



GOGODEV / @JJRuizEmpresa



Autor: Juan José Ruiz a.k.a GOGODEV

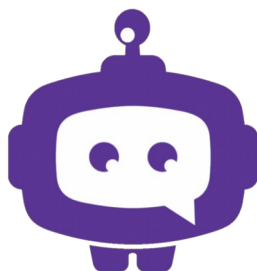
YouTube: <https://youtube.com/c/gogodev>

Twitter: <https://twitter.com/JJRuizEmpresa> (@JJRuizEmpresa)

IMPORTANTE

El contenido de este pdf lo tienes también en vídeo en el canal de YouTube <https://youtube.com/c/gogodev>, dentro de la lista de reproducción: Curso de HTML.

Todo el contenido formativo es gratuito, no olvides pasarte y suscribirte para no perderte nada, y así nos apoyas :)





GOGODEV / @JJRuizEmpresa

<CAPÍTULO .04>

DIME LO QUE PIENSAS

CONSTRUCCIÓN DE FORMULARIOS PARA INTERACTUAR CON EL USUARIO.

</CAPÍTULO .04>

<https://youtube.com/c/gogodev>



DIME LO QUE PIENSAS

<FORMULARIOS>

Hasta ahora, todas las etiquetas que hemos ido aprendiendo dentro de HTML estaban destinadas a mostrar información al usuario, manteniendo una comunicación unidireccional. Es hora de incorporar a nuestras construcciones todo el conjunto de herramientas asociadas a los formularios, que permitirán al usuario introducir datos y, por tanto, comunicarse con su entorno web.

Con los formularios no solo podemos hacer secciones de contacto, si no cajas de comentarios, editores de contenido, subir imágenes y mucho más.

A lo largo de este capítulo, vamos a desglosar su conjunto de etiquetas más importantes. Pongámonos manos a la obra:

Cada vez que vayamos a definir una zona de interacción donde vamos a recoger datos que el usuario va a tener que introducir, sean estos del tipo que sean, deberemos crear un formulario valiéndonos de la etiqueta **<form>**. Posteriormente, dentro de esta etiqueta, anidaremos a su vez las etiquetas de todos aquellos elementos que conformarán dicho formulario: Un campo de introducción de texto, una caja de comentarios, unas cajas de selección...

```
<form>
  <!-- Contenido del formulario -->
</form>
```

Una vez tenemos colocada la etiqueta, existen dos atributos esenciales que deben acompañar siempre a la etiqueta: El **method** y el **action**.

Comencemos por el action:

Este, es un atributo que indica la ruta donde se encuentra el archivo que va a procesar los datos. Y es que, cuando el usuario cumplimenta y envía unos datos, nuestro sistema debe ser capaz de recepcionar y procesar los datos recibidos, además de actuar en consecuencia.

El procesado y tratamiento de datos lo veremos más adelantes en cursos futuros del canal, en los que nos dedicaremos a trabajar el *backend* de nuestras aplicaciones web, que es la parte de la programación dedicada a tales efectos, y que estará escrito en un lenguaje de programación concreto. De momento, nos basta con saber que el **action** contiene la ruta donde se encuentra el archivo que procesará los datos recibidos.

El segundo de estos atributos es el **method**, que contiene un verbo que indica cómo viajarán los datos al fichero/función que los procesará. Existen diferentes verbos con particularidades distintas, que forman parte del temario de *backend* de la colección: GET, POST, PUT, DELETE.



De momento vamos a quedarnos con los dos principales:

- **GET:** Usamos este método cuando estamos enviando datos que NO son sensibles. Es decir, cuando la información que está escribiéndonos el usuario no es privada. Por ejemplo: una barra de búsqueda (una caja de buscar). Esto es debido a que cuando usamos GET la información viajará hacia el archivo de procesado dentro de la url, y por lo tanto es susceptible de ser leída por terceros.
- **POST:** Al contrario que ocurre con GET, POST encriptará la información en la petición, y no será visible por terceros. Usaremos esta opción cuando estamos enviando datos sensibles. Por ejemplo: un formulario de registro o acceso donde se está escribiendo la contraseña de acceso.

Así pues, la etiqueta **form** nos quedaría de la siguiente forma:

```
<form method="POST" action="miarchivo.php">  
  <!-- Contenido del formulario -->  
</form>
```

Ya conocemos cómo declarar una zona de formulario. Es hora de comenzar a cumplimentarla con los diferentes elementos de los que disponemos.

</FORMULARIOS>

<INPUTS>

Comencemos por uno de los elementos más comunes en los formularios: las cajas de texto. Seguro que, a lo largo de tu vida, en internet, has cumplimentado numerosas de ellas.

Para crear una caja de texto, usamos la etiqueta `<input>` y a dicha etiqueta, debemos darle un atributo `type`, para indicar qué tipo de caja de texto será.

Además, a todas las etiquetas que vayan dentro del formulario, tanto a las cajas de texto como las demás, deberemos darle una propiedad *name*, otorgándoles un nombre identificativo a cada una (consistente en una única palabra que no contenga tildes ni caracteres especiales. Podrá ser el que nosotros prefiramos) Esta no afecta al formulario, pero la parte de la web que se encarga de tratar los datos lo necesitará más adelante.

Puede parecer complicado, pero ya verás que es muy fácil. Veamos las diferentes opciones:

```
<input type="text" name="mi_caja_de_texto">
```

Vacía



El usuario puede escribir texto y números en ella

```
<input type="number" name="mi_caja_de_numeros">
```

Vacía

El usuario puede escribir sólo números en ella

```
<input type="password" name="caja_para_escribir_contrasenas">
```

Vacía

Lo que escribe el usuario aparece cifrado para que no pueda leerse

```
<input type="date" name="caja_para_seleccionar_fechas">  
<input type="date" name="caja_para_seleccionar_horas">
```



```
<input type="email" name="caja_para_escribir_email">
```

Vacía

El usuario solo puede introducir direcciones de email válidas en ella

```
<input type="file" name="mi_caja_para_subir_un_archivo">
```

Examinar... No se ha seleccionado ningún archivo.

Si usamos este tipo de input, destinado a subir archivos, al formulario deberemos indicárselo añadiendo a la etiqueta **form** el atributo **enctype** con el valor **"multipart/form-data"**:

```
<form method="miarchivo.php" action="POST" enctype="multipart/form-data">
```

Esto lo hacemos para que, al enviar la información, el formulario codifique correctamente la información del archivo.

¿Ves qué fácil? Ahora, deberíamos indicarle al usuario qué debe introducir en cada una de ellas. Para esto, tanto en las cajas `<input>` como en el resto de elementos de un formulario, podemos usar dos herramientas diferentes:

- La etiqueta **<label>**, para colocar un texto explicativo.
- El atributo **placeholder** en la propia etiqueta, para colocar un texto de ayuda dentro del input.

Veámoslo una vez más con un ejemplo:

```
<label>Email:</label>
<input type="email" name="email" placeholder="Ejemplo: john@doe.com">
```



Resultado:

Email:

</INPUTS>

<SELECTS>

Ya sabemos colocar en nuestros formularios cajas de textos para que el usuario introduzca información. Ahora vamos a conocer la etiqueta **select**. Esta, nos permitirá crear una caja con un menú desplegable, donde el usuario tendrá que seleccionar entre una de las opciones disponibles.

Para establecer un área de selección, marcamos la etiqueta **select**, a la cual le aplicaremos, al igual que los inputs y el resto de formularios, un atributo **name** con un nombre escogido por nosotros, para que posteriormente la información pueda ser tratada. Una vez hecho esto, anidaremos dentro del **select** diferentes etiquetas **option**, cada una de ellas con el texto de cada una de las opciones. Así mismo, las etiquetas **option** contarán con un atributo **value**, que será el valor que el formulario enviará al sistema como seleccionado. Esto hace que el valor que enviamos y el texto que se muestra al usuario puedan ser diferentes.

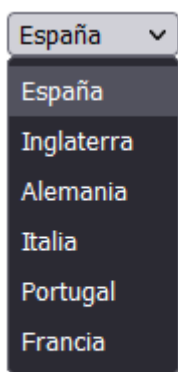
Veamos un ejemplo:

```
<select name="pais">
  <option value="es">España</option>
  <option value="en">Inglaterra</option>
  <option value="ge">Alemania</option>
  <option value="it">Italia</option>
  <option value="pt">Portugal</option>
  <option value="fr">Francia</option>
</select>
```

Resultado visual:



Y cuando pulsamos sobre el elemento:

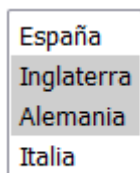


En este ejemplo, al pulsar sobre Italia, por ejemplo, el formulario enviará al sistema que el usuario en el campo **"pais"** ha seleccionado **"it"**.

Al campo select, además, podemos añadirle el atributo **multiple**. De esta forma, el usuario en lugar de poder seleccionar una única opción de entre las disponibles, podrá seleccionar múltiples de ellas.

```
<select name="pais" multiple>
  <option value="es">España</option>
  <option value="en">Inglaterra</option>
  <option value="ge">Alemania</option>
  <option value="it">Italia</option>
  <option value="pt">Portugal</option>
  <option value="fr">Francia</option>
</select>
```

Select con **multiple** en nuestra página web:



En este ejemplo, el usuario ha seleccionado Inglaterra y Alemania.

</SELECTS>



<RADIOS>

Continuamos avanzando con los diferentes elementos que podemos añadir a un formulario. En esta ocasión, vamos a ver los botones de radio. Los '**radio**' son botones circulares que podemos marcar o no. Normalmente, se usan para darnos a escoger una de entre diferentes opciones. Para ello, usamos la instrucción:

```
<input type="radio" id="arroz" name="comida_fav" value="Arroz">  
<label for="arroz">Arroz</label>
```

Esto, nos daría el siguiente resultado:

☐ Arroz

Donde la casilla circular puede ser o no marcada.

Detengámonos a analizar la instrucción:

```
input type="radio"
```

Con esta parte, indicamos que vamos a pintar un botón de radio.

```
id="arroz"
```

En HTML, a todas las etiquetas podemos darle un **id** con el valor que deseemos. En este caso, lo usamos para indicarle posteriormente a la etiqueta a qué botón de radio pertenece, a través de su atributo **for**.

```
name="comida_fav"
```

En los botones de radio, el **name**, además de servir al sistema para identificar el contenido que se le está enviando, también sirve para identificar los grupos de opciones. Por ejemplo, si tenemos cuatro botones de radio, los cuatro con el **name** de "**comida_fav**" el formulario los identificará como un grupo de opciones, y solo dejará marcar uno de ellos. Si el usuario trata de seleccionar un segundo, el primero se desmarcará.

```
value="Arroz"
```

Por último, tenemos el **value**. Al igual que ocurre con el resto de etiquetas de formulario, éste será el valor que se enviará al sistema si el usuario escoge dicha opción.

</RADIOS>



<CHECKBOX>

Pasamos ahora, de los botones de radio a su versión cuadrada, el **checkbox**. Estoy seguro que, navegando por internet te has encontrado con él en numerosas ocasiones, y es que el **checkbox** es esa cajita que marcamos cada vez que pulsamos sobre ese mensaje de "acepto los términos y condiciones". La forma de construirlo es muy similar a los botones de radio. Veamos cómo hacerlo:

```
<input type="checkbox" name="terminos" value="yes"><label>Acepto los términos y condiciones</label>
```

Resultado:

☐ Acepto los términos y condiciones

Como puedes observar, los atributos y fórmulas para construir los elementos de un formulario son siempre similares. En esta ocasión, indicamos que el input es de tipo **checkbox**, le damos un nombre y el valor que se enviará si lo marca.

Como aspecto especial, al **checkbox** podemos incluirle el atributo **checked**. Si lo hacemos, la casilla aparecerá marcada por defecto:

```
<input type="checkbox" name="terminos" value="yes" checked><label>Acepto los términos y condiciones</label>
```

Resultado:

☒ Acepto los términos y condiciones

</CHECKBOX>

<RANGOS>

Es hora de ver los rangos. Con ellos, podemos pintar una barra deslizadora para que el usuario la ajuste al valor numérico deseado. Para ello, expresaremos los atributos **min** y **max**, de cara a marcar el valor mínimo y máximo del rango respectivamente.

```
<input type="range" name="mi_porcentaje" min="0" max="100" />
```

Resultado:



</RANGOS>



<TEXTAREA>

A la hora de escribir textos, es posible que una única línea como nos ofrece el **input text** no sea suficiente. Para esas ocasiones en las que estamos esperando que el usuario introduzca una mayor cantidad de texto, como puede ser por ejemplo un comentario en una página de reseñas, usamos el área de texto. Para darle las dimensiones apropiadas a la caja, podemos usar los atributos **cols** y **rows**, columnas y filas respectivamente. Y al igual que ocurre con los **input text**, podemos aplicarle un **placeholder**. Por último, si deseamos que el texto contenga un mínimo de caracteres o un máximo, podemos usar los atributos de **minlength** (longitud mínima) y **maxlength** (longitud máxima)

Veámoslo con un ejemplo:

```
<textarea name="comentario" cols="30" rows="10" minlength="100" maxlength="500" placeholder="Escribe aquí tu comentario..."></textarea>
```

El resultado:

</TEXTAREA>

<VALIDACIÓN>

Por último, y antes de continuar con el envío, debemos saber cuáles de los campos son obligatorios que el usuario cumplimente, ya que en un formulario algunos de ellos pueden ser opcionales. Para ello, bastará con marcar con el atributo **required** a aquellos campos que vayan ser requeridos de forma obligatoria. El resto, quedarán automáticamente como opcionales por defecto. De esta forma, el navegador no dejará enviar el contenido si los campos marcados como requeridos no están debidamente rellenos, avisando al usuario de cuales de los no cumplimentados deberían estarlo, con el mensaje "debes rellenar este campo".

</VALIDACIÓN>



<ENVÍOS>

Ahora ya disponemos de un repertorio suficiente para poder construir nuestros formularios. Nos resta ahora, colocar el botón de enviar. Para ello, usaremos la etiqueta input **submit**, colocando el texto deseado del botón en el atributo **value**.

Veamos un ejemplo:

```
<input type="submit" value="Enviar" />
```

Resultado:



Y ahora sí, ya sabemos construir las estructuras HTML necesarias para que los usuarios puedan interactuar con nuestra página.

CONSEJO: Llegados a este punto, te puedes preguntar: ¿Son estas todas las etiquetas disponibles en HTML para formularios? La respuesta es un claro y rotundo no. Estas, son las más habituales, las que, en tu trabajo como desarrollador web, vas a estar escribiendo prácticamente a diario. Sin embargo, existen muchas otras. Por ello, es importante destacar, que NO debes aprenderte todas las etiquetas de memoria. Ningún desarrollador lo hace. A fuerza de usarlas, acabarás interiorizando las más comunes, pero no es posible ni recomendable tratar de conocerlas todas de memoria. El buen desarrollador siempre se preocupará de saber escribir buenas estructuras, de mantener el código limpio y de saber solucionar problemas. Para todas aquellas etiquetas más especiales que te puedan resultar de utilidad en momentos más puntuales, o para comportamientos que quieras aplicar que se salgan de la norma, recuerda consultar siempre la documentación oficial de la tecnología con la que estás trabajando. Esto, forma también parte del día a día de la profesión.

Para consultar etiquetas HTML, las páginas de documentación más usadas son:

- Mozilla: <https://developer.mozilla.org/>
- W3C: <https://www.w3schools.com/>

No obstante, no te preocupes por eso ahora. Debemos comenzar por el principio, por los cimientos, y ahora tienes un conjunto de herramientas suficientes para resolver casi cualquier necesidad en lo que a formularios se refiere. Es importante ponerlas en práctica y que comiences a familiarizarte con ellas.



Hora de practicar

Codifica el siguiente formulario. Trata de ser lo más específico posible con las etiquetas:

Registrar usuario

Nombre de usuario:

Email:

Contraseña:

Repite contraseña:

Fecha de nacimiento:

Cuenta algo sobre ti:

Cómo nos conociste:

Si quieres repasar en vídeo todo este bloque, o te has quedado atascado, puedes ver todo el proceso en el canal de YouTube GOGODEV.

Acabar de adquirir la capacidad de establecer comunicaciones bidireccionales en internet, y si bien ha sido tan solo el primer paso, será la piedra angular sobre la que cimentarás tu conocimiento. Tiempo al tiempo.

</ENVÍOS>