

# Introducción al Desarrollo de Sistemas

[Área personal](#) / [Mis cursos](#) / [IDS](#) / [Tema 3 - Formularios / Protocolo HTTP](#) / [Construcción de formularios](#)

## Construcción de formularios

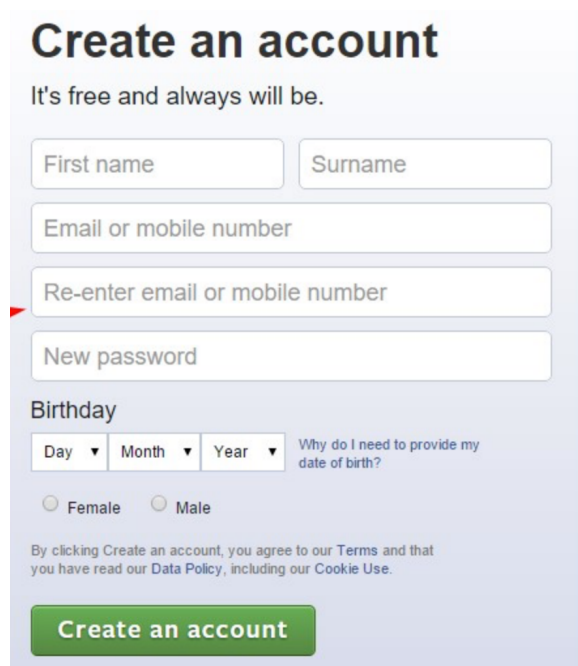
### Introducción

Los formularios Web, son un elemento esencial en el desarrollo de las aplicaciones Web.

Su funcionalidad es preparar la información que debe suministrar un usuario para que la aplicación cumpla una función determinada.

Hay numerosos ejemplos donde ésto puede visualizarse. Un caso prácticamente inevitable con el que interactuamos son los casos de inicio de sesión y/o registro en alguna página/sitio web.

Veamos el siguiente ejemplo de formulario:



The image shows a web form titled "Create an account" with the subtitle "It's free and always will be." The form includes several input fields: "First name" and "Surname" (side-by-side), "Email or mobile number", "Re-enter email or mobile number", and "New password". Below these is a "Birthday" section with dropdown menus for "Day", "Month", and "Year", and a link "Why do I need to provide my date of birth?". There are also radio buttons for "Female" and "Male". At the bottom, there is a green button labeled "Create an account". A small disclaimer text reads: "By clicking Create an account, you agree to our Terms and that you have read our Data Policy, including our Cookie Use."

En este caso, para que la funcionalidad de "Creación de cuenta" pueda efectuarse, el usuario debe introducir información como su nombre, apellido, email, etc etc. Cada uno de estos elementos de entrada de información, se los conoce como **campos del formulario**, y en el dominio de la jerga del código HTML, se los identifica como elementos **input**.

Es decir, que cada elemento de entrada de información, es a rasgos generales, un elemento **input**.

Y un **formulario**, es un contenedor de uno o más "**inputs**".

El standard de HTML ofrece una cantidad importante de elementos "input" de uso general clásico.

Más abajo referenciaré este material para que lo tengas en cuenta para desarrollar las prácticas.

Hay otras maneras también de construir inputs personalizados utilizando programación, aunque es un tema más avanzado y excede el nivel actual del curso.

De aquí en adelante, se presenta un extracto de la documentación de la MDN (Mozilla Development Network) modificado. Recuerden que para nuestro curso la MDN es bibliografía oficial recomendada.

## ¿Qué son los formularios web?

Los **formularios web** son uno de los principales puntos de interacción entre un usuario y un sitio web o aplicación. Los formularios permiten a los usuarios la introducción de datos, que generalmente se envían a un servidor web para su procesamiento y almacenamiento, o se usan en el lado del cliente para provocar de alguna manera una actualización inmediata de la interfaz (por ejemplo, se añade otro elemento a una lista, o se muestra u oculta una función de interfaz de usuario).

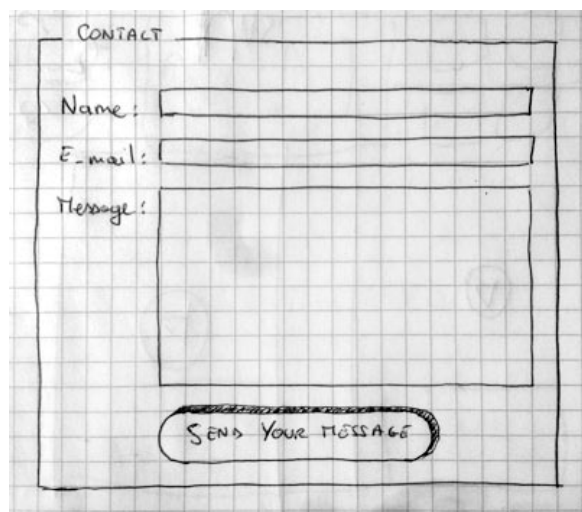
El HTML de un **formulario web** está compuesto por uno o más **controles de formulario** (a veces llamados **widgets**), además de algunos elementos adicionales que ayudan a estructurar el formulario general; a menudo se los conoce como **formularios HTML**. Los controles pueden ser campos de texto de una o varias líneas, cajas desplegables, botones, casillas de verificación o botones de opción, y se crean principalmente con el elemento `<input>`, aunque hay algunos otros elementos que también hay que conocer.

Los controles de formulario también se pueden programar para forzar la introducción de formatos o valores específicos (**validación de formulario**), y se combinan con etiquetas de texto que describen su propósito para los usuarios con y sin discapacidad visual.

## Diseñar tu formulario

Antes de comenzar a codificar, siempre es mejor dar un paso atrás y tomarte el tiempo necesario para pensar en tu formulario. Diseñar una maqueta rápida te ayudará a definir el conjunto de datos adecuado que deseas pedirle al usuario que introduzca. Desde el punto de vista de la experiencia del usuario (UX), es importante recordar que cuanto más grande es tu formulario, más te arriesgas a frustrar a las personas y perder usuarios. Tiene que ser simple y conciso: solicita solo los datos que necesitas.

En este artículo, vamos a crear un formulario de contacto sencillo. Hagamos un esbozo.



Nuestro formulario va a tener tres campos de texto y un botón. Le pedimos al usuario su nombre, su correo electrónico y el mensaje que desea enviar. Al pulsar el botón sus datos se enviarán a un servidor web.

## Implementación de nuestro formulario HTML

De acuerdo, intentemos crear el HTML para nuestro formulario. Vamos a utilizar los elementos HTML siguientes: `<form>`, `<label>`, `<input>`, `<textarea>` y `<button>`.

Antes de continuar, haz una copia local de nuestra [plantilla HTML simple](#): introduce aquí tu formulario HTML.

### El elemento `<form>`

Todos los formularios comienzan con un elemento `<form>`, como este:

```
1 <form action="/my-handling-form-page" method="post">
2
3 </form>
```

Este elemento define formalmente un **formulario**. Es un elemento **contenedor**, pero específico para contener formularios; también admite algunos atributos específicos para la configuración de la forma en que se comporta el formulario. Todos sus atributos son opcionales, pero es una práctica estándar establecer siempre al menos los atributos [action](#) y [method](#):

- El atributo `action` define la ubicación (URL) donde se envían los datos que el formulario ha recopilado cuando se validan.
- El atributo `method` define con qué método HTTP se envían los datos (generalmente `get` o `post`).

Por ahora, añade el elemento [<form>](#) anterior a tu elemento HTML [<body>](#).

## Los elementos [<label>](#), [<input>](#) y [<textarea>](#)

Nuestro formulario de contacto no es complejo: la parte para la entrada de datos contiene tres campos de texto, cada uno con su elemento [<label>](#) correspondiente:

- El campo de entrada para el nombre es un [campo de texto de una sola línea](#).
- El campo de entrada para el correo electrónico es una [entrada de datos de tipo correo electrónico](#): un campo de texto de una sola línea que acepta solo direcciones de correo electrónico.
- El campo de entrada para el mensaje es [<textarea>](#); un campo de texto multilínea.

En términos de código HTML, para implementar estos controles de formulario necesitamos algo como lo siguiente:

```
1 <form action="/my-handling-form-page" method="post">
2   <ul>
3     <li>
4       <label for="name">Nombre:</label>
5       <input type="text" id="name" name="user_name">
6     </li>
7     <li>
8       <label for="mail">Correo electrónico:</label>
9       <input type="email" id="mail" name="user_mail">
10    </li>
11    <li>
12      <label for="msg">Mensaje:</label>
13      <textarea id="msg" name="user_message"></textarea>
14    </li>
15  </ul>
16 </form>
```

Actualiza tu código de formulario para que se vea como el anterior.

Los elementos [<li>](#) están ahí para estructurar nuestro código convenientemente y facilitar la aplicación de estilo (ver más adelante en el curso). Por motivos de usabilidad y accesibilidad incluimos una etiqueta explícita para cada control de formulario. Ten en cuenta el uso del atributo `for` en todos los elementos [<label>](#), que toma como valor el `id` del control de formulario con el que está asociado; así es como asocias un formulario con su etiqueta.

Hacer esto presenta muchas ventajas porque la etiqueta está asociada al control del formulario y permite que los usuarios con ratón, panel táctil y dispositivos táctiles hagan clic en la etiqueta para activar el control correspondiente, y también proporciona accesibilidad con un nombre que los lectores de pantalla leen a sus usuarios. Encontrarás más detalles sobre las etiquetas de los formularios en [Cómo estructurar un formulario web](#).

En el elemento `<input>`, el atributo más importante es `type`. Este atributo es muy importante porque define la forma en que el elemento `<input>` aparece y se comporta. Encontrarás más información sobre esto en el artículo sobre [Controles de formularios nativos básicos](#) más adelante.

- En nuestro ejemplo sencillo, usamos el valor `<input/text>` para la primera entrada, el valor predeterminado para este atributo. Representa un campo de texto básico de una sola línea que acepta cualquier tipo de entrada de texto.
- Para la segunda entrada, usamos el valor `<input/email>`, que define un campo de texto de una sola línea que solo acepta una dirección de correo electrónico. Esto convierte un campo de texto básico en una especie de campo «inteligente» que efectúa algunas comprobaciones de validación de los datos que el usuario escribe. También hace que aparezca un diseño de teclado más apropiado para introducir direcciones de correo electrónico (por ejemplo, con un símbolo @ por defecto) en dispositivos con teclados dinámicos, como teléfonos inteligentes. Encontrarás más información sobre la validación de formularios en el artículo de [Validación de formularios por parte del cliente](#) más adelante.

Por último, pero no por ello menos importante, ten en cuenta la sintaxis de `<input>` en contraposición con la de `<textarea></textarea>`. Esta es una de las rarezas del HTML. La etiqueta `<input>` es un elemento vacío, lo que significa que no necesita una etiqueta de cierre. El elemento `<textarea>` no es un elemento vacío, lo que significa que debe cerrarse con la etiqueta de cierre adecuada. Esto tiene un impacto en una característica específica de los formularios: el modo en que defines el valor predeterminado. Para definir el valor predeterminado de un elemento `<input>`, debes usar el atributo `value` de esta manera:

```
<input type="text" value="por defecto este elemento se llena con este texto">
```

Por otro lado, si deseas definir un valor predeterminado para un elemento `<textarea>`, lo colocas entre las etiquetas de apertura y cierre del elemento `<textarea>`, así:

```
<textarea>
Por defecto, este elemento contiene este texto
</textarea>
```

## El elemento `<button>`

El marcado de nuestro formulario está casi completo; solo necesitamos añadir un botón para permitir que el usuario envíe sus datos una vez que haya completado el formulario. Esto se hace con el elemento `<button>`; añade lo siguiente justo encima de la etiqueta de cierre `</form>`:

```
<li class="button">
  <button type="submit">Envíe su mensaje</button>
</li>
```

El elemento `<button>` también acepta un atributo de `type`, que a su vez acepta uno de estos tres valores: `submit`, `reset` o `button`.

- Un clic en un botón `submit` (el valor predeterminado) envía los datos del formulario a la página web definida por el atributo `action` del elemento `<form>`.
- Un clic en un botón `reset` restablece de inmediato todos los controles de formulario a su valor predeterminado. Desde el punto de vista de UX, esto se considera una mala práctica, por lo que debes evitar usar este tipo de botones a menos que realmente tengas una buena razón para incluirlos.
- Un clic en un botón `button` no hace... ¡nada! Eso suena tonto, pero es muy útil para crear botones personalizados: puedes definir su función con JavaScript.

**Nota:** También puedes usar el elemento `<input>` con el atributo `type` correspondiente para generar un botón, por ejemplo `<input type="submit">`. La ventaja principal del elemento `<button>` es que el elemento `<input>` solo permite texto sin formato en su etiqueta, mientras que el elemento `<button>` permite contenido HTML completo, lo que permite generar botones creativos más complejos.

Fuente: [https://developer.mozilla.org/es/docs/Learn/HTML/Forms/Your\\_first\\_HTML\\_form](https://developer.mozilla.org/es/docs/Learn/HTML/Forms/Your_first_HTML_form)

Fuente: Introducción al desarrollo de Sistemas. Matías Gastón Santiago.

Última modificación: miércoles, 8 de julio de 2020, 01:53



Web (GET)

## INFO

Ir a...

## CONTACTANOS

## GET SOCIAL

[Envío de formularios ►](#)

En batan.coop educación consideramos primordial la agilidad del proceso comunicativo, cualquier inquietud, duda y aporte que quieras hacernos llegar será motivo de crecimiento para nosotros y mejorará el modo en que podamos ayudarte a través de este servicio.

[Batan.Coop](#)[Ecolan Soluciones Tecnológicas](#)[Servicio Educativo Caraludmé](#)

Julián Ríos 4215, Batán, Buenos Aires, Argentina. Julián Ríos 4215. Código Postal: 7601

☎ Phone : (0223) 464-3000

✉ E-mail :

[info@cooperativabatan.com.ar](mailto:info@cooperativabatan.com.ar)

Copyright © 2020 - Desarrollado por Batan.Coop powered by Ecolan

[Resumen de conservación de datos](#)

[Descargar la app para dispositivos móviles](#)