

FORMULARIOS

FORMULARIOS



El manejo de Formularios de datos es clave en cualquier aplicación web y Vue Cli no escapa a ello.

FORMULARIOS

Realicemos un repaso por los principales elementos de un formulario web y cómo estos interactúan con Vue.





FORM TAGS

- `<form>`
 - `<input>`
 - `<select>`
 - `<option>`
 - `<textarea>`

👉 Estos son los principales tags HTML que utilizamos para la construcción de formularios de datos.

Puntualmente, la etiqueta `<form>` es la que agrupa, entre sus tags, al resto de los elementos HTML aquí mencionados.



FORM ATTRIBUTES

```
<form action="" method="POST" name="" enctype="">  
    <!-- Aquí se ubican el resto de los campos -->  
</form>
```

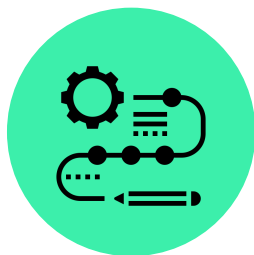
En la etiqueta **form** se utilizan una serie de **atributos**, los cuales **permiten el procesamiento de los datos** que se hayan cargado en los campos que esta contiene.

Algunos campos son opcionales, mientras que otros son requeridos u obligatorios, de acuerdo a la forma en la cual procesaremos la información cargada.

FORM ATTRIBUTES



Atributo
action



Atributo
method



Atributo
name

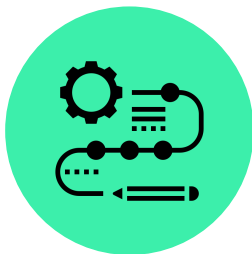


Atributo
enctype



FORM ATTRIBUTES

El atributo **action** usualmente espera recibir una URL que corresponde a una aplicación de backend (*del lado del servidor*) la cual procesa los datos cargados. También pueden especificarse una función JS que procese de otra forma la información cargada.



FORM ATTRIBUTES

El atributo `method` se debe configurar con algún método soportado por el servidor HTTP. Se suele utilizar **POST**, aunque **GET** también está permitido.

👁️ **GET** es más inseguro porque expone la información de los campos, a través de la URL de procesamiento.



FORM ATTRIBUTES

El atributo **name** funciona de manera similar al atributo global **id**, pero no se utiliza con frecuencia. En algunos casos se incluye para revisar, del lado del servidor, el nombre del formulario y así aplicar controles adicionales en base a su procesamiento.



FORM ATTRIBUTES

El atributo `enctype` se ocupa de encriptar los datos cargados. Si el formulario dispone de campos convencionales, se usa el valor `application/x-www-form-urlencoded`, pero si el formulario acepta archivos adjuntos, videos y/o imágenes, se lo configura como `multipart/form-data`.

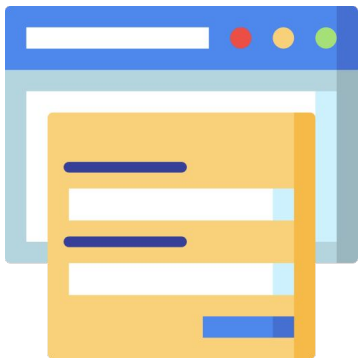
ATRIBUTOS BOOLEANOS

El atributo `autocomplete`, funciona como booleano y activa la función de autocompletado disponible usualmente en los navegadores web.

El atributo `required`, también booleano, hace que todos los campos que contiene, se tengan que completar de forma obligatoria. No se estila controlar esta acción desde aquí, sino de cada uno de los campos.

ELEMENTOS DEL FORM

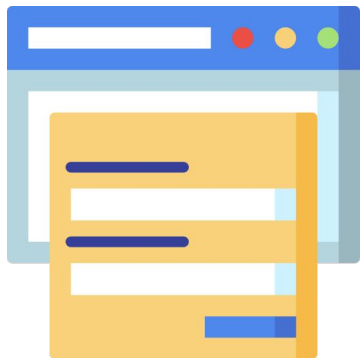
INPUT TYPES



Los campos, o elementos HTML, del tipo `input` engloban no solo elementos de entrada de datos vía teclado, sino también aquellos del tipo: `radioButton`, `checkBox`, y hasta `button`.

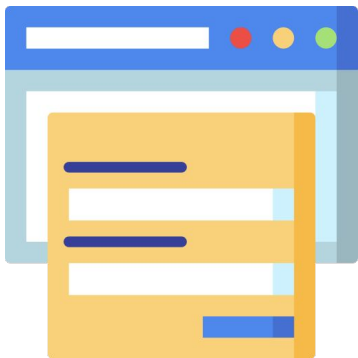
Pero, primero, nos concentraremos en los campos del tipo `entrada de datos vía teclado`.

INPUT TYPES



Originalmente, los campos de entrada de datos se regían por el atributo `type="text"`, común para cualquier información que se debiera cargar; inclusive teléfonos, email, datos numéricos en general, etcétera.

INPUT TYPES



Con la llegada de HTML5, se potenció el atributo `type`,
permitiendo que un input type pase a soportar más de
un tipo de datos.



INPUT TYPES

```
<input type="text" name="" id="nombre">
<input type="email" name="" id="email">
<input type="url" name="" id="direccionWeb">
<input type="number" name="" id="documento">
<input type="tel" name="" id="telefono">
<input type="date" name="" id="fechaNacimiento">
<input type="search" name="" id="">
```

Así nacieron los tipos de **datos nativos**, para: números de teléfono, URLs, direcciones de email, fechas, y hasta campos de búsquedas, entre otros tantos.

De esta forma, validar el dato ingresado comenzó a ser responsabilidad del motor web y dejó de ser una pesadilla para el programador y la implementación de expresiones regulares.



ATRIBUTO NAME

```
<input type="text" name="nombre" id="nombre">  
  
<input type="email" name="email" id="email">  
  
<input type="url" name="direccionWeb" id="direccionWeb">
```

En los input types sí suele ser mandatorio implementar el atributo **name**, dado que lo utiliza, habitualmente, la aplicación de backend que se ocupa de procesar los datos cargados en el form.

👉 El desarrollador de la aplicación backend debe estar en concordancia con nosotros para definir el valor de este atributo.



ATRIBUTO REQUIRED

```
<input type="text" name="" id="nombre" required>  
<input type="email" name="" id="email" required>  
<input type="url" name="" id="direccionWeb">  
<input type="number" name="" id="documento" required>
```

El atributo **required**, como ya mencionamos, es más común implementarlo en aquellos campos donde los datos solicitados son obligatorios.

Es del tipo **boolean**, lo que hace que su presencia active la notificación de campo obligatorio y bloquee el procesamiento del formulario.

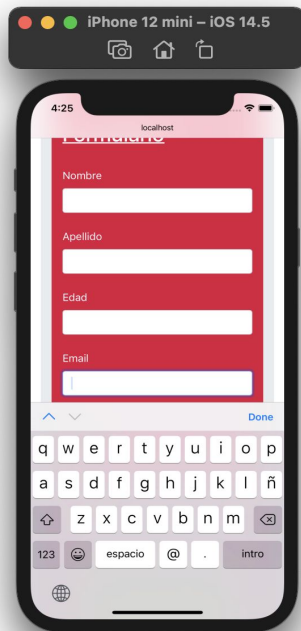
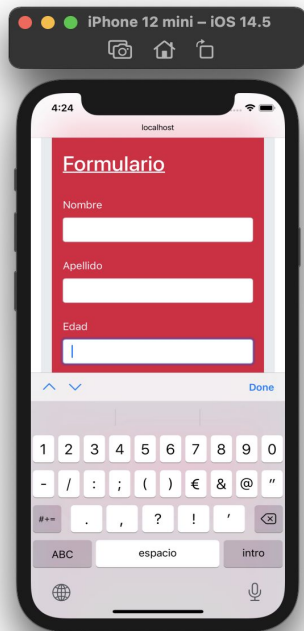
OTROS ATRIBUTOS CLAVE

OTROS ATRIBUTOS CLAVE

Aquí detallamos **otros atributos clave** para los `<input>`. Bien configurados, ayudan a mejorar la interacción del usuario y minimizar notablemente los errores de carga de información.

<i>ATRIBUTO</i>	<i>DESCRIPCIÓN</i>
min	Especifica un valor mínimo para el dato a ingresar (text, number, tel, date, month, hour, etcétera).
max	Especifica un valor máximo para el dato a ingresar (text, number, tel, date, month, hour, etcétera).
value	El dato ingresado por el usuario en el campo en cuestión.
step	Especifica un número de intervalos relacionado a datos numéricos, que el usuario pueda ingresar.
placeholder	Muestra un mensaje orientativo sobre el dato que se espera que el usuario ingrese.
disabled	Deshabilita el campo para que el usuario no pueda ingresar datos.

OTROS ATRIBUTOS CLAVE



Una correcta implementación de los input types potencian también las capacidades de los dispositivos móviles en donde, por ejemplo, un campo **number**, **email**, o **date**, adaptan el teclado para que el usuario disponga de teclas de uso obligatorio, a ingresar en el tipo de dato esperado.

OTROS INPUT



CHECKBOX

Curso(s) de su preferencia

- ✓ JAVASCRIPT
- ☐ ANGULAR
- ☐ REACT
- ✓ VUE

```
<label>  
  <input type="checkbox" checked />  
  <span class="green-text">VUE</span>  
</label>
```

El elemento checkbox utiliza el atributo **checked**, para marcar la tilde sobre la casilla de selección. El atributo es booleano, aunque si debemos manipularlo desde JavaScript o Vue, deberemos usar **checked = true || false**.



RADIO BUTTON

Tipo de documento

☒ DNI ☐ Carnet de conducir ☐ Pasaporte

```
<label>
  <input name="group1" type="radio" checked>
  <span>DNI</span>
</label>
<label>
  <input name="group1" type="radio" />
  <span>Carnet de conducir</span>
</label>
```

El elemento **radio button** funciona de manera similar a checkbox. La única diferencia es que a éste se lo suele utilizar para selección de una opción entre varias.

Si queremos lograr eso, debemos utilizar su atributo **name**, para asignarle un mismo grupo a dos o más radio buttons.



TEXTAREA

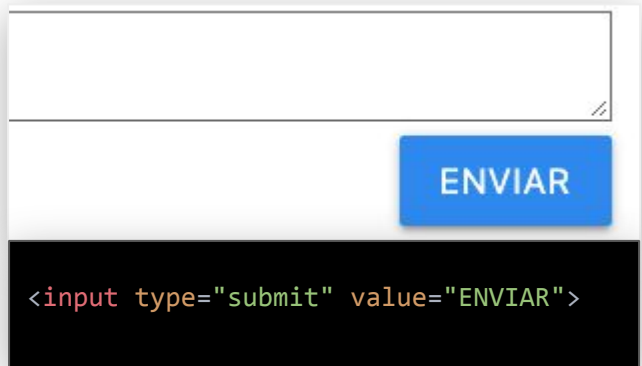
```
<textarea placeholder="Comentarios adicionales, aquí."  
            rows="3" maxlength="100"></textarea>
```

Comentarios adicionales, aquí.

Textarea tiene la capacidad de permitir textos extensos en una o múltiples líneas, tal como se comporta un campo de comentarios. Para hacer visibles múltiples líneas del input, se debe agregar el atributo **rows** y, mediante **maxlength**, limitamos el total de caracteres a ingresar. Ambos atributos esperan un valor numérico.

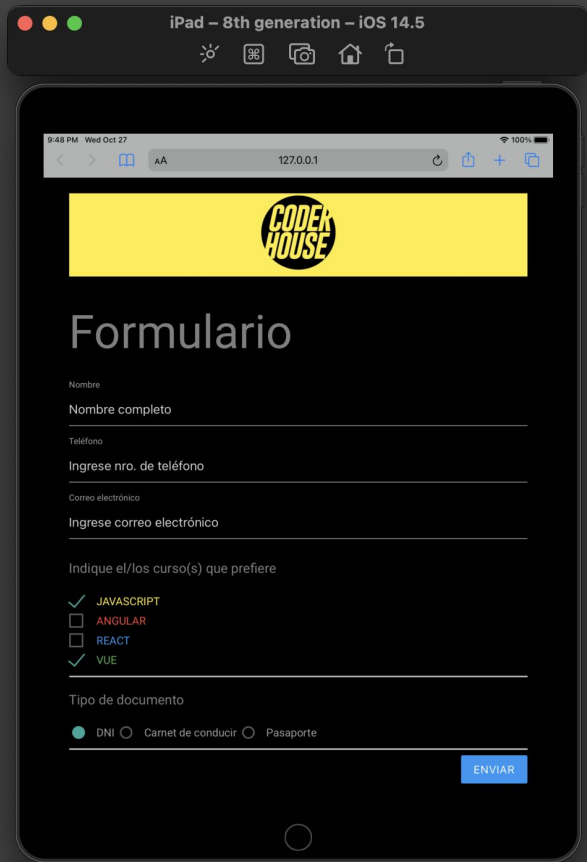


SUBMIT



El input submit despliega un botón en pantalla y su única función, cuando está contenido dentro del tag form, es ejecutar el atributo action configurado en éste último.

También existe `input type="reset"`, botón que limpia los campos del formulario, pero es poco común verlo implementado.



IMPLEMENTACIÓN

Finalmente, el conjunto de input types implementados nos brindarán resultados fabulosos gracias a sus atributos correctamente configurados y aprovechados.



CODER HOUSE

INPUT TYPES Y ATRIBUTOS

Existen otros tantos input types como así también atributos comunes y específicos de cada uno de ellos, que no mencionaremos para no desviar el objetivo de la clase.

Pero, si deseas conocer más, puedes consultar a posteriori el contenido extra que te acercamos, a través del [material ampliado](#) compartido al final de la clase.