

## 2.4. Formularios y elementos de entrada



### **LENGUAJES DE PROGRAMACIÓN NIVEL 3. UNIDAD 2**

HTML - LENGUAJE DE MARCAS PARA LA ESTRUCTURA DE CONTENIDO

## Contenido

Contenido .....	2
Introducción .....	3
Fundamentos de los formularios HTML .....	5
Introducción a los formularios.....	5
Estructura básica de un formulario .....	7
action: .....	7
method: .....	8
Construcción de un formulario .....	9
Asociación entre <label> y <input> .....	9
Mejores prácticas .....	10
Elementos de entrada comunes .....	11
Campos de texto (<input type="text">).....	11
Contraseñas (<input type="password">) .....	12
Botones (<button type="submit">, <button>).....	12
Casillas de verificación (<input type="checkbox">) y botones de radio (<input type="radio">) .....	13
Controles avanzados y estructuración .....	14
Listas desplegables (<select> y <option>).....	15
Áreas de texto (<textarea>).....	16
Agrupación de elementos (<fieldset> y <legend>).....	17
Accesibilidad y mejores prácticas.....	18
Etiquetado de elementos (<label>).....	18
Prácticas recomendadas en diseño de formularios .....	18
Conclusión .....	20
Recursos adicionales .....	21
Documentación oficial y tutoriales .....	21
Herramientas de diseño y validación de formularios .....	21

## Introducción

***Tras haber explorado cómo las imágenes, enlaces y listas enriquecen el contenido y la estructura de nuestras páginas web, nos adentramos ahora en uno de los aspectos más interactivos y fundamentales del desarrollo web: los formularios. Estos no son solo mecanismos para recopilar datos; son puentes de comunicación entre tus usuarios y tú, facilitando una interacción dinámica que va desde la suscripción a boletines hasta la realización de compras en línea.***

*Los formularios HTML son esenciales para capturar la voz del usuario, recopilando información en una variedad de formatos y contextos. Ya sea que estemos registrando nuevos usuarios, recogiendo opiniones o procesando pedidos, los formularios nos permiten recolectar esos datos de manera estructurada y significativa.*

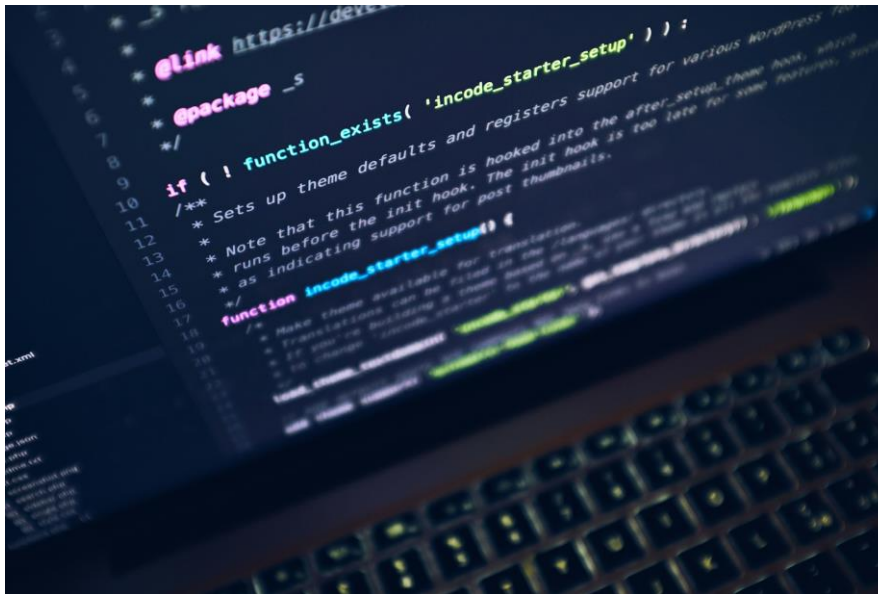
*Este módulo te guiará a través de los componentes clave que constituyen los formularios web, desde los campos de texto básicos hasta los elementos de entrada más complejos. Exploraremos cómo estructurar un formulario de manera efectiva, implementar tipos de entrada variados para diferentes necesidades de datos, y asegurarnos de que nuestros formularios no solo sean funcionales, sino también accesibles y amigables para el usuario.*

*Abordaremos la validación de formularios para mejorar la experiencia del usuario y mantener la integridad de los datos recopilados, y te daremos consejos sobre las mejores prácticas en diseño de formularios que garantizan su eficacia y usabilidad.*

*Prepárate para sumergirte en el mundo de los formularios HTML, donde la funcionalidad se encuentra con la interactividad, y cada elemento de entrada se convierte en una oportunidad para dialogar con tus usuarios.*



## Fundamentos de los formularios HTML



Los formularios son uno de los pilares de la interactividad y la funcionalidad en la web. Actúan como interfaces para que los usuarios introduzcan y envíen datos, que luego pueden ser procesados por el servidor o utilizados de manera dinámica en la propia página. Su versatilidad permite una amplia gama de aplicaciones, desde la realización de búsquedas hasta el envío de información personal.

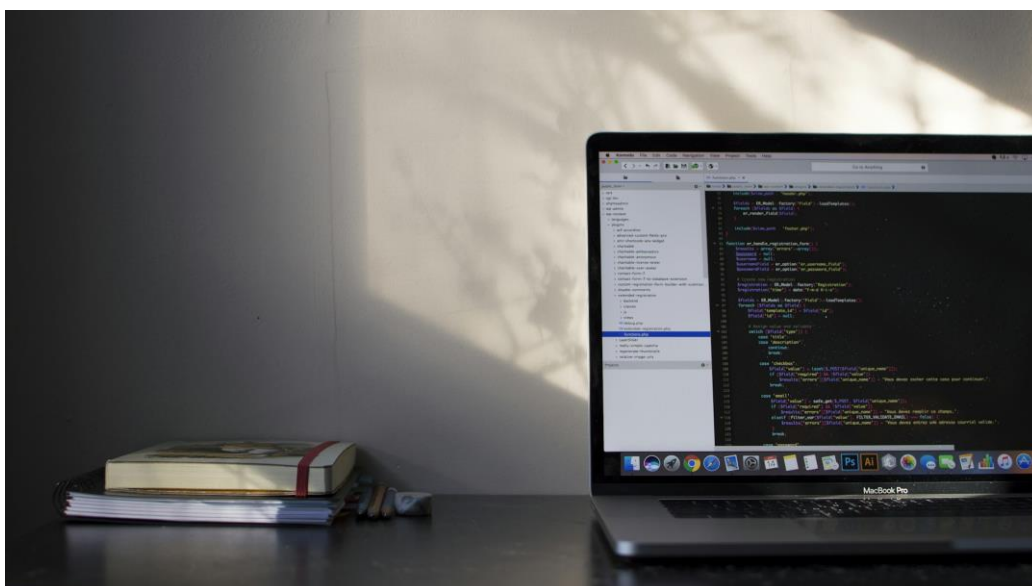
### Introducción a los formularios

Un formulario en HTML es, esencialmente, una colección de campos de entrada que permiten a los usuarios ingresar datos. Estos datos pueden ser texto, números, selecciones de opciones, marcas de verificación y más. Una vez que un usuario completa y envía un formulario, la información ingresada se envía al servidor para su procesamiento. Los formularios son fundamentales para tareas comunes en la web, como:

- **Registros y Logins:** Donde los usuarios ingresan datos personales para acceder a áreas restringidas o servicios personalizados.
- **Búsquedas:** Permitiendo a los usuarios introducir términos de búsqueda para encontrar información específica.
- **Envíos:** Como la publicación de comentarios, la carga de archivos, o la realización de pedidos en línea.

La habilidad para diseñar formularios eficientes y amigables es crucial para la experiencia del usuario, ya que una interfaz de formulario bien diseñada puede fomentar la interacción y mejorar la percepción general de un sitio web.

## Estructura básica de un formulario



La etiqueta `<form>` es el contenedor para todos los elementos de un formulario. Define el espacio donde los campos de entrada y los botones de envío se agrupan y establece cómo se enviarán los datos.

Los atributos más importantes de un `<form>` son:

### action:

Especifica el destino del servidor al que se enviarán los datos del formulario una vez que el usuario lo envíe. Este destino suele ser un script o una página capaz de procesar la información, como un archivo PHP, ASP o una URL de una API.

```
<form action="submit.php">
```

1

---

<sup>1</sup> Accesibilidad:  
`<form action="submit.php">`

## method:

Define cómo se enviarán los datos al servidor. Los métodos más comunes son GET, donde los datos se adjuntan a la URL y son visibles, y POST, donde los datos se envían en el cuerpo de la solicitud y no son visibles en la URL, siendo más seguro para datos sensibles.

```
<form action="submit.php" method="POST">
```

2

Un formulario básico puede ser tan simple como esto, pero su verdadera utilidad proviene de los elementos de entrada (<input>, <select>, <textarea>, etc.) que contiene, que son los que te permiten introducir la información que contendrá el formulario, los cuales se detallarán en las siguientes secciones.

---

<sup>2</sup> Accesibilidad:

<form action="submit.php" method="POST">



## Construcción de un formulario



Dentro del HTML de un formulario, las etiquetas `<label>` y `<input>` suelen usarse juntas para crear una interfaz de usuario clara y accesible.

- La etiqueta `<label>` proporciona una descripción o identificación del campo de entrada que se va a llenar,
- la etiqueta `<input>` define el propio campo de entrada donde los usuarios pueden introducir sus datos.

### Asociación entre `<label>` y `<input>`

Hay dos maneras principales de asociar una etiqueta `<label>` con un campo `<input>`:

- Usando el Atributo `for` en `<label>`: El atributo `for` en la etiqueta `<label>` debe coincidir con el atributo `id` del campo `<input>`. Esta es la forma recomendada de asociar una etiqueta con un campo de entrada, ya que mantiene la estructura del HTML limpia y clara.

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

3

<sup>3</sup> Accesibilidad:

`<label for="nombre">Nombre:</label>`

- Encapsulando el Campo `<input>` Dentro de `<label>`: Colocar el campo `<input>` directamente dentro de la etiqueta `<label>` también asocia ambos elementos. Esta técnica puede ser útil en ciertas situaciones de diseño o para simplificar el marcado.

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

4

## Mejores prácticas

- Siempre usa `<label>`: Asegúrate de proporcionar una etiqueta para cada campo de entrada en tus formularios para mejorar la usabilidad y accesibilidad.
- Mantén las asociaciones claras: Usa el atributo `for` y `id` para asociar claramente las etiquetas con los campos de entrada, especialmente en formularios más complejos o extensos.

La combinación de las etiquetas `<label>` y `<input>` en los formularios HTML es una práctica estándar que mejora significativamente la experiencia del usuario y la accesibilidad de los formularios web.

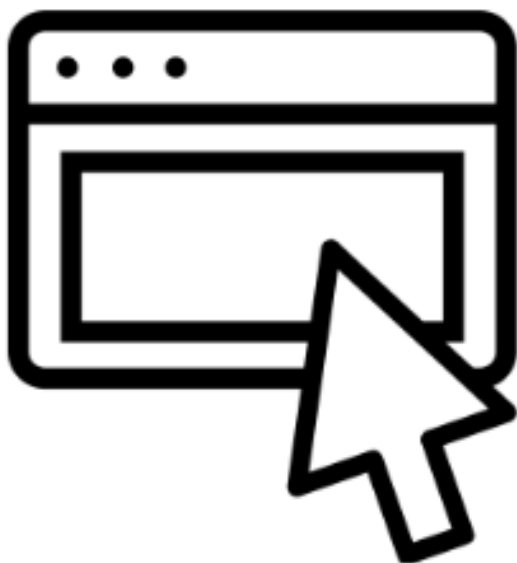
---

```
<input type="text" id="nombre" name="nombre">
```

<sup>4</sup> Accesibilidad:

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

## Elementos de entrada comunes



Los elementos de entrada son las herramientas mediante las cuales los usuarios interactúan con los formularios. Cada tipo de elemento de entrada está diseñado para recopilar un tipo específico de dato, desde textos simples hasta selecciones complejas.

Exploraremos ahora los elementos de entrada más comunes que encontrarás al construir formularios en HTML.

### Campos de texto (<input type="text">)

Los campos de texto son probablemente el tipo más básico y utilizado de elementos de entrada. Se utilizan para recopilar datos de texto cortos, como nombres, direcciones de correo electrónico o cualquier otra información que requiera una respuesta abierta y breve del usuario.

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

5

---

<sup>5</sup> Accesibilidad

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

Es recomendable acompañar cada campo de texto con una etiqueta `<label>` para mejorar la accesibilidad y proporcionar claridad sobre la información que se solicita.

### Contraseñas (`<input type="password">`)

Los campos de contraseña ocultan el texto que el usuario introduce, mostrando en su lugar símbolos como asteriscos (\*) o puntos (•). Esto es esencial para mantener la confidencialidad de datos sensibles como contraseñas y PINs.

```
<label for="contrasena">Contraseña:</label>
<input type="password" id="contrasena" name="contrasena">
```

6

Es crucial implementar medidas de seguridad adicionales en el lado del servidor para manejar las contraseñas de forma segura, como el uso de hash y salting.

### Botones (`<button type="submit">`, `<button>`)

Los botones son elementos interactivos que permiten al usuario realizar acciones como enviar el formulario, reiniciar campos o ejecutar scripts personalizados.

El elemento `<button>` ofrece más flexibilidad que `<input type="button">`, ya que puede contener contenido HTML, como imágenes o texto enriquecido.

```
<button type="submit">Enviar</button>
<button type="reset">Limpiar</button>
```

7

---

<sup>6</sup> Accesibilidad

```
<label for="contrasena">Contraseña:</label>
<input type="password" id="contrasena" name="contrasena">
```

<sup>7</sup> Accesibilidad:

```
<button type="submit">Enviar</button>
<button type="reset">Limpiar</button>
```

## Casillas de verificación (<input type="checkbox">) y botones de radio (<input type="radio">)

Las casillas de verificación permiten a los usuarios hacer selecciones múltiples de un conjunto de opciones, ideal para listas donde más de una opción puede ser relevante.

```
<label><input type="checkbox" name="interes" value="musica"> Música</label>
<label><input type="checkbox" name="interes" value="libros"> Libros</label>
```

8

Los botones de radio, por otro lado, son para selecciones exclusivas dentro de un grupo, asegurando que solo una opción pueda ser seleccionada a la vez.

```
<label><input type="radio" name="genero" value="masculino"> Masculino</label>
<label><input type="radio" name="genero" value="femenino"> Femenino</label>
```

9

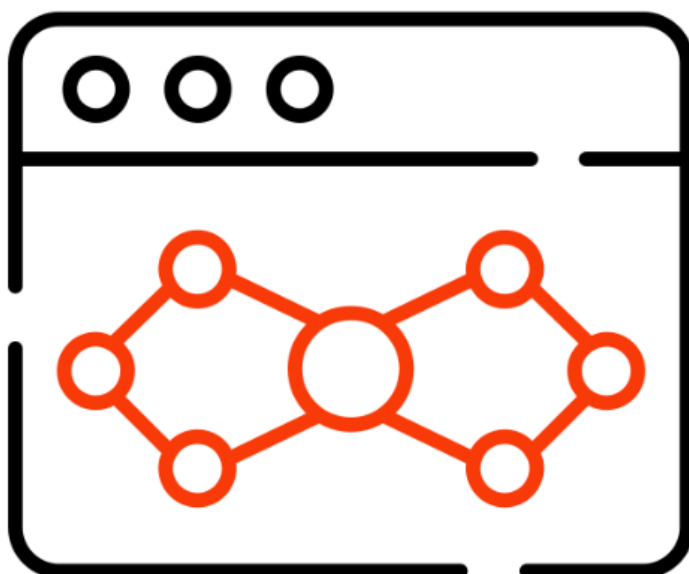
<sup>8</sup> Accesibilidad:

```
<label><input type="checkbox" name="interes" value="musica"> Música</label>
<label><input type="checkbox" name="interes" value="libros"> Libros</label>
```

<sup>9</sup> Accesibilidad:

```
<label><input type="radio" name="genero" value="masculino"> Masculino</label>
<label><input type="radio" name="genero" value="femenino"> Femenino</label>
```

## Controles avanzados y estructuración



A medida que los formularios se vuelven más complejos y recopilan una variedad más amplia de información, es esencial utilizar controles avanzados y técnicas de estructuración para mejorar la experiencia del usuario y la coherencia de los datos. En esta sección, exploraremos cómo implementar listas desplegables, áreas de texto y cómo agrupar elementos de manera lógica.

## Listas desplegables (<select> y <option>)

Las listas desplegables son una manera eficiente de presentar una lista de opciones sin abrumar al usuario. Se utilizan comúnmente cuando el espacio es limitado o cuando deseas presentar una lista de selección única o múltiple en un formato más compacto.

```
<label for="pais">País:</label>
<select id="pais" name="pais">
  <option value="">Seleccione un país...</option>
  <option value="espana">España</option>
  <option value="mexico">México</option>
  <option value="argentina">Argentina</option>
</select>
```

10

El elemento <select> crea el menú desplegable, mientras que cada <option> representa una elección disponible dentro de la lista. La atribución de un valor a cada <option> es crucial para procesar los datos del formulario.

<sup>10</sup> Accesibilidad:

```
<label for="país">País:</label>
<select id="país" name="pais">
  <option value="">Seleccione un país... </option>
  <option value="espana">España</option>
  <option value="mexico">México</option>
  <option value="argentina">Argentina</option>
</select>
```

## Áreas de texto (<textarea>)

Para recopilar entradas más extensas, como comentarios, descripciones o cualquier texto largo, las áreas de texto <textarea> son la solución ideal. A diferencia de los campos de texto <input>, que están diseñados para entradas breves, <textarea> permite entradas multilínea.

```
<label for="comentario">Comentario:</label>
<textarea id="comentario" name="comentario" rows="4" cols="50">
Escribe algo aquí...
</textarea>
```

11

Los atributos rows y cols pueden usarse para ajustar el tamaño del <textarea>, aunque también se puede redimensionar con CSS para adaptarse mejor al diseño de tu formulario.

---

<sup>11</sup> Accesibilidad:

```
<label for="comentario">Comentario:</label>
<textarea id="comentario" name="comentario" rows="4" cols="50">
Escribe algo aquí:
</textarea>
```



## Agrupación de elementos (<fieldset> y <legend>)

A medida que los formularios crecen en complejidad, organizarlos lógicamente se vuelve esencial para mantener la claridad. El elemento <fieldset> se utiliza para agrupar elementos relacionados dentro del formulario, y <legend> proporciona un título o explicación para el grupo.

```
<fieldset>
  <legend>Información de contacto</legend>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email">
  <label for="telefono">Teléfono:</label>
  <input type="tel" id="telefono" name="telefono">
</fieldset>
```

12

Esta estructuración no solo mejora la accesibilidad, facilitando la navegación por el formulario, sino que también ayuda a los usuarios a entender qué grupos de información son relevantes entre sí.

<sup>12</sup> Accesibilidad:

```
<fieldset>
  <legend>Información de contacto</legend>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email">
  <label for="telefono">Teléfono:</label>
  <input type="tel" id="telefono" name="telefono">
</fieldset>
```

## Accesibilidad y mejores prácticas



Crear formularios accesibles y bien diseñados es esencial para ofrecer una experiencia de usuario inclusiva y efectiva. Esta sección se centra en la importancia de etiquetar correctamente los elementos de los formularios y proporciona una serie de mejores prácticas para el diseño de formularios.

### Etiquetado de elementos (<label>)

Tal y como ya se ha comentado, el etiquetado adecuado de los controles de formulario es crucial para la accesibilidad. Las etiquetas <label> ofrecen una descripción clara de la función de cada campo de entrada, lo que es especialmente importante para usuarios que dependen de tecnologías de asistencia, como lectores de pantalla.

### Prácticas recomendadas en diseño de formularios

Para garantizar que los formularios no solo sean accesibles sino también intuitivos y fáciles de usar, considera las siguientes mejores prácticas:

- **Organización Lógica:** Agrupa los campos relacionados y ordénalos de manera lógica. Utiliza <fieldset> y <legend> para agrupaciones más grandes y para proporcionar contexto.
- **Instrucciones Claras:** Proporciona instrucciones claras y concisas para cada campo, especialmente si se requiere un formato específico o hay restricciones en los datos que se pueden ingresar.

- **Feedback Instantáneo:** Implementa la validación del lado del cliente para proporcionar retroalimentación inmediata sobre errores o campos incompletos, mejorando así la experiencia general del usuario.
- **Facilitar la Navegación:** Asegúrate de que los formularios se puedan navegar fácilmente con el teclado, lo cual es esencial para usuarios con limitaciones de movilidad o que prefieren no usar un ratón.
- **Usar Marcado Semántico:** Aprovecha los elementos de HTML5 como `<input type="email">`, `<input type="tel">`, etc., para que los navegadores y las tecnologías asistivas manejen los campos de manera apropiada.
- **Pruebas de Accesibilidad:** Utiliza herramientas de evaluación de accesibilidad y pruebas con usuarios reales, incluyendo aquellos que dependen de tecnologías asistivas, para identificar y corregir problemas.

## Conclusión

A lo largo de este módulo, hemos navegado por el universo de los formularios HTML, descubriendo cómo los elementos de entrada, la validación y las mejores prácticas en diseño contribuyen a crear interacciones significativas y accesibles en la web. Hemos aprendido que los formularios son mucho más que simples medios para recopilar datos; son, en esencia, una conversación entre el usuario y la página web, una oportunidad para interactuar, participar y comunicarse.

La importancia de diseñar con una mente puesta en la accesibilidad y la usabilidad no puede ser subestimada. Cada decisión que tomamos en el diseño de un formulario, desde la elección de los elementos de entrada hasta la implementación de la validación y la retroalimentación, tiene el poder de incluir o excluir a los usuarios.

*En el próximo módulo "Semántica y Accesibilidad en HTML", profundizaremos en cómo el uso adecuado de la semántica HTML no solo estructura nuestro contenido de manera lógica y significativa, sino que también mejora la accesibilidad.*

**Te espero.**

## Recursos adicionales

Para seguir profundizando en tu comprensión de los formularios HTML y mejorar tus habilidades en diseño y validación, aquí tienes algunos recursos y herramientas gratuitas que te serán de gran ayuda:

### Documentación oficial y tutoriales

- MDN Web Docs: Ofrece guías exhaustivas y documentación de referencia sobre HTML, perfectas para entender en profundidad el uso de imágenes, enlaces y listas.  
<https://developer.mozilla.org/es/>
- W3Schools: Proporciona tutoriales accesibles y ejemplos prácticos que son ideales para iniciarse y reforzar los conocimientos en HTML.  
<https://www.w3schools.com/html/>

### Herramientas de diseño y validación de formularios

- JotForm: Una herramienta en línea que permite crear formularios HTML personalizados sin necesidad de codificación, útil para entender la estructura y el diseño de los formularios. (<https://www.jotform.com/es/>)
- The W3C Markup Validation Service: Un servicio gratuito que permite comprobar la validez del marcado de tus formularios y páginas HTML, asegurando que sigan los estándares web. (<https://validator.w3.org/>)