

# Programador Full Stack - TSDWAD - 2022

[Área personal](#) / [Mis cursos](#) / [Programador Full Stack - TSDWAD - 2022](#) / [Guía HTML](#)

## Guía HTML

Hecho: Ver



### 4. Elementos y Etiquetas HTML

#### 4.7. Formularios

Cualquier página HTML puede mejorarse con formularios interactivos que piden al visitante que rellene información: introducir texto, seleccionar opciones, confirmar con un botón... **¡cualquier cosa es posible!**

Sin embargo, estamos alcanzando los límites del lenguaje HTML, ya que después deberemos analizar la información que el visitante ha introducido... y eso no se puede hacer en HTML. Como vamos a ver, los resultados tienen que procesarse en otro lenguaje, como PHP, por ejemplo.

Mientras tanto, tenemos un gran número de etiquetas HTML por descubrir. **Bienvenido al maravilloso mundo de los formularios.** *Un mundo en el que los botones, las casillas de verificación y las listas desplegadas coexisten en armonía.*

#### Crear un formulario

Cuando tienes la repentina necesidad de insertar un formulario en tu página HTML, necesitas empezar escribiendo una etiqueta `<form>` `</form>`. Esta es la etiqueta principal de los formularios, y especifica su comienzo y su fin.

```
1 <p>Texto antes del formulario</p>
2
3 <form>
4   <p>Texto dentro del formulario</p>
5 </form>
6
7 <p>Texto después del formulario</p>
```

Debes tener en cuenta que debes introducir etiquetas de bloque (como `<p>` `</p>`) dentro del formulario si quieres incluir texto en él.

Veamos un ejemplo para clarificar las cosas. Supón que un visitante acaba de incluir un comentario en el formulario, como, por ejemplo, un mensaje que les gustaría publicar en el foro. Este mensaje tiene que *enviarse* para que puedas recibirlo (lógico, ¿verdad?) y mostrarlo a otros visitantes.

Bien, ese es el problema o, más bien, esos son los problemas, que vamos a abordar:

- **Problema n.º 1:** ¿Cómo enviar el texto introducido por el visitante? ¿De qué manera?
- **Problema n.º 2:** una vez que los datos se han enviado, ¿Cómo los procesas? ¿quieres recibir el mensaje automáticamente mediante correo electrónico, o prefieres un programa que lo guarde en algún sitio y después lo muestre en una página que todo el mundo lo pueda ver?

Para aportar las soluciones a estos dos problemas, tienes que añadir dos atributos a la etiqueta `<form>`:

- **method:** este atributo especifica cómo se van a enviar los datos (solución al **problema n.º 1**). Hay dos maneras de enviar datos a la Web:
  - **method="get":** este método normalmente es menos adecuado, ya que se limita a 255 caracteres. Esta particularidad proviene del hecho de que la información se enviará a la dirección de la página (`http://...`, aunque este detalle en realidad no tiene importancia por el momento. Te recomiendo que utilices la mayoría de las veces el otro método: `post`.
  - **method="post":** es el método más usado en formularios, puesto que permite que se envíe un montón de información. Los datos introducidos en el formulario no van a través de la barra de direcciones.

- **action:** es la dirección de la página o programa que va a *procesar* la información (solución al **problema n.º 2**). Esta página te enviará un correo electrónico con el mensaje si eso es lo que quieres o, por otro lado, guardará el mensaje con todos los demás mensajes en una base de datos. Esto no se puede hacer en HTML y CSS, así que normalmente utilizamos otro lenguaje. Ej. java, php, python, etc.

Por lo tanto, ahora vamos a completar la etiqueta `<form>` con los dos atributos que acabamos de ver.

Para `method`, lo has adivinado, vamos a introducir el valor `post`.

Para `action`, vamos a introducir el nombre de una página ficticia en PHP (`tratamiento.php`). Esta es la página que será solicitada cuando el visitante haga clic en el botón de envío en el formulario.

```
1 <p>Texto antes del formulario</p>
2
3 <form method="post" action="tratamiento.php">
4   <p>Texto dentro del formulario</p>
5 </form>
6
7 <p>Texto después del formulario</p>
```

Por el momento, no abordaremos lo que pasa dentro en `tratamiento.php`. Asumimos que esta página está lista y funcionando.

Nuestra prioridad, por ahora, es descubrir en HTML/CSS cómo insertar campos de texto, botones y casillas de verificación en nuestra página web. Y eso es lo que vamos a ver ahora.

## Campos de entrada básicos

Por lo tanto, regresemos a algo tangible.

Vamos a revisar varias etiquetas HTML que necesitamos para introducir texto en un formulario.

Deberías saber que hay dos campos de texto diferentes.

- **El campo de texto de una línea:** como su nombre sugiere, solo puedes introducir una única línea en él. Se utiliza para textos cortos, por ejemplo, un nombre de usuario.
- **El campo de texto de líneas múltiples:** este campo de texto permite introducir una gran cantidad de texto en varias líneas, por ejemplo, una tesis sobre el uso de HTML en el desarrollo de países del Sudeste de Asia (tan solo una sugerencia...).

### Campo de texto de una línea

Esta imagen a continuación muestra el aspecto de un campo de texto de una línea.

\* Nombre usuario

Un campo de texto de una línea

Para insertar un campo de texto en una línea, vamos a utilizar la etiqueta `<input />`.

Para insertar un campo de texto en una línea, vamos a utilizar la etiqueta `<input />`.

Encontraremos esta etiqueta varias veces más adelante del capítulo. En cada caso, el que cambia es el valor de `type`.

Para crear un campo de texto de una línea, tienes que introducir:

```
1 <input type="text" ></code>
```

Para dar un nombre a un elemento de formulario, se utiliza el atributo `name`. En este caso, vamos a asumir que al visitante se le ha pedido que vuelva a introducir su nombre de usuario:

```
1 <input type="text" name="nombre" ></code>
```

Así que vamos a intentar crear un formulario muy básico con nuestro campo de texto:

```

1 <form method="post" action="tratamiento.php">
2   <p><input type="text" name="username" /></p>
3 </form>

```

## Etiquetas

Este campo de texto está muy bien, pero si tu visitante se lo encuentra, no sabrá lo que poner. Esta es exactamente la labor de la etiqueta <label>:

```

1 <form method="post" action="tratamiento.php">
2   <p>
3     <label>Nombre usuario</label> : <input type="text" name="username" />
4   </p>
5 </form>

```

Este código proporciona exactamente el mismo resultado que has visto en la imagen anterior.

Pero no es suficiente. Hay que enlazar la etiqueta con el campo de texto.

*¿Un name y un id en el campo? ¿No sería redundante?*

Sí, de cierta manera pero no supone ningún problema.

Para enlazar la etiqueta con el campo, se le tiene que dar un atributo for, que tiene el mismo valor que el campo id... Lo mejor es verlo con un ejemplo:

```

1 <form method="post" action="tratamiento.php">
2   <p>
3     <label for="username">Nombre usuario</label> : <input type="text" name="username" id="username"
4     />
5   </p>
6 </form>

```

## Algunos atributos adicionales

Puedes añadir varios atributos adicionales a la etiqueta <input/> para personalizar su funcionamiento:

- Puedes aumentar el tamaño del campo con size.
- Puedes limitar el número de caracteres que pueden introducirse mediante maxlength.
- Puedes prerrellenar el campo con un valor por defecto con value.
- Puedes dar algún indicio de los contenidos del campo mediante placeholder. Esta indicación desaparecerá tan pronto como el visitante haga clic dentro del campo.

En el siguiente ejemplo, el campo de texto contiene una indicación para entender lo que debería escribirse; aunque el campo mide 30 caracteres, no puedes escribir más de 10 dentro:

```

1 <form method="post" action="tratamiento.php">
2   <p>
3     <label for="username">Nombre usuario:</label>
4     <input type="text" name="username" id="username" placeholder="Zozor" size="30" maxlength="10"
5     />
6   </p>
7 </form>

```

En el navegador:

Nombre usuario

Un campo de texto con un indicador (marcador de posición)

## Campo de contraseña

Puedes hacer fácilmente que el campo de texto se comporte como un «campo de contraseña», es decir, un campo donde no puedes ver en la pantalla los caracteres que se escriban. Para crear este tipo de campo de entrada, utiliza el atributo `type="password"`.

Completaré mi formulario. Ahora le pide al visitante que escriba su nombre de usuario y su contraseña:

```

1 <form method="post" action="tratamiento.php">
2   <p>
3     <label for="username">Tu nombre de usuario:</label>
4     <input type="text" name="username" id="username" />
5
6     <br />
7     <label for="pass">Tu contraseña:</label>
8     <input type="password" name="pass" id="pass" />
9
10  </p>
11 </form>

```

Verás que los caracteres de la contraseña no se visualizan en pantalla, como se muestra en la siguiente imagen.

Nombre usuario

Contraseña

Un campo de entrada de contraseña

## Campo de texto de líneas múltiples

Para crear un campo de texto de líneas múltiples, cambia la etiqueta: vamos a utilizar `<textarea>` `</textarea>`.

Como cualquier otro elemento de formulario, tienes que proporcionarle un nombre mediante `name` y utilizar `label` para explicar qué es.

```

1 <form method="post" action="tratamiento.php">
2   <p>
3     <label for="mejora">¿Cómo crees que podría mejorar mi página web?</label><br />
4     <textarea name="mejora" id="mejora"></textarea>
5   </p>
6 </form>

```

¡Y aquí está la imagen con el resultado (siguiente captura)!

¿Cómo crees que podría mejorar mi página web?

Un campo de entrada pequeño de líneas múltiples

Como puedes ver, **¡es un poco pequeño!** Afortunadamente, puedes cambiar el tamaño de `<textarea>` de dos maneras distintas.

- **En CSS:** tan solo aplica las propiedades CSS `width` y `height` a `<textarea>`.
- **Con atributos:** puedes añadir los atributos `rows` y `cols` a la etiqueta `<textarea>`. El primero especifica el número de líneas de texto que pueden mostrarse simultáneamente y el segundo, el número de columnas.

¿Por qué abres la etiqueta `<textarea>` y la cierras de nuevo inmediatamente después?

Puedes prerrellenar `<textarea>` con un valor por defecto. En este caso, no utilizamos el atributo `value`: simplemente escribimos el texto por defecto entre la etiqueta de apertura y la de cierre.

```

1 <form method="post" action="tratamiento.php">
2   <p>
3     <label for="mejora">
4       ¿Cómo crees que podría mejorar mi página web?
5     </label>
6     <br />
7
8     <textarea name="mejora" id="mejora" rows="10" cols="50">
9       ¡¿Mejorar tu página web?!
10      ¡Venga! ¡Es tan perfecta que no necesita ninguna mejora!
11    </textarea>
12  </p>
13 </form>

```

La siguiente imagen muestra el resultado.

¿Cómo crees que podría mejorar mi página web?

¡¿Mejorar tu página web?!

¡Venga! ¡Es tan perfecta que no necesita ninguna mejora!

Un campo de entrada de líneas múltiples completado previamente

### Campos de entrada mejorados

HTML5 incluye muchas características nuevas en los formularios. De hecho, han afluado nuevos tipos de campos con esta versión. Tan solo dale al atributo type de la etiqueta <input/> uno de los nuevos valores disponibles. ¡Echemos un vistazo rápido!

*Todos los navegadores todavía no reconocen estos campos de entrada mejorados. En su lugar, los navegadores antiguos mostrarán un campo de texto simple de una sola línea (como si hubieras escrito type="text"). Que quede entre nosotros, no pasa nada: los nuevos navegadores pueden disfrutar de las últimas funciones, mientras que los navegadores antiguos muestran un campo de texto de sustitución que encaja igual de bien.*

**¡Así que te recomendamos usar estos nuevos campos de entrada desde ahora!** En el mejor de los casos, tus visitantes se beneficiarán de estas nuevas funciones y, en el peor, no verán ningún problema.

### Correo electrónico

Puedes pedir a los visitantes que introduzcan una dirección de correo electrónico:

```
1 <input type="email" ></code>
```

El campo *normalmente* te parecerá el mismo, pero el navegador ahora sabe que el usuario tiene que escribir una dirección de correo electrónico. Puede dar una indicación si la dirección no es un correo electrónico, que es lo que hace Firefox (ejemplo a continuación).

dsqfdsdfsp

Un correo electrónico escrito incorrectamente aparece enmarcado en rojo.

### Una URL

Con el tipo url, puedes pedirle al visitante que escriba una dirección absoluta (normalmente comienza con http://):

```
1 <input type="url" ></code>
```

Mismo principio: si el campo no parece diferente en tu ordenador, ten en cuenta que ha entendido eficazmente que se supone que el visitante debe escribir una dirección.

### [Número de teléfono](#)

Este campo se usa para escribir números de teléfono.

### [Número](#)

Este campo se utiliza para escribir un número entero:

```
1 <input type="number" ></code>
```

El campo normalmente se muestra con unas pequeñas flechas para cambiar el valor (imagen siguiente).

Campo de entrada de números

Puedes personalizar la manera en la que funciona el campo con los siguientes atributos:

- min: valor mínimo permitido.
- max: valor máximo permitido.
- step: es el desplazamiento. Si especificas un «step» de 2, el campo solo aceptará valores de 2 en 2 (por ejemplo: 0, 2, 4, 6, etc.).

### [Fecha](#)

Hay varios tipos de campos de selección de fecha:

- date: para la fecha (05/08/1985, por ejemplo);
- time: para la hora (13:37, por ejemplo);
- week: para la semana;
- month: para el mes;
- datetime-local para la fecha y la hora (sin administración de zona horaria).

```
1 <input type="date" ></code>
```

## Manejo de Opciones

HTML incluye un montón de opciones que pueden usarse en el formulario. Son elementos que requieren que el visitante elija a partir de una lista de posibilidades. Vamos a repasar:

- casillas de verificación;
- campos de opciones;
- listas desplegables.

### [Casillas de verificación](#)

¿Cómo se crea una casilla de verificación? ¡No podría ser más fácil! Vamos a utilizar la etiqueta `<input />`, pero esta vez especificando el tipo checkbox:

```
1 <input type="checkbox" name="opciones" ></code>
```

¡Añade `<label>` en la posición correcta, ¡y ya lo tienes!

```

1 <form method="post" action="tratamiento.php">
2   <p>
3     Marca las comidas que te gustan:<br />
4     <input type="checkbox" name="patatas fritas" id="patatas fritas" /> <label for="patatas
patatas fritas">Patatas fritas</label><br />
5     <input type="checkbox" name="hamburguesa" id="hamburguesa" /> <label
for="hamburguesa">Hamburguesa</label><br />
6     <input type="checkbox" name="espinacas" id="espinacas" /> <label
for="espinacas">Espinacas</label><br />
7     <input type="checkbox" name="ostras" id="ostras" /> <label for="ostras">Ostras</label>
8   </p>
9 </form>

```

La siguiente imagen muestra el resultado.

Marca las comidas que te gustan:

- ☐ Patatas Fritas
- ☐ Hamburguesa
- ☐ Espinacas
- ☐ Ostras

Casillas de verificación

No olvides dar un nombre diferente a cada casilla de verificación. Más tarde, te permitirá identificar las que ha seleccionado el visitante.

Por último, recuerda que puedes tener una casilla marcada por defecto con el atributo checked:

```

1 <input type="checkbox" name="opciones" checked /></code>

```

Todos los atributos normalmente tienen un valor. Sin embargo, en ese caso, añadir un valor no es necesario: la presencia del atributo es suficiente para hacer que el ordenador entienda que la casilla tiene que estar marcada.

Si te molesta, recuerda que puedes asignar cualquier valor al atributo (algunos administradores de páginas algunas veces escriben checked="checked", pero es un poco redundante). La casilla se marcará sin orden.

## Botones de radio

Los campos de opciones te permiten elegir una (y solamente una) opción de una lista de posibilidades. De alguna manera, parecen casillas de verificación pero hay una ligera dificultad adicional: tienen que estar organizadas como grupos. Aunque las opciones en el mismo grupo tienen el mismo (nombre), cada opción tienen que tener un (valor) diferente.

La etiqueta que se utiliza siempre es <input />, esta vez con el valor radio en el atributo type.

El siguiente ejemplo lo aclarará:

```

1 <form method="post" action="tratamiento.php">
2   <p>
3     Indica el grupo de edad al que perteneces:<br />
4     <input type="radio" name="edad" value="menos15" id="menos15" /> <label for="menos15">Menos de
15 años</label><br />
5     <input type="radio" name="edad" value="entre15-25" id="entre15-25" /> <label for="entre15-
25">Entre 15 y 25 años</label><br />
6     <input type="radio" name="edad" value="entre25-40" id="entre25-40" /> <label for="entre25-
40">Entre 25 y 40 años</label><br />
7     <input type="radio" name="edad" value="mas40" id="mas40" /> <label for="mas40">Incluso mayor que
eso?</label>
8   </p>
9 </form>

```

Lo que nos da el resultado de la siguiente imagen.



Indica el grupo de edad al que perteneces:

- ☐ Menos de 15 años
- ☐ Entre 15 y 25 años
- ☐ Entre 25 y 40 años
- ☐ Incluso mayor que eso?

Botones de radio

¿Por qué pusiste el mismo nombre para cada opción? Solo para que el navegador sepa a qué «grupo» pertenece el botón.

Intenta eliminar los atributos name y verás que entonces puedes seleccionar todas las opciones. Sin embargo, no es lo que queremos, por esa razón los «enlazamos» al darles el mismo nombre.

Te darás cuenta de que esta vez hemos elegido una id diferente al name. De hecho, dado que los valores de name son los mismos, no podríamos haber diferenciado las ids (y sabes muy bien que una id tiene que ser única!). Por esta razón hemos elegido dar a id el mismo valor que a value.

Si tienes dos campos de opciones diferentes, tienes que dar un name único a cada grupo, de esta manera:

```
1 <form method="post" action="tratamiento.php">
2   <p>
3     Indica el grupo de edad al que perteneces:<br />
4     <input type="radio" name="edad" value="menos15" id="menos15" /> <label for="menos15">Menos de
5     15 años</label><br />
6     <input type="radio" name="edad" value="entre15-25" id="entre15-25" /> <label for="entre15-
7     25">Entre 15 y 25 años</label><br />
8     <input type="radio" name="edad" value="entre25-40" id="entre25-40" /> <label for="entre25-
9     40">Entre 25 y 40 años</label><br />
10    <input type="radio" name="edad" value="mas40" id="mas40" /> <label for="mas40">¿Incluso mayor?
11    </label>
12  </p>
13  <p>
14    ¿En qué continente vives?<br />
15    <input type="radio" name="continente" value="europa" id="europa" /> <label
16    for="europa">Europa</label><br />
17    <input type="radio" name="continente" value="africa" id="africa" /> <label
18    for="africa">África</label><br />
19    <input type="radio" name="continente" value="asia" id="asia" /> <label for="asia">Asia</label>
20    <br />
21    <input type="radio" name="continente" value="america" id="america" /> <label
22    for="america">América</label><br />
23    <input type="radio" name="continente" value="australia" id="australia" /> <label
24    for="australia">Australia</label>
25  </p>
26 </form>
```

## Listas desplegables

Las listas desplegables son otra manera elegante de seleccionar una opción a partir de varias posibilidades. Funcionan de una manera ligeramente distinta. Vamos a utilizar la etiqueta <select> </select>, que especifica el comienzo y el final de la lista desplegable. Añadimos el atributo name a la etiqueta para darle un nombre a la lista.

Después, dentro de la etiqueta <select> </select>, vamos a insertar varias etiquetas <option> </option> (una para cada opción posible). Añadimos un atributo value a cada uno de ellas para identificar lo que el visitante ha elegido.

Aquí podéis ver un ejemplo de uso:



```
1 <form method="post" action="tratamiento.php">
2   <p>
3     <label for="pais">¿En qué país vives?</label><br />
4     <select name="pais" id="pais">
5       <option value="estadosunidos">Estados Unidos</option>
6       <option value="canada">Canadá</option>
7       <option value="reinounido">Reino Unido</option>
8       <option value="francia">Francia</option>
9       <option value="espana">España</option>
10      <option value="italia">Italia</option>
11      <option value="china">China</option>
12      <option value="japon">Japón</option>
13    </select>
14  </p>
15 </form>
```

El resultado se muestra en la siguiente imagen.

¿En qué país vives?



Lista desplegable

Si quieres que una opción esté seleccionada por defecto, esta vez utiliza el atributo selected:


```
1 <option value="canada" selected>Canadá</option>
```

También puedes agrupar las opciones con la etiqueta `<optgroup>` `</optgroup>`. En nuestro ejemplo, *¿por qué no separas países según su continente?*

```
1 <form method="post" action="tratamiento.php">
2   <p>
3     <label for="pais">¿En qué país vives?</label><br />
4     <select name="pais" id="pais">
5       <optgroup label="Europa">
6         <option value="reinounido">Reino Unido</option>
7         <option value="francia">Francia</option>
8         <option value="espana">España</option>
9         <option value="italia">Italia</option>
10      </optgroup>
11      <optgroup label="América">
12        <option value="estadosunidos">Estados Unidos</option>
13        <option value="canada">Canadá</option>
14      </optgroup>
15      <optgroup label="Asia">
16        <option value="china">China</option>
17        <option value="japon">Japón</option>
18      </optgroup>
19    </select>
20  </p>
21 </form>
```

El resultado se muestra en la siguiente imagen.

¿En qué país vives?



| Continent | Countries                            |
|-----------|--------------------------------------|
| Europe    | United Kingdom, France, Spain, Italy |
| America   | United States, Canada                |
| Asia      | China, Japan                         |

Opciones agrupadas por continente

Los grupos no pueden seleccionarse. Así que no podemos elegir «Europa» en nuestro ejemplo: solo los nombres de países pueden seleccionarse.

```

1 <form method="post" action="tratamiento.php">
2
3 <fieldset>
4   <legend>Tus datos de contacto</legend> <!-- Título del conjunto de campos -->
5
6   <label for="apellidos">¿Cuáles son tus apellidos?</label></br>
7   <input type="text" name="apellido" id="apellido" /></br>
8
9   <label for="nombre">¿Cuál es tu nombre?</label></br>
10  <input type="text" name="nombre" id="nombre" /></br>
11
12  <label for="email">¿Cuál es tu dirección de correo electrónico?</label></br>
13  <input type="email" name="email" id="email" /></br>
14
15 </fieldset>
16
17 <fieldset>
18   <legend>Tu deseo</legend> <!-- Título del conjunto de campos -->
19
20   <p>
21     Pide un deseo que quieres que se haga realidad:</br>
22
23     <input type="radio" name="deseo" value="rico" id="rico" /> <label for="rico">Ser
rico</label></br>
24     <input type="radio" name="deseo" value="famoso" id="famoso" /> <label for="famoso">Ser
famoso</label></br>
25     <input type="radio" name="deseo" value="inteligente" id="inteligente" /> <label
for="inteligente">Ser <strong>aún más</strong> inteligente</label></br>
26     <input type="radio" name="deseo" value="otro" id="otro" /> <label for="otro">Otro...
</label></br>
27   </p>
28
29   <p>
30     <label for="specs">Si has elegido «Otro», especifica cuál:</label></br>
31     <textarea name="specs" id="specs" cols="40" rows="4"></textarea>
32   </p>
33 </fieldset>
34 </form>

```

El resultado se muestra en la siguiente imagen.

### Tus datos de contacto

¿Cuales son tus apellidos?

¿Cual es tu nombre?

¿Cual es tu correo electrónico?

### Tu deseo

Pide un deseo que quieres que se haga realidad:

☐ Ser rico

☐ Ser famoso

Los campos están agrupados

## Seleccionar un campo automáticamente

Puedes situar el cursor automáticamente en uno de los campos del formulario con el atributo autofocus. Una vez que el visitante cargue la página, el cursor aparece en este campo.

Por ejemplo, para situar el cursor por defecto en el campo name:

```
1 <input type="text" name="nombre" id="nombre" autofocus ></code>
```

## Hacer un campo obligatorio

Puedes hacer un campo obligatorio al asignarle el atributo required.

```
1 <input type="text" name="nombre" id="nombre" required ></code>
```

Entonces el navegador le indicará al usuario, si el campo está vacío cuando el formulario sea enviado, que necesita completarlo.

## El botón de envío

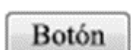
Así que ahora lo que nos queda por hacer es crear el botón de envío. De nuevo, la etiqueta `<input />` viene a nuestro rescate. Hay cuatro versiones:

- `type="submit"`: el botón de envío principal del formulario. Este es el que usarás más a menudo. El usuario será redireccionado a la página especificada en el atributo action del formulario.
- `type="reset"`: restablece el formulario
- `type="image"`: equivalente al botón submit, pero esta vez mostrado como una imagen. Añade el atributo src para especificar la URL de la imagen.
- `type="button"`: botón genérico, que no tendrá efecto alguno (por defecto). En general, este botón se maneja en JavaScript para desarrollar acciones en la página. No lo utilizaremos aquí.

Puedes cambiar el texto mostrado dentro de los botones con el atributo value.

Para crear un botón de envío, entonces escribimos, por ejemplo:

Lo que nos da el resultado de la siguiente imagen.



Un botón de envío

Cuando haces clic en el botón «Enviar», entonces el formulario te lleva a la página especificada en el atributo action. Recuerda, imaginamos una página ficticia: tratamiento.php.

*El problema es que no puedes crear esta página exclusivamente en HTML. Tienes que aprender un nuevo lenguaje, como PHP, para ser capaz de «recuperar» la información introducida y decidir lo que hacer con ella.*


Para finalizar, te invitamos a mirar un video explicativo:

HTML - Formularios

Usted se ha identificado como Claudio PLAZA (Cerrar sesión)

Resumen de retención de datos

Descargar la app para dispositivos móviles



**Desafío:** En base a lo visto en esta sección, intenta crear el siguiente formulario:

Registro de Programador

Nombre y Apellido

nombre

Email

mail

Teléfono

Su teléfono

Lenguajes que conoce

Angular

☒

Java

☐

NodeJS

☐

Python

☐

Otro

Nivel de Inglés

Bajo

☒

Medio

☐

Alto

☐

Enviar



< [Guía Introducción a Full Stack](#)

Ir a...

[Guía CSS](#) >