

HTML

2.7 Etiquetas de formulario



Autor: Lisa ERIKSEN

Fecha: 2025 / 2026

1. Descubrimos los formularios HTML

Los formularios son elementos esenciales en **HTML5** que nos permiten recopilar la información introducida por los usuarios.

1.1. Las etiquetas `<form>` y `<input>`

Aquí tienes un ejemplo de formulario:

```
<form method="get">
  <label for="email" >Email:</label>
  <input type="email" name="email">
  <label for="tel" >Teléfono:</label>
  <input type="tel" name="tel">
</form>
```

Reto nº1: Estas desarrollando una aplicación con acceso seguro. Tu tarea es crear un formulario de **login** para que el usuario pueda iniciar sesión.

The diagram shows a rounded rectangular form with an orange border. At the top center, it says "Log in". Below that are two input fields: the first contains "johndoe@gmail.com" and the second contains "*****". At the bottom is a teal-colored button with the text "Log in" in white.

Requisitos:

- El formulario debe tener **un campo para el nombre y otro para el correo del usuario.**
Pista: usa `<input type="text">` y `<input type="password">`
- El formulario debe tener **un campo para la contraseña.**
Pista: usa `<input type="password">` para ocultar los caracteres.
- Debe haber **un botón de envío** que permita enviar el formulario.
Pista: usa `<button type="submit">`
- Tienes que usar las etiquetas `<form>`, `<label>`, `<input>` y `<button>`.

Prueba:

1. Escribe un correo electrónico incorrecto (por ejemplo, sin @) y observa qué ocurre.
2. Verifique que la contraseña no se muestre en pantalla.

1.2. Atributos de la etiqueta <form> para enviar los datos

Ejemplo de los atributos que puedes usar a dentro de la etiqueta <form>:

```
<form name="miFormulario" action="/next-page" method="post"></form>
```

Aquí están unos atributos de <form>:

Atributo	Valores	Descripción
action		Define la URL donde se enviará la información recopilada.
method	GET: Se utiliza para enviar datos en la URL. POST: Envío más seguro y adecuado para datos sensibles.	Especifica el método HTTP utilizado para enviar los datos
name		Asigna un nombre único al formulario.
target	_self, _blank, _parent, _top	Define dónde se abrirá el formulario (nueva pestaña, ventana actual, etc.).
accept-charset	UTF-8, otros	Define la codificación de caracteres para los datos del formulario.
autocomplete	on, off	Habilita o deshabilita la función de autocompletar.

Reto nº2: Mejora tu formulario de login y recupera tus datos! En tu formulario de login, añade atributos de arriba en la etiqueta <form> para que:

- Al hacer clic en el botón, veas toda la información que pusiste en los campos a dentro de la URL
Pista: investiga el atributo method
- Al clic en Enviar, se abre una nueva pagina en otra pestaña con el mensaje:
"Suscripción exitosa!"
Pista: crea una nueva pagina "redireccion.html" en tu carpeta con el mensaje y añade dos atributos en la etiqueta <form> de tu login.

Prueba

1. Escribe datos en los campos e intenta enviar el formulario.
2. Observa la URL: ¿ves tus datos?
3. Comprueba si la nueva página se abre en otra pestaña.
4. ¿Qué pasa si olvidas el atributo name?
5. Cambia temporalmente method="post": ¿qué cambia?

Preguntas de reflexión:

- ¿Qué hace el atributo name en los <input>?
- ¿Qué tipo de botón se usa para enviar los datos del formulario?



¿Que tengo que saber?

1. Cómo estructurar un formulario utilizando la etiqueta <form>.
2. Los principales atributos y configuraciones de los formularios.
3. Los diferentes tipos de entradas que podemos incluir con la etiqueta <input>.
4. La importancia del atributo name para el procesamiento de datos.

2. Los campos de texto <input> y <textArea>

Reto nº3: Estás creando una comunidad online (de música, deporte...). Tu tarea es diseñar un formulario de presentación para que un nuevo miembro pueda darte información básica.

Bienvenido a la comunidad

Cuentanos quién eres y preséntate brevemente.

Nombre de usuario

Ej.: LaRojaFan99

2-10 caracteres. El navegador puede autocompletar.

Mensaje personal

Cuéntanos algo sobre ti: ¿tu equipo favorito? ¿por qué te unes?

Enviar

Requisitos:

- El primer campo « nombre de usuario »
 - tiene una cantidad minima de 2 caracteres y maxima de 10 caracteres
 - autocompleta el texto con información previamente introducida*Pista: Busca minlength, maxlength, autocomplete*
- El campo de presentación es mas largo y puede contener varias lineas.
 Pista: busca textArea
- Lo dos campos deben mostrar un texto de sugerencia para orientar al usuario.
 Pista: busca placeholder



¿Que tengo que saber?

1. Los diferentes tipos de inputs de texto y sus usos específicos.
2. Los atributos que mejoran la funcionalidad de input (autocomplete, placeholder, spellcheck)
3. Cómo utilizar <textarea> para textos largos.
4. La validación de longitud con minlength y maxlength.

3. Descubrimos los campos numéricos

3.1. Campos básicos

Reto nº4: Estás diseñando el menú de configuración de tu propio videojuego. Necesitas permitir que el jugador elija el **nivel inicial de energía** de su personaje. Crea un formulario con un campo para seleccionar el **nivel de energía inicial**.



Requisitos:

- Debe ser un campo que **solo permita números**.
Pista: busca el tipo de <input> para numero
- La energía debe ser mayor que 0 y menor que 10.
Pista: busca los atributos min y max de un input.
- El jugador no debe poder escribir cualquier número, solo valores **en incrementos de 5** (por ejemplo: 5, 10, 15, 20 ...).
Pista: investiga el atributo step.

Prueba:

1. Intenta escribir **texto** dentro del campo numérico.
2. Usa las **flechas** que aparecen en el borde del input.
3. Escribe un número **decimal** (por ejemplo, 12.5).
4. Intenta escribir un número **que no sea múltiplo de 5** (como 7) y envía el formulario.

3.2. El control deslizante (range)

Reto nº4: Ahora vas a mejorar tu formulario para controlar el nivel de “Reputación en la calle” de tu personaje usando un control deslizante.



Requisitos:

- Crea un control deslizante para la “Reputación” que se mueva como una barra.
Pista: busca el <input type=" range">.
- Personaliza los límites del control deslizante usando también min, max y step.
- Incluye un botón para enviar el formulario y observar los valores elegidos en la URL.

Prueba:

1. Mueve la barra y observa cómo cambia la posición del control.
2. Envía el formulario y revisa el valor en la **URL**.
3. Modifica los valores de min, max y step para **personalizar la escala** (por ejemplo, reputación mínima 0, máxima 200, de 10 en 10).



¿Que tengo que saber?

1. Los diferentes tipos de inputs para recolectar datos numéricos: number y range.
2. Cómo usar los atributos min, max y step para validar y personalizar los valores numéricos permitidos.
3. Ejemplos prácticos de validaciones comunes en inputs numéricos.

4. Campos de fecha y hora

Reto nº5: Estas desarrollando una aplicación de registración para un centro medico. Tu tarea es de crear formulario para **reservar una cita médica**.

The screenshot displays a user interface for scheduling medical appointments. At the top, it says "Formulario de Citas Médicas". Below that is a section labeled "Agenda" showing a calendar for November 2025. The date "11/05/2025" is selected. The calendar highlights the 5th of November. To the right of the calendar, there is a time selection grid for Wednesday, November 05, showing slots at 07:00, 08:00, 09:00, 12:00, 13:00, and 14:00. The entire interface is framed by a red border.

Requisitos:

- La fecha de la cita debe ser entre hoy y dentro de tres meses.
Pista: busca varios tipos de input (date, time) y usa min y max
- La hora de la cita debe entre 08:00 y 17:00 en pasos de una hora.
Pista: usa step
- El botón de envío que permite enviar el form y recuperar los datos en la URL
- Tienes que usar las etiquetas <form>, <label>, <input> y <button>

Prueba:

1. Intenta seleccionar una fecha fuera del rango permitido.
2. Intenta introducir una hora fuera del rango permitido.
3. Cambia step="1800" y observa la diferencia.



¿Que tengo que saber?

1. Los diferentes tipos de inputs para fechas y horas: date, month, time, y week.
2. Cómo restringir valores válidos mediante los atributos min, max, y step.
3. La utilidad de estos inputs para mejorar la experiencia del usuario y garantizar datos válidos.

5. Botones, checkbox y radio

Reto nº6: Tu tarea es crear un **formulario de quiz deportivo** (por ejemplo, sobre fútbol, baloncesto u otro deporte) que incluya **preguntas de selección múltiple y única**.

The screenshot shows a mobile application interface for a football quiz. At the top, there's a yellow header with the title "Quiz de Fútbol" and a trophy icon. Below the header, there are four numbered questions with multiple-choice options and two buttons at the bottom.

- 1. ¿Qué selecciones ganaron la Copa del Mundo?**

 - Brasil
 - Alemania
 - España
 - México

- 2. ¿En qué país se jugó el Mundial de 2022?**

 - Catar
 - Rusia
 - Brasil

- 3. ¿Qué jugadores ganaron el Balón de Oro?**

 - Lionel Messi
 - Cristiano Ronaldo
 - Kylian Mbappé
 - Luka Modrić

- 4. ¿Cuánto dura un partido de fútbol profesional?**

 - 60 minutos
 - 90 minutos
 - 120 minutos

At the bottom of the screen, there are two buttons: "Enviar respuestas" (Send answers) and "Reiniciar" (Reset).

Requisitos:

- Crea una pregunta con **varias respuestas posibles**

Usa los inputs checkbox con el mismo atributo **name** y un atributo **value** para permitir seleccionar varias opciones a la vez.

- Crea una pregunta con **una sola respuesta correcta**

Usa los inputs radio con el mismo atributo **name** y un atributo **value** para que solo se pueda elegir una sola opción.

- Añade un **botón de envío** que permita enviar el formulario y mostrar las respuestas en la URL y verifique que tus respuestas están en la URL.

Pista: usa `method="get"` y el `type="submit"`

- Añade también un **botón para reiniciar** todas las respuestas del quiz.

Pista: busca el `type="reset"`

Prueba

1. Intenta marcar varias opciones en la pregunta con checkbox.
2. Intenta seleccionar más de una opción en la pregunta con radio (¿qué pasa?).
3. Haz clic en "Enviar" y verifique que los datos en la URL coinciden con los atributos **value** de tus respuestas.
4. Pulsa "Reiniciar" y comprueba el resultado.

Preguntas de reflexión:

- ¿Qué diferencia hay entre los checkbox y los radio buttons?
- ¿Por qué es importante el atributo **name** y **value** en los checkbox y radio?
- ¿Qué función tiene el atributo **checked**?
- ¿Qué tipo de botón se usa para enviar o reiniciar los datos?



¿Que tengo que saber?

1. Cómo usar botones para realizar operaciones en un formulario.
2. Los checkbox, sus atributos, y cómo agruparlos para seleccionar múltiples opciones.
3. Los radio, su uso para opciones excluyentes y cómo configurarlos correctamente.