**Formularios:**

• Elementos de HTML que nos permiten pedirle datos al usuario.

• Los datos luego se pueden validar y procesar.

**FUNCIONAMIENTO**: se compone de 3 instancias: HTML (capturar), JS+Node.js (validar) y Node.js (procesar).

**Etiquetas**: la parte del HTML.

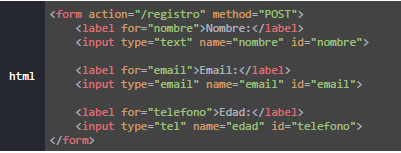
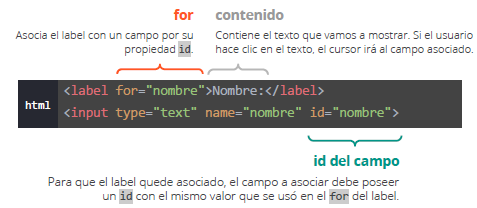
• Form: indicamos que vamos a crear un formulario en el documento HTML. Como atributos encontramos:

* *Action*: define la ruta en donde se va a procesar la información capturada.
* *Method*: define cómo se enviará la información. En la versión actual de HTML, hay solo dos valores posibles: *GET* y *POST*.

• Input: elemento multifunción. Sus atributos son:

* *Type*: distintos tipos de campos.
* *Name*: identifica el campo y lo diferencia de los demás campos. Es importante saber que los elementos que no tengan name, no se envían.

• Label: sirve para asociar un texto descriptivo a un campo determinado.



• Textarea: permite texto de gran tamaño y con múltiples líneas. A diferencia del input, el textarea tiene etiqueta de apertura y cierre.

• Select: permite agregar un componente que muestra opciones. Como el resto de los campos, debe llevar el parámetro name para enviarse.

• Option: opciones de un select son representadas por elementos option. Tienen etiqueta de apertura y cierre.



• Button: generar un botón. Con el atributo *type* definimos el tipo y los valores pueden ser:

* *Reset*: reinicia el formulario a su estado inicial.
* *Submit*: se encarga de enviar el formulario.
* *Button*: no realizará ninguna acción por defecto. Por lo general, programaremos una con ayuda de JS.

**FORMULARIOS ACCESIBLES:**

• Situar correctamente <label>:

* *Antes del campo para*: text, textarea, select, password, file. Puede ir a la izquierda del campo o arriba del campo.
* *Detrás del campo para los tipos*: checkbox (multiseleccion) o radio button (excluyente).

• Etiquetar los controles de forma correcta:

* *Label for*: relación entre la etiqueta y el control del formulario al que pertenece debe estar siempre asociada con for.
* *For*: debe ser siempre igual al id del control del formulario y cada id debe ser único.

• Label no debe usarse para los campos:

* *Image*: puede ser el caso de un botón enviar, en estos elementos utilizamos el atributo alt.
* *Submit y reset*: debemos usar value.
* *Buttons*: para los button en los que su contenido se usa como una etiqueta.

• Etiquetas de agrupamiento: cuando sea necesario y/o aclaratorio.

* *Fieldset*: permite agrupar cuando haya varios elementos de formulario asociados. Cada fieldset debe tener una leyenda (*Legend*: texto que describe el grupo de asociados de los elementos de formulario).
* *Optgroup*: permite agrupar las opciones de selección de una lista de un formulario. La leyenda es obligatoria en este elemento y debería proveer a los usuarios una pista o descripción acerca de los contenidos del grupo.
* *Recomendado*: La agrupación de opciones en un control select puede mejorar la legibilidad y la facilidad de uso.

**CAMPOS OBLIGATORIOS:**

**Identificación correcta de campos obligatorios:**

• Recomendado:

* *Obligatorio o required*: poniendo directamente el texto “obligatorio” vamos a evitar más confusiones en la interpretación. Ej.: “Email (obligatorio)”.
* Identifiquemos los errores marcando el foco con cambio de color en el campo que se ha producido, y además describiendo con texto el error cometido.
* *Textos de los mensajes de validación*: deben ser lo más descriptivos posible y aclarar al usuario lo que está sucediendo. Ej.: “El email que completaste no es correcto, por favor, introduce algo similar a mimail@gmail.com”.

• No recomendado:

* *\**: usar el \* para indicar que un campo es requerido. Se debe añadir el texto “Los campos con \* son obligatorios” o similar.
* No identifiquemos un campo obligatorio solo con color. Esto podría no funcionar para las personas con daltonismo.
* *Textos de los mensajes de validación*: ej.: “Se han dejado campos obligatorios sin completar”. Con este tipo de mensaje es complicado que el usuario sepa en qué ha fallado.

**Tabulación y selección por teclado:**

• Tabindex: una de las formas más comunes de la navegación a través de un formulario es utilizando la tecla Tab. Una persona llena un campo, presiona la tecla Tab, rellena el campo siguiente, y así sucesivamente, hasta que se alcanza el final del formulario. Para lograrlo debemos establecer el orden en que se asignan esas tabulaciones de la siguiente manera: ej.: <a href="documento2.html" tabindex="2" style="float: right;">Segundo documento.</a>

<a href="documento1.html" tabindex="1">Primer documento.</a>

* *Nº entero*: indica la posición del elemento en la navegación por tabulación. Si varios elementos comparten el mismo tabindex, su orden relativo sigue la posición relativa en el documento.
* *Valor es negativo*: el elemento se encontrará excluido de la navegación por tabulación. Ej.: usualmente tabindex="-1".
* *Atributo es omitido*: el navegador determinará automáticamente la posición de todos los elementos capaces de recibir el enfoque de acuerdo a su posición en el código.

**FORMULARIOS AVANZADOS:**

**Input de fecha**: permite ingresar una fecha incluyendo año, mes y día. Para crear un elemento de tipo date usamos un input a cuyo atributo type le damos el valor date. Dentro de los atributos del elemento, la fecha siempre se formatea como aaaa-mm-dd (ej.: 2020-10-25). Por fuera tomará el formato que corresponda según el idioma del usuario. Entre *propiedades y valores* encontramos:

• Type: crea un input de tipo date.

• Value: si necesitamos mostrar una fecha por defecto, usamos este atributo.

• Min: opcional, nos permite definir la fecha mínima que puede ser ingresada.

• Max: opcional, nos permite definir la fecha máxima que puede ser ingresada.

* Ej.: <input type="date" name="fecha" value="2021-07-22" min="2018-01-01" max="2022-12-31">.

**Input de archivos**: permite cargar uno o más archivos desde la computadora. Para crear un elemento de tipo file usamos un input a cuyo atributo type le damos el valor file. Entre *propiedades y valores* encontramos:

• Type: crea un input de tipo file.

• Accept: permite definir qué tipo de archivos serán aceptados.

• Multiple: por defecto solo permite un archivo, si deseáramos aceptar más de uno, usamos multiple.

* Ej.: <input type="file" name="avatar" accept=".jpg" multiple>.

**RADIO BUTTONS**: al hacer clic en 🔘, éste queda seleccionado. Para crear un elemento de tipo radio usamos un input cuyo atributo type tiene el valor radio. Entre *propiedades y valores* encontramos:

• Type: crea un botón de tipo radio.

• Name: para que el usuario pueda seleccionar solo una opción tenemos que asignarle el mismo nombre a todos los

inputs de tipo radio que sean del mismo grupo.

• Value: va la información que se enviará si el usuario selecciona este radio-button.

• Checked: opcional, se encarga de preseleccionar la opción.

* Ej.: <input type="radio" name="medio" value="Efectivo">

<input type="radio" name="medio" value="Débito" checked>

**CHECKBOXES**: representa una opción ⏹, al hacer clic en él queda seleccionado y con otro clic quitamos la selección.

Para crear un elemento de tipo checkbox usamos un input cuyo atributo type tiene el valor checkbox. Podemos agrupar elementos de tipo checkbox y, en ese caso, se podrán elegir múltiples opciones del conjunto. Entre *propiedades y valores* encontramos:

• Type: crea un botón de tipo checkbox.

• Name: para que el usuario pueda seleccionar más de una opción, tenemos que asignarles el mismo nombre a todos los inputs de tipo checkbox que sean del mismo grupo.

• Value va la información que se enviará si el usuario selecciona este checkbox.

• Checked: opcional, se encarga de preseleccionar la opción.

* Ej.: <input type="checkbox" name="hobbies" value="Pintura">

<input type="checkbox" name="hobbies" value="Juegos" checked>

**Uso particular**: utilizarlo para recolectar información que responde a preguntas con respuestas como sí/no porque permite tildar o destildar en un solo elemento. En ese caso no es necesario agregar un valor, ya que, en caso de seleccionarse la opción, llegará con el valor “on”. Ej.: <label for="terminos">Acepto los términos.</label>

<input type="checkbox" name="terminos">