



Nuestro compromiso es con el *futuro*.

Introducción al Front End

Clase 5

¿Qué veremos hoy?

Hoy veremos una de las utilidades más grandes que nos provee el **Desarrollo Web**: los **formularios**.

Éstos nos permiten tomar datos de los usuarios, que van desde **texto** hasta **archivos**, pasando por **números**, **fechas**, **colores**, y **datos seleccionables**, entre otros.

Vamos a ver entonces cómo es que se crean estos **formularios**, cómo se gestionan y cuáles son las distintas funcionalidades que nos ofrecen a la hora de recolectar **datos** de un **usuario**.

Formularios

Contacto

Nombre de usuario

Contraseña

Los formularios permitirán a nuestros usuarios completar su datos, registrarse en nuestra aplicación, iniciar sesión, enviarnos un mensaje, entre muchas otras cosas.

Vemos que en todo formulario existe el campo donde el usuario ingresará los datos y siempre existe un botón que nos permitirá ejecutar la acción que procesará estos datos.

Formularios

Estos campos que registran la información reciben el nombre genérico de inputs, debido a que la etiqueta más común para estos lleva ese nombre.

Ahora bien, todos estos campos deben estar contenidos por una etiqueta primordial, que es la que definirá los confines de nuestro formulario, además de darle distintas funcionalidades al mismo.

Esta etiqueta se conoce como `<form>`.

```
<form action="/contact">
  <label for="username">Nombre de usuario</label>
  <input
    type="text"
    placeholder="Ingresa un nombre de usuario..."
    name="username"
    id="username"/>
  <label for="password">Contraseña</label>
  <input
    type="password"
    placeholder="Ingresa un nombre de contraseña..."
    name="password"
    id="password"/>
  <button>Enviar</button>
</form>
```

Formularios

La etiqueta `<form>` será la etiqueta maestra en cualquier formulario.

Como atributos, esta etiqueta tiene dos atributos principales fundamentales:

- El atributo `action`, que define la dirección, ruta o destino de a dónde queremos enviar los datos de nuestro formulario.
- El atributo `method`, que definirá cuál de los métodos del protocolo **HTTP** será usado para enviar la información: **GET** o **POST**.

Estos atributos cobrarán mayor sentido e importancia cuando abordemos el módulo de Back End, ya que allí crearemos rutas que procesarán la información proveniente de nuestros formularios.

```
action="/contact" m
label_for="username"
```

```
method="|>
e">Non dialog
get
post
```

input

Como dijimos anteriormente, la etiqueta principal dentro de un **formulario** es la etiqueta `<input />`, que puede tener múltiples tipos. En las siguientes diapositivas veremos algunos de los más utilizados.

Es importante conocer los atributos más comunes de los elementos `<input />`, a continuación veremos los que aplican a la mayoría, indistintamente del tipo.

- **name**: atributo que identifica al input dentro del **formulario**, lo que hace que éste último pueda reconocerlo dentro de sí mismo y enlazarse a él, permitiendo poder acceder a su contenido y propiedades y así logrando integrar los datos con el **formulario** mismo.
- **id**: si bien ya conocemos este atributo, aquí lo destacamos porque es fundamental no solo para identificar al elemento en cuestión sino también para permitirle enlazarse con un elemento fundamental que veremos en breve: el **label** o etiqueta.

input

- **type**: atributo que justamente define el tipo de **input** que deberá representarse, el más común es el **type="text"**.
- **type="text"**: este tipo de **input** es el más común, nos permite ingresar texto en formato alfanumérico, es decir, letras, número y caracteres especiales. Es el indicado para datos generales y para cuando no haya un tipo de **input** más específico.
- **type="number"**: este **input** nos permite ingresar únicamente valores numéricos, además de que dependiendo del dispositivo, mostrará opciones acordes. Por ejemplo, en un dispositivo móvil desplegará solamente el teclado numérico.

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

```
<input  
  type="number"  
  name="age"  
  id="age"  
>
```

input

- `type="email"`: este tipo de `input` nos permite manejar datos de tipo correo electrónico. Es decir datos que contengan un `@` en su contenido y que sirvan como destino para alguna cuenta de servicio de correo electrónico.
- `type="password"`: este `input` recibe datos sensibles como contraseñas y automáticamente las va ocultando con distintos tipos de caracteres según el navegador. Es el más indicado justamente para aquellos campos que no deben mostrarse públicamente y que solo quien los escribe debería saber qué contenido tienen.

```
<input  
  type="email"  
  name="correo"  
  id="correo"  
>
```

```
<input  
  type="password"  
  name="contrasena"  
  id="contrasena"  
>
```

input

- `type="checkbox"`: este tipo de `input` nos permite realizar una selección en un casillero, es decir, no lleva contenido sino que simplemente representará si el usuario seleccionó o no este `input`. Para este tipo de `input` el atributo principal será `checked`, que retornará `true` o `false` según si el mismo está seleccionado o no.
- `type="file"`: este tipo de `input` nos permite seleccionar un archivo desde nuestra computadora para subirlo y que luego sea procesado o almacenado según corresponda.

```
<input  
  type="checkbox"  
  name="suscripcion"  
  id="suscripcion"  
>
```

```
<input  
  type="file"  
  name="fotoPerfil"  
  id="fotoPerfil"  
>
```

input

- **type="radio"**: este **input** funciona muy similarmente al de tipo **checkbox**, solamente que este nos permite seleccionar sólo uno de entre algunas opciones (usualmente dos o tres). Para lograr esto los distintos **inputs** de tipo **radio** deben compartir el atributo **name**, para así quedar unidos entre sí. Se distinguirán uno del otro gracias al atributo **value**, que cubriremos en breve.
- **type="submit"**: este **input** toma la forma de un botón y reemplaza al elemento **<button>**, al ser una etiqueta cerrada, el texto que aparecerá en el mismo debe contenerse en el atributo **value**. Se utiliza para terminar el proceso de un **formulario** y enviar sus datos.

```
<input
  type="radio"
  name="poseeLicencia"
  value="si"
/>
```

```
<input
  type="radio"
  name="poseeLicencia"
  value="no"
/>
```

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

input

- **type="search"**: este tipo de **input** nos permite crear una barra de búsqueda con algunas funcionalidades básicas como borrar el contenido y poco más, es muy similar al tipo text, pero siempre es una buena práctica usar el elemento sintáctico idóneo para cada tarea.
- **type="range"**: este **input** define una barra sobre la cual podemos deslizar un selector para definir un rango. Aquí podemos utilizar los atributos min y max para definir los valores mínimos y máximos de esta barra selectora. El atributo value será el que almacene la selección y también podríamos pre-seleccionarle un value para que se muestre al cargar el **formulario**.

```
<input
  type="search"
  name="buscador"
  id="buscador"
/>
```

```
<input
  type="range"
  min="1"
  max="10"
  value="2"
  name="calificacion"
/>
```

Más inputs

Además de todos estos inputs, existen muchos otros que podemos utilizar, para conocer más sobre ellos, dejamos la siguiente referencia donde podremos leer más sobre su utilización, que otros parámetros son útiles y otras funcionalidades interesantes.

Link: [Inputs - MDN](#)

label

Habiendo visto los distintos tipos de `input`, es hora de que los veamos en práctica, pero antes, deberíamos conocer sobre el elemento `<label>`.

La etiqueta `label` sirve como texto de apoyo o referencia para aquellos `inputs` que requieren claridad sobre de qué contenido se les debería ingresar.

Además también funcionan como enlace, ya que al colocarles en su atributo `for` el mismo contenido que el `id` de algún `input`, éstos quedarán enlazados y al hacer click sobre el `label`, el `input` correspondiente será seleccionado (lo que llamamos, *ganar foco*).

```
<label for="username">Nombre de usuario</label>
<input
  type="text"
  placeholder="Ingresa un nombre de usuario..."
  name="username"
  id="username"/>
```

Siempre es buena práctica agregar labels a los elementos de un formulario, por cuestiones de accesibilidad y usabilidad general.

select

El `<select>` es un tipo de `input` especial, ya que desplegará un listado del cual podremos elegir una opción. Se utiliza cuando necesitamos que el usuario elija una de muchas opciones, sobre todo cuando éstas pueden irse actualizando con el tiempo. Es una etiqueta abierta, que contendrá etiquetas de tipo `<option>`. Éstas son las que contendrán las opciones dentro del cuadro desplegable.

El atributo `name` identificará al `select` dentro del **formulario**, y cada option deberá tener un `value` que es lo que se enviará como valor para el `name` del select. También podemos agregar un atributo `selected`, que hará que ese option en particular sea el que está seleccionado por defecto al cargar el **formulario**.

```
<select name="pais" id="pais">
  <option selected value="arg">Argentina</option>
  <option value="arg">Uruguay</option>
  <option value="arg">Paraguay</option>
  <option value="arg">Brasil</option>
</select>
```


Poniendo todo esto en práctica

Ahora que tenemos una buena base de cómo crear nuestros formularios, pondremos todo esto en práctica creando un formulario de registro.

Este formulario utilizará distintos tipos de inputs, selects, y demás para permitirle a un usuario ingresar todos sus datos y poder procesarlos desde un Back End para almacenarlos y registrarlo en una aplicación.

Nuestro formulario tendrá los siguientes inputs:

- Nombre completo
- Nombre de usuario
- Correo electrónico
- Contraseña
- Confirmar contraseña
- Fecha de nacimiento
- Foto de perfil
- País
- Teléfono
- Género
- Suscribirse al newsletter
- Acepto los términos y condiciones

¡Vamos al código!

Muchas gracias!



ICARO Asociación Civil
CUIT 30716564815
info@icaro.org.ar
www.icaro.org.ar