

Bootcamp Java Developer

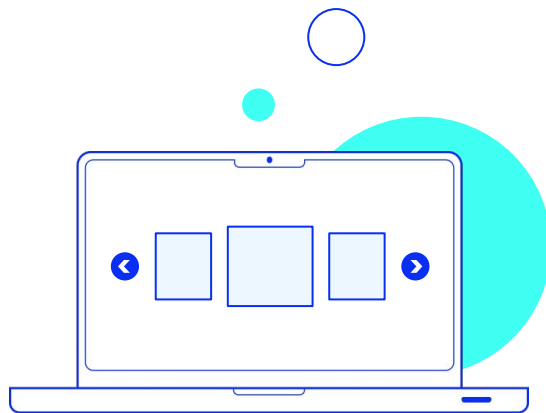
Fase 2 - Java Web Developer
Módulo 16



Carousel

Carousel

En este componente la idea es realizar un ***slider*** **de imágenes** que puedan incluirse fácilmente; para poder colocar, en general, en la parte del header de nuestra interfaz, como en cualquier otra sección de la misma.



Primer paso: vinculación a CSS y JS

1. Al igual que con cualquier otro componente, necesitamos tanto de la **vinculación a CSS, como al JS de Bootstrap**. Veamos el ejemplo siguiente:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/  
bootstrap.min.css" rel="stylesheet"  
integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLWZJEdK2Kad  
q2F9CUG65" crossorigin="anonymous">  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/  
bootstrap.bundle.min.js"></script>
```

Estructura e inclusión de imágenes

2. Vamos a dividir nuestra explicación en dos partes. En principio, vamos a **generar el html**, como se ve en la imagen del [siguiente slide](#); donde incluiremos las imágenes y la estructura principal a seguir. La clase llamada **.carousel**, la clase **.slide**, así como **.carousel-inner** nos permiten configurar la estructura principal.

Por otro lado, dependiendo de la **imagen que queramos que esté activa** cuando veamos el carousel por primera vez, sumaremos la clase **.active** a la clase generada para cada elemento del carousel: **.carousel-item**.



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

Control del carrousel

3. A la estructura anterior, vamos a agregar un **id**. Este permitirá la **interacción con los controles** (para poder ir hacia atrás o hacia adelante del carrousel).

Nota: si no se agrega la clase **.active** a ningún elemento, entonces el carrousel no se verá y por ende no funcionará.

```
<div id="miCarousel" class="carousel slide">
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
</div>
```

4. Dentro de la estructura principal ya vista, agregaremos:

```
<div id="miCarousel" class="carousel slide">
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>

  <button class="carousel-control-prev" type="button" data-bs-target="#miCarousel" data-bs-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Anterior</span>
  </button>
  <button class="carousel-control-next" type="button" data-bs-target="#miCarousel" data-bs-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Previo</span>
  </button>
</div>
```


Controles en profundidad

5. Los **botones**, tienen una relación directa con el **id** del carrousel que vimos anteriormente (“**miCarousel**”). Por otro lado, las clases tanto **.carousel-control-prev**, así como la llamada

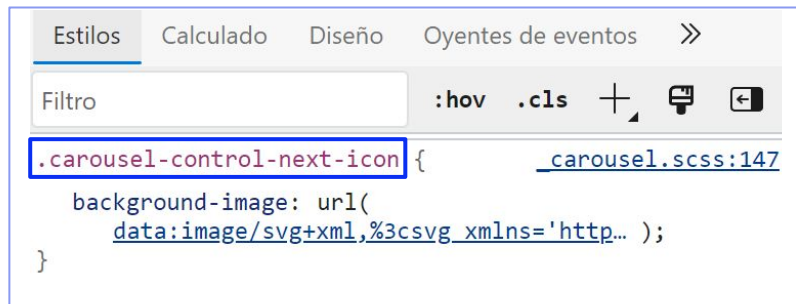
.carousel-control-next, permiten indicar cuál de los dos elementos irá hacia el **contenido posterior o anterior**. Veamos el ejemplo debajo:

```
<button class="carousel-control-prev" type="button" data-bs-target="#miCarousel" data-bs-slide="prev">
  <span class="carousel-control-prev-icon" aria-hidden="true"></span>
  <span class="visually-hidden">Anterior</span>
</button>
<button class="carousel-control-next" type="button" data-bs-target="#miCarousel" data-bs-slide="next">
  <span class="carousel-control-next-icon" aria-hidden="true"></span>
  <span class="visually-hidden">Previo</span>
</button>
```

En el ejemplo anterior vimos que el atributo **.data-bs-slide** con el valor **prev** o **next** (dependiendo el botón), realiza el resto de la acción, como veremos en el siguiente slide.

Nota: Los iconos pueden modificarse. Si notamos en la imagen, **carousel-control-prev-icon** y **carousel-control-next-icon** indican el ícono de Bootstrap a utilizar, pero podemos modificar esa clase desde nuestro css y **cambiar el ícono por el que necesitamos**.

En la imagen de la derecha, vemos una captura del **inspector de elementos**, donde claramente podemos diferenciar de dónde es traída la imagen del ícono que se visualiza y cómo cambiarla desde nuestra propia hoja de estilos.



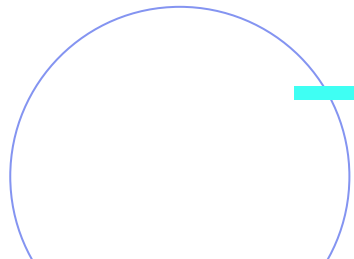
Controles individuales de imagen

6. Como en todo *slider* o carrousel, podemos **incluir un control para ir específicamente a cada imagen o elemento**.

Nota: decimos *imagen o elemento*, porque se puede incluir **texto** o **videos**, no necesariamente imágenes, en este carrousel.

Al comienzo de nuestro contenedor, se suma la clase **.carousel-indicators** con el contenido como muestra la imagen de la [pantalla siguiente](#).

Cada botón o indicador tendrá un atributo llamado **.data-bs-slide-to** con el número de elemento o slide al que apunta. Es decir, que al hacer clic nos llevará a ese número de imagen con contenido. Por otro lado, volvemos a ver la relación con el ya generado **id** del carousel, en este caso como valor del atributo **.data-bs-target**.



```

  <body>
  |
  |   <div class="container">
  |   |   <div id="miCarousel" class="carousel slide">
  |   |   |
  |   |   |   <div class="carousel-indicators">
  |   |   |   |   <button type="button" data-bs-target="#miCarousel" data-bs-slide-to="0" class="active"></button>
  |   |   |   |   <button type="button" data-bs-target="#miCarousel" data-bs-slide-to="1"></button>
  |   |   |   |   <button type="button" data-bs-target="#miCarousel" data-bs-slide-to="2"></button>
  |   |   |   |   </div>
  |   |   |   </div>
  |   |   </div>
  |   </div>
  </body>

```



Si la **imagen** es la **primera o la última**, entonces el **ícono** será el de **previo o próximo** (dependiendo el caso no se hará visible):



Descripción o caption

7. Las imágenes o contenidos del carrousel pueden tener una descripción o **caption**, que se puede incluir de forma sencilla. Veamos la imagen a continuación:



```
<div class="carousel-inner">
  <div class="carousel-item active">
    
    <div class="carousel-caption d-none d-md-block">
      <h5>Primera Imágen</h5>
      <p>Acá voy a describir esta imagen</p>
    </div>
  </div>
```

Aclaración de clases utilizadas

- En gran parte de los ejemplos hemos visto reiteradas veces **d-block** o con su breakpoint, por ejemplo, **d-md-block**. Block es un valor de la propiedad **display** y lo que indica, como se muestra en el código de la próxima pantalla, es que la imagen será un **elemento de bloque**. O por ejemplo, que el texto será de bloque en pantallas mayores o iguales a 768px.
- Se suma también **w-100** para que se ubique la **imagen al 100%** y no permita elementos flotantes a sus costados.

Veamos los ejemplos
en la próxima pantalla.



```
  
<div class="carousel-caption d-none d-md-block">
```

```

```


**¡Sigamos
trabajando!**

