DISEÑO DE INTERFACES WEB
TEMA2: BOOTSTRAP

CFGS – DIW 2

JESUITAS
Sagrado Corazón
Jesuitas · Logroño

Indice

¿Porqué BootStrap 4?

FÁCIL

De instalar, de usar...y muy bien documentado.

USADO

Millones de sites y muy importantes

¿Porqué BootStrap 4?

MANTENIDO

Continua mejora y corrección de errores

TODO IGUAL

Aunque esto tiene solución

Interfaces web

- Principios básicos sobre el diseño de interfaces de usuario
  - Interfaces claras y precisas
    - Antes que cualquier otra cosa el usuario debe reconocer bien lo que está viendo, saber para qué se usa, y entender cómo la interfaz le ayudará a interactuar con la aplicación.
    - Al diseñar elementos de manera clara creamos confianza en el usuario, pero si en vez de eso le entregas una página desordenada y además con una interfaz poco familiar, el usuario puede tener problemas para usar tu página web.

Novedades BootStrap 4

- Grandes cambios

USA

FLEX

Todo BootStrap 4 está basado en el uso de componentes CSS Flex

USA

SASS

Sustituyendo a Less como preprocesador CSS

USA

REM

En vez de em como unidad de medida para el tamaño de las fuentes

30% más ligero

Novedades BootStrap 4

- NUEVOS COMPONENTES DESTACADOS
  - Cards
  - Carousel
  - Modales
  - Popovers
  - Tooltips
- Además mejora muchos de los anteriores….

Novedades BootStrap 4

- OTRAS NOVEDADES
  - No más IE8
  - Soporte ES6
  - No Glyphycons
  - Utilidades (clases) de márgenes y paddings
  - Estilo plano, con menos gradientes
  - Nueva paleta de colores
  - Soporte para pantallas de más de 1200px col-X-xl
  - .offset-X-Y en vez de col-offset-X-Y
- Y muchos más….

Instalacción

- Para instalar BootStrap 4 podemos optar por dos opciones:
  - Usar el los enlaces que nos proporcionan en la documentación:
  - El primero de ellos es el CSS de la librería

```
<!-- Bootstrap CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css"
integrity="sha384-PsH8R72JQ3SOdhVi3uxftmaW6Vc51MKb0q5P2rRUpPvrszuE4W1povHYgTpBfshb" crossorigin="anonymous">
```

Instalacción

- Y posteriormente si vamos a usar componentes, muchos de los cuáles no son elementos estáticos y van a requerir scripts para definir su comportamiento

```
<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.3/umd/popper.min.js" integrity="sha384-
vFJXuSJphROIrBnz7yo7oB41mKfc8JzQZiCq4NCceLEaO4IHwicKwpJf9c9IpFgh" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js" integrity="sha384-
alpBpkh1PFOepccYVYDB4do5UnbKysX5WZXm3XxPqe5iKTfUKjNkCk9SaVuEZflJ" crossorigin="anonymous"></script>
```

Instalacción

- Los dos primeros scripts son para referenciar las librerías **jQuery** y **Popper** de las que depende BootStrap.
- Descargar los ficheros necesarios para no tener que pedirlos cada vez que se muestre nuestra página.
- Esos ficheros podemos descargarlos de **aquí** y en vez de usar URLs podremos referenciar los ficheros localmente.
- En cualquier caso para este curso se proporciona un fichero base HTML (template.html) con todo lo necesario para poder practicar.
- **IMPORTANTE:** Para la realización de este curso se ha usado la versión beta de BootStrap 4. Conforme los desarrolladores vayan publicando nuevas versiones habría que actualizar los enlaces.

Contenedores

- **¿Qué es un contenedor?**
- Un contenedor es el “padre” de todos los elementos de nuestra página web. Es una etiqueta que. como regla general, va a contener todas las otras etiquetas del contenido de nuestra página.

Contenedores

- **¿Qué tipos de contenedores nos ofrece BootStrap?**
- **container** : Ocupará diferentes anchos dependiendo del tamaño de la pantalla. Puede ocupar todas la pantalla o dejar unos márgenes a izquierda a derecha aunque, en este caso, siempre estará centrado.

```
<body>

<div class="container">

</div>

</body>
```

Contenedores

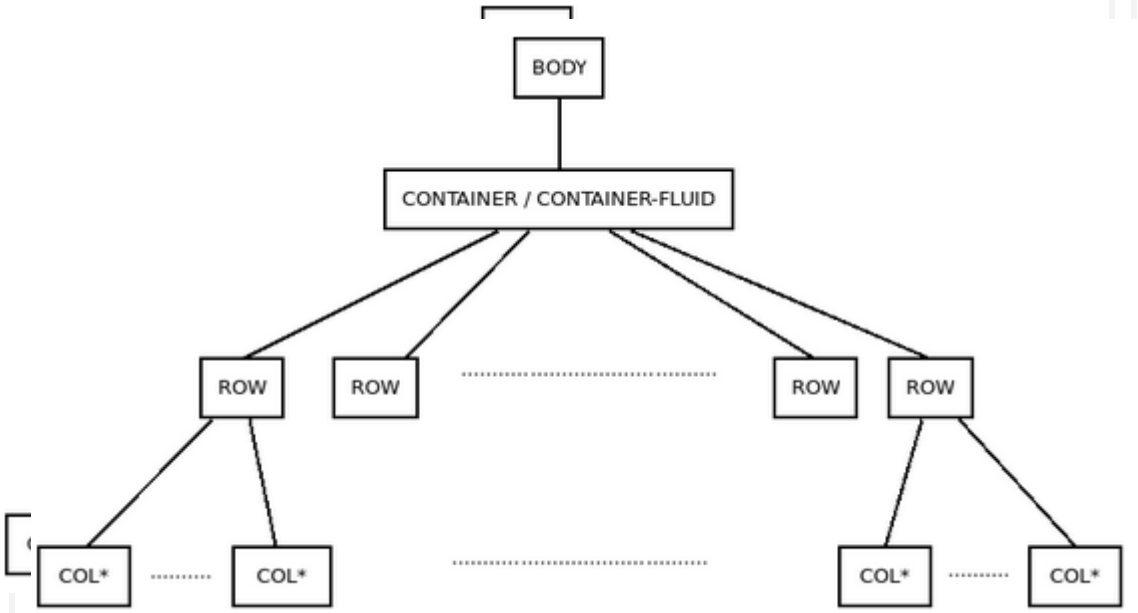
- **¿Qué tipos de contenedores nos ofrece BootStrap?**
- **container-fluid**: Ocupará todo el ancho (100%) de lo que podemos ver en el navegador.

```
<body>

  <div class="container-
fluid">

  </div>

</body>
```

- **IMPORTANTE:** Aunque los contenedores se pueden anidar unos dentro de otros son pocas las páginas que son tan complejas para necesitar contenedores anidados.

Grid de bootstrap: las 12 columnas

- Como ya hemos hablado en el capítulo anterior es muy difícil que nos encontremos con páginas cuyos layouts sean tan complejos como para necesitar más de un contenedor. Así que, si nos ponemos de acuerdo en eso, en que sólo vamos a necesitar un contenedor, podemos fijar una estructura general que va a permanecer fijas para todos nuestros layouts:



Grid de bootstrap: las 12 columnas

- Y dentro de cada fila de cada row, en principio podemos tener hasta 12 columnas, aunque ya veremos posteriormente que dado a que BootStrap 4 utiliza contenedores flex podemos usar las propiedades de estos contenedores flex para poner más columnas aunque, en realidad, es difícil que tengamos que utilizar más.
- Todas estas columnas están pegadas unas a otras y tiene siempre un padding a izquierda y a derecha de 15px del que hablaremos posteriormente.
- Otra cosa importante es que no tenemos que olvidarnos de utilizar *class="row"* ya que si nuestra estructura de columnas no está dentro de un contenedor con esa clase perderemos todas las propiedades de los contenedores flex, que son los sustentan la maquetación en BootStrap 4.

BreakPoints

- Existen multitud de tipos de dispositivos, multitud de pantallas, pantallas que giran etc. Por lo tanto la elección de los distintos breakpoints no es algo trivial.
- Sin embargo, tenemos la suerte de que los muchachos de Twitter, usando su experiencia y los datos que sus millones de usuarios les proporcionan, has decidido que la elección más adecuada son 4 breakpoints:
  - En **576px**
  - En **768px**
  - En **992px**
  - En **1200px**

BreakPoints

- Teniendo en cuenta estos 4 breakpoints podemos diferenciar entre 5 tipos de pantallas
  - **Pantallas extra pequeñas:** Entre 0 y 576px
  - **Pantallas pequeñas:** Entre 576px y 768px
  - **Pantallas medianas:** Entre 768px y 992px
  - **Pantallas grandes:** Entre 993px y 1200px
  - **Pantallas extra-grandes:** A partir de 1200px

BreakPoints

	Extra pequeñas	Pequeñas	Medianas	Grandes	Extra Grandes
Anchura máxima	576px	768px	992px	1200px	> 1200px
Prefijo de la clase	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
Número de columnas	12				
Anchura del Gutter	30px (1px a cada lado de cada columna)				
Anidar	Sí				
Reordenar columnas	Sí				

BreakPoints

- Teniendo en cuenta estos 4 breakpoints podemos diferenciar entre 5 tipos de pantallas
  - **Pantallas extra pequeñas:** Entre 0 y 576px
  - **Pantallas pequeñas:** Entre 576px y 768px
  - **Pantallas medianas:** Entre 768px y 992px
  - **Pantallas grandes:** Entre 993px y 1200px
  - **Pantallas extra-grandes:** A partir de 1200px

Grid de bootstrap: las 12 columnas

- Una cosa importante que hay que tener en cuenta es que si establecemos un tamaño para un cierto tamaño de pantallas ese tamaño se va a mantener para pantallas de mayor tamaño pero para pantallas menores ese div ocupará toda la pantalla, es decir si tenemos.
- Este div ocupará la mitad de la pantalla para pantallas de 768px (recordar los breakpoints) o superiores pero para tamaño menores ocupará todo el ancho de la fila que lo contiene.

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

Grid de bootstrap: las 12 columnas

- Podemos también fijar a la vez anchos diferentes para distintos tipos de tamaños de pantallas. Para ello añadiremos varias clases a un mismo elemento.
- En este caso ese div:
  - Ocupará toda la fila para pantallas extrapequeñas (<576px)
  - Ocupará la mitad de la fila para pantallas en 576px y 768px
  - Ocupará un tercio de la fila para pantallas entre 768px y 992px
  - Ocupará un cuarto para pantallas mayores de 992px

```
  <div class="col-sm-6 col-md-4 col-
xl-3">
  </div>
```

Grid de bootstrap: las 12 columnas

- También es importante recordar que si, en un *row* nos pasamos de 12 columnas ese elemento que provoca que nos pasemos de las 12 columnas "*pasará*" debajo de los elementos de la fila actual ocupando el tamaño establecido.
- Así por ejemplo si tenemos la siguiente estructura el tercer div irá abajo ya que 6+4+4 = 14 que es mayor que 12.

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

Grid de bootstrap: las 12 columnas

- Si mientras estamos construyendo nuestro grid y queremos que haya un salto de línea podemos forzarlo de la siguiente manera:

```
<div class="row">

  <!-- Ocupamos 5/12 de la fila -->
  <div class="col-md-3"></div>
  <div class="col-md-2"></div>

  <!-- Forzamos el salto de línea -->
  <div class="w-100"></div>

  <!-- Ocupamos otra vez 5/12 pero en una nueva línea
-->
  <div class="col-md-3"></div>
  <div class="col-md-2"></div>


</div>
```

Grid de bootstrap: las 12 columnas

- Por defecto BootStrap 4 introduce separación entre el contenido de las columnas mediante *paddings* y márgenes a derecha e izquierda. Es lo que se denomina *gutter*
- Si ése no es el efecto que queremos en nuestra maquetación podemos evitarlo añadiendo al elemento que tenga la clase *row* la clase *no-gutters*.
- Así de esta manera las columnas o elementos de la fila no tendrán ese padding con la posibilidad de que el contenido de la misma queden pegados.
- Un ejemplo sería:

```
<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">
    <div class="col-md-6"><h3>Elemento sin gutter</h3>
</div>
    <div class="col-md-6"><h3>Segundo Elemento</h3></div>
  </div>

</div>
```

Grid de bootstrap: las 12 columnas

- **Desplazamiento de columnas**
- Tal y como hemos visto en apartados anteriores podemos dar diferentes tamaños a los distintos componentes o columnas de una *row* o fila.
- Adicionalmente podemos establecer no sólo el tamaño de la columna si no también la posición a partir de la cuál se va a posicionar.
- Para conseguir eso debemos añadir la clase *offset-X* donde X es el número de columnas a la izquierda que desplazaremos el elemento que queremos posicionar.

Grid de bootstrap: las 12 columnas

- **Desplazamiento de columnas**
- 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.

Grid de bootstrap: las 12 columnas

- **Desplazamiento de columnas**
- 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.

```
<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>
```

Grid de bootstrap: las 12 columnas

- **Alineación de las filas**
- Como ya hemos descrito en apartados anteriores, BootStrap 4 utiliza contenedores flex. Precisamente utilizando las propiedades de estos elementos podremos fijar la alineación de los componentes de las *rows* o filas.
- 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.

Grid de bootstrap: las 12 columnas

- **Alineación de las filas**
- ***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 verticalemente con el borde superior de la fila.
  - align-items-center : Los elementos de la fila se alinearán centrados verticalmente.
  - align-items-center : Los elementos de la fila se alinearán centrados verticalmente.

Grid de bootstrap: las 12 columnas

- **Alineación de las filas**
- ***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.
  - **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.

Grid de bootstrap: las 12 columnas

- **Orden de las columnas**
- Por defecto,los elementos de una *row* se sitúan según el orden que hemos establecido al escribir la estructura de nuestro archivo HTML.
- Sin embargo, mediante clases de BootStrap podemos cambiar el orden en que dichos elementos van a aparecer en nuestra pantalla.
- Para ello utilizaremos la clase *.order-X* siendo X el orden en el que queremos situar dicho elemento dentro de nuestra fila.

```
<div class="row">

  <div class="col-md-4 order-3">
    Primer elemento. Posición Tercera
  </div>
  <div class="col-md-4 order-1">
    Segundo elemento. Posición Primera
  </div>
  <div class="col-md-4 order-2">
    Primer elemento. Posición Segunda
  </div>
<div>
```

Grid de bootstrap: las 12 columnas

- **Orden de las columnas**
- **Importante:** Debemos poner *order-X* a todos los elementos de la columna. Si no lo hacemos, aquellos que no lo tengan serán mostrados los primeros y en el orden establecido por defecto.
- De igual manera, nos puede interesar únicamente reordenar a partir de cierto tipo de pantallas.Para eso tendremos en cuenta los breakpoints explicados en capítulos anteriores y usaremos:
  - order-sm-X
  - order-md-X
  - order-lg-X
  - order-xl-X

Grid de bootstrap: las 12 columnas

- **Márgenes entre columnas**
- 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.

Grid de bootstrap: las 12 columnas

- **Márgenes entre columnas**

  m{lado}-sm-{tamaño}

  - 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.

Grid de bootstrap: las 12 columnas

- **Márgenes entre columnas** **m{lado}-sm-{tamaño}**
  - Y pudiendo ser tamaño de tamaño:
    - **0 :** No hay margen
    - **1 :** 0.25rem
    - **2 :** 0.5rem
    - **3 :** 1rem
    - **4 :** 1.25rem
    - **5 :** 3rem
    - **auto :** Para clases que establecen una margen auto

Contenido en bootstrap

- IMÁGENES
- Una imagen es únicamente el elemento gráfico el cual hemos añadido mediante las etiquetas img o picture.
  - <img src="ejemplo.png">
- 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>
  - <img src="ejemplo.png">
  - <figcaption>Texto descriptivo de la imagen</figcation>
  - </figure>

Contenido en bootstrap

- IMÁGENES
- **Figuras**
- 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 añadir a la etiqueta *figcation* 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.

Contenido en bootstrap

- TABLAS
- **Haciendo que la tabla sea tabla Bootstrap**
  - 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>

Contenido en bootstrap

- TABLAS
- **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>

Contenido en bootstrap

- TABLAS
- **Clases para la cabecera de las tablas**
  - Para dar estilos a las cabecera 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).

Contenido en bootstrap

- TABLAS
- **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 las clase table-hover a las etiqueta table.
    - <table class="table table-hover">
    - .....
    - </table>

Contenido en bootstrap

- TABLAS
- **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>

Contenido en bootstrap

- TABLAS
- **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 las clase BootStrap table-responsive
    -
    - <table class="table table-bordered">
    - .....
    - </table>
    -

Contenido en bootstrap

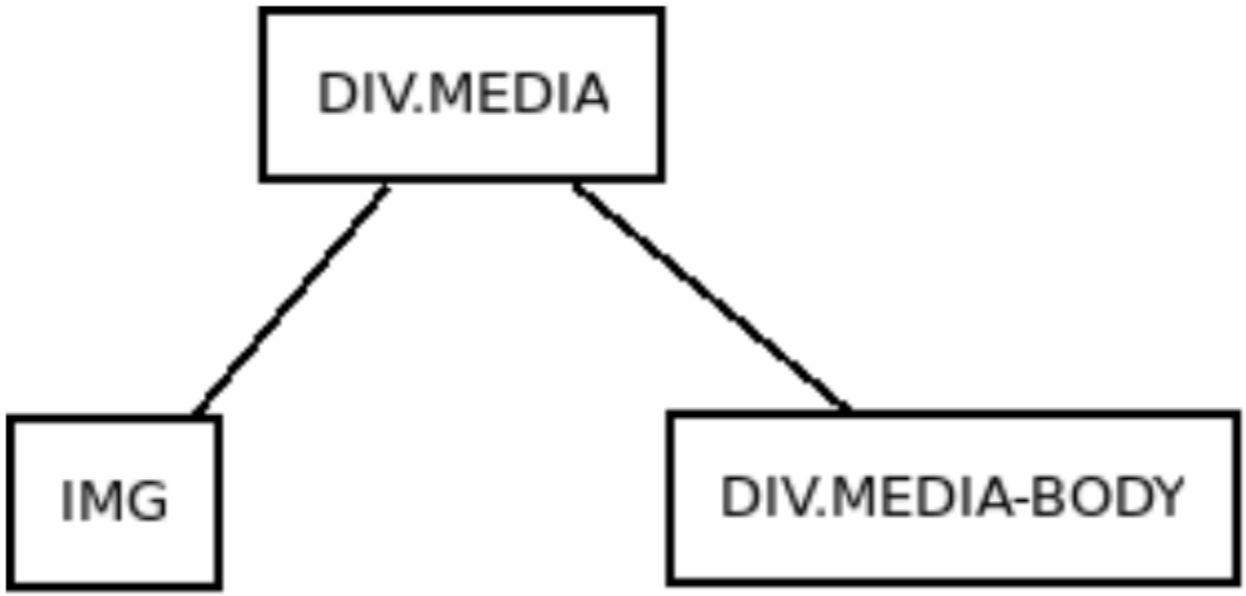
- TABLAS
  - **Haciendo las tablas responsivas**
    - También puedo fijar el breakpoint a partir de la cual quiero que la tabla sea responsiva. En esos casos usaré:
    - ***table-responsive-sm***
    - ***table-responsive-md***
    - ***table-responsive-lg***
    - ***table-responsive-xl***

Contenido en bootstrap

- ELEMENTOS MULTIMEDIA
- Para entender lo que es un objeto media lo mejor es irnos a la propio 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…

Contenido en bootstrap

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




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

Contenido en bootstrap

- ELEMENTOS MULTIMEDIA - MEDIA
- 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.

Contenido en bootstrap

- ELEMENTOS MULTIMEDIA - MEDIA
- 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).
- ***IMPORTANTE :*** Debemos darle las dimensiones a las imágenes.

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

</ul>
```

Contenido en bootstrap

- ELEMENTOS MULTIMEDIA – OBJETOS INCRUSTRADOS
- 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>
```

Contenido en bootstrap

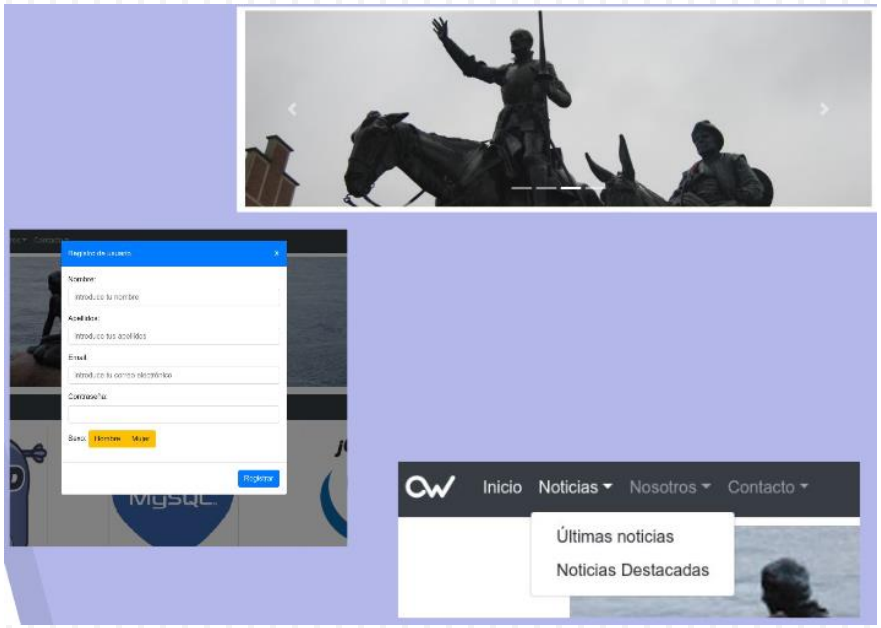
- ELEMENTOS MULTIMEDIA – OBJETOS INCRUSTRADOS
- Adicionalmente podemos modificar las proporciones elementos (su *aspect ratio*) añadiendo al elemento padre clases que lo modificarán:
  - ***embed-responsive-21by9*** **:** *Aspect Ratio 21x9*
  - ***embed-responsive-16by9*** **:** *Aspect Ratio 16x9*
  - ***embed-responsive-4by3*** **:** *Aspect Ratio 4x3*
  - ***embed-responsive-1by1*** **:** *Aspect Ratio 1x1*

Utilidades Bootstrap

- Clases que yo pongo a mis elementos para que tengan otro aspecto.
- https://getbootstrap.com/docs/4.0/utilities/display/

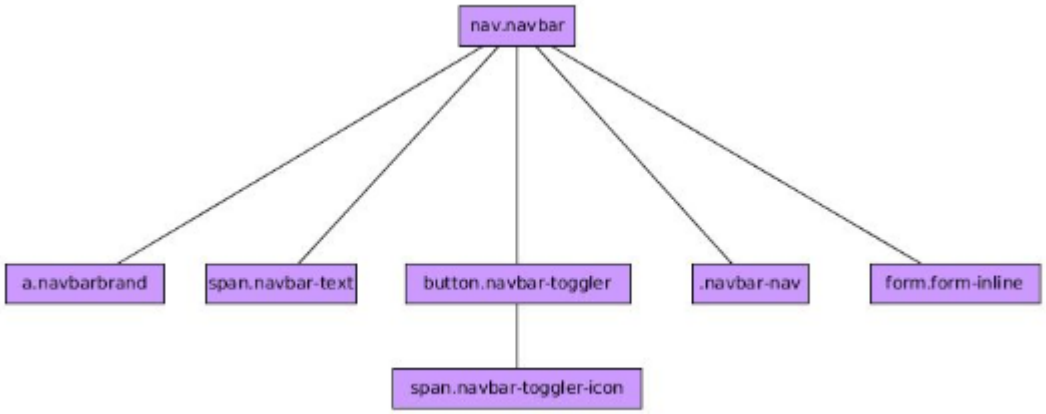
Componentes Bootstrap

- Elemento comunes en muchas web que mediante herramientas como BootStrap podemos reutilizarlos de manera fácil ya que nos proporcionan plantillas y estilos para ello



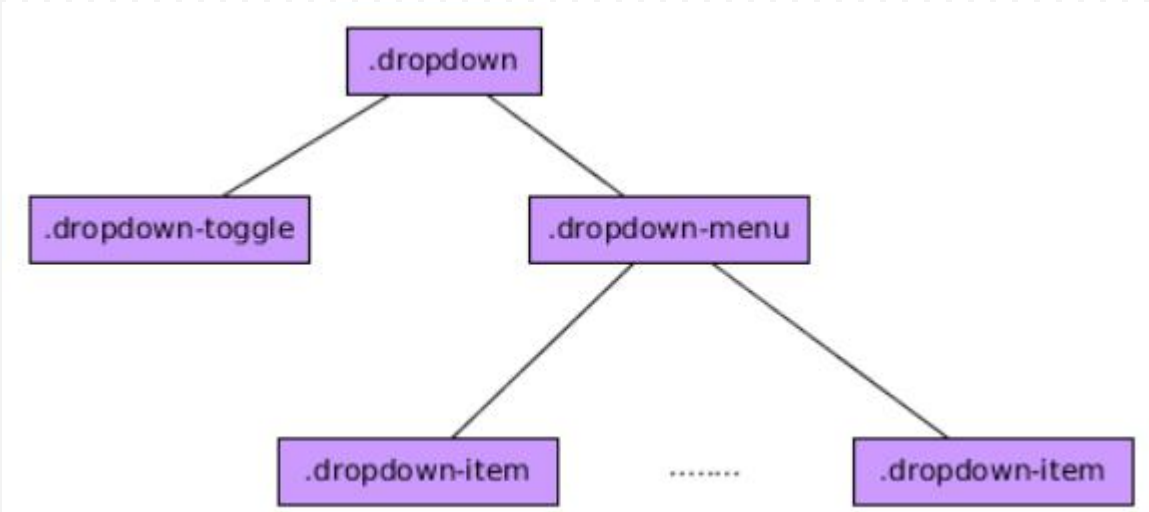
Componentes Bootstrap

- LOS COMPONENTES BOOTSTRAP
  - Son muchos.
  - Todos tienen muchas variantes y posibilidades.
- ÁRBOLDOM
  - Partiremos de un ejemplo general y sencillo e iremos añadiendo cosas. No serán todas pero será un inicio...



Componentes Navegación - 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.
- La estructura del DOM y las clases BootStrap 4 que debe tener un elemento Dropdown son las siguientes en su configuració básica:



Componentes Navegación - dropdown

- Siendo:
  - ***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.

Componentes Navegación - dropdown

- Es importante resaltar que en elemento con la clase ***dropdown-toggle*** debe también poseer el atributo ***data-toggle=”dropdown”*** para que el elemento funcione correctamente.
- Por supuesto los colores de los botones los podemos cambiar usando la paleta de colores de BootStrap 4.

```
<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>
```

Componentes Navegación - dropdown

- **Separar el botón de las opciones.**
- Si queremos dividir el botón que sirve para mostrar el submenú en dos, el texto y la flecha para desplegar tenemos que modificar un poco la estructura (aunque esta opción la encuentro de poca utilidad) y tener algo similar al siguiente ejemplo:

```
<div class="dropdown">
    <button class="btn btn-primary">Muestra el submenú</button>
    <div class="btn btn-primary dropdown-toggle dropdown-toggle-split" 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>
```

Sin título: Bloc de notas

Componentes Navegación - dropdown

- **Añadir separador en el submenú**
  - Si queremos añadir un separador dentro del submenú desplegable debemos añadir el siguiente elemento entre los grupos de opciones que queramos separar.
  -
- **Modificar el tamaño**
  - Podemos hacerlo añadiendo las clases btn-lg o btn-sm a los elementos botones que nos sirven para mostrar/ocultar el menú.
- **Dirección en la que aparece el menú.**
  - Podemos mostrar el submenú en las cuatro direcciones sólo con añadir las clases dropup, dropright, dropleft al elemento general. Si no ponemos nada la opción por defecto es hacia abajo. IMPORTANTE Tiene que caber….si no coge la opción por defecto.

Componentes Navegación - dropdown

- **Añadir separador en el submenú**
  - Si queremos añadir un separador dentro del submenú desplegable debemos añadir el siguiente elemento entre los grupos de opciones que queramos separar.
  -
- **Modificar el tamaño**
  - Podemos hacerlo añadiendo las clases btn-lg o btn-sm a los elementos botones que nos sirven para mostrar/ocultar el menú.
- **Dirección en la que aparece el menú.**
  - Podemos mostrar el submenú en las cuatro direcciones sólo con añadir las clases dropup, dropright, dropleft al elemento general. Si no ponemos nada la opción por defecto es hacia abajo. IMPORTANTE Tiene que caber….si no coge la opción por defecto.

Componentes Navegación - dropdown

- **Alineación del submenú**
  - Por defecto los submenú se posicionan a la izquierda pero si quiero puedo posicionarlos a la derecha añadiendo la clase ***dropdown-menu-right*** al elemento que ya tiene la clase ***dropdown-menu***.
- **Más consideraciones:**
  - Puedo añadir sin problemas formularios dentro de los elementos con la clase ***dropdown-menu*** aunque tendré que ajustar posteriormente su posición.
  - Para hacer que un elemento sea el elemento activo del submenú deberé añadir la clase ***active*** y para hacer que esté deshabilitado le añadire la clase ***disabled***.

Componentes Navegación - dropdown

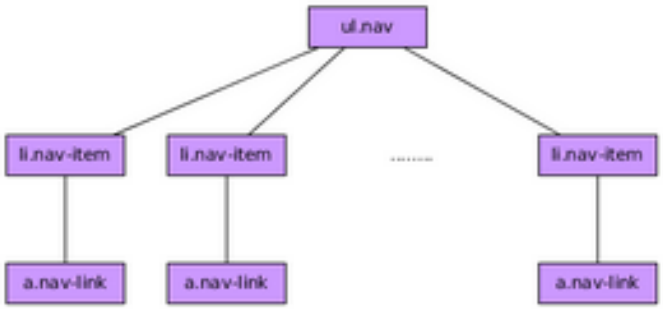
- **Funciones asociadas**
  - Las más importantes son, bajo mi punto de vista:
  - ***.dropdown('toggle')*** Cambia el estado del dropdown de un elemento de navegación
  - ***.dropdown('update')*** Actualiza la posición de un elemento.
  - ***.dropdown('dispose')*** Destruye el componente.
- **Eventos asociados**
  - La interacción con este tipo de elementos genera 4 nuevos tipos de eventos, bastante auto explicativos por su nombre y de los cuáles, para saber más detalles deberemos visitar el manual de referencia.
    - ***show.bs.dropdown***
    - ***shown.bs.dropdown***
    - ***hide.bs.dropdown***
    - ***hidden.bs.dropdown***

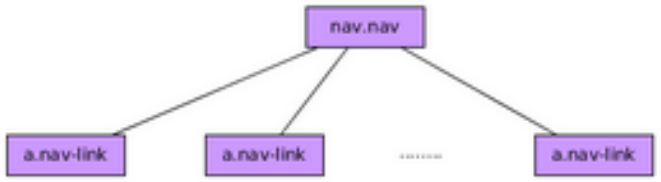
Componentes Navegación - nav

- El componente BootStrap 4 ***Nav*** es el componente básico de BootStrap 4 para formar menús y barras de navegación. Posteriormente veremos que existe otro componente que nos va a permitir realizar barras de navegación más complejas, el componente ***NavBar***.
- Este componente tiene bastantes posibilidades pero, en su estructura más sencilla, podemos optar por una jerarquía formada a partir de listas HTML o bien por una jerarquía formada a partir de la etiqueta semántica nav. Personalmente recomiendo la segunda, es más corta y aporta significado. En todo caso podemos ver esas jerarquías aquí:

Componentes Navegación - nav

- El componente BootStrap 4 ***Nav*** es el componente básico de BootStrap 4 para formar menús y barras de navegación. Posteriormente veremos que existe otro componente que nos va a permitir realizar barras de navegación más complejas, el componente ***NavBar***.
- Este componente tiene bastantes posibilidades pero, en su estructura más sencilla, podemos optar por una jerarquía formada a partir de listas HTML o bien por una jerarquía formada a partir de la etiqueta semántica nav. Personalmente recomiendo la segunda, es más corta y aporta significado. En todo caso podemos ver esas jerarquías aquí:





Componentes Navegación - nav


```html
<!-- 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>
```

- Para alinear los elementos del menú podemos usar las utilidades flex de BootStrap 4.
- 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.

Componentes Navegación - nav

- **Consideraciones adicionales.**
- Para que el menú ocupe toas la anchura del padre,si hemos construido el menú con listas debemos añadir la clase ***nav-fill*** al elemento padre que contiene la clase ***nav*** y adicionalmente, añadir la clase ***nav-item*** a los enlaces si hemos usado la etiqueta *nav*.
- Si queremos que todos los elementos del menú ocupen lo mismo en anchura debemos añadir la clase ***nav-justified*** al elemento padre y adicionalmente, añadir la clase ***nav-item*** a los enlaces si hemos usado la etiqueta *nav*.
- Podemos construir menús anidados usando el componente ***dropbdown*** que hemos visto anteriormente siendo el elemento padre del ***dropdown*** el que contiene la clase ***nav-item***..

Componentes Navegación - nav

- **Construcción de un menú dinámico de pestañas/píldoras con contenidos**
- Usando este componente podemos crear paneles de contenidos animados de tal manera que, dependiendo de en cuál de las opciones del menú hagamos click, se nos mostrará una panel de contenidos u otro.
- IMPORTANTE: En este caso no podemos meter componentes dropdown dentro del menú.
- Para todo esto necesitaremos añadir ciertos atributos a nuestros elementos. Lo mejor será verlo con un ejemplo de nav usando la etiqueta nav (con lista es similar).

Componentes Navegación - nav

- **Construcción de un menú dinámico de pestañas/píldoras con contenidos**

```
<!-- 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.

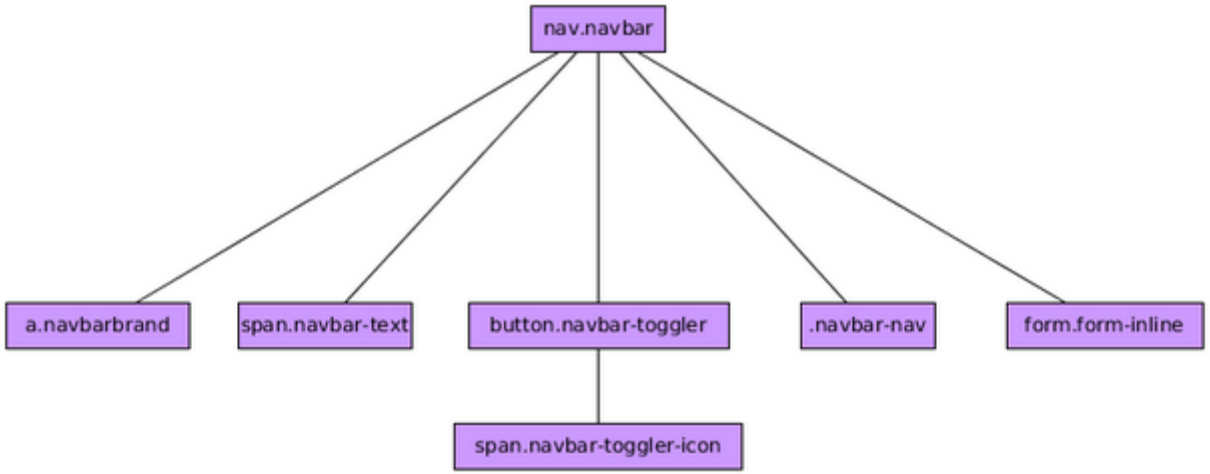
Componentes Navegación - nav

- **Funciones asociadas**
  - ***.tab()*** Activa una pestaña. Debe tener el atributo ***data-target*** o el ***href*** correcto.
  - ***.tab('show')*** Dada una pestaña muestra el panel asociado.
  - ***.tab('dispose')*** Destruye el componente.Deja de funcionar como tal.
- **Eventos asociados**
- La interacción con este tipo de elementos genera 4 nuevos tipos de eventos, bastante auto explicativos por su nombre y de los cuáles, para saber más detalles deberemos visitar el manual de referencia.
  - ***hide.bs.tab***
  - ***show.bs.tab***
  - ***hidden.bs.tab***
  - ***shown.bs.tab***

Componentes Navegación - 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*
- 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.

Componentes Navegación - navBar



Componentes Navegación - navBar


```
<nav class="navbar navbar-expand-sm navbar-dark bg-dark">
  <a class="navbar-brand" href="#"><img src="logo-openwebinars.png"/></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>
```

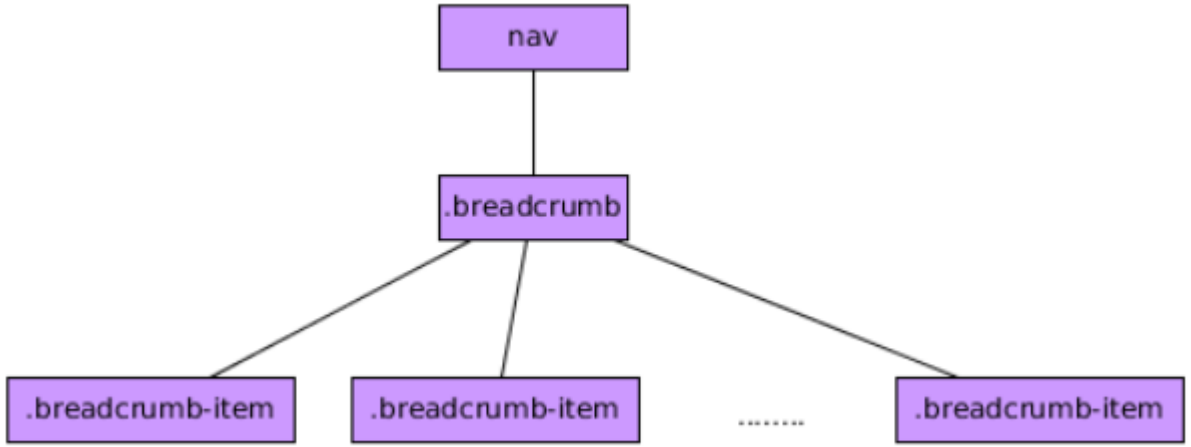
- **IMPORTANTE:** Si queremos que el menú se muestre desplegado desde el principio no debemos olvidarnos de la clase ***nav-expand-X*** siendo X uno de los breakpoints de BootStrap (sm,md,lg,xl). A partir de ese breakpoint se mostrará sin colapsar.Para más detalles ver el curso de maquetación.
- Podemos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

Componentes Navegación - navBar

- **Posicionando el menú**
  - 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).
- **NavBar Responsive.**
  - El ejemplo sugerido ya era una barra de menú responsiva. Para conseguir eso debemos hacer lo siguiente:
  - En primer luego 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 las 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***.
  - No hay funciones y eventos asociados a este componente.

Componentes Navegación - 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.



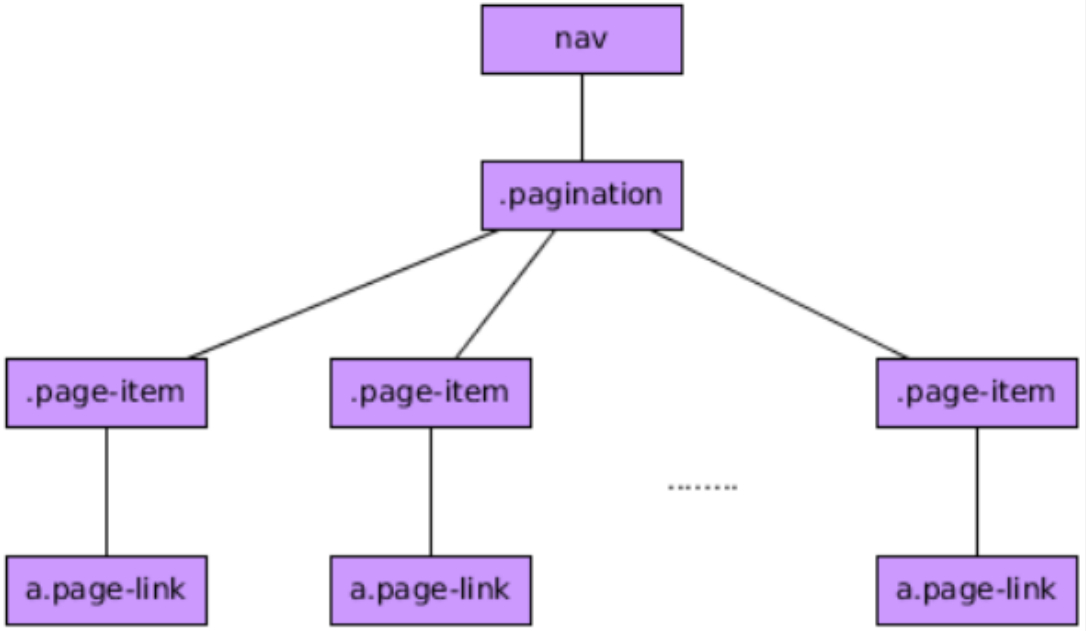
Componentes Navegación - BreadCrumbs


```
<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***.
- Este componente no posee funciones asociadas ni dispara eventos.

Componentes Navegación - 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.
- Tiene, de manera general. la siguiente jerarquía:



Componentes Navegación - Pagination

- Un ejemplo de dicho componente sería:

```
<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="#">Siguiente</a>
    </div>
  </div>
</nav>
```

Componentes Navegación - Pagination

- **Posibles modificaciones**
  - 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.
- Este componente no posee funciones asociadas ni dispara eventos.

Componentes Informar Usuario - Alert

- El componente Alert nos va a servir para mostrar mensaje destacados que queremos que el usuario vea.
- Tiene, de manera general, la siguiente estructura (no se representa el DOM por ser de un único elemento):
- Siendo ***alert-X*** la clase BootStrap para definir la gama de colores de nuestro mensaje, tanto para el fondo como para el texto que contenga la alerta. Esta gama de colores ya ha sido explicada en anteriores capítulos.

```
<div class="alert alert-x">

  Some content

</div>
```

Componentes Informar Usuario - Alert

- Al añadir colores tanto al texto como al fondo nos podemos encontrar con que los colores de los enlaces pasan desapercibidos, por ello, si queremos añadir un enlace que destaque de manera correcta dentro de una alerta debemos añadirle la clase ***alert-link***. Como ejemplo:.

```
<div class="alert alert-x">

  Some content with <a href="#" class="alert-link"> a link</a>.

</div>
```

Componentes Informar Usuario - Alert

- Adicionalmente, si quiero que la alerta desaparezca tengo que:
- Añadir la clase ***alert-dismissible*** al elmento 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.

```
<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>
```

Componentes Informar Usuario - Alert

- Puedo además interaccionar con estos elementos alert mediante el uso de una serie de métodos:
  - $().alert(). Enlaza manejadores de click con los elementos hijo que tengan ***data-dismiss="alert"*** para que al ahcer click sobre ellos desaparezcan los elementos seleccionados.
  - $().alert('close'). Cierra las alertas y las elimina del DOM.
  - $().alert('dispose'). XXXXXXX
- Y los eventos asociados son:
  - ***close.bs.alert*** que se dispara al cerrar la alerta.
  - ***closed.bs.alert*** que se dispara cuando la alerta se está cerrando del todo.

Componentes Informar Usuario - Badge

- El componente Badge es un componente que 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.

Componentes Informar Usuario - Badge

- El componente Badge es un componente que 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.

Componentes Informar Usuario - Badge

- A continuación vamos a ver un ejemplo de uso. No es necesario la estructura del DOM por ser un elemento:
  - <h3>Cesta4 productos</h3>
- Si queremos que sean aún más redondeados podemos cambiar la clase badge por la clase badge-pill.
  - <h3>Cesta4 productos</h3>
- Si el elemento que contiene el badge es un enlace BootStrap añade por defecto estilos diferentes para los estado hover y focus.
  - Cesta4 productos
- No hay métodos asociados a este elemento.

Componentes Informar Usuario - 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 no servirá para aportar explicaciones o aclaraciones sobre el elemento sobre el que hemos interaccionado.
- Para que funcionen requieren que hayamos añadido la librería popper.js a nuestro proyecto.
- Un ejemplo básico de utilización podría ser el siguiente. No es necesario mostrar el DOM por ser un único elemento:

```
<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>
```

Componentes Informar Usuario - PopOver

- Debemos fijarnos en los atributos que hemos añadido al elemento:
- ***data-toggle:*** Atributo BootStrap que indica que el elemento tiene una interacción como popover.
- ***title:*** Título del bocadillo
- ***data-content:*** Contenido del bocadillo.
- Necesitamos, para que todo funcione de manera correcta tener *data-toggle* y al menos uno de los otros dos atributos.

Componentes Informar Usuario - PopOver

- **IMPORTANTE:** Para que todo funcione correctamente el estilo de otros elementos no debe interferir con el popover. Si queremos asegurarnos de que esto suceda debemos añadir el siguiente script a nuestra página.
- 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.

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

Componentes Informar Usuario - PopOver

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

Componentes Informar Usuario - PopOver

- **Funciones asociadas**
- Las más importantes son, bajo mi punto de vista:
  - ***.popover(options)*** Habilita los popovers. Para más información sobre las posibles opciones visitar el manual.
  - ***.popover('show')*** Muestra el bocadillo asociado a un elemento.
  - ***.popover('hide')*** Oculta el bocadillo asociado a un elemento.
  - ***.popover('toogle')*** Cambio el estado de visibilidad del bocadillo asociado a un elemento.
  - ***.popover('dispose')*** El popover deja de funcionar.
  - ***.popover('enable')*** Habilita el bocadillo asociado a un elemento
  - ***.popover('disable')*** Deshabilita el bocadillo asociado a un elemento.

Componentes Informar Usuario - PopOver

- **Eventos asociados**
- La interacción con este tipo de elementos genera 5 nuevos tipos de eventos, bastante auto explicativos por su nombre y de los cuáles, para saber más detalles deberemos visitar el manual de referencia.
  - ***show.bs.popover***
  - ***shown.bs.popover***
  - ***hide.bs.popover***
  - ***hidden.bs.popover***
  - ***inserted.bs.popover***

Componentes Informar Usuario - 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.
- Un ejemplo básico de utilización podría ser el siguiente. No es necesario mostrar la estructura del DOM ya que sólo es un elemento:

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

Componentes Informar Usuario - Tooltip

- Debemos fijarnos en los atributos que hemos añadido al elemento:
- ***data-toggle:*** Atributo BootStrap que indica que el elemento tiene una interacción como tooltip.
- ***title:*** Texto del tooltip o pista.
- Si queremos añadir HTML dentro del atributo *title* debemos añadir al componente ***data-html=”true”***.

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

Componentes Informar Usuario - Tooltip

- Con BootStrap 4 podemos elegir entre 4 posibles posiciones en las que puede aparecer el componente *tooltip*:
  - Arriba o ***top***
  - Abajo o ***bottom***
  - A la derecha o ***right***
  - A la izquiera o ***left***
- Para especificar esta posición deberemos añadir el atributo el atributo ***data-placement*** con alguno de los valores anteriormente mencionados.

Componentes Informar Usuario - Tooltip

- **IMPORTANTE:** Para que todo funcione correctamente debemos añadir el siguiente script (o alguno similar) a nuestra página:

```
<script>

  $(function () {
    $('[data-toggle="tooltip"]').tooltip()
  });

</script>
```

Componentes Informar Usuario - Tooltip

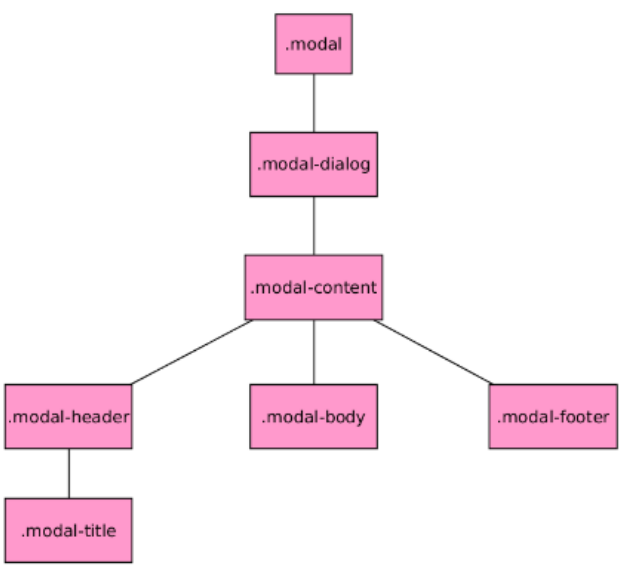
- **Funciones asociadas**
- Las más importantes son, bajo mi punto de vista:
  - ***.tooltip(options)*** Habilita los tooltips. Para más información sobre las posibles opciones visitar el manual.
  - ***.tooltip('show')*** Muestra el bocadillo asociado a un elemento.
  - ***.tooltip('hide')*** Oculta el bocadillo asociado a un elemento.
  - ***.tooltip('toogle')*** Cambio el estado de visibilidad del bocadillo asociado a un elemento.
  - ***.tooltip('dispose')*** El popover deja de funcionar.
  - ***.tooltip('enable')*** Habilita el bocadillo asociado a un elemento
  - ***.tooltip('disable')*** Deshabilita el bocadillo asociado a un elemento

Componentes Informar Usuario - Tooltip

- **Eventos asociados**
- La interacción con este tipo de elementos genera 5 nuevos tipos de eventos, bastante auto explicativos por su nombre y de los cuáles, para saber más detalles deberemos visitar el manual de referencia.
  - ***show.bs.tooltip***
  - ***shown.bs.tooltip***
  - ***hide.bs.tooltip***
  - ***hidden.bs.tooltip***
  - ***inserted.bs.tooltip***

Componentes Informar Usuario - 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.
- La estructura del DOM y las clases BootStrap 4 que deben tener una ventana modal son las siguientes:



Componentes Informar Usuario - Modal

- Un ejemplo de esta estructura sería:

```
<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>
```

Componentes Informar Usuario - Modal

- Además de esta estructura necesitaremos un elemento que sirva para mostrar dicha ventana Modal. Un ejemplo de ello sería:
- Siendo:
  - 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.
- Podemos comprobar que cuando hacemos click sobre el resto de la página la ventana modal se ocultará.

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

Componentes Informar Usuario - Modal

- **Fundido al aparecer / desaparecer**
- 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***.
- **Scroll**
- 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.
- **Centrado vertical**
- Si queremos centrar verticalmente la ventana modal debemos añadir la clase ***modal-dialog-centered*** al elemento que tiene la clase ***modal-dialog***.

Componentes Informar Usuario - Modal

- **Elementos de la venta para forzar el cierre de la misma**
- 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”*** .
- **Otras apreciaciones**
- Todos los *tooltipss* y *popover* que contenga la ventana modal se cerrarán al cerrarla.
- Puede maquetar dentro de la ventana modal usando las clases contenedor de la misma manera que si fuera una página normal. Para aprender sobre ello os recomiendo el curso ***“Maquetación con BootStrap 4”*** de OpenWebinars.
- 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***

Componentes Informar Usuario - Modal

- **Funciones asociadas**
- Las más importantes son, bajo mi punto de vista:
  - ***.modal(options)*** Habilita un elemento como ventana modal. Para más información sobre las posibles opciones visitar el manual.
  - ***.modal('show')*** Muestra la ventana modal.
  - ***.modal('hide')*** Oculta la ventana modal.
  - ***modal.('toogle')*** Cambio el estado de visibilidad de la ventana modal.

Componentes Informar Usuario - Modal

- **Eventos asociados**
- La interacción con este tipo de elementos genera 4 nuevos tipos de eventos, bastante auto explicativos por su nombre y de los cuáles, para saber más detalles deberemos visitar el manual de referencia.
  - ***show.bs.modal***
  - ***shown.bs.modal***
  - ***hide.bs.modal***
  - ***hidden.bs.modal***

Componentes Informar Usuario - ScrollSpy

- El componente BootStrap 4 ***ScroolSpy*** nos va a permtir 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.

Componentes Informar Usuario - ScrollSpy

- Para conseguir tenemos que seguir los siguiente 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 un 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 al punto 1.
  - Podemos fijar la distancia con respecto al inicio de un elemento con ***data-offet=X***. La distancia se mide en pixel.

Componentes Informar Usuario - ScrollSpy

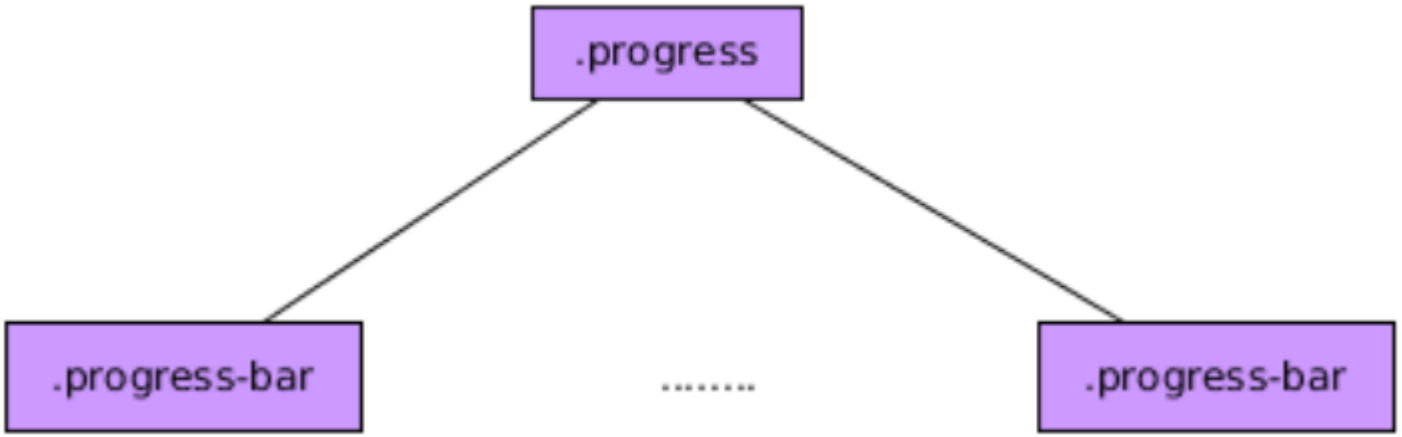

```
    <ul class="list-group" id="navegacion">
        <li class="list-group-item" data-target="#s1">Sección 1</li>
        <li class="list-group-item" data-target="#s1">Sección 2</li>
        <li class="list-group-item" data-target="#s1">Sección 3</li>
        <li class="list-group-item" data-target="#s1">Sección 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 commodo exercitation dolore enim voluptate sit laborum laboris aute.
Id laborum anim irure sint consectetur
                ullamco ullamco irure qui minim ea. Exercitation sint ut ex magna culpa elit duis officia ad
sit. Deserunt exercitation tempor quis ut. Exercitation eu aliqua labore magna duis esse in elit ullamco qui est
quis ut proident.</p>
            <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 commodo exercitation dolore enim voluptate sit laborum laboris aute.
Id laborum anim irure sint consectetur
                ullamco ullamco irure qui minim ea. Exercitation sint ut ex magna culpa elit duis officia ad
sit. Deserunt exercitation tempor quis ut. Exercitation eu aliqua labore magna duis esse in elit ullamco qui est
quis ut proident.</p>
            <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 commodo exercitation dolore enim voluptate sit laborum laboris aute.
Id laborum anim irure sint consectetur
                ullamco ullamco irure qui minim ea. Exercitation sint ut ex magna culpa elit duis officia ad
sit. Deserunt exercitation tempor quis ut. Exercitation eu aliqua labore magna duis esse in elit ullamco qui est
quis ut proident.</p>
            <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 commodo exercitation dolore enim voluptate sit laborum laboris aute.
Id laborum anim irure sint consectetur
                ullamco ullamco irure qui minim ea. Exercitation sint ut ex magna culpa elit duis officia ad
sit. Deserunt exercitation tempor quis ut. Exercitation eu aliqua labore magna duis esse in elit ullamco qui est
quis ut proident.</p>

    </div>
```

Componentes Informar Usuario - ScrollSpy

- **Funciones asociadas**
  - ***.scrollspy({target: "#idnavegación"})*** Habilita el componente para ese elemento de navegación que hemos identificado mediante un ID.
  - ***.scrollspy('refresh')*** Si añado/quito elementos al DOM para que siga funcionando el componente tengo que hacer recargar usando este método.
  - ***.scrollspy('dispose')*** Destruye el componente ScrollSpy. Deja de funcionar.
- **Eventos asociados.**
  - La interacción con este tipo de elementos genera 1 nuevos tipo de eventos, bastante auto explicativos por su nombre. Para saber más detalles deberemos visitar el manual de referencia.
  - ***activate.bs.scrollspy***

Componentes Informar Usuario - 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:



Componentes Informar Usuario - 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 ***widht*** de los posibles elementos ***.progress-bar***.

Componentes Informar Usuario - Progress

- En cada una de estos elementos ***.progress-bar*** podremos hacer cosas como:
  - 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.
  - No hay funciones y eventos asociados a este componente.

Componentes Informar Usuario - 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***.

Componentes Informar Usuario - Collapse

- Un ejemplo sería el siguiente. No se muestra la estructura del DOM por ser sólo un componente:

```
<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, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
</div>
```

Componentes Informar Usuario - Collapse

- 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.
- 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*.

```
<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>
```

Componentes Informar Usuario - Collapse

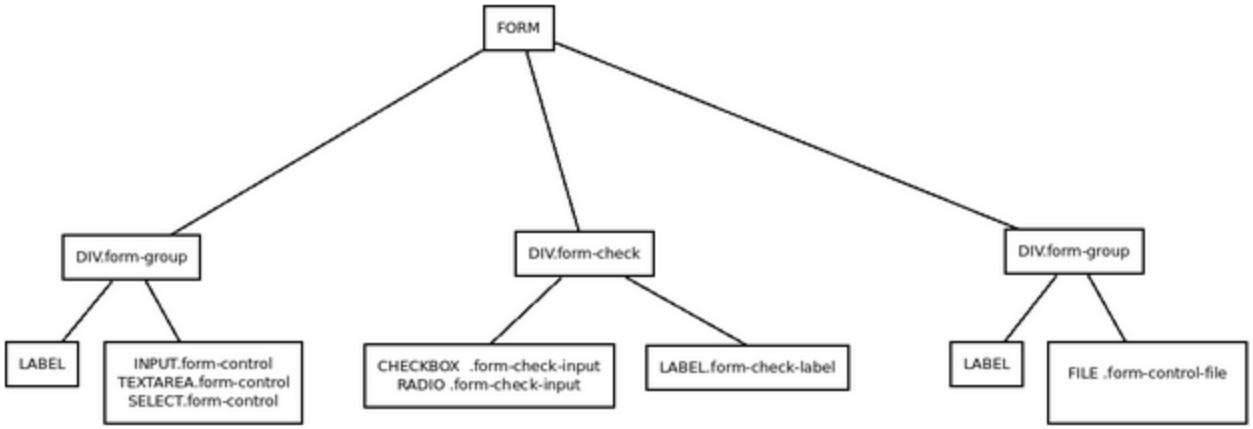
- **Funciones asociadas**
- Las más importantes son, bajo mi punto de vista:
  - ***.collapse(options)*** Habilita un elemento como colapsable. Para más información sobre las posibles opciones visitar el manual.
  - ***.collapse('show')*** Muestra los elementos seleccionados que tengan la clase *collapse*.
  - ***.collapse('hide')*** Oculta los elementos seleccionados que tengan la clase *collapse*.
  - ***.collapse('toogle')*** Cambio el estado de visibilidad los elementos seleccionados que tengan la clase *collapse*.
  - ***.collapse('dispose')*** Destruye el elemento colapsable. Deja de funcionar.

Componentes Informar Usuario - Collapse

- **Eventos asociados**
- La interacción con este tipo de elementos genera 4 nuevos tipos de eventos, bastante auto explicativos por su nombre y de los cuáles, para saber más detalles deberemos visitar el manual de referencia.
  - ***show.bs.collpase***
  - ***shown.bs.collapse***
  - ***hide.bs.collpase***
  - ***hidden.bs.collapse***

Componentes Formularios

- FORMULARIOS
  - Aunque en la documentación oficial los formularios se encuentran en la sección de componentes lo cierto es que son un elemento fundamental que se incluye en páginas de registro, de consulta, de búsqueda etc.
  - BootStrap proporciona una serie de clases para dar estilos a los distintos elementos de los formularios. De manera general podemos describir estas clases y la jerarquía que deben ocupar de la siguiente manera:



Componentes Formularios

- 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**

Componentes Formularios

- **Modificar aspectos de los formularios.**
  - 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).

Componentes Formularios

- **Tamaño de los formularios**
- 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 ves de *row*.

Componentes Formularios

- **Validación de los formularios**
- Para validar los formularios podemos:
- Recurrir a la validación normal que se hace por parte de los navegadores. Esta es una validación en cliente y los estilos y los mensajes quedan establecidos por los propios navegadores.
- Definir nuestros propios mensajes de validación en cliente. Con esto deshabilitamos los mensajes por defecto de navegador pero seguimos teniendo acceso al API de validación en JavaScript. Al hacer *submit* se añadirán los estilos correspondientes a los controles y se mostrarán los mensajes de ***valid-feedback*** o ***invalid-feedback*** . Para esto necesitamos algo de programación JavaScript.

Componentes Formularios

- **Validación de los formularios**
- Para validar los formularios podemos:
- Recurrir a la validación normal que se hace por parte de los navegadores. Esta es una validación en cliente y los estilos y los mensajes quedan establecidos por los propios navegadores.
- Definir nuestros propios mensajes de validación en cliente. Con esto deshabilitamos los mensajes por defecto de navegador pero seguimos teniendo acceso al API de validación en JavaScript. Al hacer *submit* se añadirán los estilos correspondientes a los controles y se mostrarán los mensajes de ***valid-feedback*** o ***invalid-feedback*** . Para esto necesitamos algo de programación JavaScript.
- Si validamos usando el servidor podemos indicar qué campos son válidos / inválidos añadiendo las clases ***is-valid is-invalid*** a los campos del formulario. Eso mostrará los mensajes correspondientes.
- Si queremos usar bocadillos en vez de simples debemos usar las clases ***valid-tooltip*** o ***invalid-tooltip***.
- Este componente no posee funciones asociadas ni dispara eventos.

```
<form class="validar" novalidate>
  <div class="form-group">
    <label>E-mail</label>
    <input type="email" class="form-control" required>
    <div class="valid-feedback">
      Todo parece correcto
    </div>
    <div class="invalid-feedback">
      Debe Introducir un correo
    </div>
  </div>
  <div class="form-group">
    <label>Password</label>
    <input type="password" class="form-control" required>
    <div class="valid-feedback">
      OK
    </div>
    <div class="invalid-feedback">
      Contraseña Incorrecta
    </div>
  </div>
  <button class="btn btn-primary" type="submit">Enviar</button>
</form>


<script>


  (function() {
    //Modo Estricto
    'use strict';

    // Cuando se acaba de cargar la página
    window.addEventListener('load', function() {

      //Seleccionamos los formularios que quiero validar
      var forms = document.getElementsByClassName('validar');
      console.log(forms.length);
      // Los recorro y evito que se manden los datos en caso de error
      var validation = Array.prototype.filter.call(forms, function(form) {
          form.addEventListener('submit', function(event) {
            if (form.checkValidity() === false) {
              event.preventDefault();
              event.stopPropagation();
            }
            form.classList.add('was-validated');
          }, false);
      });
    }, false);


  })();
</script>
```

Componentes Formularios

- FORMULARIOS
- 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 ves de *row*.
- Bootstrap 4 proporciona muchas más cosas en relación a los formularios, por eso, aunque con estos apuntes ya es suficiente para empezar a trabajar sin problemas, os recomiendo mirar tranquilamente la documentación.

Componentes Formularios - 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.
- No vamos a representar la jerarquía del DOM por ser sólo un elemento.
- Por ejemplo:
  - <button class="btn" value="Botón">

Componentes Formularios - Button

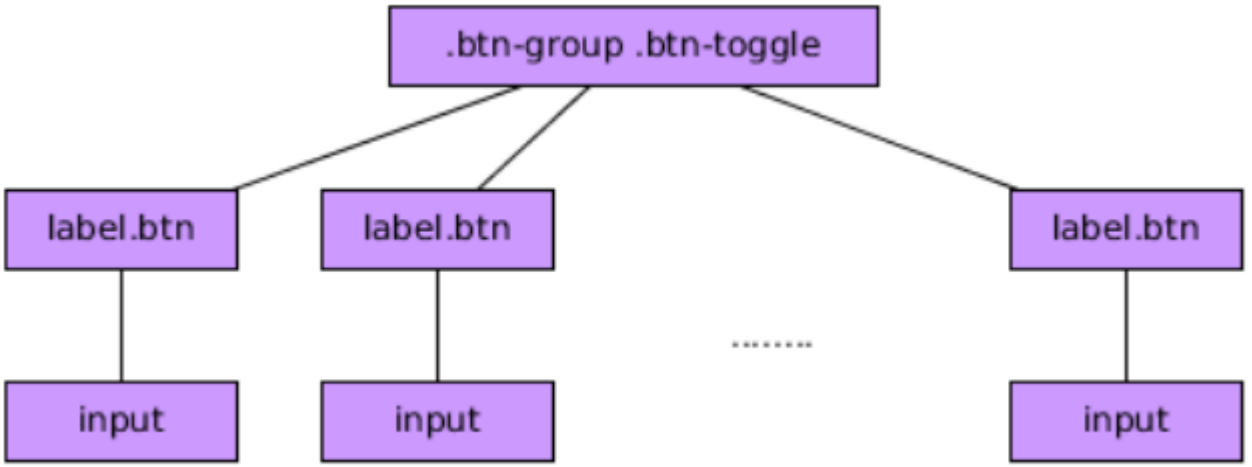
- **Modificaciones de los botones**
- Si queremos añadir colores podemos usar las clases ***bnt-X*** siendo X uno de los colores de la paleta de Bootstrap.
- Si quiero que los colores sean sólo para el borde y texto pero que no tenga fondo usaré la clase ***btn-outline-X***siendo X uno de los colores de la paleta de BooStrap.
- 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***.

Componentes Formularios - Button

- **Checkboxes y Radios usando botones.**
- Usando BootStrap 4 podemos modificar la apariencia de los elementos *radio* y *checkbox* para que se muestren como si fueran botones.
- Para simular con botones un radio group podemos utilizar una estructura similar a la siguiente:

Componentes Formularios - Button

- **Checkboxes y Radios usando botones.**
- Usando BootStrap 4 podemos modificar la apariencia de los elementos *radio* y *checkbox* para que se muestren como si fueran botones.
- Para simular con botones un radio group podemos utilizar una estructura similar a la siguiente:



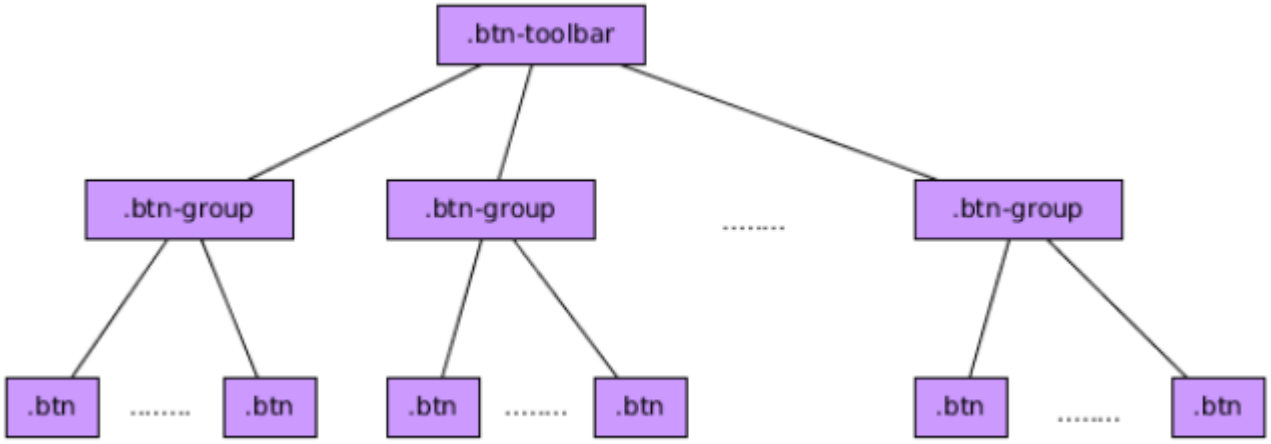
Componentes Formularios - Button

- Fijaros que en este caso.
- 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.
- Funciones asociadas.
- .button('toogle') Simula que el el botón esté seleccionado o no (activado)
- .button('dispose') Destruye el elemento botón, deja de funcionar como tal.

```
<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>
```

Componentes Formularios - Button

- **Componente Button Group**
- El componente BootStrap 4 ***Button Group*** sirve para agrupar botones en una sola linea pudiendo tener más de una agrupación para formar una barra de botones o botonera.
- De manera general la estructura del DOM sería similar a la siguiente:



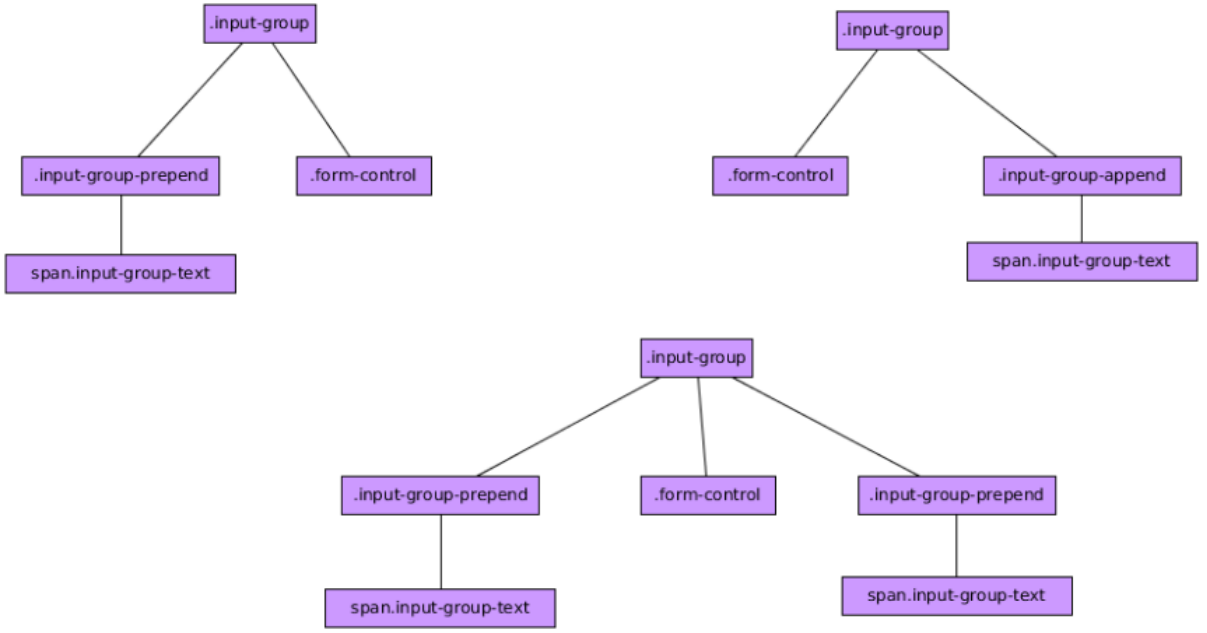
Componentes Formularios - Button

- 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***.
- Estos componentes no tienen funciones asociadas ni disparan eventos.

```
<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">G21B1</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>
```

Componentes Formularios – 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.
- Presenta varias posibilidades que comentaremos luego pero podemos generalizar varias estructuras:



Componentes Formularios – 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.
- Podemos ver un ejemplo:

```
<label>Tú página Web personal</label>
<div class="input-group">
  <div class="input-group-prepend">
    <span class="input-group-text">http://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>
```

Componentes Formularios – Input Group

- Puedo modificar el tamaño de los input-group añadiendo las clases ***input-group-lg*** (grande) o ***input-group-sm***(pequeño).
- En vez de únicamente sólo texto, en el elemento con la clase ***input-group-text*** puedo añadir etiquetas input con el tipo *checkbox* o *radio*. Un ejemplo de esto:

```
<label>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>
```

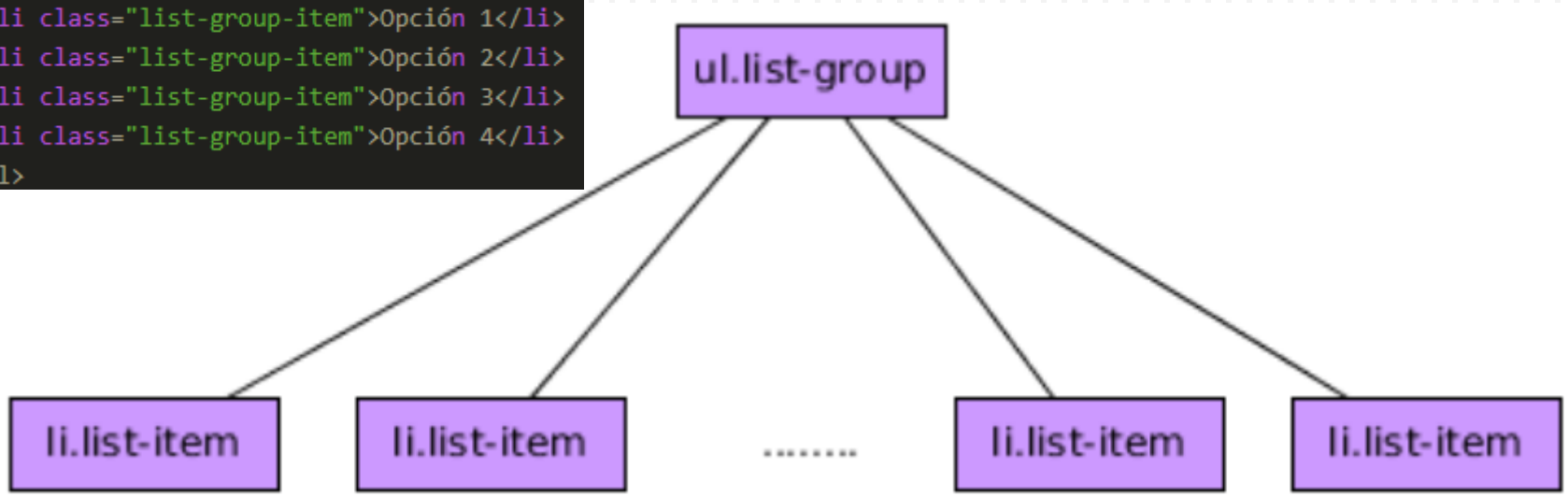
Componentes Formularios – Input Group

- **Más consideraciones sobre esos add-ons.**
  - Puedo tener varios elementos con la clase ***input-group-text*** detro de los elementos con las clases ***input-group-prepend*** y ***input-group-append***.
  - Puedo añadir botones y *dropdowns* dentro de elementos con las clases ***input-group-prepend*** y ***input-group-append***.
- Este componente no tiene funciones asociadas ni dispara eventos.

Otros Componentes – List Group

- El componente BootStrap 4 es un componente que nos va a permitir mostrar una serie de contenidos.
- En su ejemplo más básico la jerarquía que debe presentar este componente es la siguiente:

```
<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>
```



Otros Componentes – List Group

- 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 redondeadeas 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 paletas de colores como fondo de los elementos ***list-group-item*** debeo 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.

Otros Componentes – List Group

- **List-Group con efecto hover.**
- Si queremos conseguir el efecto hover tendremos que cambiar la estructura dejando de usar listas,usando etiquetas como ***a*** o ***buttom*** como elementos ***list-group-item*** y adicionalmente agregando a esos elementos la clase ***list-group-item-action***.
- Un ejemplo de esto sería:

```
<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>
```

Otros Componentes – List Group

- **Simulando paneles con el componente list-group**
- Usando el componente list-group y el código javascript que viene con BootStrap 4 podemos simular paneles de contenidos.
- Como se puede apreciar para conseguir esto:
- 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.

```
<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>
```

Otros Componentes – List Group

- **Funciones asociadas.**
  - ***.tab()*** Activa el funcionamiento de los paneles.
  - ***.tab('show')*** Muestra el panel asociado al elemento (por id o data-target).
  - ***.tab('hide')*** Oculta el panel asociado al elemento (por id o data-target).

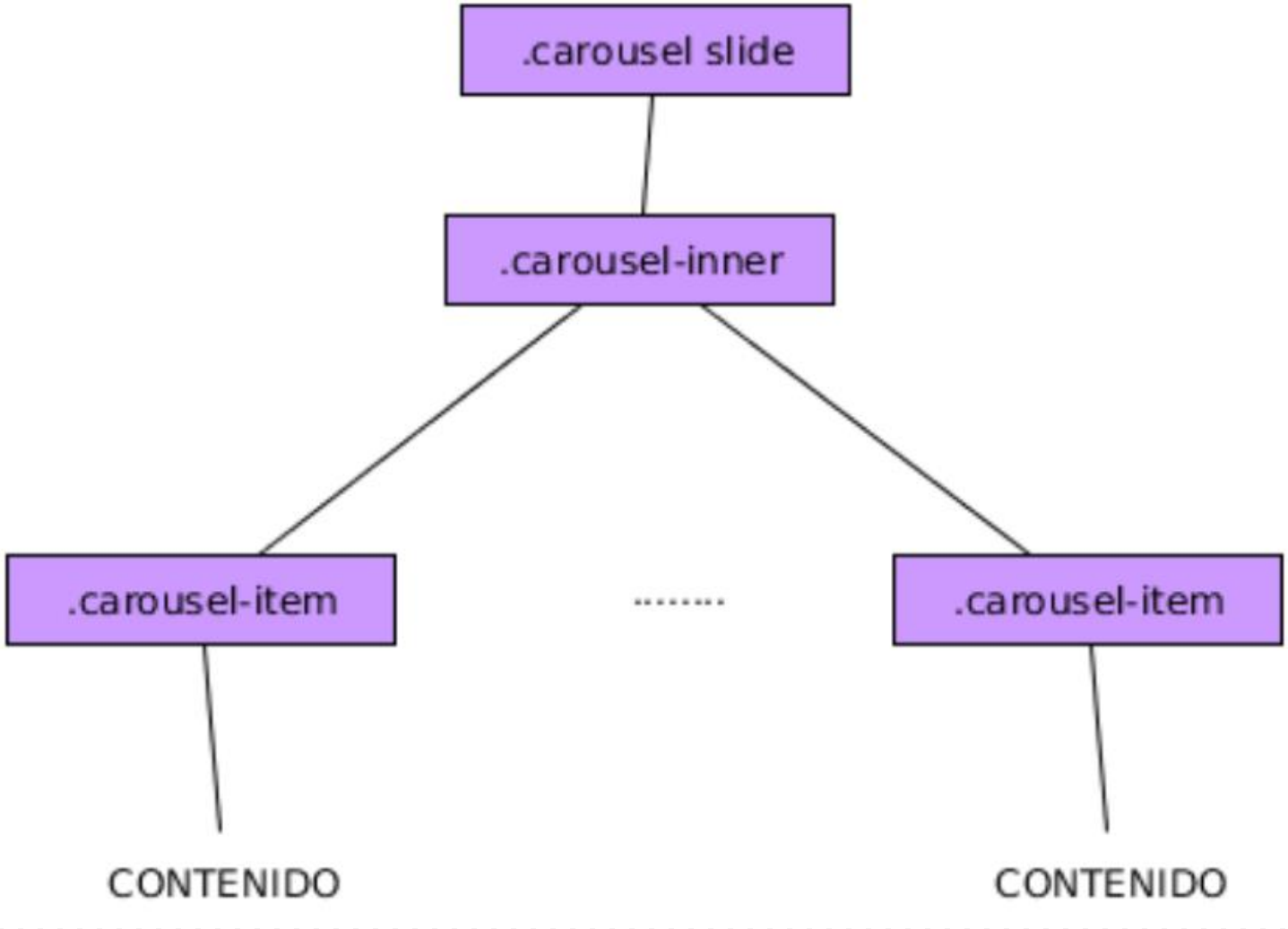
Otros Componentes – List Group

- **Eventos asociados**
- La interacción con este tipo de elementos genera 4 nuevos tipos de eventos. Su nombre es auto-explicativo y para más detalles deberíamos consultar el manual.
  - ***hide.bs.tab***
  - ***show.bs.tab***
  - ***shown.bs.tab***
  - ***hidden.bs.tab***

Otros Componentes – Carrousel

- 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.
- BootStrap 4 nos proporciona una manera fácil de conseguir este tipo de estructuras y, en su configuración más sencilla necesitaremos una jerarquía similar a la siguiente:

Otros Componentes – Carrousel



Otros Componentes – Carrousel

- Siendo:
  - 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 la 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.

Otros Componentes – Carrousel

- Siendo:
  - 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 la 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">
        <img  class="d-block w-100" src="img/img01.jpg" alt="Primer elemento" />
      </div>
      <div class="carousel-item">
        <img class="d-block w-100" src="img/img02.jpg" alt="Segundo elemento" />
      </div>
      <div class="carousel-item">
        <img class="d-block w-100" src="img/img03.jpg" alt="Tercer elemento" />
      </div>
  </div>
</div>
```

Otros Componentes – Carrousel

- **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.
- Adicionalmente, a esta estructura por defecto podemos añadirle varias cosas:
  - Controles para navegar por la galería a izquierda y a derecha.
  - Indicadores que nos señalan en qué posición nos encontramos dentro de la galería.
  - Leyendas o *captions* para cada uno de los elementos de la galería.

Otros Componentes – Carrousel

- **Añadir controles al *Carousel***
- 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.
- **IMPORTANTE** #MyCarouselId debe ser el id del elemento que posea la clase *carousel*.

```
<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>
```

Otros Componentes – Carrousel

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

```
<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>
```

Otros Componentes – Carrousel

- **Añadir leyendas a los elementos del *Carousel***
- 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>
```

Otros Componentes – Carrousel

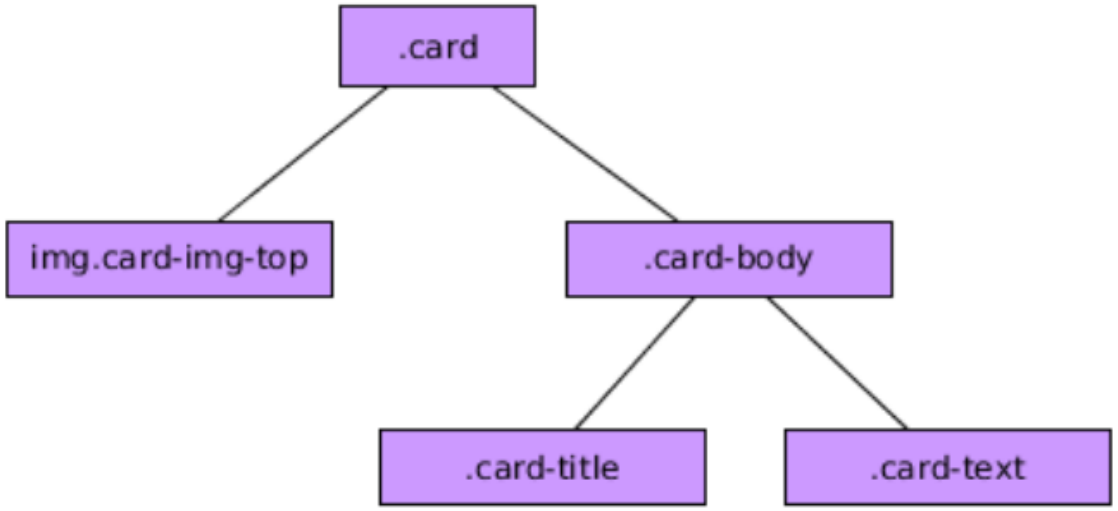
- **Funciones asociadas**
  - ***.carousel(options)*** Inicializa con las opciones pertinentes (ver el manual para más detalles).
  - ***.carousel('cycle')*** Comienza el ciclo del carrusel.
  - ***.carousel('pause')*** Pausa el ciclo del carrusel.
  - ***.carousel(number)*** Sitúa el carrusel en esa posición.
  - ***.carousel('prev')*** Fuerza al carrusel a mostrar el elemento anterior.
  - ***.carousel('next')*** Fuerza al carrusel a mostrar el siguiente elemento.
  - ***.carousel('dispose')*** Destruye el carrusel (deja de funcionar).

Otros Componentes – Carrousel

- **Eventos asociados**
  - La interacción con este tipo de elementos genera 2 nuevos tipos de eventos con propiedades adicionales (ver manual). Dichos eventos son:
    - ***slide.bs.carousel*** Cuando comienza una transición del carrusel.
    - ***slid.bs.carousel*** Cuando finaliza una transición del carrusel.

Otros Componentes – Card

- El componente BootStrap 4 ***Card*** es uno de las novedades de este nueva versión y viene a sustituir a los viejos componentes *panels*, *wells* y *thumbnails*.
- Es una construcción muy flexible ya que tiene como objetivo encapsular todo tipo de contenido. Sin embargo, en su estructura más básica y común podemos decir que su estructura sigue la siguiente jerarquía:



Otros Componentes – Card

- No todos los elementos son obligatrios y podemos jugar a añadir y quitarlos para ver la apariencia final. Además podemos añadir otros elementos (según necesitemos) dentro de esta estructura.
- Un ejemplo de aplicación de esta jerarquía sería:

```
<div class="card">
  <img src="img/img01.jpg" class="card-img-top">
  <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>
```

Otros Componentes – Card

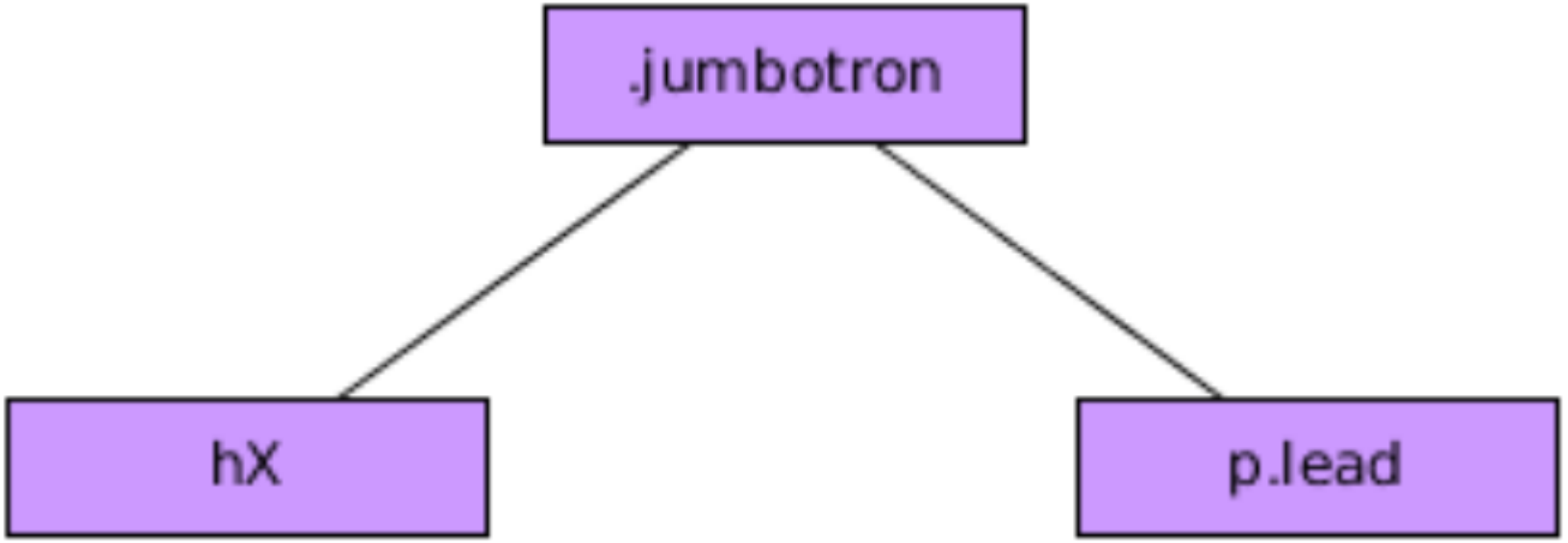
- **Más consideraciones sobre las *Cards***
  - Podemos también añadir un subtítulo a la card con un elemento que tenga la clase ***class-subtitle*** justo detrás del título.
  - Los enlaces tienes una clase especial con estilos propios ***card-link***.
  - Puedo añadir cabeceras y pies a la *Card* posicionando 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***.

Otros Componentes – Card

- Adicionalmente puedo usar todas las utilidades de BootStrap 4 para colores, fondos etc..para mejorar la apariencia de las mismas.
- **IMPORTANTE*** BootStrap 4 presenta tres layouts exclusivos para *Cards* y que nos permiten agrupar estos componentes: ***card-group***, ***card-deck*** y ***card-columns***. De momento estos no son responsivos y por eso se han dejado fuera de este curso. Pero os recomiendo que lo miréis, seguramente en el futuro se mejorarán.
- Este componente no posee funciones asociadas ni dispara eventos.

Otros Componentes – Jumbotron

- El componente BootStrap ***Jumbotron*** es un componente sencillo que se utiliza para mostrar mensajes y/o títulos.
- En su forma más común el DOM sigue la siguiente estructura:



Otros Componentes – Jumbotron

- Un ejemplo sencillo del uso de Jumbotron sería el siguiente.
- Como podemos ver, este componente modifica las tipografías, añade un color de fondo y unos bordes redondeados.

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

Otros Componentes – Jumbotron

- Si, por el contrario, preferimos que ocupe todo y no tener bordes redondeados debemos añadir la clase jumbotron-fluid al div con la clase jumbtron y añadir como hijo de este elemento un elemento con las clase container o container-fluid.
- Por supuesto podemos añadir más elementos dentro, lo que necesitemos y se nos ocurra.
- Este componente no posee funciones asociadas ni dispara eventos.

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

It’s your turn

A2.2.1. Proyecto BootStrap