Bootstrap

Escuela Técnica Superior de Ingeniería de Telecomunicación
Universidad Rey Juan Carlos

gsyc-profes (arroba) gsyc.urjc.es

Marzo de 2021

Derivado a partir de material de
Jesús M. González-Barahona y Gregorio Robles.
El original está disponible en
`http://cursosweb.github.io`

¿Qué es Bootstrap?

- Bootstrap es un framework libre para desarrollo web
- Desarrollado inicialmente en 2011 por ingenieros de Twitter
- Incluye plantillas HTML y CSS con tipografías, formas, botones, cuadros, barras de navegación, carruseles de imágenes y muchas otras
- También existe la posibilidad de utilizar plugins de JavaScript
- Aunque su preferencia es *mobile first*, permite crear diseños que se ven bien en múltiples dispositivos (*responsive design*)
- Orientado a programadores, no a diseñadores gráficos
- Es posiblemente la herramienta más popular para este fin, aunque hay alternativas como Foundation

Características de Bootstrap

Ventajas

- Resulta sencillo y rápido escribir páginas con muy buen aspecto
- Se adapta a distintos dispositivos (*responsive design*)
- Proporciona un diseño consistente
- Es compatible con los navegadores modernos
- Es software libre

Inconvenientes

- Al ser una herramienta muy popular, las páginas web que no estén personalizadas *quedan iguales que las de todo el mundo*
- No es especialmente fácil personalizar los estilos (Foundation puede ser más adecuado para esto)

Ficheros de Bootstrap

```
bootstrap/
|--- css/
|    |--- bootstrap.css
|    |--- bootstrap.css.map
|    |--- bootstrap.min.css
|    |--- bootstrap-theme.css
|    |--- bootstrap-theme.css.map
|    |--- bootstrap-theme.min.css
|--- js/
|    |--- bootstrap.js
|    |--- bootstrap.min.js
|--- fonts/
     |--- glyphicons-halflings-regular.eot
     |--- glyphicons-halflings-regular.svg
     |--- glyphicons-halflings-regular.ttf
     |--- glyphicons-halflings-regular.woff
     |--- glyphicons-halflings-regular.woff2
```

Holamundo en Bootstrap

```
<!DOCTYPE html>
<html lang="es-ES">
<head>
  <meta charset="utf-8">
  <title>Hola mundo en bootstrap</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet"
     href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
  <script
     src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js">
  </script>
  <script
     src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js">
  </script>
</head>
<body>
  <div class="container">
    <h1>
      Hola, bootstrap
    </h1>
  </div>
</body>
</html>
```

http://ortuno.es/hola_bootstrap.html

Bootstrap en CDN

- Con un CDN (Content Delivery Network) no hace falta tener Bootstrap en nuestros archivos. Además, si un usuario ya ha descargado esas URLs, probablemente las tenga ya en la caché del navegador (con el consiguiente ahorro de tiempo).

```
 <!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="http://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">

<!-- jQuery library -->
<script

 ↪  src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js">
</script>

<!-- Latest compiled JavaScript -->
<script

 ↪  src="http://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js">
</script>
```

Adaptación del contenido a la pantalla

Ya desde su diseño original, un requisito importante para el web es que las páginas se pueda representar en pantallas de cualquier tamaño. Con la aparición de los *stmart phones*, esto es aún más necesario y más complicado. A lo largo de los años se han usado varias técnicas para conseguir esto, cada vez mejores

1. Técnica inicial
   Viewport. Barras de desplazamiento horizontal y vertical, recomposición de los elementos sobre el *viewport*
2. Primeros *smart phones*
   Viewport virtual
3. Teléfonos móviles actuales
   Diseño *responsive* basado en *grid*

Viewport

Para diseñar webs en dispositivos móviles, es importante tener claro qué es el *viewport* y cómo se comporta

- *Viewport* es la zona visible de una página web. En los navegadores tradicionales de escritorio, coincide con la ventana del navegador
- Supongamos una página web grande y compleja, como la portada de un periódico. La página no cabrá en la ventana del navegador, el usuario usará las barras de scroll para mover el *viewport* sobre el documento. Al redimensionar la ventana, cambiará el tamaño del *viewport*
- Cambiar el tamaño del viewport reposiciona el texto y todos los elementos: las líneas se truncan, las imáges se recolocan, etc

Viewport es un rectángulo donde se compone un fragmento (tal vez completo) de la página web para presentarla al usuario

Viewport virtual

Con la aparición de los navegadores en teléfonos móviles, los cambios del tamaño de la pantalla son mucho más drásticos. La técnicas tradicionales siguen funcionando, pero proporcionan una experiencia de uso muy poco satisfactoria

- El área visible de un móvil es demasiado pequeña, componer una página web tradicional en ese *viewport* queda mal. Observa lo que sucede en esta página antigua cuando la ventana es muy grande o muy pequeña
  `https://tinyurl.com/y7e77lvw`
- Además, en un navegador para móvil no hay barras de scroll, ocuparían un espacio demasiado valioso. Ni ventanas, serían demasiado pequeñas

Para solucionar este problema, aparece el concepto del *viewport virtual*, mayor que el *viewport* ordinario (la pantalla)

- Lo introduce Apple para Safari en iOS, luego pasa a ser estándar
- El ancho del *viewport* virtual es razonablemente grande, por ejemplo 980 pixeles en el navegador safari para iPhone
- El navegador compone la página sobre este viewport virtual, ya no hacen falta barras desplazamiento horizontal
- El usuario arrastra el *viewport* (la pantalla, más pequeña) sobre el viewport virtual, para que le muestre una zona u otra del documento. También se le puede permitir hacer zoom
  - Redimensionar este viewport ya no provoca la recomposición de la página

Páginas responsive

Una página web moderna con un minimo de calidad se entiende que tiene que ser *responsive*

- La página se adapta al tamaño de la pantalla (escritorio, tablet, móvil), sin usar la barra de desplazamiento horizontal, que es muy incómoda. La barra de desplazamiento vertical se sigue usando, no es molesta
- El diseño *responsive* tal y como lo conocemos en la actualidad se basa en el uso de un *grid*. En español se traduce por cuadrícula o rejilla. Aquí veremos el *grid* de Bootstrap, hay otros pero siempre son similares
- En estas páginas ya no hace falta un *viewport* virtual, porque la página está diseñada para adaptarse al *viewport* ordinario (la pantalla pequeña)

La misma cuadrícula de 12 elementos se presenta de forma distinta en un ordenador

XXX
XXX

En un tablet

XX
XX
XX

En un móvil

X
X
X
X
X
X

Responsive design

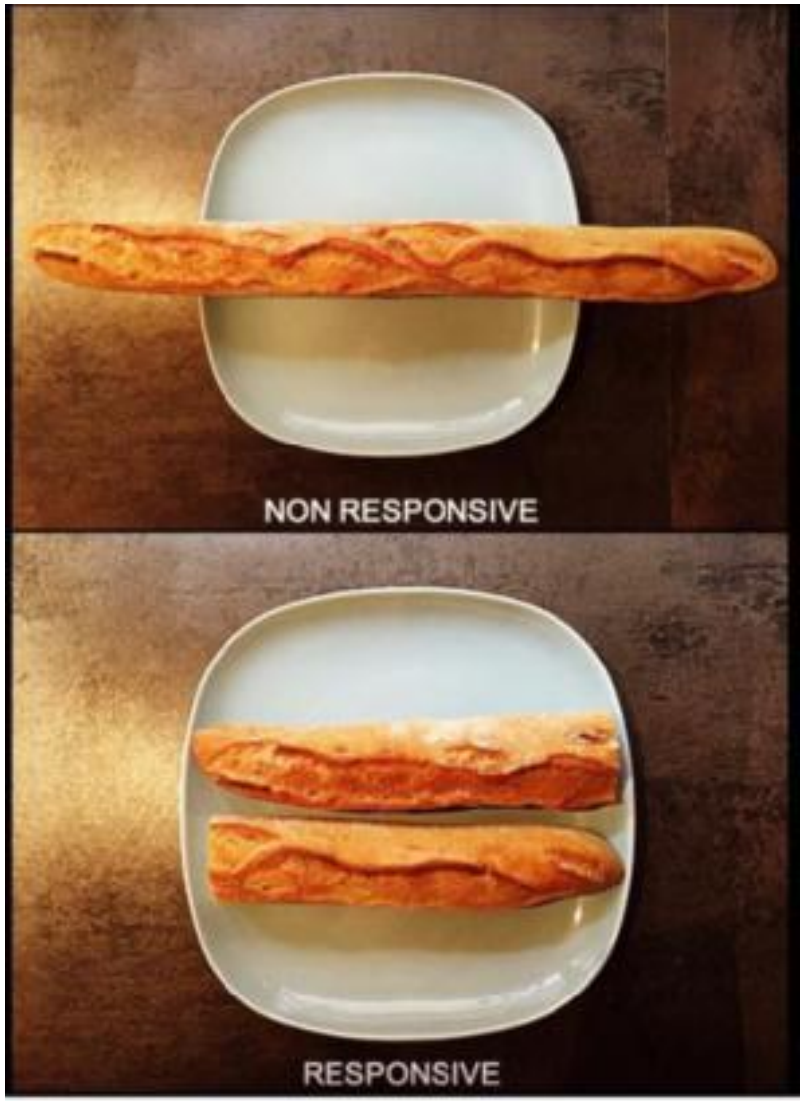


Foto: https://image-store.slidesharecdn.com/420d15aa-cbf0-4ded-ac42-fcf728610bc1-original.jpeg

Bootlint

- Herramienta que detecta algunos errores comunes el HTML de diseños `Bootstrap`
- Comprueba que las instancias de componentes Bootstrap han sido correctamente estructurados
- Analiza también la inclusión de ciertas etiquetas $< meta >$, la declaración DOCTYPE HTML5, etc.
- Supone que el HTML es correcto, así que deberíamos usar previamente alguna herramienta como el W3C validator o similar
- Página web: `https://github.com/twbs/bootlint`

- Instalación:

```
sudo apt install npm
sudo npm install -g bootlint
```

- bootlint mostrará un warning

```
missing X-UA-Compatible <meta> tag that disables
old IE compatibility modes
```

  - Internet Explorer 6 (años 2001-2008) fue un navegador muy usado, que no seguía las normas HTML
  - Durante algún tiempo pudo ser recomendable que quien usase HTML estándar (y no HTML IE 6), lo indicara con

    ```
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    ```

  - En la actualidad, añadir esta indicación se considera obsoleto. Podemos ignorar este warning de bootlint

Mobile first

- Con una propiedad de etiqueta meta, podemos indicar la escala inicial del *viewport*
- Como las páginas con bootstrap son *responsive*, especificamos que el *viewport* virtual coincida con el ancho de la pantalla, esto es, con el *viewport* ordinario. En otras palabras: que no haya un *viewport* virtual

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

- También se puede inhabilitar el zoom en dispositivos móviles con `user-scalable=no`
- Los usuarios sólo podrán hacer *scroll* y tendrá una apariencia nativa.

```
<meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, user-scalable=no">
```

Contenedores

- Todos los elementos de Bootstrap deben estar dentro de un elemento contenedor
- Para un contenedor responsivo de tamaño fijo, se usa `.container`

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

- Si se desea un contenedor con el ancho total (del *viewport*), se ha de usar `.container-fluid`

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

- Los contenedores no se pueden anidar

El sistema de rejilla (I)

La página se organiza en filas. Cada fila es un *div* de HTML con la clase *row*

```
<div class="row">
</div>
```

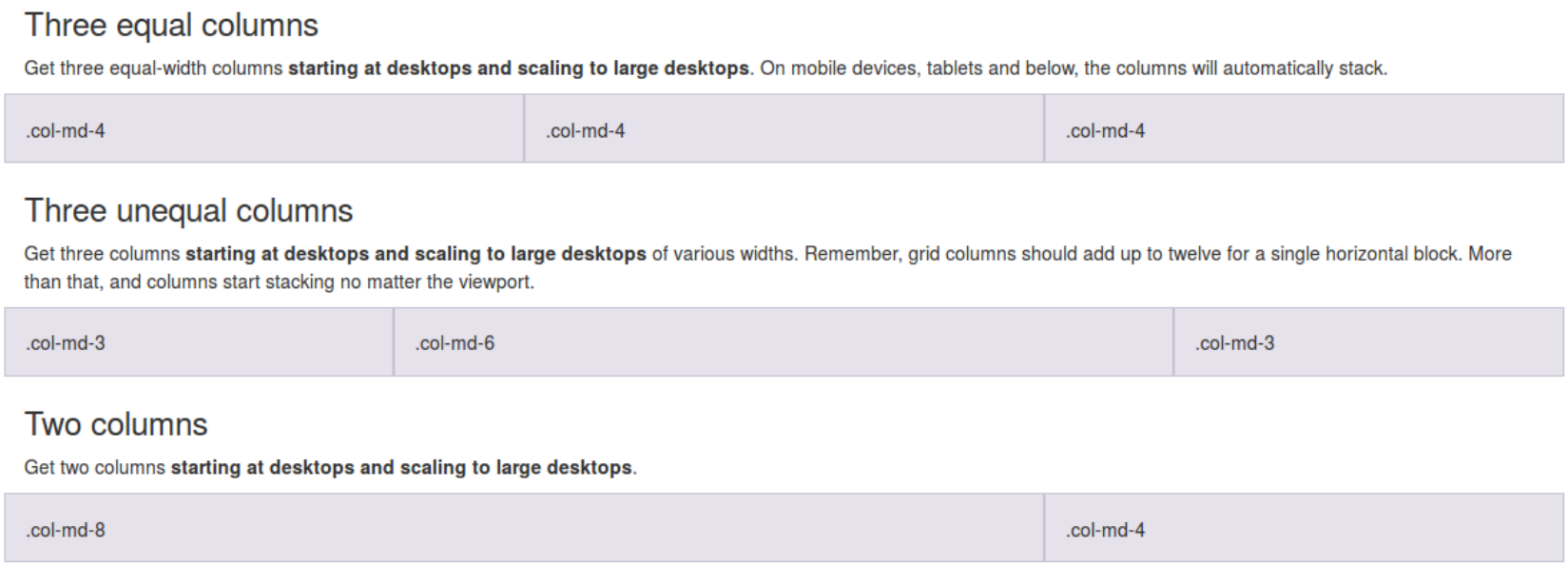
Dentro de cada fila hay columnas. Cada columna es un *div* de alguna de estas clases

```
col-lg-N

col-md-N

col-sm-N

col-xs-N
```

Donde N es un número entre 1 y 12, que indica el ancho de cada columna. El total de las columnas de cada fila tiene que sumar 12

El sistema de rejilla (II)

Esto son tres ejemplos típicos de organización de la rejilla



Podemos usar cualquier número de columnas, de cualquier ancho, con tal de que el total ocupe 12 cuadrículas

El sistema de rejilla (III)

Dicho de otro modo

- El ancho de cada columna se mide en casillas
- En Bootstrap, la pantalla ocupa un ancho total de 12 casillas, que repartimos entre el número de columnas que deseemos
- Si el *viewport* (la pantalla) es lo bastante ancho, las columnas se muestran en su disposición normal, esto es, cada una al lado de la otra
- Si el *viewport* es demasido pequeño, entonces las casillas se apilan verticalmente

Hay 4 tipos de casillas, según el ancho del *viewport* disponible en el dispositivo

- lg
  Large. Pantallas alta resolución. 1200 pixels y más
- md
  Medium. Pantallas tradicionales. 992-1199 pixeles
- sm
  Small. Tablets. 768-991 pixeles
- xs
  Extra small. Teléfonos moviles. 767 pixeles o menos

La frontera entre cada uno de estos tamaños se denomina *breackpoint*

- Columnas *lg*
  Disposición normal en pantallas grandes
  se apilan en: pantallas medianas, pequeñas o muy pequeñas
- Columnas *md*
  Disposición normal en pantallas medianas o grandes
  Se apilan en pantallas pequeñas o muy pequeñas
- Columnas *sm*
  Disposición normal en pantallas pequeñas, medianas o grandes
  Se apilan en pantallas muy pequeñas
- Columnas *xs*
  Disposición normal en pantallas muy pequeñas, medianas o grandes.
  Nunca se apilan

Dicho de otro modo

- Cada tipo de columna se muestra en su disposición normal, esto es, horizontalmente, si la pantalla es de su tipo o de un tipo mejor
- En otro caso, las casillas se apilan verticalmente

Esto parece un poco complicado, pero con el siguiente ejemplo verás que no:

1. Vete a `http://ortuno.es/grid`
2. Maximiza la ventana
3. Vete reduciéndola gradualmente

Columnas desplazadas

Además de indicar el ancho de una columna, podemos especificar que se desplace un cierto número de casillas a la derecha, esto es, que se dejen casillas en blanco
Basta añadir un nuevo valor al atributo class del div

- `col-lg-offset-N`
- `col-md-offset-N`
- `col-sm-offset-N`
- `col-xs-offset-N`

Donde N es el número de casillas, entre 1 y 11
Ejemplo

```
<div class="col-md-4 col-md-offset-4">
```

Componentes de Bootstrap

Bootstrap viene con una serie de estilos (generalmente en formato de clase CSS) y componentes en JavaScript.

- `panel`
- `btn`
- `nav`
- `dropdown`
- `carousel`
- y otras utilidades responsivas

panel

Un panel es un componente de bootstrap consistente en una caja redondeada donde se pueden insertar otros elementos

- Sintácticamente, es un div cuyo atributo class tiene el valor `panel` y el valor del tipo de panel, que puede ser
  - `panel-default`
  - `panel-primary`
  - `panel-success`
  - `panel-info`
  - `panel-warning`
  - `panel-danger`
- Cada panel se compone de
  - Cabecera: un div de clase `panel-heading`
  - Cuerpo: un div de clase `panel-body`
  - Pie: un div de clase `panel-footer`

Ejemplo:
`http://ortuno.es/panel`

btn

A partir de los elementos HTML `<a>`, `<button>` o `<input>`, se puede generar un botón de Bootstrap, basta añadir la clase btn

- Adicionalmente, se puede especificar el tipo de botón: btn-default btn-xs btn-sm btn-lg btn-block btn-primary btn-success btn-info btn-warning btn-link

Ejemplo:
`http://ortuno.es/button`

nav

`nav` es un componente de Bootstrat formado por un conjunto de enlaces para navegar dentro de un web. Hay tres tipos

- `nav-stacked`
  Tipo por omisión. Enlaces apilados verticalmente
- `nav-tabs`
  Enlaces en forma de pestaña
- `nav-pills`
  Enlaces en forma de botón, llamados *pills*

Sintácticamente, un `nav` es

- Un elemento `<ul>` al que se añade el atributo `class` con el valor `nav` más el valor del tipo de nav:
  `nav-stacked`, `nav-tabs`, `nav-pills`
- Cada uno de los enlaces es un `<li>` que contiene el `<a>` correspondiente
- Cada `<li>` puede incluir, adicionalmente, el atributo `class` con el valor
  - `active` para indicar que se trata del botón/pestaña actual
  - `disabled`

Ejemplo:
`http://ortuno.es/nav`

dropdown

Un `dropdown` es un menú desplegable: un menú que aparece al pulsar *algo*, llamado *toggle*

- El `toggle` puede ser
  - `<a>`
  - `<btn>`, aislado o dentro de `<btn-group>`
  - Una pestaña o un botón de un `nav` ya sea `nav-stacked`, `nav-tabs` o `nav-pills`
  - Algunos otros elementos
- El menú es un `<ul>` con sus `<li>`

El `toggle` y el `<ul>` tienen que estar contenidos dentro de un elemento

- Que tendrá la clase `<dropdown>`
- Podrá ser un `div` genérico o un `btn-group`, o un `nav`

- El toggle tiene que tener los atributos

  `data-toggle="dropdown" class="dropdown-toggle"`

- Para que el usuario sepa que el toggle es desplegable, se añade el símbolo *caret*, también llamado acento circunflejo (un pequeño triángulo apuntando hacia abajo)

  `<span class="caret"></span>`

- El `<ul>` del menú tendrá los atributos

  `<ul class="dropdown-menu">`

Ejemplo:
`http://ortuno.es/dropdown`

Formularios

Bootstrap incluye clases para mejorar el aspecto y usabilidad de los formularios

- El uso de `<label>` es necesario, no es válido escribir texto HTML para identificar los elementos del formulario
- Cada par `<label>` - elemento de entrada (`<input>`, `<select>`, `<textarea>`, etc) tiene que ir dentro de un `<div>` de clase `form-group`
- Los elementos de entrada de texto tienen que llevar la clase `form-control` (los checkbox, radiobutton y similares, no)

Esto hace, entre otras cosas, que los elementos de entrada ocupen todo el ancho del container

```
<form action="/action_page.html">

  <div class="form-group">
    <label for="usuario"> Nombre de usuario:</label>
    <input type="text" id="usuario" class="form-control"
    ↪  name="usuario">
  </div>

  <div class="form-group">
    <label for="contrasenya"> Contraseña:</label>
    <input type="password" name="contrasenya" id="contrasenya"
    ↪  class="form-control">
  </div>

  <input type="submit">
</form>
```

Formulario inline

La clase `form-inline` ofrece una presentación más compacta del formulario, con los campos en horizontal

- Basta añadir `class="form-inline"` al elemento `<form>`

```
<form action="/action_page.html" class="form-inline">
   ...
</form>
```

Ejemplo con diversos formularios Bootstrap:
`http://ortuno.es/bform`

carousel

El componente `carousel` muestra fotografías como un pase de diapositivas, desplazándose horizontalmente y con un texto en cada una de ella
Un carousel es un div con los siguientes atributos

- Clase `carousel slide`
- Atributo `data-ride` con el valor `carousel` para especificar que el carrusel se ponga en marcha en cuanto se cargue la página
  - Si lo omitimos, el carrusel solo se mueve al hacer clic sobre controles izquierda y derecha
- Atributo `id` con el identificador de carousel
- La velocidad de transición de las diapositivas se puede especificar, en milisegundos, con el atributo `data-interval`
  Su valor por omisión es 5000

Dentro del `div` principal de carousel habrá:

- Indicators
  Son los círculos que aparecen debajo del texto para representar la diapositiva activa respecto a todas las demás
- Un `<div>` de clase `<carousel-inner>` con las fotos y el texto
- Controles izquierda y derecha

Los indicators son una `<ol>` de clase `carousel-indicators`

- Cada `<li>` del `<ol>` contiene un atributo `data-target` con el identificador del carrusel y un `data-slide-to` con el ordinal de la imagen

```
<ol class="carousel-indicators">
  <li data-target="#carrusel01" data-slide-to="0" class="active"></li>
  <li data-target="#carrusel01" data-slide-to="1"></li>
  <li data-target="#carrusel01" data-slide-to="2"></li>
</ol>
```

Cada diapositiva es un `div` de clase `item`
Contiene:

- Un elemento `img` con la imagen
- Un `div` de clase `carousel-caption` con los textos

```
<div class="item">
   <img src="images/foto02.jpg" alt="Segunda foto">
   <div class="carousel-caption">
      <h3>Título de la foto 2</h3>
      <p>Lorem Ipsum</p>
   </div>
</div>
```

Una forma conveniente de redimensionar las fotos es especificar el atributo `width` directamente en cada imagen, dejando que el navegador fije el alto manteniendo las proporciones

Los controles izquierda y derecha son elementos <a>

- Contienen en el atributo `href` el identificador del carousel
- En el resto de atributos se indica qué caracter usar para el control

```
<a class="left carousel-control" href="#carrusel01"
↪  data-slide="prev">
  <span class="glyphicon glyphicon-chevron-left"></span>
  <span class="sr-only">Previous</span>
</a>
<a class="right carousel-control" href="#carrusel01"
↪  data-slide="next">
  <span class="glyphicon glyphicon-chevron-right"></span>
  <span class="sr-only">Next</span>
</a>
```

Ejemplo:
`http://ortuno.es/carousel`

Deshabilitar elementos

Como hemos visto, muchos elementos bootstrap admiten la clase *disabled* para indicar que tengan un aspecto gráfico distinto, deshabilitado

- Pero esto no los deshabilita realmente. (Bootlint nos avisa con un warning)
- Para deshabilitar por completo un elemento, usamos el atributo *disabled*

```
<button type="button" class="btn btn-lg" disabled>Botón</button>
<input type="text" name="lname" disabled><br>
```

Enlaces relacionados

- S.F.Rahman. Jump Start Bootstrap. Ed SitePoint, 2014

  `http://proquest.safaribooksonline.com/book/web-design-and-development/9781457174346`
- Listado de recursos sobre Bootstrap
  `http://bootsnipp.com/resources`
- Bootsnipp: Cientos de componentes adicionales
  `http://www.bootsnipp.com`
- Startboostrap: Plantillas y temas Bootstrap (gratis)
  `http://startbootstrap.com/`
- WrapBootstrap: Plantillas y temas Bootstrap (de pago)
  `https://wrapbootstrap.com/`
- BootTheme: Generador (no libre) de diseños Bootstrap
  `http://www.boottheme.com/`