

Parte 3 de 4: Bootstrap

LISTADOS
MODAL
NAVS
NAVBAR
ETC.

Apuntes de: **Rosa María Medina Gómez**

Adaptados a Power Point por José Jesús Torregrosa García

Curso de Formación del Profesorado a distancia
Cefire Específico de FP de Chestre

Generalitat Valenciana Curso 2018 - 2019



Bootstrap

INTRODUCCIÓN

En la unidad anterior vimos los componentes: alertas, badge, migas de pan, botones, grupo de botones, tarjetas, carousel, collapse, desplegables, formularios y grupos de inputs. En esta veremos:

[Jumbotron](#)

[Grupo de listados](#)

[Modal](#)

[Navs](#)

[Navbar](#)

[Paginación](#)

[Popovers](#)

[Barra de progreso](#)

[Scrollspy](#)

[Tooltips](#)

JUMBOTRON

[Jumbotron](#) es un ligero y flexible componente que puede opcionalmente extenderse para mostrar un cierto mensaje.

Ver ejemplo
[jumbutron.html](#)

Hello, world!

This is a simple hero unit, a simple jumbotron-style component for calling extra attention to featured content or information.

It uses utility classes for typography and spacing to space content out within the larger container.

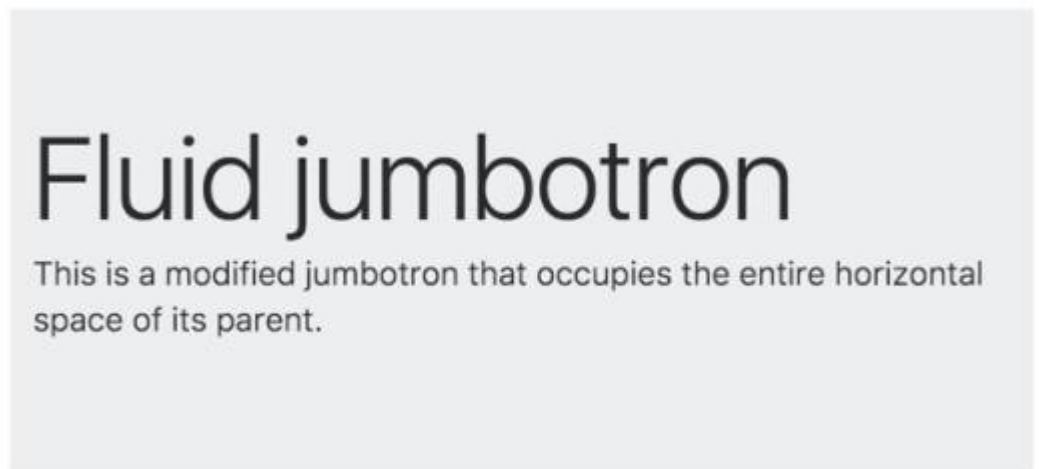
[Learn more](#)

JUMBOTRON

Para hacer que el jumbotron ocupe todo el ancho y sin tener los bordes redondeados añadiremos la clase `.jumbotron-fluid` y también

- la `.container` o `.container-fluid`

Ver ejemplo
`jumbutron2.html`



GRUPO DE LISTADOS

Hay varios modelos básicos de listados predefinidos en Bootstraps, veamos poco a poco cada uno de ellos cómo hacerlos (ver **IGroup.html**)

```
<ul class="list-group">  
  <li class="list-group-item">Cras justo  
odio</li>  
  <li class="list-group-item">Dapibus ac  
facilisis in</li>  
  <li class="list-group-item">Morbi leo  
risus</li>  
  <li class="list-group-item">Porta ac  
consectetur ac</li>  
  <li class="list-group-item">Vestibulum  
at eros</li>  
</ul>
```

Cras justo odio
Dapibus ac facilisis in
Morbi leo risus
Porta ac consectetur ac
Vestibulum at eros

GRUPO DE LISTADOS

Si queremos indicar qué sección se encuentra activa, lo que podemos hacer es añadir la clase **.active** al ítem correspondiente (ver **IGroup2.html**)

```
<ul class="list-group">  
  <li class="list-group-item active">Cras  
justo odio</li>  
  <li class="list-group-item">Dapibus ac  
facilisis in</li>  
  <li class="list-group-item">Morbi leo  
risus</li>  
  <li class="list-group-item">Porta ac  
consectetur ac</li>  
  <li class="list-group-item">Vestibulum  
at eros</li>  
</ul>
```

Cras justo odio
Dapibus ac facilisis in
Morbi leo risus
Porta ac consectetur ac
Vestibulum at eros

GRUPO DE LISTADOS

Si queremos indicar qué sección se encuentra desactiva, lo que podemos hacer es añadir la clase **disabled** al ítem correspondiente (ver **IGroup3.html**)

```
<ul class="list-group">
  <li class="list-group-item
disabled">Cras justo odio</li>
  <li class="list-group-item">Dapibus ac
facilisis in</li>
  <li class="list-group-item">Morbi leo
risus</li>
  <li class="list-group-item">Porta ac
consectetur ac</li>
  <li class="list-group-item">Vestibulum
at eros</li>
</ul>
```

Cras justo odio
Dapibus ac facilisis in
Morbi leo risus
Porta ac consectetur ac
Vestibulum at eros

GRUPO DE LISTADOS

Usaremos el elemento `<a>` o `<button>` si queremos que al pulsar sobre algún ítem realice alguna acción. (ver **lGroup4.html**)

Código:

```
<div class="list-group">  
  <a href="#" class="list-group-item list-group-item-action active">  
    Cras justo odio  
  </a>  
  <a href="#" class="list-group-item list-group-item-action">Dapibus ac  
facilisis in</a>  
  <a href="#" class="list-group-item list-group-item-action">Morbi leo  
risus</a>  
  <a href="#" class="list-group-item list-group-item-action">Porta ac  
consectetur ac</a>  
  <a href="#" class="list-group-item list-group-item-action  
disabled">Vestibulum at eros</a> </div>
```


GRUPO DE LISTADOS

Otro ejemplo sería los ítems contextuales, es decir con colores para que sean identificativos (ver **IGroup5.html**)

```
<ul class="list-group">  
  <li class="list-group-item">Dapibus ac facilisis in</li>  
  <li class="list-group-item list-group-item-success">Dapibus ac facilisis in</li>  
  <li class="list-group-item list-group-item-info">Cras sit amet nibh libero</li>  
  <li class="list-group-item list-group-item-warning">Porta ac consectetur ac</li>  
  <li class="list-group-item list-group-item-danger">Vestibulum at eros</li> </ul>
```



GRUPO DE LISTADOS

Otro ejemplo sería los ítems contextuales, es decir con colores para que sean identificativos (ver **IGroup5.html**)

```
<ul class="list-group">  
  <li class="list-group-item">Dapibus ac facilisis in</li>  
  <li class="list-group-item list-group-item-success">Dapibus ac facilisis in</li>  
  <li class="list-group-item list-group-item-info">Cras sit amet nibh libero</li>  
  <li class="list-group-item list-group-item-warning">Porta ac consectetur ac</li>  
  <li class="list-group-item list-group-item-danger">Vestibulum at eros</li> </ul>
```



GRUPO DE LISTADOS

Para consultar más ejemplos sobre los list-group, podéis consultarlo aquí:

<https://getbootstrap.com/docs/4.1/components/list-group/#contextual-classes>

MODAL

Antes de empezar con el componente [modal](#) de bootstrap, debemos asegurarnos de que conocemos algunas de las opciones que han cambiado:

- Modal está hecho con **HTML**, **CSS** y **JS**. Están posicionados sobre todo lo demás, y eliminan el scroll del `<body>` para que el scroll sea del contenido del modal en su lugar.
- Al hacer clic sobre el modal “backdrop” automáticamente se cerrará el modal.

MODAL

- Bootstrap solo soporta **una ventana** de un modal a la vez. Nested modals no están soportados, de hecho, se considera que es una mala práctica utilizarlo.
- Modal usa `position: fixed`, el cual a veces puede ser un poco particular sobre todo si se renderiza. Siempre que sea posible, el lugar de nuestro modal irá en el top de nuestro HTML evitando interferencias con cualquier otro elemento. Hay problemas al hacer un nesting con `.modal` sin algún otro elemento fijo.

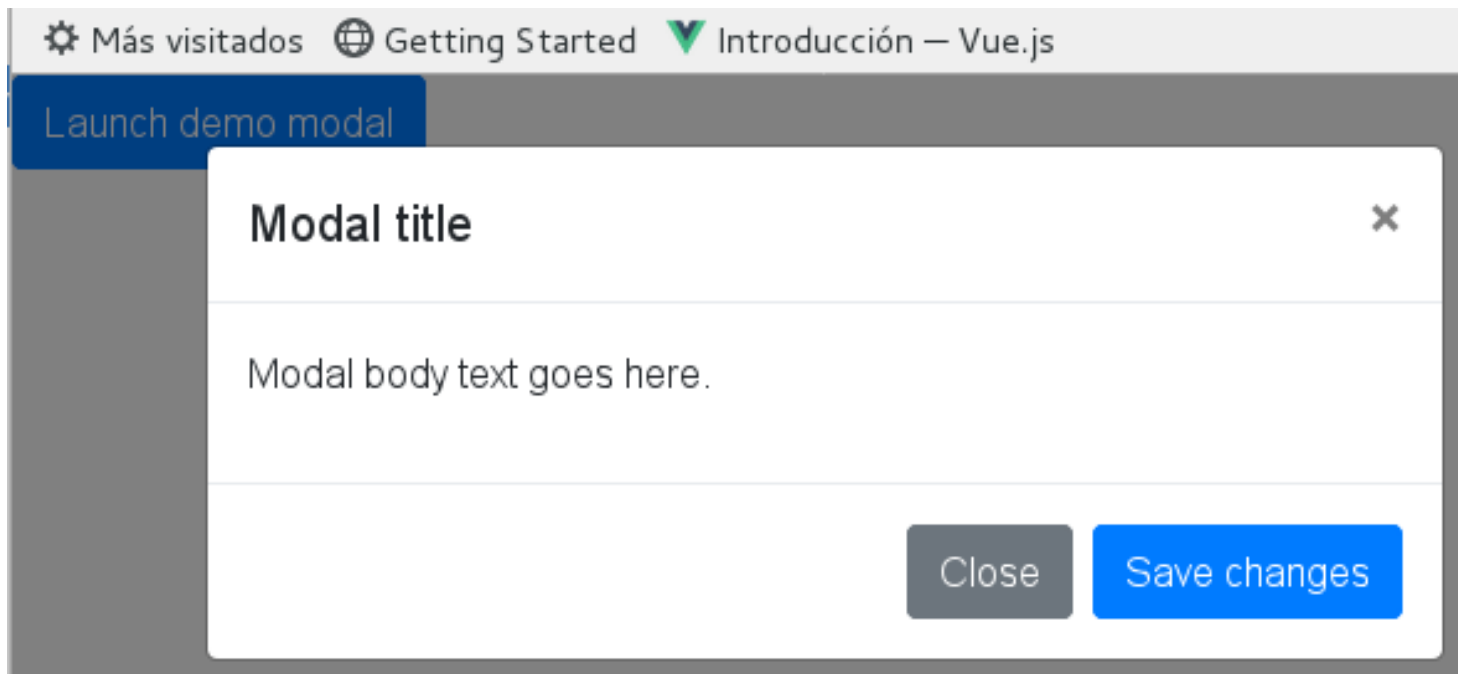
MODAL

- Además debido a `position: fixed`, hay advertencias a la hora de utilizar los modals en los móviles. Podemos consultar con más detalle la documentación aquí ([_https://getbootstrap.com/docs/4.0/getting-started/browsers-devices/#modals-and-dropdowns-on-mobile_](https://getbootstrap.com/docs/4.0/getting-started/browsers-devices/#modals-and-dropdowns-on-mobile))
- Por último, indicar que el atributo HTML `autofocus` no se aplica en los modals, este efecto lo puedes conseguir si quieres mediante **JS**

MODAL

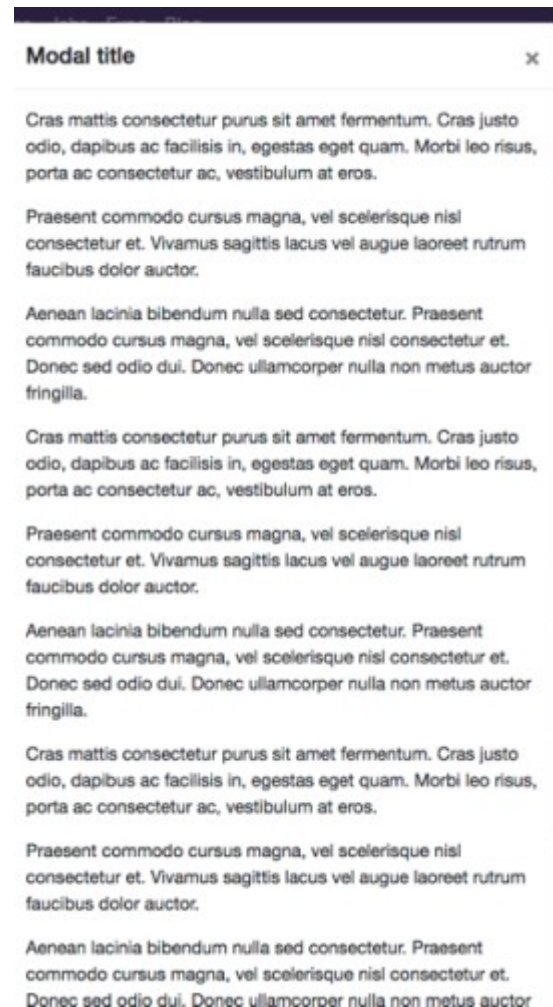
Ejemplos de uso del componente modal

- Ejemplo de un modal que aparece al pulsar el botón (dinámico): **ver modal.html**



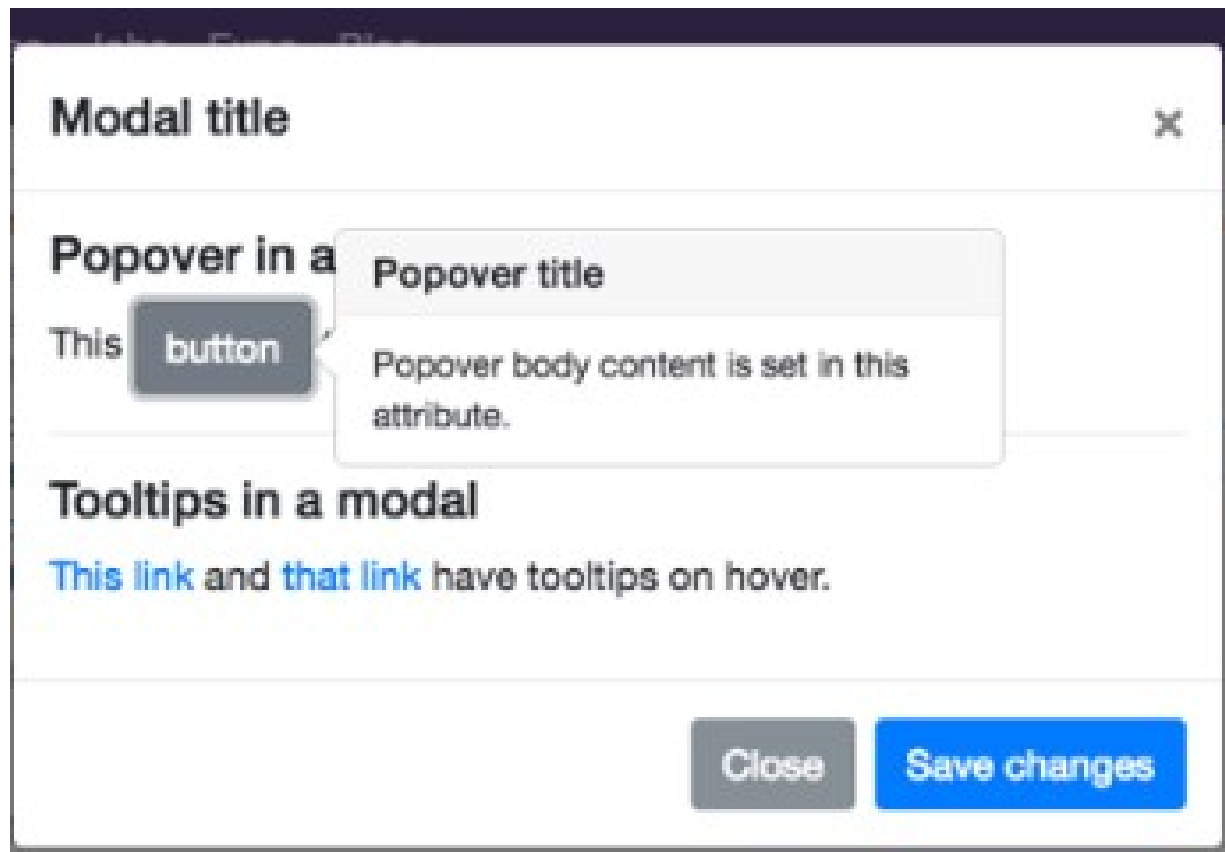
MODAL

Ejemplo de Modal con scroll: [ver modal2.html](#)



MODAL

Mas ejemplos de Modals con tooltips y popovers:ver **modals3.html**



MODAL

Del mismo modo en el modal podemos utilizar el contenido en un grid, para ver un ejemplo pulsad [aquí](#).

Del mismo modo, si queremos conocer más variedades de modal, funcionalidades, etc. Podemos consultarlo [aquí](#).

NAVS

Los **navs** se utilizan para navegar a lo largo de nuestra web. Para conseguirlo con Bootstrap debemos utilizar al menos la clase **.nav** que nos va a permitir activar y desactivar los estados.

El componente básico utiliza la clase **.nav** como ya se ha mencionado, y este no incluye el estado **.activo** si nosotros no lo ponemos por defecto. En el siguiente ejemplo, podemos ver cómo usar el modelo básico.

NAVS

- Ver el ejemplo **navs.html**

```
<ul class="nav">  
  <li class="nav-item">  
    <a class="nav-link active" href="#">Active</a>  
  </li>  
  <li class="nav-item">  
    <a class="nav-link" href="#">Link</a>  
  </li>  
  <li class="nav-item">  
    <a class="nav-link" href="#">Link</a>  
  </li>  
  <li class="nav-item">  
    <a class="nav-link disabled" href="#">Disabled</a>  
  </li>  
</ul>
```

Active

Link

Link

Disabled

NAVS

Estas clases las podemos utilizar en cualquier parte, son bastantes flexibles, de modo que podemos utilizarla o bien con los `` como se hacía hace unos años o con los elementos `<nav>`. Debido a que `.nav` utiliza `display:flex`, los vínculos de navegación se comportan igual que los elementos de navegación pero sin el marcado adicional.

NAVS

- Ver el ejemplo **navs2.html**

```
<nav class="nav">
```

```
<a class="nav-link active" href="#">Active</a>
```

```
<a class="nav-link" href="#">Link</a>
```

```
<a class="nav-link" href="#">Link</a>
```

```
<a class="nav-l
```

```
</nav>
```

Active

Link

Link

Disabled

NAVS

Estilos disponibles que podemos usar con el nav de Bootstrap

Vamos a ver alguno de los estilos más usados como:

- Alineación horizontal:
 - Si queremos que esté centrado justificado a la derecha usaremos la clase: **.justify-content-end**
 - Si queremos que esté centrado usaremos: **.justify-content-center.**
- Ejemplo: **navs3.html**

Active

Link

Link

Disabled

NAVS

En cambio, si queremos que en lugar de tener el nav en horizontal lo queremos en vertical, usaremos la clase `.flex-column` y para versiones responsive `.flex-sm-column`. En el siguiente ejemplo estamos viendo cómo sería utilizando el elemento `ul`, pero sería igual si usáramos el `nav`.

Ejemplo: **navs4.html**

Active

Link

Link

Disabled

NAVS

Otra forma habitual que podemos encontrar los navs es mediante tabs, para ello añadiremos la clase `.nav-tabs`. Ejemplo: **navs5.html**

Active

Link

Link

Disabled

NAVS

Aprovechando el mismo código anterior, vamos a ver cómo convertir esos tabs en pills, donde la clase que se utiliza pasa a ser **.nav-pills**. Ver ejemplo **navs6.html**



Para consultar más estilos disponibles, usad esta [página](#). En ella podremos encontrar cómo justificar cada uno de los elementos del nav, cómo hacer que uno o varios de los elementos del nav sea un desplegable, etc.

NAVBAR

Antes de nada, vamos a ver cómo funciona el [navbar](#), y lo que necesitamos conocer antes de empezar:

- Navbars require un empaquetado `.navbar` con `.navbar-expand{sm|-md|-lg|-xl}` para un responsive collapse y esquema de clases de color.
- Navbars y sus contenidos son **fluid** por defecto. Por tanto podemos utilizar contenedores de forma opcional para limitar el ancho horizontal que tienen.
- Los navbar y sus contenidos están contruidos con flexbox, de forma que obtenemos un fácil alineamiento mediante las utility clases.

NAVBAR

- Los navbar y sus contenidos están contruidos con flexbox, de forma que obtenemos un fácil alineamiento mediante las utility clases.
- **Nabvars son responsive** por defecto, pero podemos de forma fácil modificarlos y cambiarlos. El comportamiento responsive depende del plugin Collapse de JavaScript.
- Debemos asegurarnos una accesibilidad usando el element `<nav>`, o si usamos un elemento más genérico como `<div>`, debemos añadir `role="navigation"` en cada navbar para identificarlos de forma explícita, es como una region de referencia para los usuarios que necesitan de una tecnología de asistencia.

NAVBAR

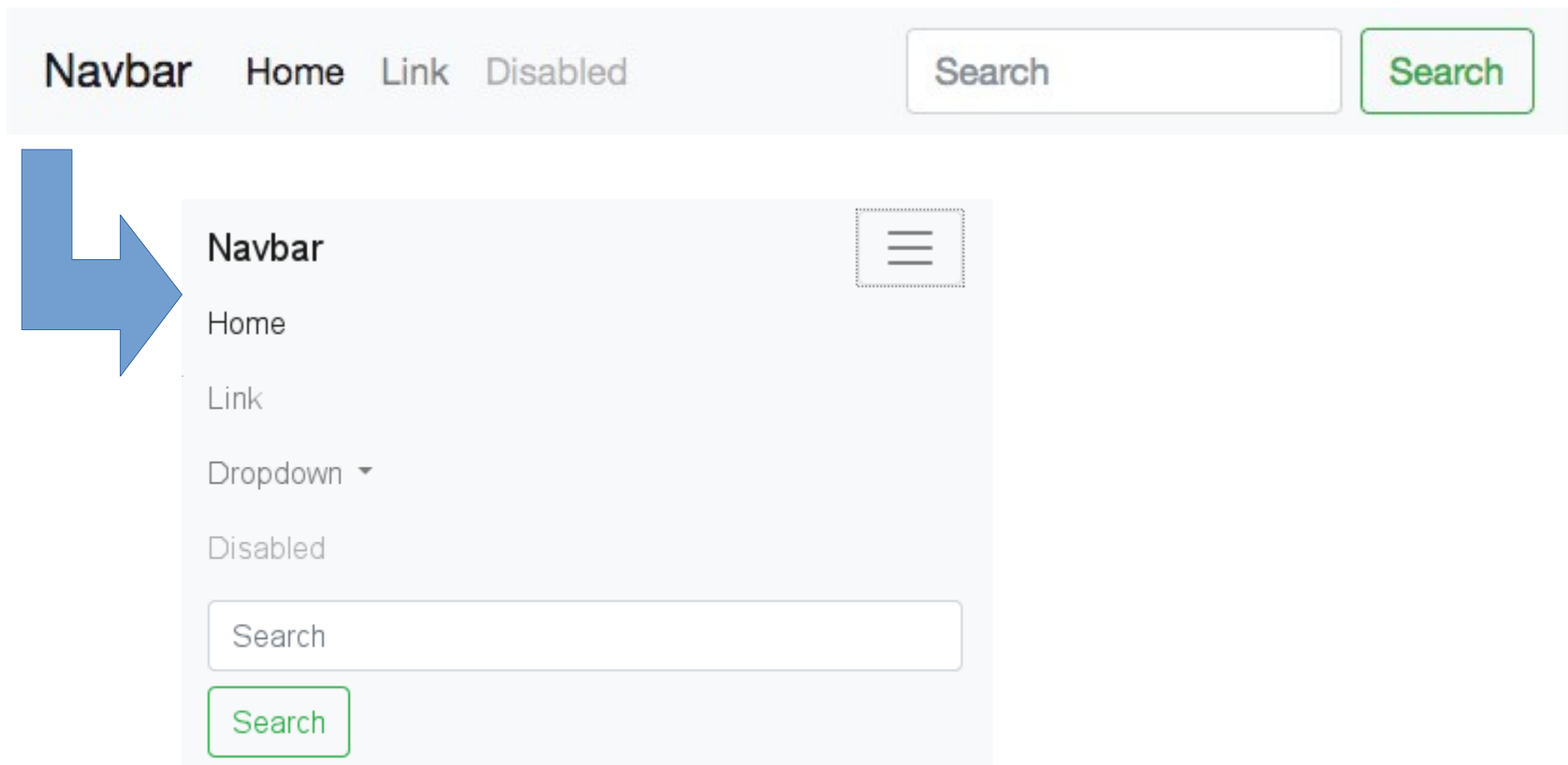
El contenido que soporta es:

- **.navbar-brand** para nuestra compañía, producto o nombre de nuestro proyecto.
- **.navbar-nav** para darle toda la altura y anchura de la navegación (incluye el soporte para dropdowns).
- **.navbar-toggler** Para usarlo con el plugin de collapse de Bootstrap y otros comportamientos de navegación.
- **.form-inline** Para cualquier control y acción de los formularios
- **.navbar-text** Para añadir un centrado vertical a las cadenas de texto
- **.collapse.navbar-collapse** Para agrupar y ocultar el contenido del navbar utilizando un breakpoint.

NAVBAR (con Responsive)

Vamos a ver un ejemplo de todos los componentes que podemos incluir en un tema responsive del navbar que de forma automática se collapse si no estamos con una resolución lg (grande) y no cabe.

Código (navbar.html)



NAVBAR

Estilo Brand

La clase `.navbar-brand` se puede aplicar a la mayoría de los elementos, pero debemos tener en cuenta que algunos elementos pueden requerir de utility clases o de los estilos personalizados. Ver ejemplo

navbar2.html

Navbar

Navbar

NAVBAR

Del mismo modo que hemos puesto un texto, podíamos haber puesto la combinación de un texto y una imagen o simplemente la imagen. Veamos el **ejemplo navbar3.html**



Si queremos por ejemplo ver cómo crear un navbar en nuestro nav, cómo incorporar en nuestro navbar elementos de un formulario, utilizar desplegados, o los colores que podemos emplear para nuestros navbar, haremos uso de la documentación oficial de aquí: _

<https://getbootstrap.com/docs/4.1/components/navbar/#nav>

PAGINACIÓN

La [paginación](#) se utiliza para tener un bloque de enlaces y poder acceder a ellos de forma fácil. La forma de construirlo es mediante una lista de elementos en HTML:

- 1) Estos enlaces deben estar agrupados dentro del elemento `<nav>`
- 2) Cada uno de los elementos tendrá la clase `.page-item`
- 3) Cada enlace la clase `.page-link`, veamos un ejemplo tanto si usamos iconos como si usamos texto.

PAGINACIÓN

Ver ejemplo **pagination.html**



```
<nav aria-label="Page navigation example">
  <ul class="pagination">
    <li class="page-item"><a class="page-link"
href="#">Previous</a></li>
    <li class="page-item"><a class="page-link" href="#">1</a></li>
    <li class="page-item"><a class="page-link" href="#">2</a></li>
    <li class="page-item"><a class="page-link" href="#">3</a></li>
    <li class="page-item"><a class="page-link"
href="#">Next</a></li>
  </ul>
</nav>
```

PAGINACIÓN

Con iconos:



Paginación con iconos, donde para ello deberemos utilizar los atributos **aria** y utilizar la clase **.sr-only**:

ver pagination2.html



Para desactivar algún elemento de la paginación porque por ejemplo no hay una página previa usaremos la clase **.disabled**, y para indicar que estamos en una página en concreto usaremos la clase **active**.

PAGINACIÓN

Mientras que la clase `.disabled` utiliza `pointer-events: none` para intentar desactivar la funcionalidad del enlace del element HTML `<a>`, esa propiedad todavía no está estandarizada y por tanto no tiene en cuenta la navegación por el teclado. Como tal, siempre debemos añadir **`tabindex="-1"`** en los enlaces deshabilitados y el uso de JS personalizado para desactivar completamente su funcionalidad (ver ejemplo `pagination3.html`).



POPOVERS

El siguiente componente a estudiar será el [popover](#):

- Para poder utilizar los popovers en Bootstrap, debemos incluir la librería de terceros [Popper.js](#), y esta debe ser incluida antes de incluir la de Bootstrap para que funcione.
- Necesita el `requires` [util.js](#)
- El uso de un popover es opcional, por tanto en el caso de querer usarlo debemos ser nosotros quien lo inicialicemos.
- Un título y contenido vacío, nunca se mostrará como popover

POPOVERS

- Especifica `container:'body'` para evitar problemas de renderizados en muchos components complejos (como en los grupos de inputs que tiene Bootstrap, grupos de botones, etc.)
- No funciona disparar un popover en elementos ocultos.
- Los popovers en atributos o clases `disabled` deben ser activados en un elemento envuelto
- Cuando disparamos un popover desde un hyperlink que contiene multiples líneas este debería ser centrado. Para ello usaremos `white-space: nowrap;` en nuestro elemento `<a>` para evitar este comportamiento

POPOVERS

- Popovers deberían ser ocultos antes que sus correspondientes elementos HTML hayan sido eliminados del DOM. Una forma de inicializar cualquier popover en una página web puede ser seleccionandolos con el atributo `data-toggle`:

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

POPOVERS

Ejemplo usando la opción container

Cuando tenemos algún estilo en un elemento padre que interfiere con un popover, querrás especificar un contenedor personalizado, de modo que el popover aparezca dentro de ese elemento.

```
$(function () {  
  $('.example-popover').popover({ container: 'body' })  
})
```

Los popovers se pueden posicionar arriba, abajo, a la izquierda y a la derecha alineados, veamos un ejemplo a la derecha.

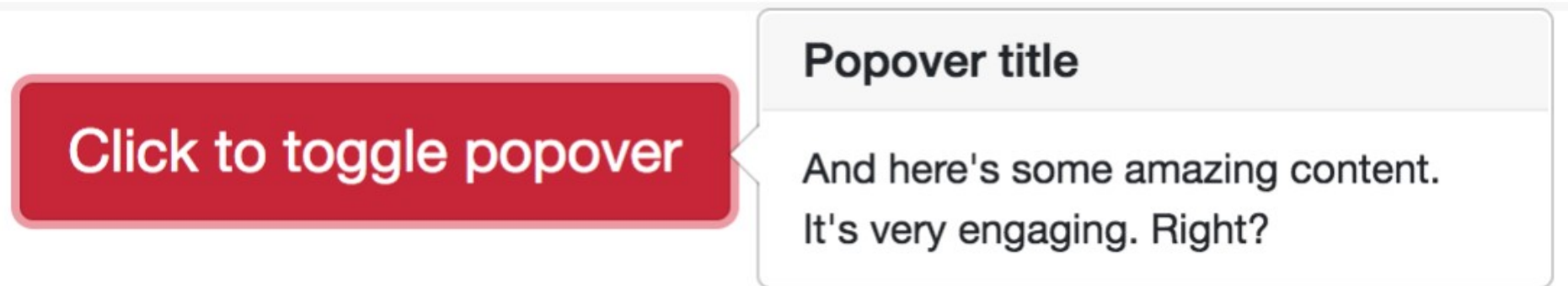
POPOVERS

Los popovers se pueden posicionar arriba, abajo, a la izquierda y a la derecha alineados, veamos un ejemplo a la derecha. Ver el código **popover.html**

```
<button type="button" class="btn btn-lg btn-danger" data-  
toggle="popover" title="Popover title" datacontent="And here's  
some amazing content. It's very engaging. Right?">
```

Click to toggle popover

```
</button>
```



Para ver más propiedades como que sólo funcione en el primer click, o el resto de opciones que tiene, consultar [aquí](#).

BARRA DE PROGRESO

El componente de [barra de progreso](#) se construye con dos elementos HTML, algunas propiedades CSS permiten establecer el ancho y algunos atributos. Eso lo véis en programación en el entorno del cliente (Javascript)

Nosotros no vamos a utilizar el element `<progress>` de HTML sino la clase que ha diseñado Bootstrap para ello.

BARRA DE PROGRESO

Ver ejemplo **progress.html**

```
<div class="progress">  
  <div class="progress-bar" role="progressbar" aria-valuenow="0" aria-valuemin="0" aria-  
valuemax="100"></div>  
</div>
```

..... el resto de barras

```
</div>  
<div class="progress">  
  <div class="progress-bar" role="progressbar" style="width: 100%" aria-valuenow="50"  
aria-valuemin="0" aria-valuemax="100">  
</div>
```



BARRA DE PROGRESO

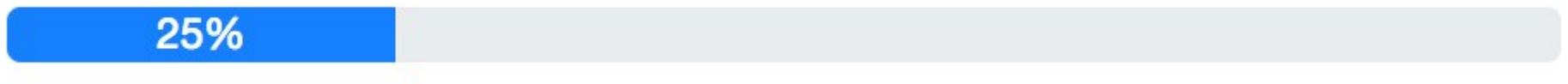
Bootstrap tiene un conjunto de herramientas que nos permite configurar de forma rápida y fácil las propiedades de la barra de progreso, podemos verlo aquí.

Del mismo modo que podemos jugar con la anchura, podemos probar a modificar la altura, el color del fondo, que nos muestre el % que lleva realizado, mostrar múltiples barras en una o animación en la barra de progreso.

BARRA DE PROGRESO

Ejemplo del % que ha realizado: ver **progress2.html**:

```
<div class="progress">  
  <div class="progress-bar" role="progressbar" style="width: 25%;" aria-  
valuenow="25" aria-valuemin="0" aria-valuemax="100">25%</div>  
</div>
```



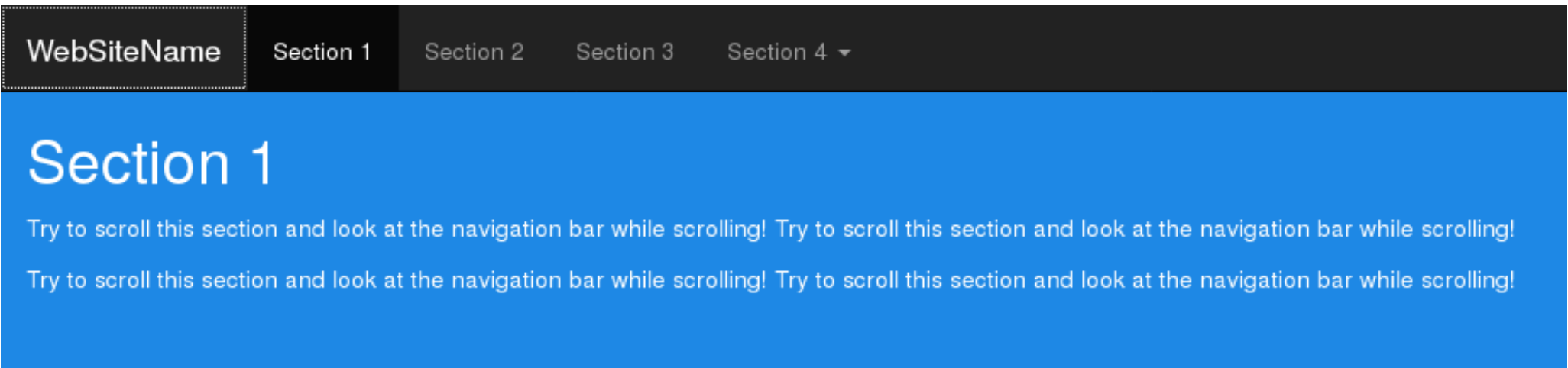
SCROLLSPY

El componente [Scrollspy](#) tiene las siguientes propiedades:

- Debemos utilizarlo con un componente nav o list group
- Requiere la propiedad **position: relative;** en el element que estamos observando, normalmente es el **<body>**
- Cuando observamos elementos diferentes del **<body>**, debemos asegurarnos que tiene un altura establecida y la propiedad **overflow-y: scroll;**
- La etiqueta **<a>** es requerida y con el elemento **id**

SCROLLSPY

Cuando lo tengamos realizando, nuestro nav o list group se irá actualizando moviendo la clase `.active` de un item a otro basado en el target que tenga asignado. **Ver ejemplo scrollspy.html**



SCROLLSPY

El scrollspy también funciona con nav nested. Si un nested .nav está active, su padre también lo estará. Al hacer scroll en el área que hay al lado del navbar podemos ver cómo las clases del navbar van cambiando en función del punto en el que nos encontramos.

El complemento Scrollspy se utiliza para actualizar automáticamente los enlaces en una lista de navegación basada en la posición de desplazamiento.

SCROLLSPY

Se debe de agregar `data-spy = "scroll"` al elemento que debe usarse como área desplazable (a menudo este es el elemento `<body>`).

Luego, agregue el atributo de destino de datos con un valor de id o el nombre de clase de la barra de navegación (.navbar). Esto es para asegurarse de que la barra de navegación esté conectada con el área desplazable.

Tenga en cuenta que los elementos desplazables deben coincidir con el ID de los enlaces dentro de los elementos de la lista de la barra de navegación (`<div id = "section1">` coincide con ``).

SCROLLSPY

El atributo de desplazamiento de datos opcional especifica el número de píxeles que se deben desplazar desde arriba cuando se calcula la posición del desplazamiento. Esto es útil cuando sientes que los enlaces dentro de la barra de navegación cambian el estado activo muy pronto o al saltar a los elementos desplazables. El valor predeterminado es 10 píxeles.

Requiere posicionamiento relativo: el elemento con `data-spy = "scroll"` requiere la propiedad de posición CSS, con un valor de "relativo" para que funcione correctamente.

Tooltips

Un **Tooltip** es un pequeño cuadro emergente que aparece cuando el usuario mueve el puntero del mouse sobre un elemento.

Se debe de agregar el atributo `data-toggle = "tooltip"` a un elemento, y usar el atributo `title` para especificar el texto que se debe mostrar dentro de la información sobre herramientas.

Debe de ser inicializado con JQuery. **Ver ejemplo [tooltip.html](#)**

Bibliografía

Página oficial desde donde poder consultar toda la documentación y poder descargar la librería: <http://getbootstrap.com>

Blog oficial de Bootstrap donde consultar las últimas novedades:

<http://blog.getbootstrap.com/>

Temas y plantillas gratuitas para Bootstrap:

o <http://startbootstrap.com> o <http://bootstrapzero.com> o
<http://bootswatch.com> o <http://www.bootbundle.com>

Ejemplos y trozos de código útiles para Bootstrap: <http://bootsnipp.com>

Ejemplos de uso: <http://expo.getbootstrap.com/>

Listado de los recursos disponibles: <https://startbootstrap.com/bootstrap-resources/>