

Lenguajes de marcas y sistemas de gestión de la información



UT02 – HTML

9 – Agrupación de campos – Otros tipos de campos – Validación

Agrupación de controles

A veces necesitamos agrupar varios controles de un formulario, para organizar o "dividir" un formulario con gran número de controles, o, aunque sea un formulario pequeño, para agrupar controles relacionados.

Si pensamos en accesibilidad, es de obligado cumplimiento cuando tenemos conjuntos de `input type="radio"` o de `input type="checkbox"`.

A los grupos de campos se les añade una especie de etiqueta, que define todo el grupo de campo.

Para agrupar se utiliza el elemento `"fieldset"`.

Para etiquetar el `fieldset` se utiliza `"legend"`. Debe ser el primer elemento dentro de un `fieldset`.

Agrupación de controles – Ejemplos

Formulario de registro, en el que se pide al usuario un nombre de usuario, una contraseña, y repetir la contraseña.

Usuario:

Contraseña:

Repetir Contraseña:

¿Cómo nos ha conocido?

☐ Prensa ☐ Internet ☐ Amigos

Los fieldset son elementos "de bloque", por lo que siempre ocupan todo el ancho disponible.

Agrupación de controles – Ejemplos

El código para el anterior formulario sería:

```
<p><label>Usuario: <input type="text" name="usr"></label></p>
<p><label>Contraseña: <input type="password" name="pwd"></label></p>
<p><label>Repetir Contraseña: <input type="password"
name="pwd2"></label></p>
<fieldset>
  <legend>¿Cómo nos ha conocido?</legend>
  <label><input type="radio" name="cto" value="P"> Prensa</label>
  <label><input type="radio" name="cto" value="I"> Internet</label>
  <label><input type="radio" name="cto" value="A"> Amigos</label>
</fieldset>
<button type="submit">Registrarme</button>
```

Anidando grupos de controles

Los fieldsets pueden "anidarse". Puede haber fieldsets dentro de otro fieldset.

Ejemplo: parte de un formulario que pide detalles del cliente y del equipamiento de un coche, para poder dar un presupuesto:

Datos del comprador

Nombre:

Apellidos:

Color y equipamiento

Color

☐ Blanco ☐ Gris ☐ Rojo

Equipamiento

☐ Llantas de aleación ☐ Climatizador bizona ☐ Faros Xenon

Solicitar presupuesto

Anidando grupos de controles – Ejemplos

```
<fieldset>
  <legend>Datos del comprador</legend>
  <p><label>Nombre: <input type="text" name="nombre"></label></p>
  <p><label>Apellido: <input type="text" name="apellido"></label></p>
</fieldset>
<fieldset>
  <legend>Color y equipamiento</legend>
  <fieldset>
    <legend>Color</legend>
    <label><input type="radio" name="color" value="B"> Blanco</label>
    <label><input type="radio" name="color" value="G"> Gris</label>
    <label><input type="radio" name="color" value="R"> Rojo</label>
  </fieldset>
  <fieldset>
    <legend>Equipamiento</legend>
    <label><input type="checkbox" name="equipo" value="LL">
```

Fieldsets con radio buttons y checkboxes

Cuando usamos radiobuttons y checkboxes podemos encontrarnos:

- Un checkbox aislado: no es necesario rodearlo con fieldset.
- Un radio button aislado: es imposible. No se deben usar radio buttons aislados. Están pensados para su uso en grupo, de forma excluyente.
- Un conjunto de checkboxes: de deben rodear con un fieldset, y añadir en el legend la descripción de para que se usen los checkboxes.
- Un conjunto de radio buttons: de deben rodear con un fieldset, y añadir en el legend la descripción de para que se usen los radio buttons.

Para mejorar su accesibilidad, cuando tenemos un conjunto de radio buttons o de checkboxes, se recomienda, además, usar una lista no ordenada (ul).

Fieldsets con listas de controles – Ejemplo

En el ejemplo anterior, si añadimos una lista para los colores, tendríamos:

```
<fieldset>
  <legend>Color</legend>
  <ul>
    <li><label><input type="radio"
      name="color" value="B">Blanco</label></li>
    <li><label><input type="radio"
      name="color" value="G"> Gris</label></li>
    <li><label><input type="radio"
      name="color" value="R"> Rojo</label></li>
  </ul>
</fieldset>
```


Uso de "section" en formularios grandes

Además de utilizar fieldset, cuando se tiene un formulario muy grande, dividido en muchas partes, se puede utilizar "section", para dividir el formulario en secciones.

Otros tipos de campos HTML / HTML 5

Los campos más básicos, que vienen existiendo desde versiones anteriores de HTML, los más utilizados y que ya hemos visto son:

- Input, tipos "text", "password", "checkbox", "radio".
- Textarea
- Select, en sus tres variantes (selector, lista, lista con selección múltiple)

Vamos a ver rápidamente otros tipos de campo:

- Email (input type="email")

Otros campos HTML 5 – Email

Se consigue con `input type="email"`.

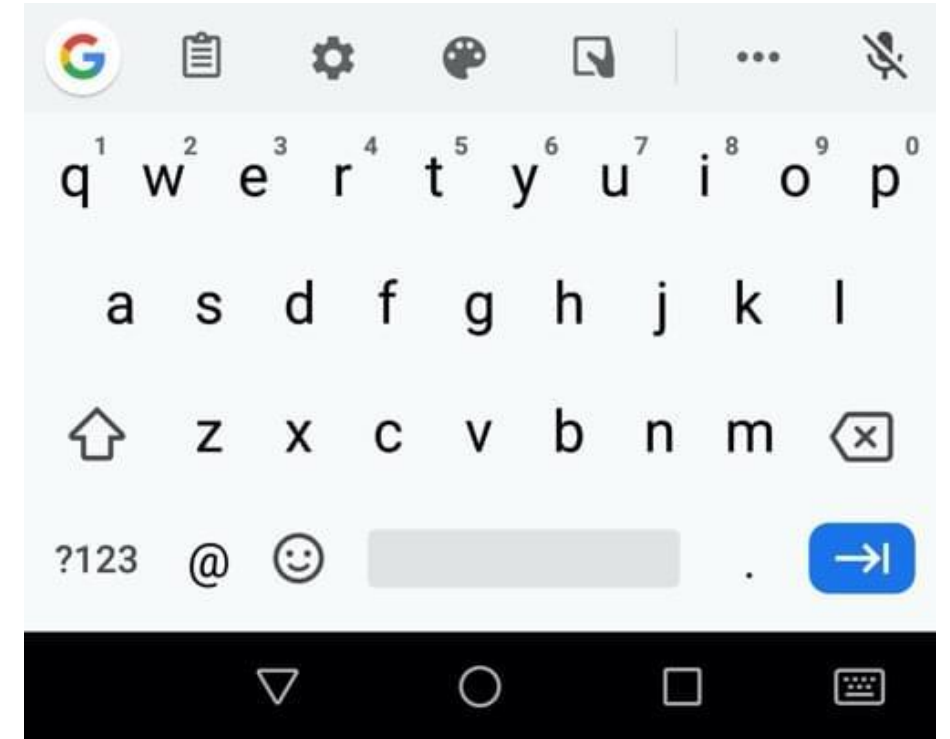
Permite pedir al usuario un correo electrónico.

Añade automáticamente validación para que el valor introducido tenga la forma de una dirección de correo electrónico.

La validación es limitada, porque el estándar para direcciones de correo permite direcciones como, por ejemplo, `juan@casa`.

Para validación más exhaustiva hay que recurrir a otras técnicas.

Algunos dispositivos, sobre todo móviles, mostrarán un teclado específico, en el que la `@` está disponible directamente.



Otros campos HTML 5 – Teléfono

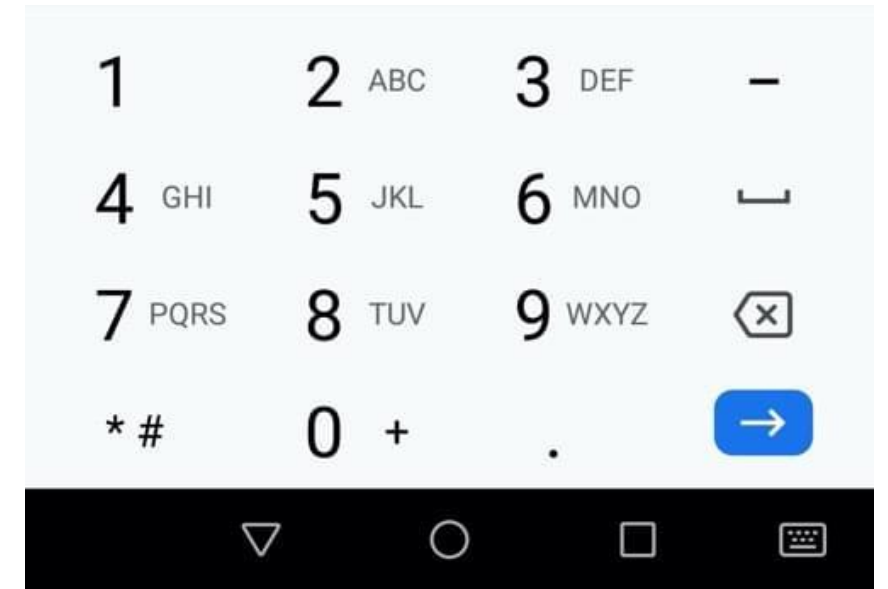
Se consigue con `input type="tel"`.

Permite pedir al usuario un teléfono.

Este control no añade validación, porque en el mundo hay gran variedad de formatos de número de teléfono, así que no tiene mucho sentido forzar un formato por defecto.

Otra vez, si deseamos una validación más exhaustiva tendremos que recurrir a otras técnicas, que veremos más adelante.

Algunos dispositivos mostrarán un teclado específico, similar al de los antiguos teléfonos con teclado, en el que sólo habrá números y los caracteres #, *, etc.



Otros campos HTML 5 – Número

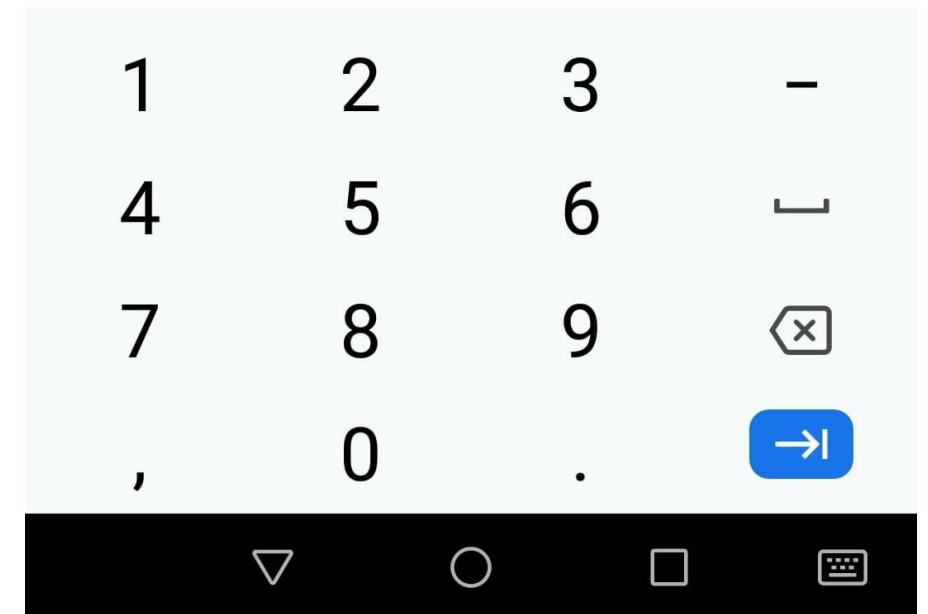
Se consigue con `input type="number"`.

Permite pedir al usuario un número sin decimales.

El navegador puede mostrar en el campo unas flechas para aumentar o reducir el valor. Por defecto de uno en uno, pero se puede cambiar el valor del salto con el atributo "step".

Algunos dispositivos mostrarán un teclado específico, sólo con números y símbolos relacionados..

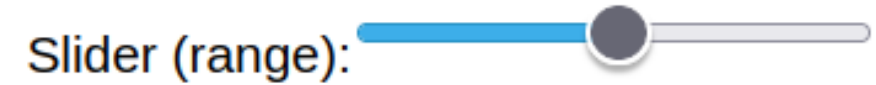
Número:



Otros campos HTML 5 – Range (slider)

Se consigue con `input type="range"`.

Permite elegir un valor numérico entre un mínimo y un máximo.



Por defecto permite elegir un entero entre 0 y 100 (ambos incluidos), pero puede cambiarse con los atributos:

- `min`: mínimo valor que se puede elegir.
- `max`: máximo valor que se puede elegir
- `step`: "salto" entre valores. Si es un número con decimales (ej. 0,1) se podrán especificar valores decimales.

También puede modificarse su orientación, para hacerlo vertical.

Otros campos HTML 5 – Fecha y hora


Hay varios tipos de controles para fecha, hora, y otras medidas de tiempo. Se definen con un elemento input, y dependiendo del atributo type tendrán distinto comportamiento:


- `datetime-local`: fecha y hora sin información de zona horaria
- `date`: sólo fecha
- `time`: sólo hora
- `month`: mes y año. No en Firefox, sí en Chromium y derivados.
- `week`: semana del año. Igual que month, no en Firefox.

Se puede restringir el rango de fechas con `min` y `max`, y se puede usar `step` para limitar los valores a elegir.

Otros campos HTML 5 – Fecha y hora

Campos de fecha / hora

Fecha y hora local: 

Fecha: 

Hora:

Mes (no funciona en Firefox):

Semana (no funciona en Firefox):

Campos de fecha / hora

Fecha y hora local: 

Fecha: 

Hora: 

Mes (no funciona en Firefox): 

Semana (no funciona en Firefox): 

Otros campos HTML 5 – Selector de color

Se consigue con `input type="color"`.

Permite pedir al usuario un número sin decimales.

Número:

El navegador puede mostrar en el campo unas flechas para aumentar o reducir el valor. Por defecto de uno en uno, pero se puede cambiar el valor del salto con el atributo "step".

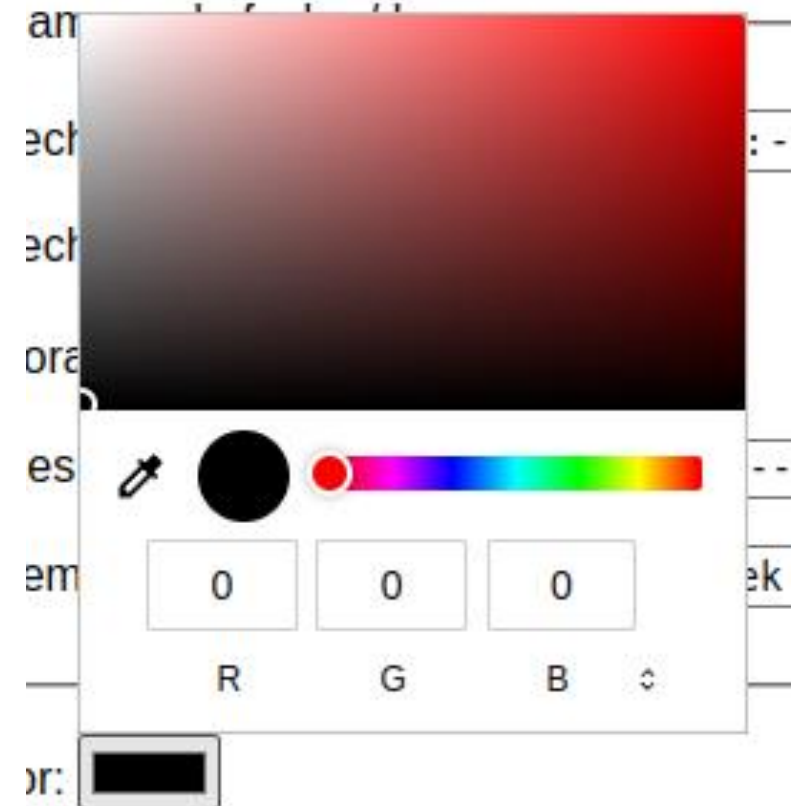
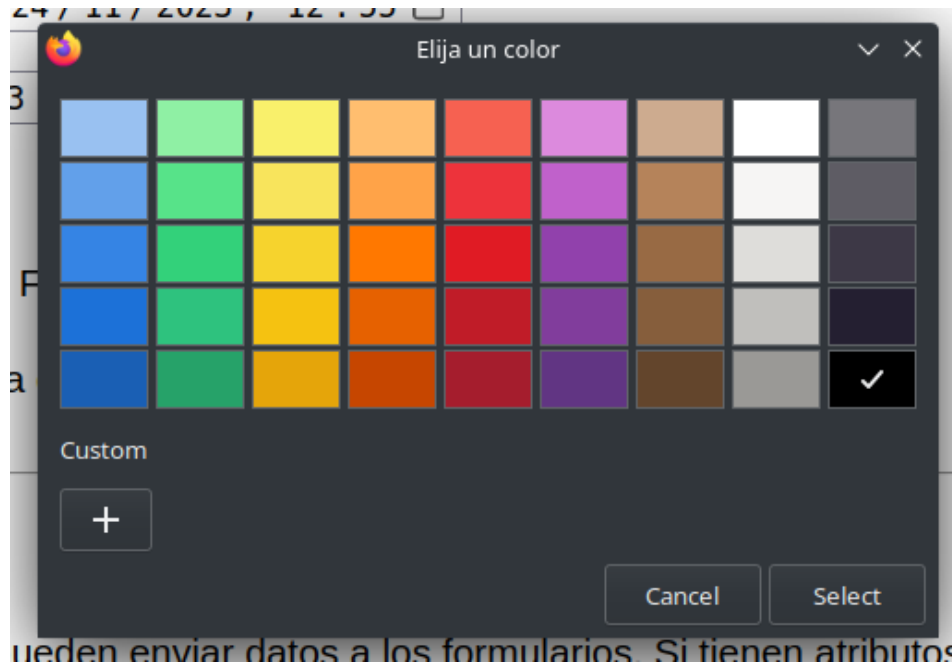
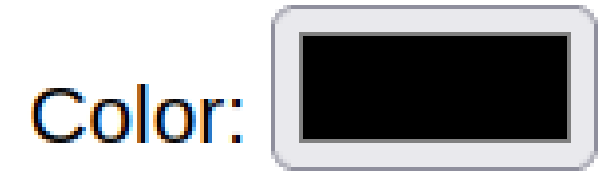
Algunos dispositivos mostrarán un teclado específico, sólo con números y símbolos relacionados..

Otros campos HTML 5 – Número

Se consigue con `input type="color"`.

Permite elegir un color.

El aspecto del selector de color variará en función del navegador y/o del sistema operativo del cliente.



Algunos atributos útiles

Los siguientes atributos se pueden usar con la mayoría de los controles, pero algunos sólo se pueden utilizar con ciertos controles, como placeholder, que sólo tiene sentido en controles de tipo texto.

- autofocus: al cargar la página, el control con este atributo será el que tenga el foco del teclado. Sólo un control de toda la página debería tener este atributo.
- disabled: el usuario no puede modificar el valor, y además este valor nunca se envía al enviar el formulario.
- readonly: el usuario no puede modificar el valor, pero este sí se envía cuando se envía el formulario.

Algunos atributos útiles

- placeholder: muestra un texto que aparece dentro de la caja de texto, y desaparece cuando empezamos a escribir en ella. Importante: no es un sustituto de label, es más bien un complemento.
- spellcheck: permite habilitar / deshabilitar la corrección ortográfica si el navegador la tiene y admite el atributo.
- multiple: ya hemos visto que sirve para que en un select en modo lista podamos elegir varios valores, pero también sirve, por ejemplo, en los controles de tipo email, para introducir varios valores separados por comas, o para admitir varios ficheros en un input type="file".
- size: en los select indicaba el número de elementos visibles en modo lista. En los campos de texto indica el ancho de la caja. En general, es preferible usar CSS en lugar de utilizar este atributo.

Algunos atributos útiles

- rows y cols. En los controles textarea, indican el ancho y alto del control. Otra vez, siempre que sea posible, es mejor usar CSS.
- wrap. En textarea determina como gestionar saltos de línea:
 - soft (valor por defecto): el texto se parte en líneas para que quepa en el textarea, pero el texto que se envía al servidor no está partido.
 - hard: el texto se parte en líneas, y se envía partido al servidor.
 - off: el texto no se parte, y se envía sin partir.
- resize. En textarea indica si se puede cambiar el tamaño del control.
- accept. Permite especificar el tipo de contenido que admiten los campos `input type="file"`. Admite varios tipos separados por comas. Pueden ser la extensión (ej. .xml) o el content-type (ej. text/xml).

Validación HTML 5

HTML 5 permite validar un formulario antes de enviarlo al servidor, para intenta evitar los errores de usuario más habituales.

Hay una serie de atributos específicos relacionados con la validación, como, entre otros: required, minlength, maxlength, min, max, pattern.

Cuando veamos CSS, veremos cómo dar estilos a campos válidos con las pseudoclasas :required para campos obligatorios, :valid para campos válidos e :invalid para campos no válidos.


Validación – Campos obligatorios

Para hacer un input, select, o textarea obligatorio se añade el atributo "required". Al enviar el formulario, si no se ha completado el campo, se mostrará un mensaje indicando que se debe completar

```
<label>Nombre: <input type="text" name="nombre" required></label>
```

```
<label>Apellidos: <input type="text" name="apellidos" required></label>
```


Nombre:

Apellido  Completa este campo

País de residencia:

Nombre:

Apellidos:

País de res  Completa este campo

Para los campos select, radio button y checkbox hay que tener algunas cosas más en cuenta.

Validación – Select obligatorio

Por su naturaleza, un campo "select" de tipo lista desplegable siempre tiene al menos una opción seleccionada, la primera de las opciones de la lista:

```
<select name="residencia" required>  
  <option value="E">España</option>  
  <option value="F">Francia</option>  
  <option value="A">Alemania</option>  
  <option value="P">Portugal</option>  
</select>
```

País de residencia:

En este caso, el atributo required no hace nada, porque siempre hay una opción seleccionada.

Para poder hacer validación de campo obligatorio en un select, tenemos que hacer algunos ajustes.

Validación – Select obligatorio

Para que el navegador interprete que no hay opción seleccionada, tenemos que añadir una opción vacía (con value=""), la primera de todas:

```
<select name="residencia" required>  
  <option value="">Seleccione...</option>  
  <option value="E">España</option>  
  <option value="F">Francia</option>  
  <option value="A">Alemania</option>  
  <option value="P">Portugal</option>  
</select>
```

País de residencia:

El texto de la opción puede tener algo como "Seleccione una opción", "Seleccione...", "Elige una opción", etc.

O puede dejarse en blanco.

País de residencia:

En este caso el select se vería vacío.

Validación – Radio button obligatorio

Como hemos visto, los radio button se utilizan para que el usuario elija entre varias opciones excluyentes entre sí, cuando sólo puede elegir una de ellas.

Lo conseguimos haciendo que todos los input type="radio" tengan el mismo nombre.

Si queremos obligar al usuario a que elija una de las opciones, basta con que añadamos el atributo "required" a uno de los radio buttons. No es necesario hacerlo en todos.

```
<label><input type="radio" name="color" value="B" required> Blanco</label>  
<label><input type="radio" name="color" value="G"> Gris</label>  
<label><input type="radio" name="color" value="R"> Rojo</label>
```

Validación – Checkbox obligatorio

Los checkboxes se usan en dos contextos: como elemento aislado, o como grupo de checkboxes.

Como elemento aislado puede ser, por ejemplo, cuando se pide aceptación de condiciones legales.

```
<label>
```

```
  <input type="checkbox"  
    name="legal" required>
```

```
  Acepto las condiciones de contratación
```

```
</label>
```

☐ Acepto las condiciones de contratación

En este caso, el atributo `required` obliga al usuario a marcar ese checkbox, si no, no podrá enviar el formulario.

Validación – Checkbox obligatorio

Como grupo de checkboxes podría ser, por ejemplo, cuando se eligen opciones para algo, que se pueden combinar entre sí, y que unas no excluyen las otras.

```
<label><input type="checkbox"
  name="equipo" value="LL">
  Llantas de aleación</label>
```

```
<label><input type="checkbox"
  name="equipo" value="CB"> Climatizador bizona</label>
```

```
<label><input type="checkbox"      name="equipo" value="FX"> Faros
Xenon</label>
```

Equipamiento

☐ Llantas de aleación

☐ Climatizador bizona

☐ Faros Xenon

En estos casos podemos querer que el usuario elija al menos una de las opciones. Esto no es posible sólo con HTML. Tendríamos que utilizar JavaScript.

Validación – Otros atributos

- minlength y maxlength: longitud mínima y/o máxima para un campo de texto.
- min y max: valor mínimo y/o máximo para un campo numérico
- step: el valor del campo numérico tiene que ser múltiplo del paso (incluido el cero).
- pattern: especifica un patrón (usando expresiones regulares) que define un patrón que los datos introducidos deben cumplir. Por ejemplo, si queremos que se introduzcan cuatro dígitos, podríamos usar pattern="[0-9]{4}".

Más información de expresiones regulares en:

- <https://regexone.com/>
- <https://regexr.com/>

Validación cliente vs servidor

Las aplicaciones web suelen validar la información en dos momentos:

- En el cliente, en el formulario, antes de hacer el envío al servidor.
- En el servidor, al recibir la información del cliente.

Es fundamental entender que la validación en cliente es vulnerable a multitud de ataques, porque se puede añadir código que permita enviar un formulario, aunque no sea válido.

Por ese motivo la validación en cliente no es un sustituto de la validación en servidor. Es un complemento, que mejora la experiencia de usuario.

El servidor SIEMPRE tiene que verificar la validez de los datos al recibirlos, aunque haya una validación en cliente.