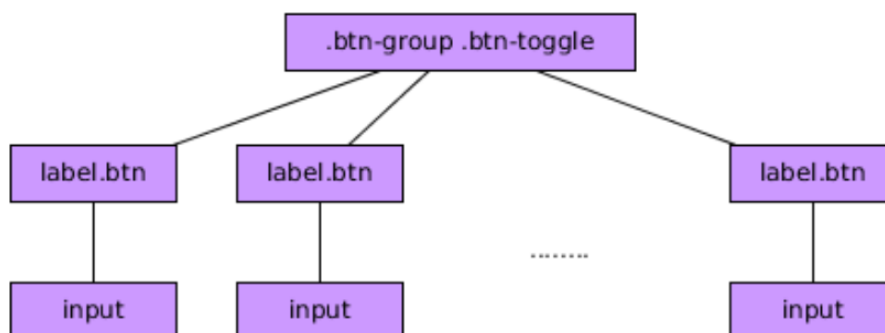
El componente Bootstrap 4 **Button** nos sirve para representar botones. Esto se consigue añadiendo la clase **btn** a las etiquetas **button**, **a** e **input**.

```
<button class="btn" value="Botón">
```

- Si queremos añadir colores podemos usar las clases **btn-X** siendo X uno de los colores de la paleta de Bootstrap.
- Si quiero que los colores sean sólo para el borde y texto pero que no tenga fondo usaré la clase **btn-outline-X** siendo X uno de los colores de la paleta de Bootstrap.
- Si quiero modificar el tamaño puedo añadir las clases **btn-lg** (grandes), **btn-sm** (pequeños) o hacer que los botones se comporten como elementos de bloque ocupando todo el ancho del padre **btn-block**.

Usando Bootstrap 4 podemos modificar la apariencia de los elementos *radio* y *checkbox* para que se muestren como si fueran botones.

Estructura



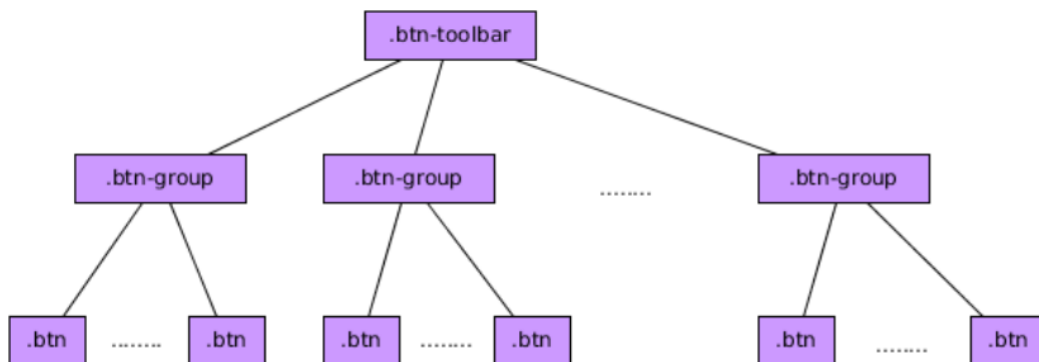
```
<div class="btn-group btn-group-toggle" data-toggle="buttons">
  <label class="btn btn-primary">
    <input type="radio" name="options" id="opA" > Opción A
  </label>
  <label class="btn btn-primary active">
    <input type="radio" name="options" id="opB" checked> Opción B
  </label>
  <label class="btn btn-primary">
    <input type="radio" name="options" id="opC"> Opción C
  </label>
</div>
```

- Hemos añadido al padre de la estructura las clases **btn-group** y **btn-group-toggle** y el atributo **data-toggle="buttons"**.
- Hemos convertido, en este caso, las etiquetas en botones con las clase **btn**.

En caso de que quisieramos hacer lo mismo con un *checkbox* sólo tenemos que cambiar el atributo *type* del input.

El componente Bootstrap 4 **Button Group** sirve para agrupar botones en una sola línea pudiendo tener más de una agrupación para formar una barra de botones o botonera.

Estructura



```

<div class="btn-toolbar">
  <div class="btn-group">
    <button type="button" class="btn btn-primary">G1B1</button>
    <button type="button" class="btn btn-primary">G1B2</button>
    <button type="button" class="btn btn-primary">G1B3</button>
  </div>
  <div class="btn-group">
    <button type="button" class="btn btn-warning">G2B1</button>
    <button type="button" class="btn btn-warning">G2B2</button>
    <button type="button" class="btn btn-warning">G2B3</button>
  </div>
  <div class="btn-group">
    <button type="button" class="btn btn-danger">G3B1</button>
    <button type="button" class="btn btn-danger">G3B2</button>
    <button type="button" class="btn btn-danger">G3B3</button>
  </div>
</div>

```

- Hemos añadido al elemento padre de la botones la clase **btn-toolbar**.
- Hemos añadido cada grupo de botones dentro de un elemento con la clase **btn-group**.

Puedo modificar el tamaño del grupo añadiendo la clase **btn-group-lg** (grande) o **btn-group-sm** (pequeño) al elemento con la clase **btn-group**.

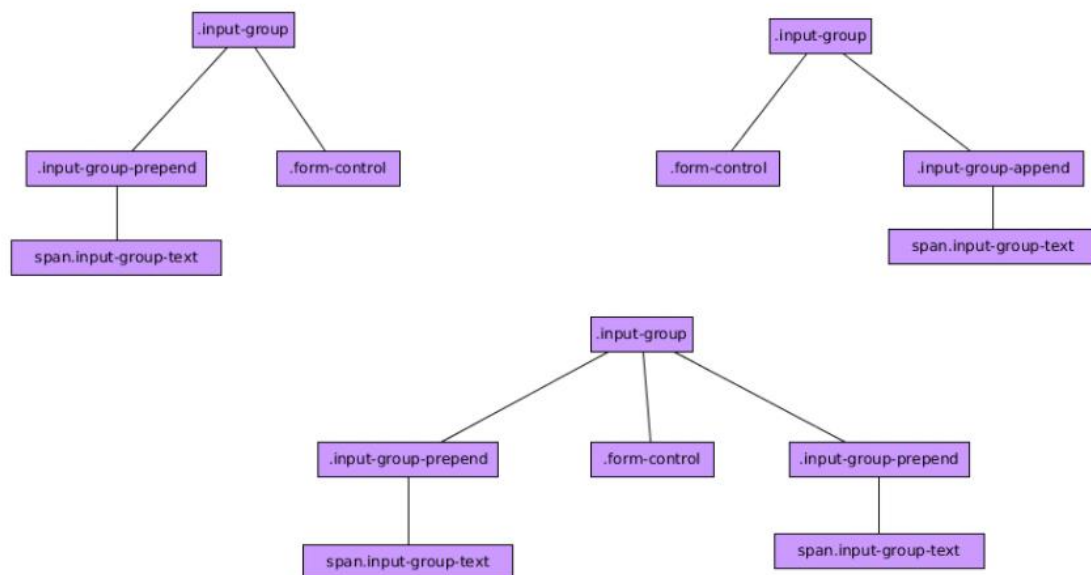
Si quiero tener *dropdown* (desplegables) tengo que anidar elementos **btn-group**.

Si quiero que los botones aparezcan en vertical sólo tengo que añadir la clase **btn-group-vertical** al elemento con las clase **btn-group**.

Input-Group

El componente Bootstrap 4 **Input-Group** es un componente que nos va servir para extender los controles de los formularios pudiendo añadir *add-ons* como texto, botones o grupos de botones a ambos lados de los campos del mismo.

Estructuras dentro de un Input-Group



IMPORTANTE: Sólo podemos tener un **form-control** en cada **input-group** y en caso de querer etiqueta debe estar fuera del **input-group**. Aunque visualmente puede haber varios, sólo se validará uno.

```

<div class="container mt-5">
  <form>
    <!-- INPUT GROUP CON TODOS LOS ELEMENTOS -->
    <label>Tú página Web personal</label>
    <div class="input-group input-group-lg">
      <div class="input-group-prepend">
        <span class="input-group-text">http://</span>
        <span class="input-group-text">www.mysite.edu/</span>
      </div>
      <input type="text" class="form-
control" placeholder="Tu nombre de usuario">
      <div class="input-group-append">
        <span class="input-group-text">/site</span>
      </div>
    </div>

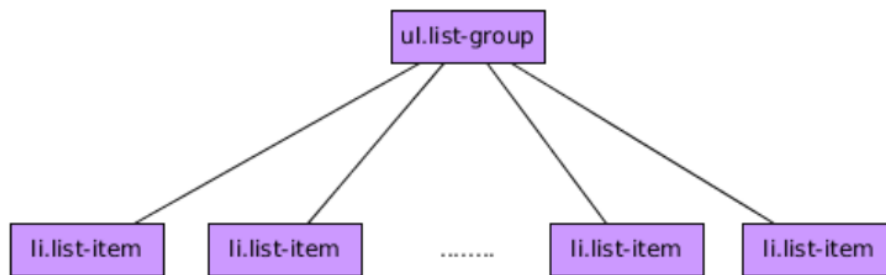
    <label class="mt-5">Input con Radio</label>
    <div class="input-group">
      <div class="input-group-prepend">
        <div class="input-group-text">
          <input type="radio" />
        </div>
      </div>
      <input type="text" class="form-
control" placeholder="Campo de texto asociado al radio">
    </div>
  </form>
</div>

```

List-Group

El componente List-Group es un componente que nos va a permitir mostrar una serie de contenidos.

Estructura dentro de un List-Group



```
<ul class="list-group">
  <li class="list-group-item">Opción 1</li>
  <li class="list-group-item">Opción 2</li>
  <li class="list-group-item">Opción 3</li>
  <li class="list-group-item">Opción 4</li>
</ul>
```

Usando las clases **active** y **disabled** puedo indicar qué elemento se destaca como activo o se muestra como deshabilitado.

Si quiero evitar los bordes y las esquinas redondeadas tengo que añadir la clase **list-group-flush** al elemento principal.

No debo usar las clases **btn** y **btn-X** en este componente pero si quiero usar la paleta de colores como fondo de los elementos **list-group-item** debo añadir a estos elementos alguna de las clases **list-group-item-X** siendo X uno de los colores de la paleta de colores de Bootstrap 4.

IMPORTANTE: Dentro de estos elementos **list-group-item** puedo añadir casi cualquier elemento HTML.

Si queremos conseguir el efecto hover tendremos que cambiar la estructura dejando de usar listas, usando etiquetas como **a** o **button** como elementos *list-group-item* y adicionalmente agregando a esos elementos la clase *list-group-item-action*.

```
<div class="list-group">
  <a href="#" class="list-group-item list-group-item-
action active">Enlace 1</a>
  <a href="#" class="list-group-item list-group-item-action">Enlace 2</a>
  <a href="#" class="list-group-item list-group-item-
action disabled">Enlace 3</a>
  <a href="#" class="list-group-item list-group-item-action">Enlace 4</a>
</div>
```

Usando el componente list-group y el código javascript que viene con Bootstrap 4 podemos simular paneles de contenidos.

```
<div class="list-group">
  <a class="list-group-item list-group-item-action active"
data-toggle="list" href="#panel1">Panel 1</a>
  <a class="list-group-item list-group-item-action"
data-toggle="list" href="#panel2">Panel 2</a>
  <a class="list-group-item list-group-item-action"
data-toggle="list" href="#panel3">Panel 3</a>
  <a class="list-group-item list-group-item-action"
data-toggle="list" href="#panel4">Panel 4</a>
</div>

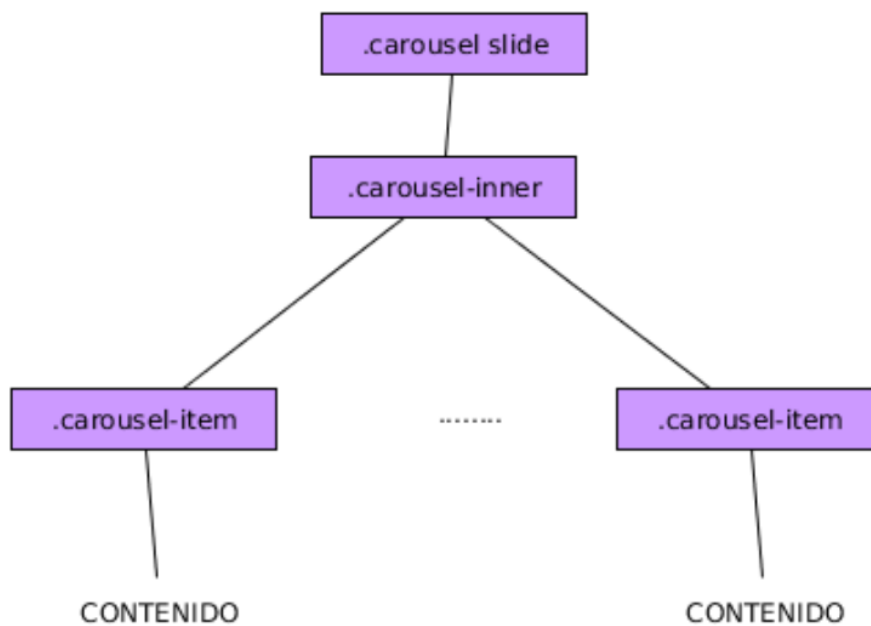
<div class="tab-content">
  <div class="tab-
pane fade show active" id="panel1"><h1>Contenido Panel 1</h1></div>
  <div class="tab-
pane fade" id="panel2" ><h1>Contenido Panel 2</h1></div>
  <div class="tab-
pane fade" id="panel3" ><h1>Contenido Panel 3</h1></div>
  <div class="tab-
pane fade" id="panel4" ><h1>Contenido Panel 4</h1></div>
</div>
```

- Se ha añadido al elemento con la clase *list-item-group* el atributo *data-toggle="list"* y se ha establecido el *href* al id del panel que queremos mostrar (se puede hacer también con *data-target*).
- Los paneles se han puesto dentro del elemento con las clase *tab-content* y cada uno de los paneles tiene la clase *tab-pane*.
- La clase *fade* es para conseguir un efecto de desvanecimiento al usarla.

Carousel

El componente Bootstrap 4 **carousel** es un componente que hemos visto en muchas páginas web y que se encarga de ir mostrándonos diversos contenidos uno tras otro y efectuando un efecto cuando se produce este cambio. Normalmente suelen ser galerías de imágenes.

Estructura dentro de un Carousel



- El elemento padre de todo el que tiene las clases **carousel** y **slide** teniendo además que tener establecido el atributo **data-ride=carousel**. Si no tenemos la clase **slide** no habrá un efecto de deslizamiento.
- **carousel-inner** es la clase del elemento que contiene los distintos elementos que mostrará el carrusel.
- **carousel-item** es la clase que debe tener cada uno de los elementos que mostrará el carrusel. Lo que pongamos dentro de cada uno de ellos depende de nosotros.

```
<div class="carousel slide" data-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
```



```

    <div class="carousel-item">
      
    </div>
  </div>
</div>

```

IMPORTANTE: Un error muy común es no elegir uno de los elementos como **active**. Si no lo hacemos el carrusel no funcionará.

IMPORTANTE: El componente no se encarga de dimensionar los contenidos. Debemos hacerlo nosotros usando los estilos necesarios o las utilidades BootStrap 4 que consideremos.

Si queremos añadir controles a nuestro *carousel* debemos añadir como hijos del elemento con la clase **carousel**, y después del elemento con la clase **carousel-inner** los siguientes elementos

```

<a class="carousel-control-prev" href="#MyCarouselId" data-slide="prev">
  <span class="carousel-control-prev-icon" aria-hidden="true"></span>
  <span class="sr-only">Anterior</span>
</a>
<a class="carousel-control-next" href="#MyCarouselId" data-slide="next">
  <span class="carousel-control-next-icon" aria-hidden="true"></span>
  <span class="sr-only">Posterior</span>
</a>

```

IMPORTANTE #MyCarouselId debe ser el id del elemento que posea la clase *carousel*.

Si queremos añadir indicadores al *carousel*, para poder movernos con total libertad por él, deberemos añadir como hijo del elemento con la clase **carousel** una estructura similar al siguiente ejemplo:

```

<ol class="carousel-indicators">
  <li data-target="#MyCarouselId" data-slide-to="0" class="active"></li>
  <li data-target="#MyCarouselId" data-slide-to="1"></li>
  <li data-target="#MyCarouselId" data-slide-to="2"></li>
</ol>

```

Si queremos añadir leyendas a los elementos del *carousel* deberemos añadir lo siguiente detrás del contenido de los elementos con la clase **carousel-item**.

```

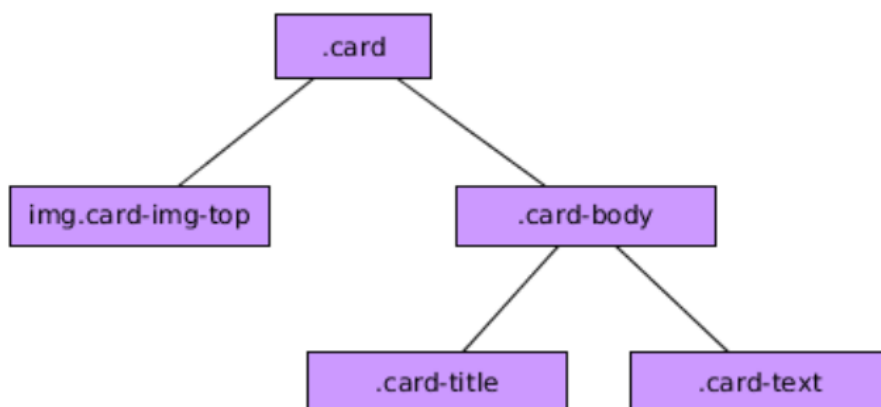
<div class="carousel-caption d-none d-md-block">
  ...
  Contenido de la leyenda
  ...
</div>

```

Card

El componente Bootstrap 4 **Card** es uno de las novedades de esta nueva versión y viene a sustituir a los viejos componentes *panels*, *wells* y *thumbnails*.

Estructura dentro de un Card



- El elemento con la clase **card** es el elemento padre que contiene todo.
- La imagen con la clase **img-card-top** es una imagen que se muestra en la zona superior del componente.
- El elemento con la clase **card-body** es el contenido del componente. Estará compuesto a su vez con el título **card-title** (normalmente etiquetas hX) y el texto de la card **card-text**.

No todos los elementos son obligatorios y podemos jugar a añadir y quitarlos para ver la apariencia final. Además podemos añadir otros elementos (según necesitemos) dentro de esta estructura.

```
<div class="card">
  
  <div class="card-body">
    <h3 class="card-title">Ejemplo de Card</h3>
    <div class="card-text">
      Texto descriptivo que queremos añadir al card
    </div>
  </div>
</div>
```

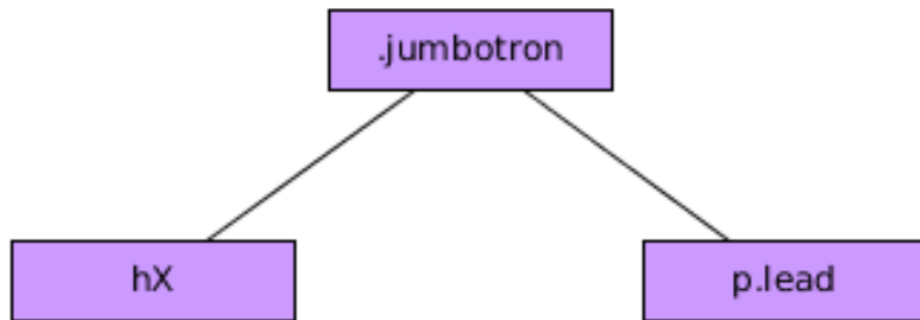
- Podemos también añadir un subtítulo a la card con un elemento que tenga la clase **class-subtitle** justo detrás del título.
- Los enlaces tienen una clase especial con estilos propios **card-link**.
- Puedo añadir cabeceras y pies a la Card posicionando los elementos en el lugar correcto y usando las clases **card-header** y **card-footer**.

- Las *Cards* por defecto ocupan todo el ancho del elemento padre pero esto podemos variarlo usando clases Bootstrap 4 para maqueta y/o utilidades Bootstrap 4 (w-25,w-50, ...).
- Puedo situar la imagen al final de la *card* usando la clase ***img-card-bottom*** y moviendo la etiqueta al final.
- Puedo hacer que la imagen se use de fondo poniéndole la clase ***img-card*** y cambiando la clase del ***card-body*** por ***card-img-overlay***.

Jumbotron

El componente Bootstrap ***Jumbotron*** es un componente sencillo que se utiliza para mostrar mensajes y/o títulos

Estructura dentro de un Jumbotron



```

<div class="jumbotron">
  <h1>Título de Jumbotron</h1>
  <p class="lead">
    Párrafo con una clase <i>lead</i> para mensajes secundarios.
  </p>
</div>
  
```

Como podemos ver, este componente modifica las tipografías, añade un color de fondo y unos bordes redondeados.

Si, por el contrario, preferimos que ocupe todo y no tener bordes redondeados debemos añadir la clase ***jumbotron-fluid*** al div con la clase ***jumbotron*** y añadir como hijo de este elemento un elemento con las clases ***container*** o ***container-fluid***.

```
<div class="jumbotron jumbotron-fluid">
  <div class="container">
    <h1>Otro Jumbotron</h1>
    <p class="lead">
      Sin border y ocupando todo.
    </p>
  </div>
</div>
```