
FORMULARIOS

Un formulario es un conjunto de controles que permiten al usuario interactuar con el documento o página. El objetivo de esta interacción suele ser, a menudo, la solicitud de información adicional o un mero intercambio de datos a petición del usuario.

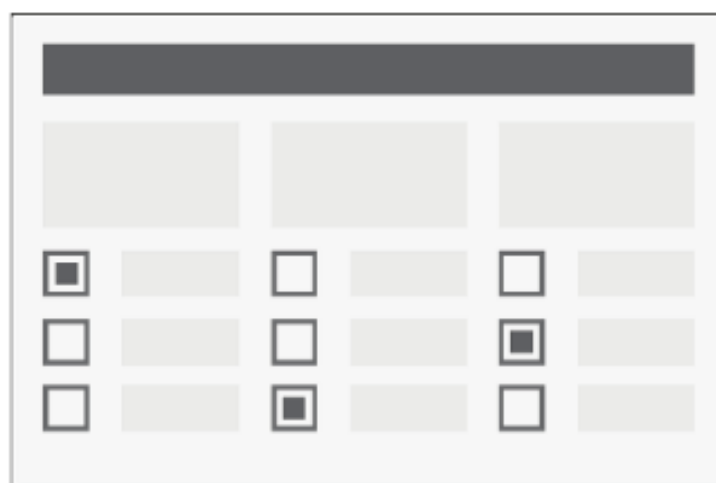


Diagrama de un formulario web. Incluye un encabezado oscuro, tres campos de texto grandes, y una sección de botones y casillas de verificación. Los botones están representados por cuadrados con esquinas redondeadas, algunos con un pequeño cuadrado negro en el centro. Las casillas de verificación son cuadrados pequeños con un borde negro.



Entre los diferentes controles que se pueden insertar o agregar a los formularios podemos encontrar acciones directas vinculadas a botones, solicitud de entradas de texto de una única línea, solicitud de entradas de texto multilínea, casillas de verificación, botones de única elección o tipo radio, selección de objetos y/o ficheros, entre otros.

En este capítulo vamos a ver los tipos básicos de formulario, cómo definirlos, cómo enviarlos, cómo hacerlos receptivos y, gran medida, cómo hacerlos usables y accesibles.

7.1 TIPOS DE FORMULARIO

Existen, fundamentalmente, cinco tipos de formulario que están definidos en función de su objetivo. Formularios de contacto, acceso, registro, suscripción y de entrada general.

No obstante, sea cual sea el tipo de formulario y su objetivo, para que cumpla con las expectativas de los clientes y con los requisitos establecidos por el RGPD (Reglamento General de Protección de Datos), deben incluir:

- Una casilla de verificación explícita con la que, los usuarios, puedan aceptar la política de privacidad del sitio. Por defecto, no puede estar seleccionada.
- Los enlaces pertinentes con la política de privacidad, el aviso legal, política de cookies, límite y responsabilidad, etcétera.
- Un texto, no demasiado largo, ni demasiado escueto, sobre qué datos se van a almacenar, quién será el responsable y cuál es el objetivo de dicho almacenamiento.
- Algún método para que el usuario ejerza su derecho a la eliminación de datos.
- Algún algoritmo de cifrado para que la información se gestione y manipule de forma encriptada. Esto es, que los datos que se envíen y/o almacenen estén codificados para evitar usos fraudulentos.

Únicamente se deben solicitar los campos que sean necesarios y estén bien justificados, es decir, no se debe solicitar una información concreta, como pueda ser tus hobbies, a no ser que se tenga una buena razón y esté justificada. Además, se debe tratar de conseguir que el medio sea lo suficientemente seguro y confiable, como para que invite a introducir los datos solicitados.

7.1.1 Formularios de contacto

Los formularios de contacto pueden llegar a ser un elemento clave en un sitio web porque permiten, o hacen posible, que la comunicación entre los usuarios y los proveedores sea directa y privada.

En general, un formulario de contacto debe solicitar sólo la información esencial para el proceso de comunicación. Esto es, no se deben pedir datos como la razón social, sitio web o teléfono sólo por fines comerciales o estadísticos.

Por ejemplo, el siguiente formulario podría ser considerado no fiable, además de ser ilegal.

CONTACTAR CON NOSOTROS

NOMBRE

Introduzca su nombre

EMAIL

Introduzca su correo electrónico

TELÉFONO

Opcional

SITIO WEB

Opcional

MENSAJE

ENVIAR MENSAJE

Como se puede apreciar en la ilustración anterior, el sitio web no es un dato necesario y, aunque sea un dato opcional, puede suscitar desconfianza e impedir que el usuario haga uso de él. Además, como se ha mencionado anteriormente, es ilegal puesto que no presenta el checkbox de Política de Privacidad, entre otras cosas.

A continuación, se muestra el mismo ejemplo, pero corregido.

CONTACTAR CON NOSOTROS

NOMBRE

Introduzca su nombre

EMAIL

Introduzca su correo electrónico

MENSAJE

☐

Sí, acepto la política de privacidad

ENVIAR MENSAJE

Texto explicativo sobre qué datos se almacenan, quién es el responsable, cuál es su objetivo y legitimación y, cómo se pueden eliminar los datos.

7.1.2 Formularios de suscripción

Los formularios de suscripción son, esencialmente, lo mismo que los formularios de contacto, pero más sencillos. Su objetivo es recuperar correos electrónicos para luego utilizarlos con fines comerciales o informativos.

En general, un formulario de suscripción sólo requiere de una caja de texto para recuperar el email, no obstante, es habitual pedir también el nombre para dirigirse a él. Por ejemplo, la siguiente ilustración podría una buena opción como formulario de suscripción.

SUSCRIPCIÓN A LA NEWSLETTER

| NOMBRE | EMAIL |
|---|---|
| <input type="text" value="Sólo tu nombre"/> | <input type="text" value="Introduzca su correo electrónico"/> |

☐ **Sí, acepto la política de privacidad**

SUSCRIBIRSE

INFORMACIÓN BÁSICA SOBRE PROTECCIÓN DE DATOS

Texto explicativo sobre qué datos se almacenan, quién es el responsable, cuál es su objetivo y legitimación y, cómo se pueden eliminar los datos.

7.1.3 Formularios de acceso

Los formularios de acceso son, junto con los formularios de contacto, los más recurrentes y utilizados en el mundo web. Su objetivo, como su propio nombre indica, es proporcionar acceso a información, productos o servicios que no están disponibles por vía pública o sin proporcionar una identidad.

En general, un formulario de acceso se caracteriza por tener dos cajas de texto para el nombre de usuario y contraseña, una casilla de verificación o botón de tipo interruptor para mantener la sesión iniciada, un enlace para recuperar la contraseña en supuesto caso de olvido y, evidentemente, un botón para acceder.

Los nombres de usuario se presentan con el texto visible y, a menudo, son el correo electrónico de los usuarios, aunque puede valer cualquier tipo de identificador único como un DNI, código de cliente, nombre de usuario, etcétera.

Las contraseñas suelen presentarse con el texto oculto, con asteriscos o puntos y, a menudo, suelen proporcionar un modo de cambiar a modo visible para ver el texto escrito.

A continuación, se muestra un ejemplo:

El diagrama muestra un formulario de acceso con el título "FORMULARIO DE ACCESO". Hay dos campos de entrada: "Email o NIF" y "Contraseña". El campo de contraseña tiene una máscara de puntos y un icono de ojo para alternar la visibilidad. Debajo de estos campos hay un control de "Mantener la sesión iniciada" con el valor "NO" seleccionado. A la izquierda hay un enlace "RECORDAR CONTRASEÑA" y a la derecha un botón "REGISTRARSE". En la parte inferior hay un botón ancho "ACCEDER".

Si observamos la ilustración anterior, veremos que el campo contraseña tiene un icono de ojo que funciona como botón para cambiar el modo de presentación de ese campo. Si pulsamos una vez se pondrá en modo visible y seremos capaces de leer lo escrito, pero si pulsamos otra vez, volverá a ocultarse, mostrándose como la vemos ahora mismo.

Además, se ha puesto un botón de acceso al formulario de registro. Esta funcionalidad también es habitual, sobre todo, porque un registro también suele ser un acceso, una vez que ha finalizado el proceso.

7.1.4 Formularios de registro

Los formularios de registro suelen ir de la mano con los de acceso. Si bien, no siempre se ofrece la posibilidad de registrarse online, sí es lo más habitual.

En general, un formulario de registro se caracteriza por la presencia de campos propietarios como el nombre o email del usuario, pero también pueden contener elementos considerados información sensible como son la edad, género o país de nacimiento. La decisión de qué campos se deben solicitar al usuario debe estar regida por la lógica de negocio y el RGPD (Reglamento General de Protección de Datos). No obstante, recordemos que, cuantos menos campos se solicite, más cómodo y confiable podrá sentirse el usuario.

Los nombres de usuario para dicho registro son, a menudo, el correo electrónico de los usuarios, aunque puede valer cualquier tipo de identificador único como un DNI o código de cliente.

Las contraseñas suelen seguir un patrón estricto de definición para evitar la fácil recuperación a través de métodos como la fuerza bruta y uso fraudulento.

En general, podríamos decir que un buen patrón de contraseña es aquel que tiene, al menos, un carácter en mayúsculas, un carácter en minúsculas, un dígito o número, un carácter especial como pueda ser el símbolo de almohadilla o interrogación y con una longitud mínima de ocho caracteres.

Sobra decir que no se deben utilizar sustituciones de carácter a número como pueda ser sustituir la letra A por el número cuatro, ni fechas especiales como cumpleaños o aniversarios y, por supuesto, nada de nombres propios. Eso sí, hay que tratar de que sea fácil de recordar y difícil de adivinar.

A continuación, se muestra un ejemplo:



FORMULARIO DE REGISTRO

Nombre de usuario Email

Contraseña Repita contraseña

Nombre Completo

☐ He leído y acepto los **Términos y Condiciones** y **la Política de Privacidad**.

☐ No quiero recibir correos electrónicos sobre nuevos productos, ofertas, ...

ACCEDER

Cabe destacar que, es bueno que los campos de un formulario de registro se soliciten de forma que el usuario vaya tomando confianza de menos a más. Es decir, se debe tratar de conseguir que los usuarios confíen un poco más cada vez que se avanza en el proceso de registro. Esto es así porque puede pasar que un usuario no se sienta del todo cómodo si se le solicita primero la edad antes que el nombre de usuario y contraseña.

Por último, sólo destacar una cosa más. Si el formulario de registro requiere de muchos campos, es mejor que sean agrupados por contexto y solicitados en varios pasos o pantallas.

7.1.5 Formularios de entrada general

Los formularios de entrada general son aquellos formularios que tienen el propósito de recuperar una información concreta y que, por lo general, están pensados para cubrir las necesidades que no están cubiertas en las casuísticas anteriores.

Por ser algo más explícito, un formulario de propósito general podría ser un formulario de inserción de comentarios, de eventos o de inscripción a sorteos, concursos o juegos.

7.2 ELEMENTOS DISPONIBLES EN HTML5

Los elementos disponibles son bastantes y todos pueden manejar diferentes atributos, sin embargo, podemos destacar algunos por gran presencia en la definición y creación de formularios:

| Atributo | Descripción |
|---------------------------------|--|
| disabled | Especifica si el elemento está deshabilitado o no. Esta propiedad tendrá efecto con sólo declararla, es decir, en cuanto esté presente e independientemente de su valor, el elemento aparecerá deshabilitado, no obstante, suele declararse sin valor asignado. Es válido para todos los elementos, a excepción de form , datalist , label , legend , meter , output y progress . |
| id , autofocus | Estos atributos son globales y son válidos para cualquier elemento. Sin embargo, con respecto a los formularios el atributo id resulta fundamental para vincular los elementos label con los elementos de entrada como input , select o textarea . |
| form | Especifica el formulario al que pertenece el elemento cuando se declaran sus elementos fuera del ámbito del formulario. Por ello, el valor de este atributo debe ser el mismo que el indicado por el atributo id del elemento form , o de lo contrario, no se hará efectivo. Es válido para todos los elementos, a excepción de datalist , form , optgroup , option y progress . |
| sname | Especifica o asigna el nombre del elemento y se utiliza para recuperar el valor de los campos del formulario desde el lado del servidor. Si el valor de este atributo lleva asignado como sufijo [] (corchetes), el nombre actuará como si de un array numérico se tratase, en donde los distintos elementos a través de un índice, siendo, el índice CERO el primero y, N-1, el último. Es válido para todos los elementos a excepción de label , fieldset , legend , meter , optgroup , option , output . |
| value | Especifica el valor textual del elemento, por lo que no puede contener nada que no sea texto. Este atributo está en la definición de todos los elementos de formulario a excepción de los elementos form , label , fieldset , legend , output y textarea . |

7.2.1 Elemento form

El elemento **FORM** especifica que el contenido que se va a representar es un formulario, es decir, una estructura de interacción, formada por uno o varios elementos, que permiten introducir datos y enviarlos al servidor para su procesamiento.

Los formularios pueden albergar muy diversos elementos con diferentes formatos y estructuras, pero los más comunes quizás sean **INPUT**, **TEXTAREA**, **BUTTON**, **SELECT**, **OPTION**, **OPTGROUP**, **FIELDSET**, **LABEL** y **OUTPUT**, los cuales se pasarán a ver a continuación.

Además de los comentados al comienzo de este apartado, entre los atributos que admite en su configuración, se deben destacar los siguientes:

| Atributo | Descripción |
|-----------------------|--|
| accept-charset | Especifica la codificación de caracteres que se debe usar cuando se realice el envío del formulario. Su valor predeterminado es UNKNOWN, que indica que la codificación de caracteres a utilizar debe ser la misma que la del documento actual. Entre sus valores más frecuentes podemos encontrar la codificación ISO-8859-1 y la codificación UTF-8. |
| action | Especifica el destino dónde enviar los datos, es decir, la dirección a la que se redirigirá cuando se vaya a realizar la acción de enviar los datos de formulario. |
| autocomplete | Especifica si el contenido que se va a representar debe ser manipulado por el navegador y herramientas de asistencia de manera automática. Sus posibles valores son off , que le indica al navegador que no se deben sugerir opciones automáticamente, on , que le indica al navegador que se deberían sugerir opciones automáticamente, pero sin indicar el tipo de dato que se espera recibir y [valor] , que le indica al navegador que se deberían sugerir opciones automáticamente y especificando el tipo de dato que se espera recibir, todas ellas descritas en la web de MDN Web Docs. |
| enctype | Especifica cómo se deben codificar los datos a enviar. Sus posibles valores son application/x-www-form-urlencoded , que es la opción por defecto e indica que se codifiquen todos los caracteres antes de enviarlos, multipart/form-data , que indica que no se codifiquen los caracteres y text-plain , que indica que sólo los espacios sean codificados como símbolos de suma. |
| method | Especifica el método de envío por el que se deben enviar los datos. Sus posibles son GET , que envía los datos del formulario como parámetros de la propia dirección o URL y POST , que los agrega como parte del cuerpo de la solicitud HTTP. |
| novalidate | Especifica si se deben validar los elementos del formulario o no y tendrá efecto con sólo declararlo, es decir, en cuanto esté presente, e independientemente de su valor, desactivará la validación de todos los elementos del formulario. |
| target | Especifica a qué contexto o ventana se enviará la información. Entre sus posibles valores los más frecuentes son _blank , que indica que se abra en una nueva pestaña o contexto, _self , que indica que se abra en la misma pestaña o contexto, _parent , que indica que se abra en la pestaña o contexto padre, _top , que indica que se abra en el primer elemento BODY de la ventana y [nombre] , que indica que se abra en el marco con ese nombre. |

```
<form action="/action_page.php" method="get" name="frm">
<!-- ... -->
</form>
```


7.2.2 Elemento button

El elemento **BUTTON** especifica que el contenido que se va a representar es una acción. Como veremos, a diferencia del elemento **INPUT**, el elemento **BUTTON** puede albergar una gran variedad de contenidos como, por ejemplo, una imagen, un texto o, incluso, otros elementos **HTML**.

Además de los comentados al comienzo de este apartado, entre los atributos que admite en su configuración, se deben destacar los siguientes:

| Atributo | Descripción |
|-----------------------|---|
| formaction | Especifica el destino dónde enviar los datos, es decir, la dirección a la que se redirigirá cuando se vaya a realizar la acción de enviar los datos de formulario. Si cuando este atributo está presente, el atributo action del elemento form será anulado, pero puede ser una buena idea cuando se desea que el formulario que se está definiendo pueda ejecutar varias acciones que llevan a objetivos distintos. |
| formenctype | Especifica cómo se deben codificar los datos a enviar. Sus posibles valores son los mismos que los definidos en el atributo enctype del elemento form . Si cuando este atributo está presente, el atributo enctype del elemento form será anulado, pero puede ser una buena idea cuando se desea enviar un mismo formulario con diferentes tipos de codificación. |
| formmethod | Especifica el método de envío por el que se deben enviar los datos. Sus posibles valores son los mismos que los definidos en el atributo method del elemento form . Si este atributo está presente, el atributo method del elemento form será anulado, pero puede ser una buena idea cuando se desea enviar un mismo formulario con diferentes métodos. |
| formnovalidate | Especifica, con sólo declararlo, si se deben validar los elementos del formulario o no, por lo que, en cuanto esté presente, e independientemente de su valor, desactivará la validación del elemento de formulario. Si este atributo está presente, el atributo novalidate del elemento form será anulado, pero puede ser una buena idea cuando se desea validar el formulario en función de si se ejecuta una u otra acción. |
| formtarget | Especifica a qué contexto o ventana se enviará la información. Si este atributo está presente, el atributo target del elemento form será anulado, pero puede ser una buena idea cuando se desea enviar un mismo formulario a diferentes ventanas o contextos. Sus posibles valores son los mismos que los definidos en el atributo target del elemento form . |
| type | Especifica el tipo de botón. Sus posibles valores son submit , que sirve para indicar que se proceda al envío de los datos del formulario, button , para que permita hacer clic sin enviar los datos del formulario y, reset , para indicar que se restablezcan todos los elementos del formulario a sus valores por defecto o preestablecidos. |

```
<button type="submit"
formaction="./pages/login-test-2.php"
formmethod="POST"
formenctype="multipart/form-data"
formtarget="_blank">
value="Probar test 2"
</button>
```

7.2.3 Elemento datalist

El elemento **DATALIST** especifica que el contenido que se va a representar es una lista de opciones predefinidas para un elemento **INPUT**.

La razón para utilizar este elemento es para proveer de una funcionalidad de autocompletado a los elementos **INPUT**. Los usuarios podrán ver las diferentes opciones como si de un desplegable se tratase, pero con la opción de ir buscando a través de coincidencias parciales proporcionadas mediante teclado.

Cabe destacar que el único atributo que necesita para funcionar es el atributo **ID**. El valor de este atributo debe coincidir exactamente con el valor del atributo **LIST** declarado en el elemento **INPUT** al que está asociado.

```
<input list="technologies">

<datalist id="technologies">
<option value="HTML5">
<option value="JavaScript">
<option value="CSS3">
<option value="SVG">
</datalist>
```

7.2.4 Elemento fieldset

El elemento **FIELDSET** especifica que el contenido que se va a representar es una agrupación de elementos relacionados. En general, y salvo excepciones, todos los agentes de usuario dibujan un cuadro alrededor de este elemento, equivalente a un estilo de borde.

```
<form action="./login.php">
<fieldset>
<legend>Acceso Privado</legend>

<!-- ... -->
</fieldset>
</form>
```

7.2.5 Elemento input

El elemento **INPUT** especifica que el contenido que se va a representar es una entrada de datos.

Una de las cualidades más importantes que tiene el elemento **INPUT** es que tiene una gran variedad de validaciones nativas. Por ejemplo, es posible definir un elemento de entrada que sólo admita números, que sólo admita fechas en un formato específico o que valide las reglas de nombres de los correos electrónicos.

Además de los comentados al comienzo de este apartado, entre los atributos que admite en su configuración, se deben destacar los siguientes:

| Atributo | Descripción |
|---------------------|--|
| accept | Especifica los tipos de archivo válidos cuando se utiliza un elemento input de tipo file . Todos los posibles valores que puede tomar este atributo están disponibles en la página web de en IANA Media Types (http://www.iana.org/assignments/media-types/), aunque, lo frecuente, es que tome un valor comodín como pueda ser audio/* , video/* o image/* , los cuales indican la aceptación de todas las extensiones de archivo para cada tipo. |
| alt | Permite especificar un texto alternativo para las imágenes que son definidas a través de un input tipo image . |
| autocomplete | Especifica si el contenido que se va a representar debe ser manipulado por el navegador y herramientas de asistencia de manera automática. Sus posibles valores son off , que le indica al navegador que no se deben sugerir opciones automáticamente, on , que le indica al navegador que se deberían sugerir opciones automáticamente, pero sin indicar el tipo de dato que se espera recibir y [valor] , que le indica al navegador que se deberían sugerir opciones automáticamente y especificando el tipo de dato que se espera recibir, todas ellas descritas en la web de MDN Web Docs. |
| checked | Especifica si el elemento está chequeado o no y sólo es válido para los elementos input de tipo radio o checkbox . Tendrá efecto con sólo declararlo, es decir, en cuanto esté presente e independientemente de su valor, el elemento se marcará como chequeado o seleccionado. |
| formaction | Especifica el destino dónde enviar los datos, es decir, la dirección a la que se redirigirá cuando se vaya a realizar la acción de enviar los datos de formulario. Si cuando este atributo está presente, el atributo action del elemento form será anulado, pero puede ser una buena idea cuando se desea que el formulario que se está definiendo pueda ejecutar varias acciones que llevan a objetivos distintos. |
| formenctype | Especifica cómo se deben codificar los datos a enviar. Sus posibles valores son los mismos que los definidos en el atributo enctype del elemento form . Si cuando este atributo está presente, el atributo enctype del elemento form será anulado, pero puede ser una buena idea cuando se desea enviar |

| | |
|-----------------------|---|
| formmethod | <p>Especifica el método de envío por el que se deben enviar los datos. Sus posibles valores son los mismos que los definidos en el atributo method del elemento form.</p> <p>Si este atributo está presente, el atributo method del elemento form será anulado, pero puede ser una buena idea cuando se desea enviar un mismo formulario con diferentes métodos.</p> |
| formnovalidate | <p>Especifica, con sólo declararlo, si se deben validar los elementos del formulario o no, por lo que, en cuanto esté presente, e independientemente de su valor, desactivará la validación del elemento de formulario.</p> <p>Si este atributo está presente, el atributo novalidate del elemento form será anulado, pero puede ser una buena idea cuando se desea validar el formulario en función de si se ejecuta una u otra acción.</p> |
| formtarget | <p>Especifica a qué contexto o ventana se enviará la información.</p> <p>Si este atributo está presente, el atributo target del elemento form será anulado, pero puede ser una buena idea cuando se desea enviar un mismo formulario a diferentes ventanas o contextos.</p> <p>Sus posibles valores son los mismos que los definidos en el atributo target del elemento form.</p> |
| height | Especifica el alto del elemento en píxeles. |
| list | <p>Especifica la lista de opciones predefinidas que se debe asociar a un elemento input.</p> <p>Este atributo sólo tiene sentido en combinación con el elemento datalist y su uso requiere que el valor del atributo id del datalist sea el mismo que el valor del atributo list del elemento input.</p> |
| max | Especifica el valor máximo que puede aceptar o se puede introducir en el elemento. |
| maxlength | Especifica la longitud máxima, en caracteres, del valor que puede ser ingresado en el elemento. |
| min | Especifica el valor mínimo que puede aceptar o se puede introducir en el elemento. |
| multiple | <p>Especifica que el elemento admite la selección o inserción de más de un valor.</p> <p>Sólo es válido para los elementos input de tipo file. Además, tendrá efecto con sólo declararlo, es decir, en cuanto esté presente e independientemente de su valor, el elemento permitirá la selección de múltiples valores.</p> |
| pattern | <p>Especifica la expresión regular con la que se validará la entrada de datos en el elemento.</p> <p>La expresión regular que utiliza es similar a la de JavaScript, no obstante, en la dirección http://html5pattern.com/, se pueden encontrar multitud de ejemplos aptos para ser utilizados.</p> <p>Sólo será válido para los elementos input de tipo text, date, search, url, tel, email y password.</p> |

| | |
|--------------------|---|
| placeholder | <p>Especifica el texto que se debe mostrar cómo pista de datos válidos cuando el elemento no contenga un valor.</p> <p>En general es una buena idea usarlo para ayudar a la usabilidad y accesibilidad web y, si va asociado con otro atributo como pattern, la pista a proporcionar debe estar en concordancia con la validez de la entrada.</p> |
| readonly | <p>Especifica que el elemento no permitirá la inserción de nuevos valores ni la modificación de su valor actual.</p> <p>En cuanto esté presente, e independientemente de su valor, provocará que se ponga en modo de sólo lectura, no obstante, permitirá las acciones de seleccionar, resaltar o copiar.</p> |
| required | Especifica que el elemento debe contener valor, aunque este sea un espacio en blanco, es decir, que no puede ser nulo ni vacío. |
| size | Especifica la ubicación del recurso externo que se desea cargar o mostrar y sólo será válido para los elementos input de tipo image . |
| src | Especifica la ubicación del recurso externo que se desea cargar o mostrar. Este atributo sólo es válido para los elementos input de tipo image . |
| step | <p>Especifica el intervalo de incremento o decremento sobre el valor actual del elemento, es decir, el número de pasos que se deben sumar o restar al valor actual del elemento.</p> <p>Sólo será válido para los elementos input de tipo datetime, datetime-local, date, month, number, range, time y week.</p> |
| type | <p>Especifica el tipo de elemento input. Sus posibles valores son:</p> <ul style="list-style-type: none"> • button: para indicar que es un botón. • checkbox: para indicar que es una casilla de verificación. • date: para indicar que es una fecha. • datetime: para indicar que es una fecha con hora. Sólo está soportado por Safari y Opera. • Datetime-local: para indicar que es una fecha con hora sin zona horaria. Sólo está soportado por Chrome, Microsoft Edge y Opera. • email: para indicar que es un correo electrónico. • file: para indicar que es un control para subir archivos al servidor. • hidden: para indicar que es un campo oculto. • image: para indicar que es una imagen. • month: para indicar que es un mes. Sólo está soportado por Chrome, Microsoft Edge y Opera. • number: para indicar que es un número real. • password: para indicar que es una contraseña. • radio: para indicar que es un botón de radio. • reset: para indicar que es un botón de restauración que reestablece los valores por defecto o preestablecidos. • search: para indicar que es un campo de búsqueda. • tel: para indicar que es un teléfono Actualmente sólo soportado por Safari. • text: para indicar que es una caja o campo de texto. • time: para indicar que es una hora. Actualmente no soportado por Safari. • url: para indicar que es una dirección web o URL. • week: para indicar que es un valor de semana. Sólo está soportado por Chrome, Microsoft Edge y Opera. |
| width | Especifica el ancho del elemento en píxeles. |

```

<form action="/action_page.php" method="get" name="frm">
<label for="field1">Campo de texto:</label>
<input type="text" id="field1" name="field1">

<label for="field2">Campo sólo números:</label>
<input type="number" id="field2" name="field2">

<label for="field3">Campo fecha:</label>
<input type="date" id="field3" name="field3">

<label for="field4">Campo URL:</label>
<input type="url" id="field4" name="field4">

<label for="field5">Campo para emails:</label>
<input type="email" id="field5" name="field5">

<label for="field6">Campo para teléfonos:</label>
<input type="tel" id="field6" name="field6">

<label for="field7">Campo para imágenes:</label>
<input type="image" id="field7" name="field7">

<label for="field8">Elemento para adjuntar archivos:</label>
<input type="file" id="field8" name="field8">

<label for="field9">Casilla de verificación:</label>
<input type="checkbox" id="field9" name="field9">

<label for="field10">Campo tipo radio:</label>
<input type="radio" id="field10" name="field10">

<label for="field11">Campo oculto:</label>
<input type="hidden" id="field11" name="field11">

<label for="submit">Botón enviar:</label>
<input type="submit" id="submit" name="submit">

<button type="submit">
Guardar datos
</button>
</form>

```

7.2.6 Elemento label

El elemento LABEL especifica que el contenido que se va a representar es una etiqueta para un elemento de formulario INPUT, METER, PROGRESS, SELECT o TEXTAREA.

Cuando se utiliza elemento LABEL, la estructura del documento se vuelve más legible y consistente. Además, beneficia tanto a la usabilidad web, como a la accesibilidad web porque los elementos pequeños como son las casillas de verificación pueden manipularse a través del elemento LABEL.

Además, al tener el control de formulario una etiqueta asociada, las herramientas de asistencia, como los lectores de pantalla, pueden leerla y comunicárselo a los usuarios con discapacidad visual total o parcial.

Además de los comentados al comienzo de este apartado, hay que destacar:

| Atributo | Descripción |
|------------|--|
| for | Especifica el id del elemento de formulario con el que está vinculada la etiqueta y funcionará como texto descriptivo de la entrada de datos. |

```
<label for="name">Nombre:</label>
<input type="text" id="name" name="name">
```

7.2.7 Elemento legend

El elemento **LEGEND** especifica que el contenido que se va a representar es un título para un elemento **FIELDSET**.

```
<form action="./login.php">
<fieldset>
<legend>Acceso a ejemplo.com</legend>

<!-- ... -->
</fieldset>
</form>
```

Su uso es necesario para la semántica web y la accesibilidad web, por lo que siempre debería estar presente, aunque no esté visible.

7.2.8 Elemento meter

El elemento **METER** especifica que el contenido que se va a representar es una medición escalar, es decir, como una barra de progreso con rango conocido o valor fraccional.

Además de los comentados al comienzo de este apartado, entre los atributos que admite en su configuración, se deben destacar los siguientes:

| Atributo | Descripción |
|----------------|---|
| high | Especifica el valor a partir del cual se considera que es un valor alto, siendo este, igual o menor que el valor del atributo max . |
| low | Especifica el valor a partir del cual se considera que es un valor bajo, siendo este, igual o menor que el valor del atributo min . |
| max | Especifica el valor máximo que puede aceptar o se puede introducir en el elemento. |
| min | Especifica el valor mínimo que puede aceptar o se puede introducir en el elemento. |
| optimum | Especifica el valor a partir del cual se considera que es un valor optimo, siendo este, mayor que el valor del atributo min y menor que el atributo max . |

```
<meter id="available-space" value="324" min="0" max="1024">  
324MB libres de 1024 MB  
</meter>
```

Cabe destacar que, aunque pueden parecerse, una medición escalar no es una barra de progreso. Por esta razón, el elemento `METER` no debe utilizarse para mostrar valores de progreso. El correcto uso de este elemento es para uso o capacidad de disco, para indicar el valor de tareas que tuvieron éxito en un conjunto definido o situaciones similares.

7.2.9 Elemento `optgroup`

El elemento `OPTGROUP` especifica que el contenido que se va a representar es un grupo de opciones relacionadas de un elemento `SELECT`.

Además de los comentados al comienzo de este apartado, hay que destacar:

| Atributo | Descripción |
|--------------|---|
| label | Especifica la etiqueta que se mostrará para diferenciar el grupo de opciones relacionadas y sólo será aplicable para los elementos optgroup que sean descendientes directos de un elemento select . |

```
<select id="motorcycles">  
<optgroup label="Harley Davidson">  
<option value="01">Sportster</option>  
<option value="02">Softail</option>  
<option value="03">Touring</option>  
</optgroup>  
<optgroup label="Indian">  
<option value="04">Chief</option>  
<option value="05">Springfield</option>  
<option value="06">Scout Sixty</option>  
</optgroup>  
</select>
```

7.2.10 Elemento `option`

El elemento `OPTION` especifica que el contenido que se va a representar es una opción para elemento `SELECT`.

Además de los comentados al comienzo de este apartado, entre los atributos que admite en su configuración, se deben destacar los siguientes:

| Atributo | Descripción |
|-----------------|---|
| label | Especifica la etiqueta que se mostrará para diferenciar el grupo de opciones relacionadas y sólo será aplicable para los elementos optgroup que sean descendientes directos de un elemento select . |
| selected | Especifica si la opción debe marcarse como seleccionada. |

```
<select id="rate">
<option value="01">Mala</option>
<option value="02">Media</option>
<option value="03">Buena</option>
</select>
```

7.2.11 Elemento output

El elemento **OUTPUT** especifica que el contenido que se va a representar es el resultado de una operación. Esto puede ser una buena opción a implementar cuando se trata de mostrar el resultado de una operación en dónde el usuario introduce varios valores a través de elementos **INPUT**.

Además de los comentados al comienzo de este apartado, hay que destacar:

| Atributo | Descripción |
|------------|--|
| for | Especifica los id de los elementos de formulario con los que se operará para mostrar el resultado en el elemento output . |

```
<form oninput="res.value=parseInt(op1.value) + parseInt(op2.value)">
<label>Suma de dos operandos</label>

<input type="range" id="op1" value="50">100
+
<input type="range" id="op2" value="50">
=
<output name="res" for="op1 op2">100</output>
</form>
```

7.2.12 Elemento progress

El elemento **PROGRESS** especifica que el contenido que se va a representar es una barra de progreso.

Además de los comentados al comienzo de este apartado, hay que destacar:

| Atributo | Descripción |
|------------|--|
| max | Especifica el valor máximo que puede aceptar o se puede introducir en el elemento. |

```
<form oninput="res.value=parseInt(op1.value) + parseInt(op2.value)">0
<label>Progreso de instalación</label>
<progress id="progress" value="54" max="100"> 54% </progress>
</form>
```

Cabe destacar que, aunque puedan parecerse, una barra de progreso no es una medición escalar. Por esta razón, el elemento **PROGRESS** no debe utilizarse para mostrar valores indicadores. El correcto uso de este elemento es únicamente para mostrar cómo progresa, o ha progresado, una tarea o proceso.

7.2.13 Elemento select

El elemento `SELECT` especifica que el contenido que se va a representar es un desplegable con una lista de opciones predefinida.

Además de los comentados al comienzo de este apartado, entre los atributos que admite en su configuración, se deben destacar los siguientes:

| Atributo | Descripción |
|-----------------|---|
| multiple | Especifica que el elemento admite la selección o inserción de más de un valor. |
| required | Especifica que el elemento debe contener valor, aunque este sea un espacio en blanco, es decir, que no puede ser nulo ni vacío. |
| size | Especifica el ancho en caracteres que debe tener el elemento. |

```
<select id="rate">
<option value="01">Mala</option>
<option value="02">Media</option>
<option value="03">Buena</option>
</select>
```

7.2.14 Elemento textarea

El elemento `TEXTAREA` especifica que el contenido que se va a representar es una caja de texto con la opción de multilínea, es decir, un control de entrada de datos de múltiples líneas.

Además de los comentados al comienzo de este apartado, entre los atributos que admite en su configuración, se deben destacar los siguientes:

| Atributo | Descripción |
|--------------------|--|
| cols | Especifica la anchura, en caracteres, del elemento. |
| maxlength | Especifica la longitud máxima, en caracteres, del valor que puede ser ingresado en el elemento. |
| placeholder | Especifica el texto que se debe mostrar como pista de datos válidos cuando el elemento no contenga un valor. En general es una buena idea usarlo para ayudar a la usabilidad y accesibilidad web y, si va asociado con otro atributo como pattern , la pista a proporcionar debe estar en concordancia con la validez de la entrada. |
| readonly | Especifica que el elemento no permitirá la inserción de nuevos valores ni la modificación de su valor actual. En cuanto esté presente, e independientemente de su valor, provocará que se ponga en modo de sólo lectura, no obstante, permitirá las acciones de seleccionar, resaltar o copiar. |

| | |
|-----------------|--|
| required | Especifica que el elemento debe contener valor, aunque este sea un espacio en blanco, es decir, que no puede ser nulo ni vacío. |
| rows | Especifica la altura del elemento en líneas de caracteres. |
| wrap | <p>Especifica si el texto debe incluir o no la definición de nuevas líneas cuando se realiza el envío del formulario para que se ajuste al ancho del elemento establecido.</p> <p>Sus posibles valores son hard y soft, los cuales indican que se agreguen nuevas líneas al texto enviado o no, respectivamente.</p> <p>El carácter de nueva línea añadido al texto será en base al valor del atributo cols, es decir, cuando el texto llegue a la longitud establecida por cols, agregará o no un nuevo salto de línea en función de si el atributo wrap está o no establecido a hard.</p> <p>El uso de este atributo no se recomienda si se desea mantener un buen nivel de usabilidad y accesibilidad web puesto que puede provocar pérdidas en la legibilidad de los datos si no se utiliza adecuadamente.</p> |

```
<textarea id="desc" rows="24" cols="80"></textarea>
```

7.3 ELEMENTOS DISPONIBLES EN CSS

CSS no presenta ninguna propiedad específica para elementos de formulario, sin embargo, cabe destacar que estos elementos pueden aprovecharse de las mismas ventajas que cualquier otro elemento de bloque o caja. Es decir, pueden ser personalizados con propiedades de todo tipo, como son FONT-FAMILY, TEXT-ALIGN, POSITION, DISPLAY, WIDTH, HEIGHT, PADDING, MARGIN, BORDER, BACKGROUND, ...

No obstante, entre todas ellas, cabe destacar dos:

7.3.1 Propiedad box-sizing

Especifica cómo deben asignarse y calcularse el alto y ancho de los elementos. Esto es, si deben incluir los márgenes internos (padding) y/o los bordes, o no. Entre sus posibles valores podemos encontrar **CONTENT-BOX**, que es el valor por defecto e indica que se debe incluir sólo el contenido, e ignorar los márgenes internos y bordes y, **BORDER-BOX**, que indica que se deben incluir el contenido, padding y bordes.

NOTA

En general, se puede afirmar que, trabajar con cajas o capas incluyendo los márgenes internos y los bordes es más fácil de manejar y facilita el diseño adaptativo, aunque no siempre.

7.3.2 Propiedad resize


Especifica si los elemento TEXTAREA deben permitir la manipulación del tamaño del elemento desde la interfaz que presenta el agente de usuario. Sus posibles valores son **NONE**, que indica que el usuario no puede cambiar el tamaño del elemento, **HORIZONTAL**, que indica que el usuario puede cambiar, únicamente, el tamaño del elemento horizontalmente, **VERTICAL**, que indica que el usuario puede cambiar, únicamente, el tamaño del elemento verticalmente y **BOTH**, que indica que el usuario puede cambiar el tamaño del elemento en ambas direcciones.

NOTA

Aunque esta propiedad puede resultar muy útil, su soporte está bastante limitado. De hecho, no está soportado por Microsoft Edge 78 o inferior, ni Opera 14 o inferior.

```
textarea { resize: none; }
```

7.4 PRACTICA Y JUEGA

| Prueba de formulario | Código QR |
|---|---|
| <p>Averigua qué elementos de formulario no se deberían solicitar en la página registro del siguiente ejemplo.</p> <p>https://codepen.io/pefc/full/PodjamM</p> |  |