educación IT
CODE YOUR FUTURE

Bootcamp Java Developer

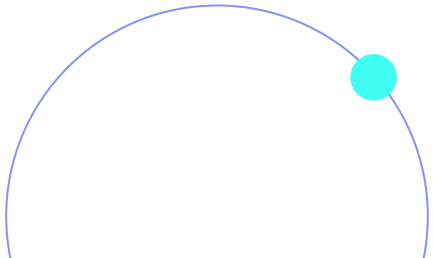
Fase 2 - Java Web Developer

Módulo 16

Validación JS

Validación de formularios

Para comenzar a trabajar, es importante entender que en este caso, trabajamos con **JavaScript** incluso en un Framework de CSS. Por esta razón, uno de los pasos más importantes, es **vincular no sólo Bootstrap CSS sino también Bootstrap JS**.



Dos JavaScript unificados

Si bien en versiones anteriores, se trabajaban por separado y existen en sí dos JavaScript a vincular (uno relacionado con los *popovers*, es decir, cualquier elemento que se solape y surja de nuestra interfaz; y otro relacionado con temas generales de JavaScript), en este caso vamos a optar por la **versión unificada**, como se muestra en el bloque de código de la próxima pantalla.

```
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Validación</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
    rel="stylesheet"
        integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65"
        crossorigin="anonymous">


    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.
    js"
        integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4"
        crossorigin="anonymous"></script>
</head>
```

Luego de vincular ambos elementos, podemos generar un *formulario tipo* teniendo en cuenta dos elementos básicos en su construcción.

Validación original de HTML5

La validación que se efectúa al utilizar (como la mayoría de las validaciones actuales) la **validación original de HTML5**. A diferencia de lo que sucedía hace un tiempo, **no es que tomamos desde cero todo el trabajo con Javascript**, sino que sólo lo utilizamos para hacer mucho más **dinámica la validación original**.

Manteniendo esa premisa en mente, vamos a, por ejemplo, generar un formulario con un atributo de requerimiento obligatorio, como solemos hacer. Veamos la próxima pantalla.

Ejemplo: formulario con atributo de requerimiento obligatorio

Vemos en la imagen de abajo, un label con un **`for`** para mejorar la **accesibilidad**. Un campo que forma parte de un **sistema de grillas** (**`col-md-4`**) y la clase que utilizamos de manera usual para trabajar con formularios, **`form-control`**. Nada adicional a lo ya visto, salvo el haberle agregado, en este caso, el atributo **`required`**, ya integrado en los cursos iniciales de HTML.

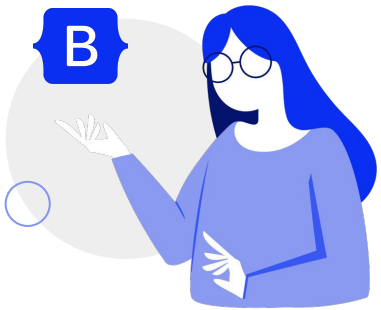
```
<div class="col-md-4">
  <label for="nombreEtiqueta" class="form-label" >  Nombre</label>
  <input type="text" class="form-control" id="nombreEtiqueta"  required>
```

Si es importante, obviamente incorporar este campo dentro de un elemento **`form`**, donde a su vez contemos con ciertos elementos importantes.

```
<form class="row needs-validation" novalidate>
```

El formulario, utiliza una **clase asociada a las grillas** (**`row`**) para luego incorporar una clase específica de la validación **`needs-validation`** y un atributo conocido pero que es importante recordar su uso.

`novalidate` asegura que **el navegador no efectúe su usual validación automática, para dejar paso** en ese caso, (si bien utilizando la validación de HTML5 original) **a nuestros propios mensajes personalizados**.



Código de jQuery

Para ir un paso más allá, es importante mencionar que Bootstrap utiliza una **librería de base de JS, llamada *jQuery***. Al momento de vincular (primer slide de este manual) Bootstrap JS hemos entre otras, incluido esa librería.

En el caso de jQuery, **trabajamos con JavaScript pero a través de selectores muy similares** (casi iguales) **a CSS**. Por eso su facilidad para aprender esa librería, dado su similitud en muchos aspectos y fácil manipulación.

Procedimiento para comenzar la validación

Para que el código de la diapositiva 7 pueda empezar a funcionar, vamos a trabajar con un **código que inicializa el proceso de validación**. Siempre recordando también agregar un ***botón de envío***, para que al momento de presionarlo, se comience el proceso de validación.

```
<div class="col-12">
  <button class="btn btn-warning" type="submit">Enviar</button>
</div>
```

En base a esto, incorporaremos el **código de jQuery** de la siguiente pantalla. Así, generamos el hecho de que para cada elemento de formulario (por eso el **`array`**) que no sea validado, se agregará la clase **`was-validated`**, que permitirá el efecto que veremos luego.

```
<script>

// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
  'use strict'

  // Fetch all the forms we want to apply custom Bootstrap validation styles to
  var forms = document.querySelectorAll('.needs-validation')

  // Loop over them and prevent submission
  Array.prototype.slice.call(forms)
    .forEach(function (form) {
      form.addEventListener('submit', function (event) {
        if (!form.checkValidity()) {
          event.preventDefault()
          event.stopPropagation()
        }
        form.classList.add('was-validated')
      }, false)
    })
})()</script>
```

Este código pueden copiarlo y pegarlo. O tomarlo directamente de la página oficial de Bootstrap.

En base a sólo lo realizado anteriormente, el resultado será el siguiente:

Podemos modificar el **css**, si es que no nos gusta, utilizando las pseudo clases:

```
:invalid { border-color: orange;}
:valid { border-color: purple !important;}
```

El **`!important`** se agregó para pisar un estilo más específico.

Resultados:

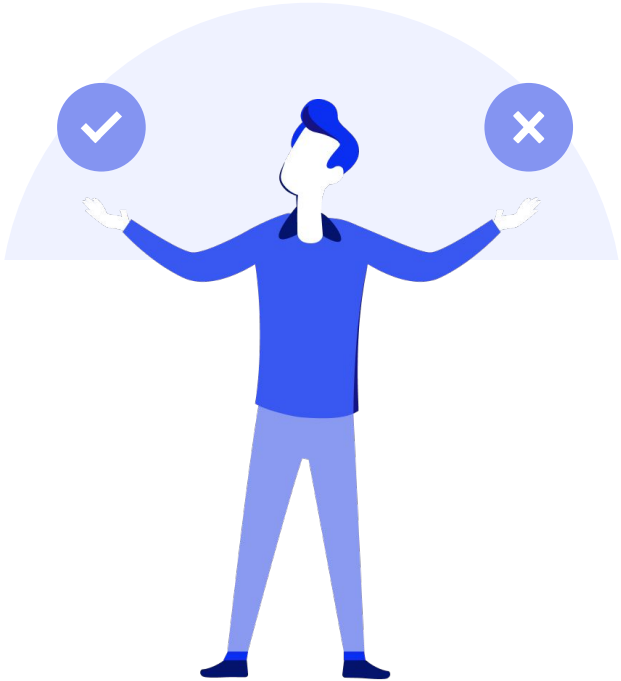
En el caso de **válido**:

En el caso de **inválido**:

Validación: mensajes personalizados

El trabajo con validación puede pasar al siguiente nivel, si sumamos mensajes personalizados. Veremos las clases siguientes, para tal fin:

- **`valid-feedback`**.
- **`invalid-feedback`**.



.valid-feedback: mensaje positivo

Al utilizar esta clase, desde la estructura del HTML se generaría un **mensaje positivo**, en caso de que el **campo haya sido completado de forma correcta**. Veamos un ejemplo con su resolución, a continuación:

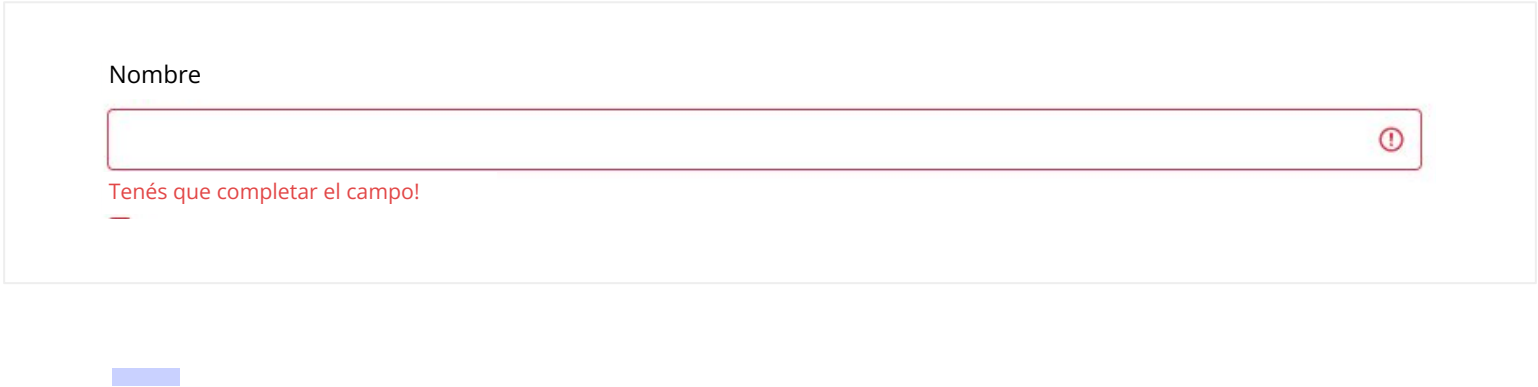
```
<div class="col-md-12">
  <label for="nombreEtiqueta" class="form-label" > Nombre</label>
  <input type="text" class="form-control" id="nombreEtiqueta"  required>
  <div class="valid-feedback">
   Muy bien!
  </div>
</div>
```

.invalid-feedback: mensaje negativo

El trabajo con una **respuesta errada o no respuesta** (es decir, por ejemplo, en un `required` el campo no se completó), con esta clase. Veamos un ejemplo:

```
<form class="row needs-validation" novalidate>
    <div class="col-md-12">
      <label for="nombreEtiqueta" class="form-label" > Nombre</label>
      <input type="text" class="form-control" id="nombreEtiqueta"  required>
      <div class="valid-feedback">
       Muy bien!
      </div>
      <div class="invalid-feedback">
       Tenés que completar el campo!
       </div>
    </div>
```

Resolución:



Otro ejemplo

Veamos un ejemplo con un ***check*** y **un solo mensaje**:

```
<div class="col-md-12">

  <div class="form-check">
    <input class="form-check-input" type="checkbox" value="" id="invalidCheck"
    required>
    <label class="form-check-label" for="invalidCheck">
      Estoy de acuerdo con los términos y condiciones
    </label>
    <div class="invalid-feedback">
    Tenés que estar de acuerdo con los términos y condiciones para avanz
    </div>
  </div>
</div>
```

educación IT
CODE YOUR FUTURE

¡Sigamos trabajando!