

En la lección anterior, presentamos los formularios. Pero no lo vimos todo. Ahora, vamos a examinar cómo solicitar al navegador que compruebe los datos introducidos por el usuario, para comprobar si cumplen determinadas reglas, antes de su envío al servidor.

La lección comienza introduciendo el concepto de validación y regla de validación. También presentamos cómo configurar el formulario para que el navegador no realice la validación predeterminada. A continuación, presentamos varias reglas de validación: la que obliga al usuario a introducir un dato, la que obliga a que el dato cumpla un determinado patrón, la que obliga a que el dato tenga un determinado número de caracteres y, finalmente, la que obliga a que el dato se encuentre en un determinado rango.

Al finalizar la lección, el estudiante sabrá:

- Cómo configurar el formulario para que el navegador no realice la validación.
- Cómo indicar que el usuario debe introducir un dato en un determinado elemento de entrada.
- Cómo indicar que el dato introducido por el usuario se encuentra en un rango.
- Cómo indicar que el dato introducido por el usuario tiene un determinado número de caracteres.
- Cómo indicar que el dato introducido por el usuario cumple un determinado patrón.

Introducción

La **validación** (*validation*) es la operación mediante la cual se comprueba que los datos introducidos por los usuarios son válidos, antes de su remisión al servidor web. Esta validación se puede hacer básicamente de dos formas: mediante **JavaScript** o delegándolo al navegador. En esta lección, vamos a ver cómo hacerlo mediante el navegador.

Para que el navegador pueda realizar la validación, es necesario que definamos **reglas de validación** (*validation rules*), las restricciones que deben cumplir los datos introducidos por el usuario. Para ello, debemos definir los elementos de entrada usando atributos especiales.

Cuando el usuario hace clic en el botón de envío, el proceso que se sigue es el siguiente:

1. Si no se ha deshabilitado la comprobación del navegador, se realiza ésta.
2. Se genera el evento **Submit**.
3. Se envía los datos al servidor, si el controlador del evento **Submit** no ha cancelado la acción predeterminada.

Atributo novalidate

De manera predeterminada, el navegador valida los datos cuando el usuario hace clic en el botón de envío. Si por cualquier razón no deseamos que lo haga, hay que definir el atributo booleano **novalidate** en el formulario:

```
<form novalidate ...>  
...  
</form>
```

De esta manera, le estamos indicando al navegador que los datos introducidos no los debe validar él. En este caso, generalmente la validación se hace mediante **JavaScript**. Más concretamente, mediante el controlador del evento **Submit**.

Valor obligatorio

Algunos datos pueden ser obligatorios y otros opcionales. De manera predeterminada, el usuario no debe proporcionar datos para ninguno de los elementos de entrada. Pero si definimos el atributo booleano **required**, el navegador pedirá al usuario, cuando esté realizando la validación, que inserte un valor.

A continuación, se muestra un ejemplo de formulario con los dos elementos de entrada como obligatorios:

```
<form>
  <input type="text" name="username" required placeholder="Nombre de usuario">
  <input type="password" name="password" required placeholder="Contraseña">
  <input type="submit" value="Entrar">
</form>
```

Un formulario web con dos campos de entrada: "Nombre de usuario" y "Contraseña", ambos con el atributo required. A la derecha hay un botón "Entrar".

Cuando el navegador realice la validación, si el usuario no ha introducido nada, mostrará un mensaje como el siguiente:

El formulario muestra el valor "elvis" en el campo "Nombre de usuario". El campo "Contraseña" está resaltado con un borde azul. Debajo de él aparece un mensaje de error: "Please fill out this field." con un icono de advertencia.

Patrón de valor

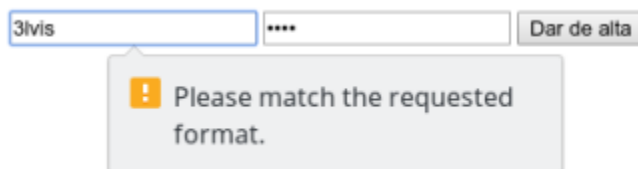
Cuando el usuario debe introducir un dato mediante un cuadro de texto, en cualquiera de sus variantes, **text**, **number**, **email**, etc., podemos definir una regla que especifique el formato que debe seguir el valor. Esto se conoce formalmente como **patrón** (*pattern*) y se especifica mediante el atributo **pattern**, cuyo valor debe ser una expresión regular **JavaScript**.

Veamos un ejemplo:

```
<form>
  <input type="text"
    name="username"
    required
    pattern="[a-zA-Z].*"
    placeholder="Nombre de usuario">
  <input type="password"
    name="password"
    required
    pattern="[a-zA-Z0-9]{8,}"
    placeholder="Contraseña">
  <input type="submit" value="Dar de alta">
</form>
```

En el elemento **username**, el usuario debe introducir un nombre de usuario que comience por una letra, seguido de tantos caracteres como quiera. Pero necesariamente, debe comenzar por letra. En el caso de la contraseña, debe contener como mínimo ocho letras y/o dígitos.

Si el valor introducido por el usuario no cumple el patrón indicado, el navegador mostrará un mensaje que lo indique en el momento de la validación, por ejemplo, como el siguiente:

El formulario muestra el valor "3lvis" en el campo "Nombre de usuario". El campo "Contraseña" está oculto con cuatro puntos. Debajo de él aparece un mensaje de error: "Please match the requested format." con un icono de advertencia.

Longitudes mínima y máxima

Si es necesario que el usuario introduzca un determinado número de caracteres mínimo y/o máximo, podemos utilizar los atributos **minlength** y **maxlength**, tal como muestra el siguiente ejemplo:

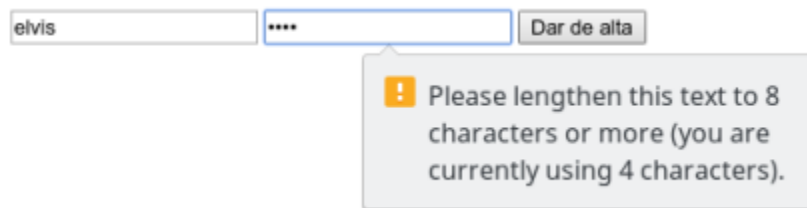
```
<form>
```

```

<input type="text" name="username" required placeholder="Nombre de usuario">
<input type="password" name="password" required minlength="8" placeholder="Contraseña">
<input type="submit" value="Dar de alta">
</form>

```

Si el valor no cumple lo indicado, el navegador mostrará un mensaje como el siguiente:



The screenshot shows a web form with three input fields: a text field containing 'elvis', a password field with four asterisks, and a submit button labeled 'Dar de alta'. The password field is highlighted with a blue border, and a gray tooltip message is displayed below it. The message contains an orange exclamation mark icon and the text: 'Please lengthen this text to 8 characters or more (you are currently using 4 characters).'

Valores mínimo y máximo

Cuando indicamos como tipo de entrada de datos **number** o **date**, podemos indicar los valores mínimo y máximo que puede introducir el usuario. Para este fin, se utilizan los atributos **min** y **max**. Ejemplo:

```

<form>
<input type="text" name="username" required placeholder="Nombre de usuario">
<input type="password" name="password" required minlength="8" placeholder="Contraseña">
<input type="number" name="birthYear" required max="1998" placeholder="Año de nacimiento">
<input type="submit" value="Dar de alta">
</form>

```

El navegador mostrará un mensaje, similar al siguiente, si no se cumple la regla de validación:



The screenshot shows a web form with three input fields: a text field containing 'elvis', a password field with eight asterisks, and a number field containing '2000'. The number field is highlighted with a blue border, and a gray tooltip message is displayed below it. The message contains an orange exclamation mark icon and the text: 'Value must be less than or equal to 1998.'