

# Bootcamp Java Developer

Fase 2 - Java Web Developer  
Módulo 16

# Formularios en Bootstrap

# Introducción a formularios

Bootstrap nos permite de forma simple trabajar con **clases asociadas a formularios**.

## Dar estilos a un formulario

Existen diversas maneras de dar estilos a un formulario. Este puede ser parte de una grilla trabajada con flex desde Bootstrap, o simplemente un contenedor con ciertas clases de margen y padding. Cualquiera sea la propuesta más o menos compleja, **.form-control** es la **clase a utilizar por excelencia en un formulario**.

Veamos un ejemplo en la próxima pantalla.

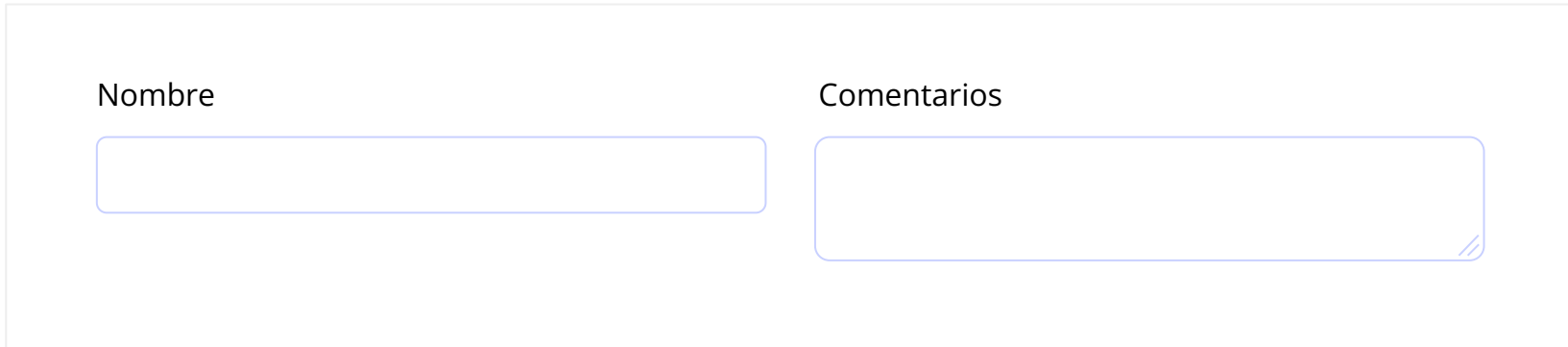


```
<main class="container mb-5">
  <form action="codigo.php">
    <div class="row">
      <div class="col-md-6">
        <label class="form-label">Nombre</label> <input type="text"
          class="form-control">
      </div>
      <div class="col-md-6">
        <label class="form-label">Comentarios</label>
        <textarea class="form-control"></textarea>
      </div>
    </div>
  </form>
</main>
```

## Etiqueta form-label

Es importante también incluir, por cuestiones vinculadas con la accesibilidad, la etiqueta **form-label** para anidar el texto que **indicará el contenido a escribir en ese elemento del formulario**.

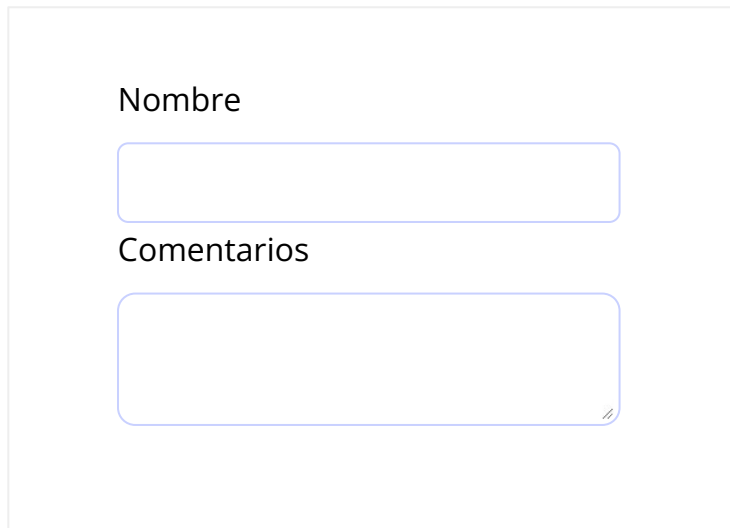
Vista pantalla desktop:



A screenshot of a desktop form layout. It features two input fields side-by-side. The first field is labeled 'Nombre' and the second is labeled 'Comentarios'. Both fields are empty and have a light blue border. The 'Comentarios' field has a small icon in the bottom right corner.

Nombre	Comentarios
<input type="text"/>	<input type="text"/>

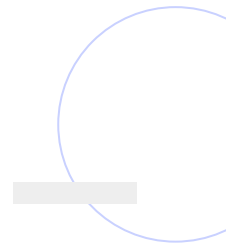
Vista pantalla mobile:



A mockup of a mobile screen view showing a form with two input fields. The first field is labeled "Nombre" and the second field is labeled "Comentarios". Both fields are empty and have a light blue border. The "Comentarios" field has a small icon in the bottom right corner.

Nombre

Comentarios



## Tamaño de campos

Los campos también pueden tomar un **tamaño mayor o menor**, **utilizando clases** que indiquen tal cambio.

El sumar **lg** o **sm** al final de la clase **form-control** modificará el valor por defecto del mismo.

```
<main class="container">
  <div>
    <label class="form-label">Nombre</label>
    <input type="text" class="form-control form-control-lg">
  </div>
  <div class="mb-2">
    <label class="form-label">Email</label>
    <input type="email" class="form-control
      form-control-sm">
  </div>
</main>
```

El **resultado** del paso anterior, será el siguiente:

Nombre

Email



## Transformar un campo en texto plano

Se puede integrar una clase llamada **plaintext** a **form-control** para lograr que la misma se muestre no como un campo común, sino como un **texto plano**; que según el diseño del formulario, puede resultar interesante. Siempre esta característica debe ir acompañada del atributo booleano **readonly** y un **value**, dado que es funcional a aquellos **campos donde el usuario sólo puede leer el contenido**.

Ese contenido a su vez es pasado a través del formulario a quien deba recibirlo, o guardarlo en una base de datos, pero **no puede ser modificado por el usuario; por eso su denominación de readonly**.



**Ejemplo** (código y resultado):

```
<div class="mb-2">  
  <label class="form-label">Email</label>  
  <input readonly type="email"  
    class="form-control-plaintext form-control-sm"  
    value="info@mimail.com">  
</div>
```

Email

info@mimail.com

## Campo disabled

En formularios donde cierta información se carga de forma dinámica o interactúa de forma especial con el usuario, podemos necesitar elementos que, en ciertos contextos, se encuentren

**deshabilitados**. Por ejemplo, un campo que no debe completar el usuario, dado que no corresponde quizás a su rol en una empresa, o trabajo a realizar. En estos casos, en Bootstrap, muestra tales elementos de forma diferente si es que se utilice o no el atributo **placeholder** o **value**.

**Nota:** todas las características pueden modificarse luego desde nuestra propia hoja de estilos, sobrescribiendo el valor original de la propiedad que existe en Bootstrap.



**Ejemplo** (código  
y resultado):

```
<div>
  <label class="form-label">Gente a cargo</label>
  <input value="0" disabled type="number"
    class="form-control">
</div>
<div>
  <label class="form-label">Código especial</label>
  <input placeholder="No corresponde" disabled readonly
    type="text" class="form-control">
</div>
```

Gente a cargo

0

Código especial

No corresponde

## Formularios & Columnas

Siguiendo la lógica de trabajo con grillas en Bootstrap, obviamente podemos integrar lo que ya hemos utilizado al uso de formularios.

```
<main class="container mb-5">
  <div class="row">
    <label class="form-label col-md-2
col-form-label">Nombre</label>
    <div class="col-md-10">
      <input type="text" name="nombre"
class="form-control">
    </div>
  </div>
</main>
```

Incluso contamos con **clases especiales**, como vemos en la imagen de abajo a la izquierda, para **encolumnar formularios** de forma simple.

La clase **col-form-label** agrega utilizada de forma conjunta con las clases **row** y **col** un **margin-bottom**, como se puede ver en el inspector de elementos:

```
.form-label {
  margin-bottom: .5rem;
}
```

El **resultado** del trabajo anterior será el siguiente:

Podemos combinar de diversas maneras nuestros elementos, para lograr resultados distintos según el diseño a replicar.

#### Vista desktop

Nombre

#### Vista mobile

Nombre

# Filas, columnas y formularios

Veamos un ejemplo más sencillo con columnas y filas:

```
<form class="row">
  <div class="col-md-6">
    <label for="email" class="form-label">Email</label>
    <input type="email" class="form-control" id="email">
  </div>
  <div class="col-md-6">
    <label for="password" class="form-label">Password</label>
    <input type="password" class="form-control" id="password">
  </div>
  <div class="col-12">
    <label for="direccion" class="form-label">Address</label>
    <input type="text" class="form-control" id="direccion" placeholder="Av Rivadavia
4567">
  </div> </form>
```

El **resultado** del paso anterior será el siguiente:

Email

Password

Address

Av Rivadavia 4567





## Sass y variables

Recordemos que siempre todos **los elementos trabajados en Bootstrap** están inicialmente **declarados en variables de Sass**; que nos permiten luego, modificar estos elementos desde nuestro propio SCSS o nuestro CSS, de forma indistinta.



## Variables

**\$input-\*** se comparten en la mayoría de nuestros controles de formulario (y no en los botones):

<code>\$input-padding-y;</code>	<code>\$input-btn-padding-y;</code>
<code>\$input-padding-x;</code>	<code>\$input-btn-padding-x;</code>
<code>\$input-font-family;</code>	<code>\$input-btn-font-family;</code>
<code>\$input-font-size;</code>	<code>\$input-btn-font-size;</code>
<code>\$input-font-weight;</code>	<code>\$font-weight-base;</code>
<code>\$input-line-height;</code>	<code>\$input-btn-line-height;</code>
<code>\$input-padding-y-sm;</code>	<code>\$input-btn-padding-y-sm;</code>
<code>\$input-padding-x-sm;</code>	<code>\$input-btn-padding-x-sm;</code>
<code>\$input-font-size-sm;</code>	<code>\$input-btn-font-size-sm;</code>
<code>\$input-padding-y-lg;</code>	<code>\$input-btn-padding-y-lg;</code>
<code>\$input-padding-x-lg;</code>	<code>\$input-btn-padding-x-lg;</code>
<code>\$input-font-size-lg;</code>	<code>\$input-btn-font-size-lg;</code>

## Datalist en Bootstrap

Las **listas o menús** de formularios también trabajan con las **mismas clases que los campos**, y se obtienen estilos rápidamente. Veamos un ejemplo de una lista:



```
<form action="codigo.php">
  <label for="listaPaíses" class="form-label">Países</label>
  <input class="form-control" list="opciones" id="listaPaíses" placeholder="Elegir...">
  <datalist id="opciones">
    <option value="Argentina">
    <option value="Brasil">
    <option value="Paraguay">
    <option value="Uruguay">
    <option value="Chile">
  </datalist>
</form>
```

El **resultado** del paso anterior será el siguiente:

Países

Elegir...

Argentina

Brasil

Paraguay

Uruguay

Chile

## Select, ¿cómo trabajarlo?

Los menús que acostumbramos a ver usualmente son *select*, y no es la excepción en Bootstrap.

Podemos trabajarlos de forma simple con la clase **form-select** como vemos en la imagen:

```
<main class="container mb-5">
  <form action="codigo.php">
    <label class="form-label">Países</label>
    <select class="form-select" name="países">
      <option selected>Elegir un País</option>
      <option value="argentina">Argentina</option>
      <option value="brasil">Brasil</option>
      <option value="chile">Chile</option>
    </select>
  </form>
</main>
```

El **resultado** del paso anterior, que hemos acompañado también del trabajo con **label1**, puede verse en el navegador de la siguiente forma:

Países

Elegir un País



## Menús

Se puede, así como en los casos anteriores, modificar el **tamaño de los menús** con **sm** o **lg**. Veamos un ejemplo a continuación, con el uso de **lg**:

```
<main class="container mb-5">
  <form action="codigo.php">
    <label class="form-label">Países</label>
    <select class="form-select form-select-lg" name="paises">
      <option selected>Elegir un Pais</option>
      <option value="argentina">Argentina</option>
      <option value="brasil">Brasil</option>
      <option value="chile">Chile</option>
    </select>
  </form>
</main>
```



Resultado:

Países

Elegir un País





## Menú múltiple

También podemos colocar un menú múltiple para que permita **opciones múltiples**.

```
<main class="container mb-5">
  <form action="codigo.php">
    <label class="form-label">Países</label>
    <select class="form-select form-select-sm multiple" name="países">
      <option selected>Elegir un País</option>
      <option value="argentina">Argentina</option>
      <option value="brasil">Brasil</option>
      <option value="chile">Chile</option>
    </select>
  </form>
```

**Nota:** en el ejemplo, hemos incluido **form-select-sm** para poder mostrar otro tamaño de menú posible.

Resultado:

Países



Elegir un País

Elegir un País

Argentina

Brasil

Chile

## Radios

Los radios también pueden trabajarse desde Bootstrap de forma muy simple.

Por ejemplo, utilizando las clases denominadas **form-check-input** y **form-check-label**. Es importante a su vez, que se encuentren anidados en un contenedor con la clase **form-check** para lograr el resultado de la imagen de la derecha.

Se utilizan para opciones **exclusivas**, es decir, donde el usuario debe elegir **sí o sí una opción entre varias**.

☒ Casado  
☐ Soltero

Para poder generar el resultado anterior, vamos a trabajar de la siguiente manera:

```
<form action="codigo.php">
  <div class="form-check">
    <input name="estado" class="form-check-input" type="radio" value="" id="casado">
    <label class="form-check-label" for="casado">
      Casado
    </label>
  </div>
  <div class="form-check">
    <input name="estado" class="form-check-input" type="radio" value="" id="soltero" >
    <label class="form-check-label" for="soltero">
      Soltero
    </label>
  </div>
</form>
```

## Checkboxes

Los *checkboxes* trabajan de manera muy similar, siempre comprendiendo que **las opciones pueden en este caso, ser inclusivas**.

(Recordemos que para opciones *exclusivas*, debemos de utilizar los radios).

Veamos el ejemplo de la próxima pantalla, donde desde nuestro editor, se verán las **similitudes con los radios, en cuanto a utilización de clases**.



A rectangular box containing two checked checkboxes. The first checkbox is labeled 'Correr' and the second is labeled 'Nadar'. Both checkboxes are blue with a white checkmark.



A simple rectangular box with a thin blue border, representing an empty input field.

```
<form action="codigo.php">
  <div class="form-check">
    <input name="eleccion" class="form-check-input" type="checkbox" value="" id="correr">
    <label class="form-check-label" for="correr">
      Correr
    </label>
  </div>
  <div class="form-check">
    <input name="eleccion" class="form-check-input" type="checkbox" value="" id="nadar" >
    <label class="form-check-label" for="nadar">
      Nadar
    </label>
  </div>
</form>
```

## Switch

Si bien podemos desde CSS utilizando un poco de JavaScript generar nuestros propios *switches*, desde Bootstrap es mucho más fácil.

Simplemente, al **div** que anida al checkbox y a su correspondiente **label** debemos agregar **form-switch**, como vemos en la imagen:



```
<div class="form-check form-switch">  
  <input name="eleccion" class="form-check-input" type="checkbox" value="" id="correr">  
  <label class="form-check-label" for="correr">  
Correr  
  </label>  
</div>
```

## Switch en línea

Se pueden sumar a las clases anteriores el trabajo en línea, es decir, **un checkbox al lado del otro**. Por ejemplo:

```
<div class="form-check form-switch form-check-inline">  
  <input name="eleccion" class="form-check-input" type="checkbox" value="" id="correr">  
  <label class="form-check-label" for="correr">  
Correr  
  </label>  
</div>
```

Resultado:

☐

Correr

☐

Nadar



## Aclaración: Switch en reversa

Si al paso anterior le agregamos la clase **.form-check-reverse** veremos que en contextos donde deseamos que el usuario **no seleccione**, es una opción ideal de uso.

Sin embargo, como vemos, esta opción no funciona aún correctamente en los navegadores, dado que el resultado termina siendo el mismo que en los otros casos.

```
<main class="container mb-5">
  <form action="codigo.php">
    <div class="form-check form-switch form-check-inline
      form-check-reverse">
      <input class="form-check-input" type="checkbox" >
      <label class="form-check-label">No estoy de Acuerdo</label>
    </div>
  </form>
</main>
```

No estoy de Acuerdo



**¡Sigamos  
trabajando!**