

Validaciones HTML5

PUBLICIDAD

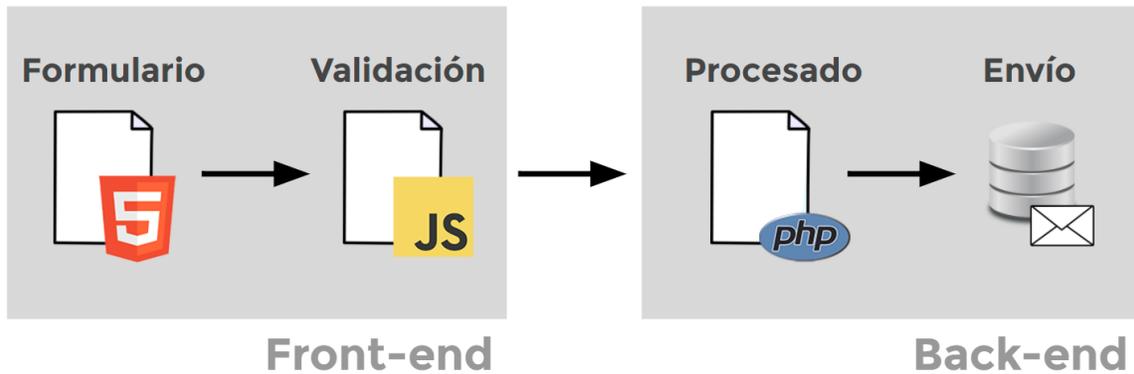
[Medidores y barras de progreso «](#)[» Etiquetas de scripting](#)

Al crear un formulario en HTML, debemos ser conscientes de un detalle ineludible: **los usuarios se equivocan al rellenar un formulario**. Ya sea por equivocación del usuario, ambigüedad del formulario, o error del creador del formulario, el caso es que debemos estar preparados y anticiparnos a estos errores, para intentar que los datos lleguen correctamente a su destino y evitar cualquier tipo de moderación o revisión posterior.

Para evitar estos casos, se suele recurrir a un tipo de proceso automático llamado **validación**, en el cuál, establecemos unas pautas para que si el usuario introduce alguna información incorrecta, deba modificarla o en caso contrario no podrá continuar ni enviar el formulario correctamente.

Un esquema tradicional de validación de ejemplo podría ser el siguiente, donde tenemos un formulario diseñado en HTML5 y mediante [javascript](#) realizamos una validación para comprobar que los datos son correctos.

Una vez superada, el formulario se envía al back-end de la página para ser procesado y enviarlo por email (por ejemplo).



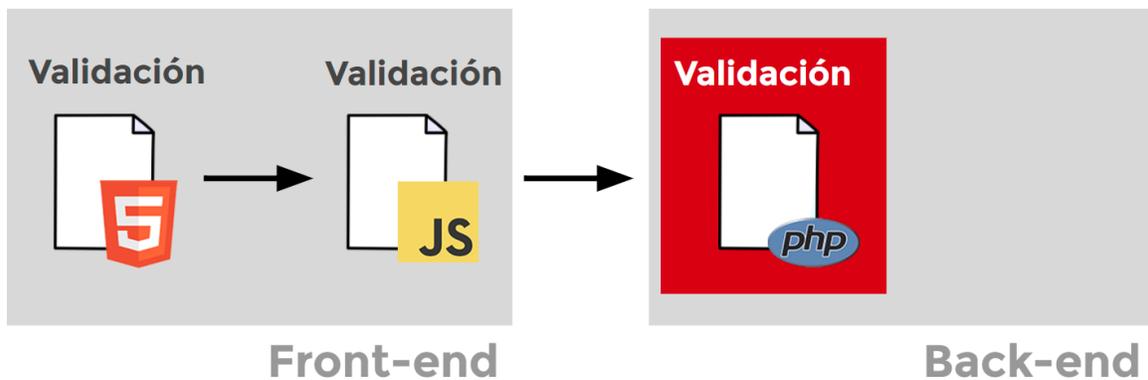
Tipos de validación

Cada vez que creamos un formulario, la **validación** de los datos introducidos estará situada en uno de los siguientes casos (colocados de peor a mejor):

1. En este primer caso, **el formulario no tiene validación** de ningún tipo. El usuario puede escribir la información y el sistema no comprobará los datos, ni realizará ningún tipo de validación. Es el peor escenario posible, puesto que el usuario podría enviar desde información incorrecta, hasta datos malintencionados que podrían comprometer la seguridad de la página.
2. Otro caso podría ser que **el formulario tiene validación sólo en el front-end** (cliente). De esta forma, los datos son verificados en el navegador del usuario antes de enviarse, pero carecen de validación en el **back-end**, por lo que un usuario malintencionado podría eliminar la validación del front-end y saltársela, enviando datos malintencionados que comprometan la seguridad de la página.

3. [LenguajeHTML](#) [Documentación](#) [¡Aprende HTML5!](#) [Acerca de...](#) **ón**
SOLO EN EL BACK-END. De esta forma, garantizamos que un usuario malintencionado no podrá eliminar el proceso de validación, y los datos siempre se comprobarán. Sin embargo, la desventaja de este método es que el usuario puede rellenar un formulario y es necesario que lo envíe (con la tardanza que eso puede acarrear), se procese en el back-end y al devolver un error, el usuario tenga que retroceder al formulario y en algunos casos, incluso tener que volver a rellenar todos los campos de nuevo.
4. Por último, tendríamos el **caso ideal**, donde el formulario **tiene validación en el front-end y en el back-end**, también denominado **dobles validación**. En este caso, el formulario es sometido a un proceso de validación en la parte del front-end, y si lo supera, vuelve a pasar otro proceso de validación en el back-end. La desventaja de este método es que conlleva más trabajo de validación, pero es el sistema recomendado, puesto que es más estricto y sobre todo, más seguro.

Tradicionalmente, la validación de un formulario se ha hecho siempre en Javascript, sin embargo, HTML5 introduce unos nuevos atributos para formularios que permiten realizar la validación del formulario directamente en HTML5, sin necesidad de recurrir a Javascript (aunque es posible hacerlo si se desea):



Recuerda siempre realizar la **validación en el servidor o back-end** (en

Atributos básicos de validación

En nuestros campos de entrada de datos, se pueden utilizar ciertos **atributos** para realizar validaciones sencillas. Algunos de estos atributos ya lo hemos visto en apartados anteriores, sin embargo, vamos a comentarlos uno por uno:

Atributo	Valor	Puede actuar sobre...	Descripción
minlength	<u>número</u>	Campos de texto	Establece la longitud mínima del texto requerida.
maxlength	<u>número</u>	Campos de texto	No permite escribir textos superiores a <u>número</u> caracteres.
min	<u>número</u>	Campos numéricos	Establece el número mínimo permitido.
	<u>fecha</u>	Campos de fecha	Establece la fecha mínima permitida.
	<u>hora</u>	Campos de hora	Establece la hora mínima permitida.
max	<u>número</u>	Campos numéricos	Establece el número máximo permitido.
	<u>fecha</u>	Campos de fecha	Establece la fecha máxima permitida.

 LenguajeHTML	 Documentación sobre...	 ¡Aprende HTML5!	 Acerca de...
	<u>hora</u>	Campos de hora	Establece la hora máxima permitida.
step	<u>número</u>	Campos numéricos	Establece el salto de números permitido. Por defecto, 1.
	<u>fecha</u>	Campos de fecha	Establece el salto de días permitido. Por defecto, 1.
	<u>hora</u>	Campos de hora	Establece el salto de segundos permitido. Por defecto, 1.
required		Campos en general	Campo obligatorio. Se debe rellenar para enviar formulario.
disabled		Campos en general	Campo desactivado. No se puede modificar. No se envía.
readonly		Campos en general	Campo de sólo lectura. No se puede modificar. Se envía.

Con estos atributos, podemos crear validaciones básicas en nuestros campos de entrada de datos, obligando al usuario a rellenar un campo obligatorio, forzando a indicar valores entre un rango numérico o permitiendo sólo textos con un tamaño específico, entre otros.

```
HTML <form name="formulario" method="post"
action="http://pagina.com/send.php">
  <!-- Nombre de usuario.
  Campo obligatorio, entre 5-40 caracteres -->
  Usuario: <input type="text" name="nombre" placeholder="Por
ejemplo, Juan"
           minlength="5" maxlength="40" required>
  <!-- Contraseña.
  Campo obligatorio, mínimo 11 caracteres -->
```

```
🏠 LenguajeHTML 📄 Documentación 🗉 ¡Aprende HTML5! 👤 Acerca de...  
minlength="11" required>  
</form>
```

Además, utilizando las [pseudoclasas CSS de validación](#) `:valid` e `:invalid` podemos aplicar estilos a los campos `<input>` y `<textarea>` teniendo en cuenta su validación. Aplicando el siguiente código CSS se mostrará un fondo verde o rojo, dependiendo de si cumple o no la validación, respectivamente:

```
CSS input:valid, textarea:valid {  
    background:green;  
}  
input:invalid, textarea:invalid {  
    background:red;  
}
```

En el caso particular de un campo `<input type="email">` se puede indicar el atributo **multiple** para permitir varios emails y que valide correctamente en el caso de estar realizando una validación HTML5.

Patrones de validación HTML5 🔗

No obstante, aunque los atributos de validación básicos son muy interesantes y pueden facilitarnos la tarea de validación, en muchos casos son insuficientes. Para ello tenemos los **patrones de validación HTML5**, mucho más potentes y flexibles, que nos permitirán ser mucho más específicos utilizando expresiones regulares para validar datos.

Una **expresión regular** es una cadena de texto que representa un posible patrón de coincidencias, que aplicaremos mediante el atributo **pattern** en los campos que queramos validar.

Para ello hay que conocer algunas características básicas de las

exp [🏠 LenguajeHTML](#) [📄 Documentación](#) [🗣️ ¡Aprende HTML5!](#) [👤 Acerca de...](#)

Expresión regular	Carácter especial	Denominación	Descripción
.	Punto	Comodín	Cualquier carácter (o texto de tamaño 1)
A B	<i>Pipe</i>	Opciones lógicas	Opciones alternativas (o A o B)
C(A B)	Paréntesis	Agrupaciones	Agrupaciones alternativas (o CA o CB)
[0-9]	Corchetes	Rangos de caracteres	Un dígito (del 0 al 9)
[A-Z]			Una letra mayúscula de la A a la Z
[^A-Z]	^ en corchetes	Rango de exclusión	Una letra que no sea mayúscula de la A a la Z
[0-9]*	Asterisco	Cierre o clausura	Un dígito repetido 0 ó más veces (vacío incluido)
[0-9]+	Signo más	Cierre positivo	Un dígito repetido 1 ó más veces
[0-9]{3}	Llaves	Coincidencia exacta	Cifra de 3 dígitos (dígito repetido 3 veces)
[0-9]{2,4}		Coincidencia	Cifra de 2 a 4 dígitos

🏠 LenguajeHTML regular	📄 Documentación especial	📖 ¡Aprende HTML5!	👤 Acerca de...
		(rango)	(rep. de 2 a 4 veces)
b?	Interrogación	Carácter opcional	El carácter b puede aparecer o puede que no
\.	Barra invertida	Escape	El carácter . literalmente (no como comodín)

➤ Ejemplos de patrones HTML5 📌

Para clarificar estos conceptos y entenderlos mejor, vamos a mostrar algunos ejemplos de campos con validaciones HTML5 en casos reales, y así afianzar conceptos sobre expresiones regulares y su aplicación en los atributos **pattern**:

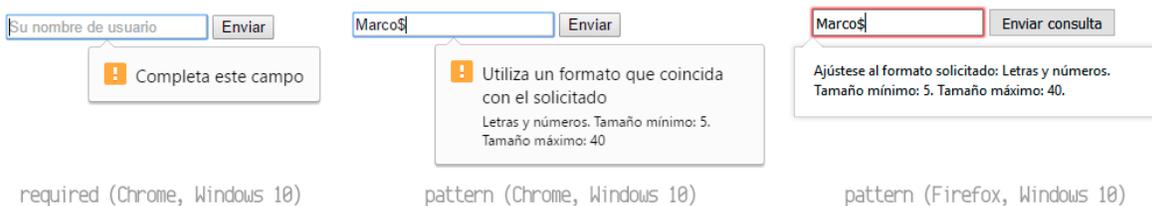
- Tipo de campo: **Nombre de usuario**
- Campo obligatorio: **required**.
- Entre 5-40 caracteres: **minlength="5" maxlength="40"**
- Sólo se permiten letras (mayúsculas y minúsculas) y números: **pattern="[A-Za-z0-9]+"**

```
HTML <form name="formulario" method="post"
      action="http://pagina.com/send.php">
      <!-- Nombre de usuario.
           Campo obligatorio, entre 5-40 caracteres
           Sólo se permiten letras y números -->
      <input type="text" name="nombre" placeholder="Su nombre de
      usuario"
           minlength="5" maxlength="40" required pattern="[A-Za-
      z0-9]+">
```

Nótese que de no incluir los atributos **minlength** y **maxlength** el usuario no tendría limitación en cuanto al tamaño. Esto también puede incorporarse en la propia expresión regular, y prescindir de dichos atributos:

```
HTML <form name="formulario" method="post"
      action="http://pagina.com/send.php">
  <!-- Nombre de usuario.
        Campo obligatorio, entre 5-40 caracteres
        Sólo se permiten letras y números -->
  <input type="text" name="nombre" placeholder="Su nombre de
usuario"
        required pattern="[A-Za-z0-9]{5,40}"
        title="Letras y números. Tamaño mínimo: 5. Tamaño
máximo: 40">
</form>
```

Sin embargo, en este caso, no se limitará al usuario a la hora de escribir, como hace **maxlength**, sino que permitirá al usuario escribir la información que desee y en caso de no pasar la validación, mostrará un mensaje de advertencia y no lo dejará continuar hasta que termine. Podemos ampliar el mensaje de advertencia incluyendo el texto en el atributo **title**.



En el siguiente caso, se pide al usuario que indique el modelo de coche que posee, en un posible formulario de servicio técnico. Los modelos posibles son A1, A3, A4 y A15. En lugar de mostrar una **lista de selección**, podemos mostrar un campo de texto y colocar una validación como la siguiente:

- [LenguajeHTML](#) [Documentación](#) [¡Aprende HTML5!](#) [Acerca de...](#)
- **campo obligatorio. required.**
- Sólo se permiten las opciones: **A1, A3, A4 y A15**

```
HTML <form name="formulario" method="post"
      action="http://pagina.com/send.php">
      <!-- Modelo de coche
           Campo obligatorio, opciones posibles: A1, A3, A4 y A15
           Sólo se permiten letras y números -->
      <input type="text" name="coche" placeholder="Su modelo de
      coche"
           required pattern="A|a(1|3|4|15)"
           title="Modelos posibles: A1, A3, A4 y A15">
</form>
```

Nótese que se permite tanto el formato **a1** como el formato **A1**.

Herramientas para expresiones regulares

A continuación, algunas herramientas útiles para crear o probar **expresiones regulares**:

Herramienta	Descripción
Debuggex 	Herramienta de creación gráfica de expresiones regulares.
RegExR 	Herramienta para construir y testear expresiones regulares.
RegEx101 	Herramienta para construir y testear expresiones regulares.
RegExBuddy 	Herramienta profesional para el trabajo con expresiones regulares.

LenguajeHTML	Documentación	¡Aprende HTML5!	Acerca de...
RegExplained	Herramienta visual para explicar el funcionamiento de una expresión regular.		
RegExper	Herramienta para generar diagramas visuales de expresiones regulares.		
Rubular	Editor de expresiones regulares.		

PUBLICIDAD



Publicado por Manz

Docente, divulgador informático y freelance. Trabaja como autónomo, escribe en [Emezeta.com](#) (sitio web de tecnología con +10.000 visitas diarias) y es profesor en la [Oficina de Software Libre de la Universidad de La Laguna](#), en Tenerife (Islas Canarias).



 Pulsa aquí para dejar un comentario

[Medidores y barras de progreso](#) «

[Índice](#)

» [Etiquetas de scripting](#)

 [LenguajeHTML](#)

 [Documentación](#)

 [¡Aprende HTML5!](#)

 [Acerca de...](#)

Lenguaje HTML

Sitio web de documentación, información, docencia y herramientas acerca de diseño web y front-end (HTML5 / CSS3 / Javascript).

Creado y mantenido por [@Manz](#) con ❤️

