



Formularios

- Los formularios son elementos que tienen campos que nos permiten obtener datos del usuario
- Podemos encontrarlos por todo internet en diferentes formatos:

Buscadores:

Google:



Buscar con Google

Me siento con suerte

Google.com.ar ofrecido en: [Español \(Latinoamérica\)](#)

Twitter:  Twitter search

Youtube:



Comentarios/posts:



Twitter post:

The image shows a Twitter post composition interface. At the top, it says 'Redactar un nuevo Tweet' with a close button (X). Below this is a large text input area with the placeholder text '¿Qué está pasando?'. To the right of the input area are two icons: a green 'G' (Giphy) and a smiley face (emojis). Below the input area is a row of icons: a picture icon, a GIF icon, a list icon, and a location pin icon. To the right of these icons is a character count '140' and a blue 'Tuitear' button.

Facebook:  FB post

Log in / sign up

Facebook:

The image shows the Facebook login and sign-up page. At the top, there's a dark blue header with the Facebook logo on the left. On the right, there are two input fields: 'Email or Phone' and 'Password', followed by a 'Log In' button and a link 'Forgot account?'. Below the header, the page is split into two columns. The left column has the heading 'Connect with friends and the world around you on Facebook.' and three items: 'See photos and updates from friends in News Feed.', 'Share what's new in your life on your Timeline.', and 'Find more of what you're looking for with Facebook Search.' The right column has the heading 'Sign Up' with the subtext 'It's free and always will be.' Below this are four input fields: 'First name', 'Last name', 'Mobile number or email', and 'New password'. Under these is a 'Birthday' section with dropdowns for 'Month', 'Day', and 'Year', and a link 'Why do I need to provide my birthday?'. Below the birthday section are two radio buttons for 'Female' and 'Male'. At the bottom of the sign-up section is a green 'Create Account' button. Below the button is a link 'Create a Page for a celebrity, band or business.' At the very bottom right, there's a hamburger menu icon.

- Para definir un formulario utilizamos la etiqueta **form**

- Los formularios tienen 3 atributos importantes:
 - action: url a donde se van a enviar o submitear los datos del formulario
 - method: acepta como valor un método de HTTP para establecer la forma en que se van a enviar los datos: get o post
 - Get: manda los datos por query string (en la url, ej: nombre=valor&apellido=valor)
 - Post: los datos viajan en el cuerpo del mensaje no se pueden ver
 - enctype: establece el tipo de media que vamos a enviar
 - Por defecto: 'application/x-www-form-urlencoded'
 - Para enviar archivos: 'multipart/form-data'
 - Texto plano: 'text/plain'

```
<form action="guardar_usuario.html" method="get" enctype="application/x-www-form-urlencoded" >
```

```
  Mi form
```

```
</form>
```

- El formulario va a tener un montón de elementos HTML que le permitan al usuario ingresar datos, por ejemplo nombre de usuario y contraseña
- Utilizando un botón vamos a ejecutar el formulario y enviar todos los datos a otro documento en el cual tenemos programado la forma de analizarlos y actuar en consecuencia
- Para recibir y utilizar los datos podemos utilizar varios lenguajes de programación como C#, Java, JavaScript, PHP, Ruby, etc
- Cada campo de texto va a tener un nombre y de esta forma vamos a reconocer que tipo de dato es
- Al terminar de utilizar los datos enviamos un archivo de respuesta, por ejemplo al verificar que el usuario nos envió credenciales válidas, le podemos mostrar la página que antes no podía

Inputs

- Usando los **inputs** le permitimos al usuario que ingrese o seleccione datos
- Existen distintos tipos de inputs y cada uno de ellos tiene una funcionalidad distinta
- Por medio del atributo **type** definimos que tipo de campo es



Texto

- Se utilizan para ingresar una línea de texto como puede ser un nombre, apellido, email, dirección, etc
- Para definir este tipo de input utilizamos el **type** text
- Con el atributo **name** definimos el nombre de la variable

```
<form>  
  <input type="text" name="apellido" />  
</form>
```

- Con el atributo **value** podemos definir el valor del contenido
- El usuario va a ver este texto pre ingresado en el campo de texto
- En caso de querer remplazarlo deberá borrar el valor actual y luego ingresar uno nuevo

```
<input type="text" name="apellido" value="Lovelace" />
```

- Por medio del atributo **placeholder** podemos establecer un texto dentro del input a modo de ejemplo. Este texto no es el valor final del campo y solo lo utilizamos para ayudar al usuario a entender que tipo de valor debe ingresar
- Si utilizamos el atributo **value** y **placeholder** al mismo tiempo vamos a ver el valor asignado dentro del atributo **value**

```
<input type="text" name="apellido" placeholder="Ej: Lovelace" />
```

- Existe un atributo **size** que nos permite establecer el tamaño del campo de texto
- Este atributo acepta un valor numérico con la cantidad de caracteres que queremos ver en el campo
- Este atributo solo limita la forma en que se ve el campo y no la cantidad de caracteres que puede ingresar el usuario
- Dado que es un atributo visual lo vamos a cambiar desde CSS en el futuro

```
<input type="text" name="apellido" value="Lovelace" size="20" />
```



- Para limitar la cantidad de caracteres que puede ingresar un usuario utilizamos el atributo **maxlength** que acepta un valor numérico con la cantidad máxima de caracteres esperados

```
<input type="text" name="apellido" value="Lovelace" maxlength="20" />
```

Password

- Permite ingresar un password o contraseña
- Es como el campo de texto pero no se ven los caracteres ingresados
- Para definir este tipo de input utilizamos el **type** password
- Podemos utilizar los atributos **name**, **value**, **size** y **maxlength** al igual que los campos de texto
- Si bien el usuario no puede ver los caracteres ingresados esto no significa que sea seguro ya que es solo un efecto visual
- Utilizamos HTTPS para enviar los datos de forma segura entre nuestro cliente al servidor

```
<form>  
  <input type="password" name="password" />  
</form>
```

Submit buttons

- Permite enviar o submitear los datos del formulario a otro documento
- Para definir este tipo de input utilizamos el **type** submit
- Podemos utilizar el atributo name pero no es necesario ya que es tan solo un botón y no nos brinda ningún dato extra importante
- El atributo **value** acepta un texto y nos es útil para mostrar un mensaje al usuario

```
<form>  
  <input type="submit" value="Crear usuario" />  
</form>
```

Radio buttons

- Permite que el usuario seleccione sólo un valor entre algunas opciones
- Para definir este tipo de input utilizamos el **type** radio



- Utilizamos el atributo **name** para identificar este elemento y luego obtener su valor
- Dado que este elemento se utiliza junto a otros iguales y la idea es que el usuario seleccione uno solo debemos nombrar a todos los elementos con el mismo **name**
- Para dar un contexto del valor de este elemento debemos escribir algún texto descriptivo junto a ellos

<form>

Acepta términos y condiciones:

```
<input type="radio" name="terminos" value="1" /> Si
```

```
<input type="radio" name="terminos" value="2" /> No
```

</form>

- Podemos seleccionar inicialmente un radio button utilizando el atributo **checked** con el mismo valor **checked**
- Sólo un elemento radio button tiene que tener este valor ya que se utilizan para selección única como verdadero o falso

<form>

Acepta términos y condiciones:

```
<input type="radio" name="terminos" value="1" /> Si
```

```
<input type="radio" name="terminos" value="2" checked="checked" /> No
```

</form>

Checkboxes

- Permite que el usuario seleccione/deseleccione multiples valores entre algunas opciones
- Para definir este tipo de input utilizamos el **type** checkbox
- Utilizamos el atributo **name** para identificar este elemento
- Dado que la selección es múltiple debemos utilizar el mismo **name** para todos los checkboxes de un mismo criterio (ejemplo: elegir super heroes que te gustan)
- Por medio del atributo **value** establecemos el valor de cada uno de estos elementos
- El atributo **checked** nos permite activar o seleccionar inicialmente este elemento

<form>

Estado Civil:

```
<input type="checkbox" name="superheroe" value="batman" checked="checked" />
Batman

<input type="checkbox" name="superheroe" value="superman" checked="checked"
/> Superman
<input type="checkbox" name="superheroe" value="flash" checked="checked" />
Flash
<input type="checkbox" name="superheroe" value="mujer maravilla"
checked="checked" /> Mujer Maravilla
</form>
```

Hidden

- En algunos casos podemos necesitar agregar algunos datos que no queremos que el usuario vea en el formulario, para esto utilizamos inputs ocultos
- Para definir este tipo de input utilizamos el **type** hidden
- Utilizamos el atributo **name** para identificar este valor y el atributo **value** para establecer el valor del mismo
- Si el usuario ve el código fuente si puede ver estos campos y valores ya que son parte del código de la página
- Un caso de uso puede ser ocultar el id de un producto

```
<form>
  <input type="hidden" name="id-de-producto" value="20" />
</form>
```

Image buttons

- Permite enviar o submitar los datos del formulario a otro documento utilizando una imagen
- Para definir este tipo de input utilizamos el **type** image
- Al igual que las imágenes este elemento tiene el atributo **src** que nos permite establecer cuál es la fuente

```
<form>
  <input type="image" src="crear_usuario.png" />
</form>
```



- Podemos utilizar los atributos **height** y **width** para establecer el alto y ancho respectivamente
- Al ser un atributo visual, en el futuro, lo vamos a controlar desde CSS

Campos especiales en HTML5

- HTML5 incorpora nuevos tipos de inputs
- Si el browser no soporta estos nuevos tipos de inputs, muestra un input de texto normal
- Todos ellos tienen el atributo **name** que nos permite identificar el valor ingresado (variable)

Date

- En muchos casos necesitamos pedirle al usuario que ingrese una fecha, puede ser para saber su nacimiento, vencimiento de la tarjeta de crédito, año en el que termino el colegio o para saber su experiencia laboral
- Para definir este tipo de input utilizamos el **type** date

```
<form>  
  <input type="date" name="fechadenacimiento" />  
</form>
```

Email

- Este nuevo tipo de input permite que el usuario ingrese su email o correo electrónico
- Para definir este tipo de input utilizamos el **type** email

```
<form>  
  <input type="email" name="email" />  
</form>
```

URL

- Este nuevo tipo de input permite que el usuario ingrese una url, por ejemplo el usuario puede ingresar la dirección de su sitio personal, trabajos realizados online o cuenta de twitter con la url completa
- Para definir este tipo de input utilizamos el **type** url




```
<form>
  <input type="url" name="sitiopersonal" />

</form>
```

Search

- Utilizando este nuevo tipo de input podemos pedirle al usuario que ingrese un criterio por el cual vamos a realizar algún tipo de búsqueda
- Para definir este tipo de input utilizamos el **type** search
- Algunos browsers agregan la funcionalidad de borrar el contenido por medio de algún botón
- Esta funcionalidad la podemos construir con css y js sin problema

```
<form>
  <input type="search" name="busqueda" />
</form>
```

File upload

- Permite seleccionar un archivo que será enviado con el resto de los datos del formulario
- Para definir este tipo de input utilizamos el **type** file
- Al utilizar este tipo de inputs debemos establecer el atributo **method** del formualrio en **post**
- Tambien debemos establecer el atributo del fomr **enctype** en 'multipart/form-data'
- Utilizamos el atributo name para identificar este archivo

```
<form action="archivo.php" method="post" enctype="multipart/form-data">
  <input type="file" name="archivo" />
</form>
```

Text area

- Permite ingresar texto multilinea, se puede utilizar para notas, una bio o cualquier texto que ocupe más de una linea
- A diferencia del resto de las etiquetas el text area tiene contenido, es decir que tiene una etiqueta de apertura y una de cierre



- También utilizamos el atributo **name** para identificar a este elemento y obtener su valor

```
<textarea name="bio">Ada Lovelace es la 1er programadora.</textarea>
```

- Podemos utilizar el atributo placeholder para establecer un valor inicial al igual que en el resto de los inputs

```
<textarea name="bio" placeholder="Ingrese su bio"></textarea>
```

- Podemos limitar el ancho de este elemento por medio del atributo **cols** que acepta un valor numérico con la cantidad de caracteres que queremos ver
- También podemos cambiar la cantidad de líneas que muestra el textarea utilizando el atributo **rows** y acepta un valor numérico con la cantidad de líneas que queremos ver

```
<textarea name="bio" cols="10" rows="2" >Texto inicial de mi text  
area</textarea>
```

Drop-down

- Permite seleccionar una o varias opciones de una lista
- Se utiliza cuando tenemos que mostrar muchas opciones
- Este elemento está compuesto de dos partes:
 - Etiqueta **select** para establecer que es un drop down
 - Etiquetas **option** para establecer las opciones y valores del **select**
- Identificamos el elemento **select** con el atributo **name**

```
<form>  
  <select name="opciones">  
    <option>Opción 1</option>  
    <option>Opción 2</option>  
  </select>  
</form>
```



- Establecemos el valor de la etiqueta **option** con el atributo **value**
- Por medio del atributo **selected** y el valor **selected** podemos seleccionar inicialmente un **option** dentro del **select**
- Si no utilizamos el atributo **selected** el drop down utiliza el primer elemento

```
<form>
  <select name="paises">
    <option value="">Seleccione un país</option>
    <option value="arg">Argentina</option>
    <option value="bra">Brasil</option>
    <option value="chl">Chile</option>
    <option value="col" selected="selected">Colombia</option>
    <option value="ven">Venezuela</option>
  </select>
</form>
```

- Podemos transformar el **select** en un selector de múltiples valores utilizando el atributo **multiple** con el valor **multiple**
- Podemos establecer el atributo **size** con un valor numérico con la cantidad de elementos que queremos mostrar en la lista
- Para seleccionar múltiples opciones utilizamos la tecla **control**

```
<form>
  <select name="paises" multiple="multiple" size="5">
    <option value="arg">Argentina</option>
    <option value="bra">Brasil</option>
    <option value="chl">Chile</option>
    <option value="col">Colombia</option>
    <option value="ven">Venezuela</option>
  </select>
</form>
```

Buttons

- También contamos con botones personalizados utilizando el elemento **button**
- Dado que podemos establecer el contenido de este elemento tenemos que utilizar la



etiqueta de apertura y cierre

```
<form>
  <button>+ Usuario</button>
</form>
```

- Podemos agregar cualquier contenido, por ejemplo una imagen

```
<form>
  <button>
    
    Usuario
  </button>
</form>
```

Describir elementos

- Para comunicarle al usuario que valor tiene que ingresar en cada campo podemos utilizar el elemento **label** o etiqueta
- Estas etiquetas también nos permiten describirle el contenido de los campos a las personas no videntes
- Este es un elemento en línea

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

- También se puede utilizar de la siguiente forma

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



```

    </label>
    <label>

        Apellido:
        <input type="text" name="apellido" />
    </label>
</form>

```

- La etiqueta **label** tiene un atributo **for** que acepta el nombre del id de un input
- Si el **label** tiene establecido un atributo **for** el usuario puede hacer click en el texto del label para hacer foco en el input

```

<form>
    <label for="idnombre">Nombre:</label>
    <input type="text" name="nombre" id="idnombre" />
    <label for="idapellido">Apellido:</label>
    <input type="text" name="apellido" id="idapellido" />
</form>

```

Agrupar campos

- Si el formulario es largo está bueno agrupar los campos por algún criterio, por ejemplo datos del usuario, de su auto, etc
- Para agrupar campos utilizamos la etiqueta **fieldset**

```

<form>
    <fieldset>
        <label for="idnombre">Nombre:</label>
        <input type="text" name="nombre" id="idnombre" />
        <label for="idapellido">Apellido:</label>
        <input type="text" name="apellido" id="idapellido" />
    </fieldset>
</form>

```

- Podemos agregar un título a la sección utilizando la etiqueta **legend**

```

<form>

```



```
<fieldset>
  <legend>Datos del usuario</legend>
  <label for="idnombre">Nombre:</label>
  <input type="text" name="nombre" id="idnombre" />
  <label for="idapellido">Apellido:</label>
  <input type="text" name="apellido" id="idapellido" />
</fieldset>
</form>
```

Validar campos

- Es común tener que validar los datos ingresados por el usuario para verificar que tengamos todo lo que necesitamos y en el formato correcto
- Validar los formularios en el browser se conoce como validar los datos del lado del cliente
- Para asegurarnos que los datos son correctos y que no nos interceptaron el request vamos a validar los datos también del lado del servidor
- Si bien la doble validación es más trabajo, hace que nuestro código sea más seguro
- Por otro lado evita enviar datos al servidor que no estén bien y de esta forma ahorrar idas y vueltas entre cliente y servidor
- HTML5 incorpora algunas funcionalidades que nos permiten validar nuestros formularios
- Generalmente esta tarea se realiza utilizando JavaScript y lo vamos a ver más adelante
- Por ahora no podemos customizar o personalizar el mensaje de error que muestran los campos (lo podemos hacer desde JavaScript)

Campos obligatorios

- Podemos hacer que un campo sea obligatorio utilizando el atributo **required**
- Browsers más viejos necesitan que se agregue el valor **required**
- El form no va a submitar los datos hasta que no esté completo

```
<form>
  <input type="text" name="nombre" required />
  <input type="text" name="nombre" required="required" />
  <input type="submit" value="Registrar" />
</form>
```



Campos con el formato correcto

- Distintos tipos de datos necesitan cumplir con un formato para ser válidos por ejemplo un email, ruta o fecha
- Para validar el formato de un input podemos utilizar el atributo **pattern** y una **expresión regular** como valor
- El campo no va a ser válido hasta que cumpla con el formato esperado
- Los nuevos campos de HTML5 como date, email y url ya vienen con una expresión regular incluida
- El elemento **textarea** no soporta este atributo

```
<form>
  <input type="text" name="nombre" pattern="^[a-zA-Z]+$" placeholder="Ingrese sólo letras" />
  <input type="text" name="edad" pattern="\d+" placeholder="Ingrese sólo números" />
  <input type="submit" value="Registrar" />
</form>
```

- Para saber si podemos utilizar una feature nueva y entender que browsers la soportan podemos utilizar caniuse.com tanto para html, css o js
- Si quieren leer un poco más sobre este tema pueden entrar al sitio de [MDN](https://developer.mozilla.org/)

 [Edit this page](#)

Previous
[« Tablas](#)

Next
[HTML5 »](#)

Inputs

Hidden

Text area



Drop-down

Buttons

Describir elementos

Agrupar campos

Validar campos

Docs

Frontend

Community

Discord

Social

Linkedin

Instagram

Facebook

Copyright © 2021 Ada Itw.

