

## EXERCISES QUE TRABAJAREMOS EN LA CUE

- EXERCISE 1: ETIQUETAS PARA CREAR UN FORMULARIO.
- EXERCISE 2: TIPOS DE DATOS DE UN FORMULARIO.
- EXERCISE 3: VALIDACIÓN DE FORMULARIO CON EXPRESIONES REGULARES.

### EXERCISE 1: ETIQUETAS PARA CREAR UN FORMULARIO

El uso de los formularios nos permitirá crear ese espacio para interactuar con el usuario, que nos otorgará mayor valor a nuestras páginas web.

Varias son las etiquetas que se utilizan para crear un formulario, cada una de ellas con sus respectivos atributos obligatorios o no.

Para ejemplificarlo mejor, vamos a realizar nuestro primer formulario web en el cual solicitaremos al usuario ingresar su nombre y su dirección y enviar aquella información.

Para eso, lo primero que debemos hacer es generar un proyecto con la estructura básica y luego, nos centraremos en conocer las etiquetas a utilizar.

#### ETIQUETAS PARA FORMULARIO

La primera etiqueta que debemos tener en cuenta es la etiqueta `<form>`. Esta etiqueta es la que englobará todo el resto de las etiquetas que componen un formulario.

```
1 <form action="">
2
3 </form>
```

Esta etiqueta cuenta con la posibilidad de utilizar en ella dos atributos: `action` y `method`. Estos atributos van situados en la etiqueta de apertura (como ocurre en todos los casos que usemos atributos en etiquetas) y si bien `action` es un atributo que usaremos siempre, `method` puede ser ignorado.

Para verlos un poco más en detalle diremos que:

- **Action:** Establece la acción que se realizará una vez enviado el formulario. Puede ser enviar un mensaje, direccionar a otra página, descargar un archivo, cambiar la interfaz gráfica o cualquier acción asociada al envío de formulario.

- **Method:** Este atributo indica la forma en que se envían los datos recogidos en el formulario. Esto quiere decir, la forma en que la información que el usuario ingresó es enviada a capas posteriores que incluyan algún lenguaje de programación. Los métodos para enviar la información son **GET**, que lleva la información en la **URL** y **POST**, que los lleva ocultos en la petición. Si no se define nada, por defecto será **GET**.

Dentro de las etiquetas `<form>` y tal como explicábamos previamente, vamos a colocar todo el cuerpo del formulario, partiendo por la etiqueta `<label>`. Esta etiqueta tiene la función de identificar o relacionar el campo `<input>` (que veremos a continuación) con su texto, es decir, con el texto que el usuario ve como guía, indicando que información debe ingresar. Cuenta con el atributo `for`, este atributo será el que “enlazaremos” con el atributo `name` de la siguiente etiqueta `<input>` para indicar que se relacionan.

```
1 <label for="nombre">NOMBRE</label>
```

Luego, debajo de esta etiqueta, colocaremos la etiqueta `<input>`. Esta etiqueta nos genera el cuadro de texto donde el usuario ingresará la información solicitada.

```
1 <input type="text" name="nombre" id="nombre">
```

Acá podemos observar que tiene tres atributos (pares nombre – valor). El primero de ellos es el atributo `type`, cuya función es indicar el tipo de dato que será permitido ingresar en ese campo (lo veremos en detalle más adelante, por el momento mantendremos el valor por defecto “**text**”).

Luego tenemos el atributo `name`, el cual se relaciona con el atributo `for` de la etiqueta `<label>` para indicar texto e `input` al que pertenece. Finalmente, el atributo `id`, utilizado como identificador único para poder, en caso de ser requerido, capturar el dato ingresado a ese `<input>` o capturar el elemento como tal.

Abramos nuestro proyecto en el navegador y veamos que tenemos hasta ahora.

NOMBRE

Tenemos nuestra primera caja de texto en la cual podemos ingresar información y que, gracias a nuestro `<label>` nos indica que lo que debemos ingresar es un nombre.

Si hacemos clic sobre el texto “nombre”, nos enviará de inmediato al campo destinado para escribir ese dato. Esa es la función de los atributos `for` y `name` que hemos utilizado para “relacionar” tags.

Continuaremos creando el segundo campo, en el cual solicitaremos la dirección, haciéndolo de la misma forma que en el campo nombre.

El código quedará de la siguiente forma:

```
1 <label for="direccion">DIRECCIÓN</label>
2 <input type="text" name="direccion" id="direccion">
```

Finalmente, para poder hacer el envío, debemos colocar un botón, así que, debajo del último `<input>`, colocaremos la etiqueta `<button>` y entre apertura y cierre escribiremos “enviar”. El código completo quedará de la siguiente forma:

```
1 <!DOCTYPE html>
2 <html lang="es">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-
8 scale=1.0">
9     <title>Primer Formulario</title>
10 </head>
11
12 <body>
13     <!-- MI PRIMER FORMULARIO SOLICITA NOMBRE Y DIRECCIÓN-->
14     <form action="">
15         <label for="nombre">NOMBRE</label>
16         <input type="text" name="nombre" id="nombre">
17         <label for="direccion">DIRECCIÓN</label>
18         <input type="text" name="direccion" id="direccion">
19         <button>ENVIAR</button>
20     </form>
21
22 </html>
```

Guardaremos todos los cambios realizados. Debemos tener siempre presente que cada cambio que realicemos debemos ir guardándolo, de lo contrario, no se verán reflejados en nuestra página web.

Actualizaremos nuestra página web.

NOMBRE  DIRECCIÓN

Finalmente, intentemos enviar el formulario.

← → ↻ file:///D:/Users/Catalina/Downloads/PruebaModuloTres/bla.html?nombre=Catalina&direccion=Siempre+Viva+123  
 NOMBRE  DIRECCIÓN

Si observamos la parte destacada en la **URL**, podremos ver los datos “nombre = Catalina & dirección = Siempre + Viva + 123”.

Cuando escribimos nuestro código fuente para crear el formulario, no asignamos información para el método **HTTP** que se utilizará, lo que hace que por defecto haya tomado el método **GET** y los datos se envíen a través de la **URL**.

Cambiemos en nuestro formulario el **method**, eligiendo **POST**.

```
1 <form action="" method="post">
```

Guardaremos los cambios, actualizaremos nuestra página web y volveremos a enviar la información.

Si observamos la **URL** ahora, no nos mostrará la información, ya que esta se envía en el cuerpo y no en la **URL** cuando usamos **POST**.

file:///D:/Users/Catalina/Downloads/PruebaModuloTres/bla.html?

## OTRAS ETIQUETAS DE FORMULARIO

Las etiquetas que vimos hasta el momento representan las etiquetas básicas para trabajar con formularios, pero, sin embargo, existen otras etiquetas que nos pueden permitir dar aun mejores servicios a los usuarios que eventualmente visiten nuestro sitio web.

Si quisiéramos, por ejemplo, exponer una lista para que un usuario pueda elegir un valor, tenemos la opción de utilizar la etiqueta **<datalist>**. Este elemento representa una lista de elementos que se colocan a disposición utilizando la etiqueta **<option>** en conjunto.

Para usarla vamos a colocar otra etiqueta **<label>** indicando “color favorito” y una etiqueta **<input>** que dentro de sus atributos tendrá el atributo **list** con el valor “colores”. A través de este valor se relacionará con la lista, que tendrá **id** “colores”.

Vamos a usar la etiqueta **<datalist>** con el **id** “colores” y entre sus etiquetas de apertura y cierre, la etiqueta **<option>** con el valor que será opción para el usuario.

Veamos cómo quedará el código:

```
1 <label for="lista">COLOR FAVORITO</label>
```

```
2 <input type="text" name="lista" list="colores">
3 <datalist id="colores">
4   <option label="Rojo" value="rojo"></option>
5   <option value="azul"></option>
6   <option value="amarillo">Amarillo</option>
7 </datalist>
```

Si observamos en detalle, podremos ver que las tres etiquetas `<option>` están escritas de manera distinta.

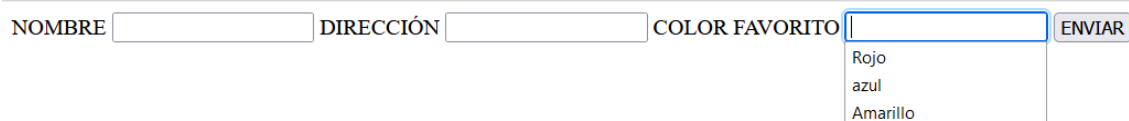
Las tres coinciden en el atributo `value` y el valor asignado. Este atributo es fundamental y es el que indicará el real valor de la etiqueta (con este valor podremos procesar la información seleccionada).

Luego, podemos observar que el primer `<option>` tiene el atributo `label` con valor "Rojo". Este es un texto que verá el usuario.

En el segundo `<option>` solo colocamos el `value` y veremos posteriormente que hace que se vea distinto uno y otro para el usuario.

Finalmente, en el tercer `<option>` utilizamos la opción de colocar la información entre las etiquetas para ser mostrada al usuario.

Si observamos como se ve nuestro formulario en el navegador de **Mozilla Firefox**, encontraremos lo siguiente:



NOMBRE  DIRECCIÓN  COLOR FAVORITO   
Rojo  
azul  
Amarillo

ENVIAR

Mientras que en navegador **Google Chrome**, se verá de la siguiente manera:



NOMBRE  DIRECCIÓN  COLOR FAVORITO   
rojo  
Rojo  
azul  
amarillo  
Amarillo

ENVIAR

En un navegador no tenemos ninguna diferencia con las tres formas de escribir la información, pero en el otro navegador sí.

Comúnmente cuando desarrollamos estaremos atentos a otorgar una buena experiencia para el usuario en cualquiera sea el navegador que decida usar.

Ahora bien, con esto ejemplificamos como usar una lista, mostrando otras etiquetas posibles en un formulario.

Pero no es la única etiqueta que podemos utilizar para crear una lista. Escribiremos nuevamente un **<label>** que indicará ahora seleccionar a un animal favorito y luego haremos uso de la etiqueta **<select>** dentro de la cual usaremos nuevamente la etiqueta **<option>** con las opciones a desplegarse.

El código quedará de la siguiente forma:

```
1 <label for="animales">ANIMALES FAVORITOS</label>
2   <select name="animales" id="animales">
3     <option value="perro">Perro</option>
4     <option value="gato">Gato</option>
5     <option value="conejo">Conejo</option>
6   </select>
```

Y nuestro navegador se verá de la siguiente manera:

NOMBRE  DIRECCIÓN  COLOR FAVORITO  ANIMALES FAVORITOS 

Perro

Perro

Gato

Conejo

## ACCION DEL FORMULARIO

El último punto que nos queda por analizar entre los puntos básicos de un formulario es la acción que se realizará al ser enviado.

Para este caso, vamos a crear un segundo archivo **HTML** al que llamaremos **redireccion.html**, y al cual accederemos cuando el formulario sea enviado.

En ese archivo solo colocaremos la estructura básica de **HTML** y una etiqueta **<h1>** que dirá "gracias por enviar el formulario".

En nuestro archivo principal, donde hemos estado trabajando el formulario, vamos a modificar el atributo **action** de la etiqueta **<form>** y colocaremos ahí la dirección a la cual enviaremos al usuario una vez enviado su formulario: **redirección.html**.

El código fuente completo quedará así:

```
1 <!DOCTYPE html>
```

```
2 <html lang="es">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-
8 scale=1.0">
9     <title>Primer Formulario</title>
10 </head>
11
12 <body>
13     <!-- MI PRIMER FORMULARIO SOLICITA NOMBRE Y DIRECCIÓN-->
14     <form action="redireccion.html" method="post">
15         <label for="nombre">NOMBRE</label>
16         <input type="text" name="nombre" id="nombre">
17         <label for="direccion">DIRECCIÓN</label>
18         <input type="text" name="direccion" id="direccion">
19         <label for="lista">COLOR FAVORITO</label>
20         <input type="text" name="lista" list="colores">
21         <datalist id="colores">
22             <option label="Rojo" value="rojo"></option>
23             <option value="azul"></option>
24             <option value="amarillo">Amarillo</option>
25         </datalist>
26         <label for="animales">ANIMALES FAVORITOS</label>
27         <select name="animales" id="animales">
28             <option value="perro">Perro</option>
29             <option value="gato">Gato</option>
30             <option value="conejo">Conejo</option>
31         </select>
32         <button>ENVIAR</button>
33     </form>
34
35 </html>
```

Y en nuestro navegador actualizaremos el formulario, escribiremos los datos y cliquearemos el botón “enviar”, resultando en una redirección a la página que hemos indicado:

## GRACIAS POR ENVIAR EL FORMULARIO

De esta forma podemos decir que hemos aprendido a crear formularios **HTML**.

## EXERCISE 2: TIPOS DE DATOS DE UN FORMULARIO

El atributo **type** es el atributo que utilizamos para indicar el tipo de dato que queremos que sea ingresado en el formulario y nos permite de esta forma validar que se ingrese aquel que estamos solicitando.

Hasta ahora, hemos trabajado con el valor por defecto **text**, que es aquel valor que permite el ingreso de cualquier tipo de carácter.

Veamos los distintos tipos disponibles:

- **Text:** un campo de texto. Permite un campo de texto que acepta cualquier carácter.

```
1 <label for="nombre">NOMBRE</label>
2 <input type="text" name="nombre" id="nombre">
```

Nombre

- **Button:** nos crea un botón. Para darle el texto "Click me" usamos **<label>** y para dar el valor texto "Botón" utilizamos el atributo **value**.

```
1 <label for="">Click me</label>
2 <input type="button" value="Botón">
```

Click me

- **Checkbox:** para que el usuario seleccione una o varias opciones disponibles. Se coloca un **<input>** con **value** distinto por cada opción, además, un **<label>** con cada opción que será visualizada por el usuario.

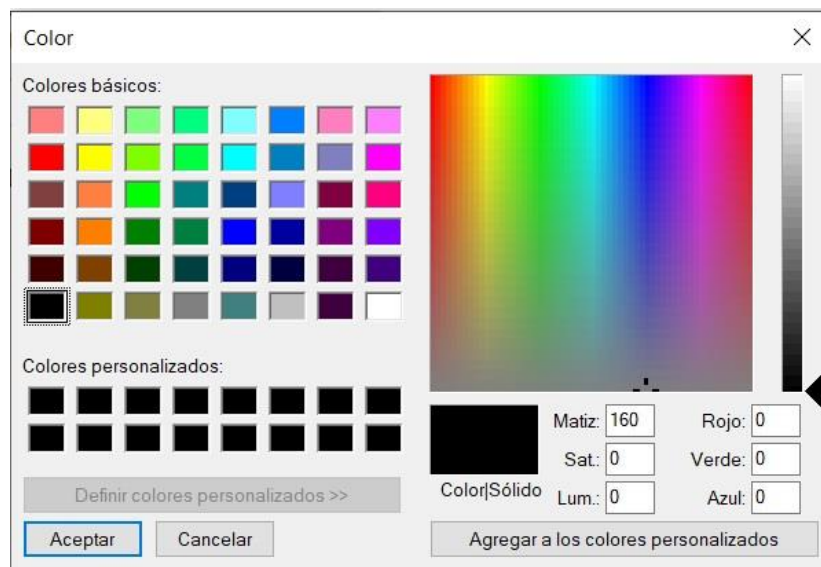
```
1 <label for="">Verde</label>
2   <input type="checkbox" value="verde" name="color">
3   <label for="">Amarillo</label>
4   <input type="checkbox" value="amarillo" name="color">
5   <label for="">Rojo</label>
6   <input type="checkbox" value="rojo" name="color">
```



Verde ☒ Amarillo ☐ Rojo ☐

- **Color:** genera un selector de colores. Este tipo se verá distinto en cada navegador.

```
1 <input type="color" >
```



- **Date:** obliga al usuario a incorporar una fecha en formato día – mes – año seleccionando la fecha en el calendario que se despliega.

```
1 <input type="date" >
```

DATO

<

noviembre de 2019

>

lun.	mar.	mié.	jue.	vie.	sáb.	dom.
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

- **Datetime o datetime-local:** igual que el tipo date, pero obliga también a seleccionar una hora.

```
1 <input type="datetime-local" >
```

DATO

enero de 2016

↑

↓

lu	ma	mi	ju	vi	sá	do	19	00
28	29	30	31	1	2	3	20	01
4	5	6	7	8	9	10	21	02
11	12	13	14	15	16	17	22	03
18	19	20	21	22	23	24	23	04
25	26	27	28	29	30	31	00	05
1	2	3	4	5	6	7	01	06

Hoy

- **Email:** fuerza a que el dato ingresado contenga como formato de email un signo @. Si no se ingresa el dato como corresponde marca la casilla en rojo.

```
1 <input type="email" >
```

DATO

- **Password:** Se usa para campos que tienen que ir cifrados, al escribir se muestra puntos no lo que se escribe.

```
1 <input type="password" >
```

DATO

- **Number:** Fuerza a que la entrada sea un número

```
1 <input type="number" >
```

DATO

- **Submit:** envía los datos de la consulta.

```
1 <input type="submit" value="Enviar consulta" >
```

DATO

- **Radio:** Los botones de opción se presentan normalmente en grupos de opciones (una colección de botones de opción que describen un conjunto de opciones relacionadas). Solo se puede seleccionar un botón de opción en un grupo al mismo tiempo. Deben compartir el mismo **name** para ser un grupo.

```
1 <input type="radio" value="Verde" name="color" >  
2 <input type="radio" value="Amarillo" name="color" >  
3 <input type="radio" value="Rojo" name="color" >
```

Verde ☐ Amarillo ☐ Rojo ☒


- **File:** Permite subir un archivo, como una imagen o documento.

```
1 <input type="file" >
```

Ingrese archivo adjunto  No se eligió archivo

- **URL:** Permite ingresar una URL, impidiendo el envío de datos si no cumple con el formato.

```
1 <input type="url">
```

URL    
 Ingresa una URL.

- **Time:** obliga al usuario a ingresar una hora seleccionada.

```
1 <input type="time">
```

HORA    

15	53
16	54
17	55
18	56
19	57
20	58
21	59

- **Month:** permite seleccionar un mes y un año específico.

```
1 <input type="month">
```

MES  de  

2021

ene.	feb.	mar.	abr.
may.	jun.	jul.	ago.
sept.	oct.	<b>nov.</b>	dic.

[Borrar](#) [Este mes](#)

- **Range:** permite ingresar un rango. Trabaja con los atributos **max** y **min** para indicar un máximo y un mínimo.

```
1 <input type="range" max="10" min="1">
```

RANGO

Cada una de las opciones de los types que muestran un cuadro de selección, como color, date, range, etc. Se mostrarán distintos en cada navegador. Es por esta razón que de manera correcta debemos evaluar la grafica para cada una de las opciones.

Podemos encontrar algunos tipos más, con los cuales podemos jugar para conocerlos mejor.

### ATRIBUTOS ÚTILES EN UN FORMULARIO

Vimos previamente algunos atributos útiles para un formulario, pero lo hicimos a grandes rasgos. Veamos algunos otros atributos:

- **Maxlength:** nos permite definir el número máximo de caracteres permitidos.
- **Pattern:** nos permite definir los caracteres que serán permitidos, por ejemplo, solo letras. En ese caso si un usuario ingresa números será erróneo. Se utiliza en conjunto con una expresión regular.
- **Checked:** usado con los **type radio** y **checkbox**, nos permite indicar que opción esta seleccionada por defecto.

- **Min y max:** establece límite superior e inferior. Puede ser utilizado para fechas o números.
- **Placeholder:** nos permite definir un valor que se verá previamente en la caja de texto y que servirá para guiar al usuario en el valor que debe ingresar.
- **Required:** permite indicar que un valor es requerido, es decir, que ese campo debe ser completado obligatoriamente. Impide el envío del formulario si el campo no se ha completado.
- **Title:** nos permite indicar un texto que se verá al posar el mouse sobre el elemento. Generalmente se utiliza para guiar al usuario.

Ahora que conocemos estos atributos podemos usarlos para practicar su función.

### EXERCISE 3: VALIDACIÓN DE FORMULARIOS CON EXPRESIONES REGULARES

Continuando con el trabajo con formularios y pensando en la validación de los datos que el usuario ingresa, es decir, considerando la posibilidad de que el usuario no ingrese los datos seleccionados y debamos revisarlo para evitar errores al momento del almacenamiento es que trabajaremos con expresiones regulares.

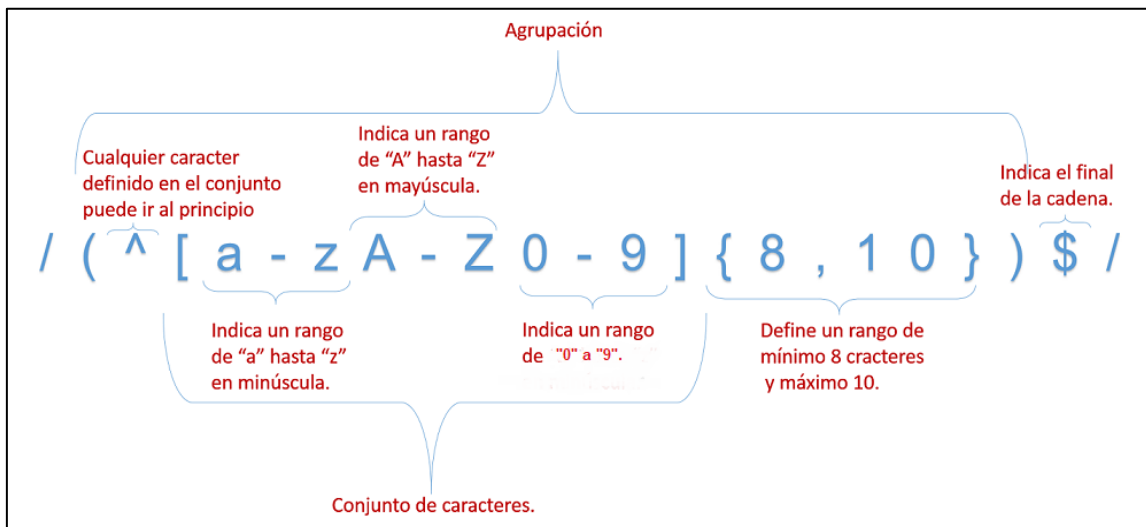
Las expresiones regulares son cadenas de texto que definen el formato de texto que queremos que se pueda ingresar en un **input** determinado. Permiten filtrar textos para encontrar coincidencias.

#### SINTAXIS DE LAS EXPRESIONES REGULARES

Lo primero que conoceremos será algunos de los caracteres más utilizados y su significado.

CARÁCTER	SIGNIFICADO
<b>^</b>	Indica el principio de una cadena
<b>\$</b>	Indica el final de una cadena
<b>()</b>	Agrupamiento de parte de una expresión
<b>{}</b>	Indica número o intervalo de longitud de la expresión
<b>[]</b>	Un conjunto de caracteres de la expresión
<b>.</b>	Cualquier carácter con excepción del salto de línea
<b>\</b>	Para escribir un carácter especial y que sea tratado como un literal

Continuamos conociendo la sintaxis de una expresión regular, observemos la siguiente imagen.



- **a-z:** Nos indica que es aceptado el rango de letras de la a hasta la z en minúsculas.
- **A-Z:** Nos indica que es aceptado el rango de letras de la A hasta la Z en mayúsculas.
- **0-9:** Indica que es aceptado un rango de números de 0 a 9.
- **{8,10}:** Define la cantidad de caracteres que son aceptados como mínimo y como máximo.

Hasta ahora, hemos definido que es una expresión regular y su sintaxis, pero, nos queda llevar a cabo la validación.

### ¿CÓMO PODEMOS LLEVAR A CABO UNA VALIDACIÓN CON EXPRESIÓN REGULAR?

En este punto conoceremos el atributo **Pattern**. Este atributo lo utilizaremos en la etiqueta **<input>** y nos permitirá definir un patrón de ingreso de datos ayudándonos de expresiones regulares.

El punto negativo de la validación es que puede ser violada por los usuarios, es por lo que, al momento de trabajar en un programa complejo, al lado del servidor también se deben realizar validaciones, además de las validaciones realizadas en la capa cliente.

Para conocer como trabajar con este atributo y las expresiones regulares, veremos un pequeño ejemplo.

Crearemos un proyecto y en él trabajaremos en un archivo de extensión **HTML**. Luego de tener escrita la estructura básica del documento, procederemos a utilizar las etiquetas **<form>** y dentro las etiquetas **<label>** e **<input>**.

Veremos una de las validaciones más comunes que necesitaremos realizar: un correo electrónico.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-
7 scale=1.0">
8     <title>Document</title>
9 </head>
10 <body>
11
12     <form action="">
13
14         <label for="correo">correo</label>
15         <input type="email" name="correo" required>
16
17
18         <input type="submit" value="Boton">
19
20     </form>
21
22 </body>
23 </html>
```

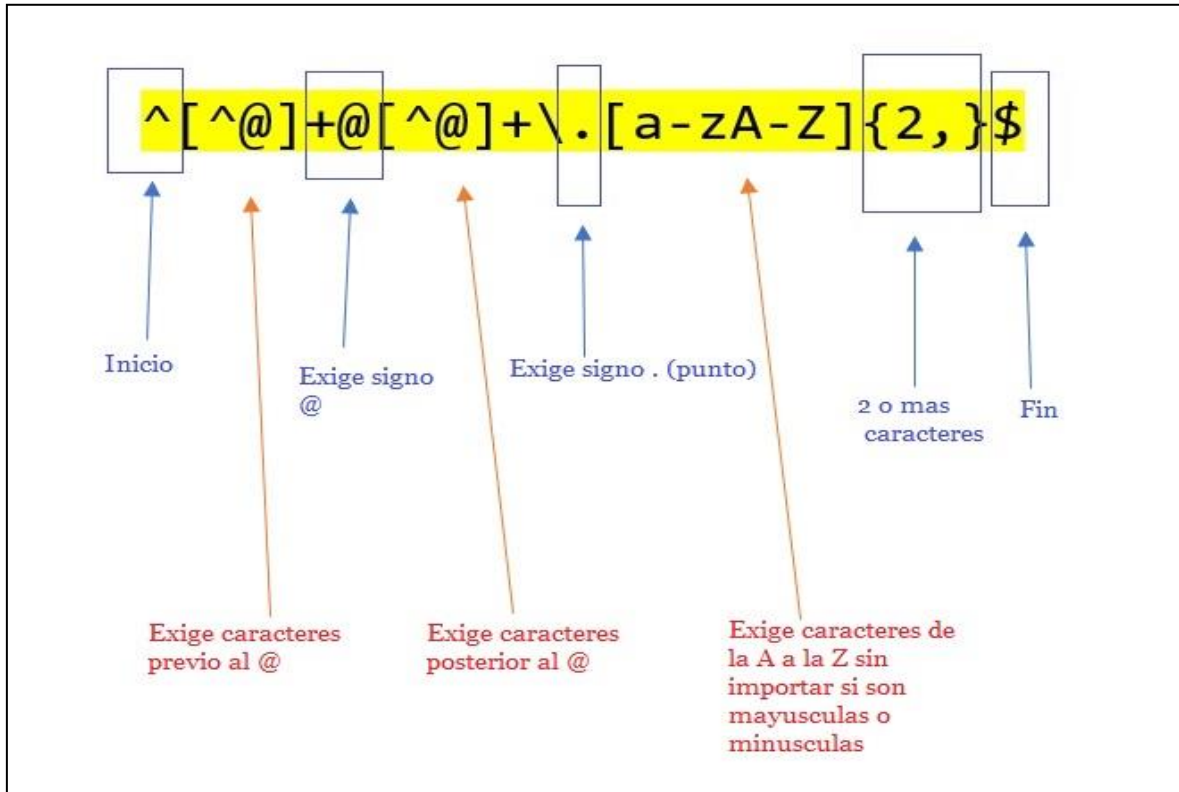
Si observamos nuestro código, podemos notar que hemos utilizado el **type email** que, como vimos previamente, nos exige ingresar un correo electrónico en el formato **xx@xx** sin exigirnos un dominio.

Es este el momento en que podemos hacer uso de una expresión regular, incorporando el atributo **pattern** seguido de la expresión regular que deseemos.

```
1 <form action="">
2
3     <label for="correo">correo</label>
4     <input type="email" name="correo" pattern="^[^@]+@^[^@]+\.[a-
5 zA-Z]{2,}$" required>
6
7
8     <input type="submit" value="Boton">
9
10 </form>
```



En la expresión regular que estamos utilizando indicamos:



*Desglose de nuestra expresión regular*

Veamos que ocurre en el navegador al implementar esta expresión regular:

correo  Boton

! Haz coincidir el formato solicitado.

A diferencia de cuando usábamos el atributo `type="email"` solo, en esta ocasión no nos permite hacer el `submit` sin colocar un dominio.

correo

Boton

Al completar el formato solicitado, nuestra información es aceptada. De esta forma hemos validado que ingrese un correo electrónico valido, valga la redundancia, según nuestra indicación.

Podemos utilizar esta expresión regular con el type email o text de igual manera.

Conozcamos una expresión regular más: En todos los países existe un código único para ciudadano, en Chile el RUT (que es el código único) tiene el formato XX.XXX.XXX-X o simplemente XXXXXXXX-X.


Para validar que se ingrese adecuadamente este formato, podemos utilizar la siguiente expresión regular.

```
1 <form action="">
2
3     <label for="rut">Rut</label>
4     <input type="text" name="rut" pattern="^[0-9]+-[0-9kK]{1}$"
5 required >
6
7
8     <input type="submit" value="Boton">
9
10 </form>
```

En el navegador nos encontramos con lo siguiente:

Rut

Boton

 Haz coincidir el formato solicitado.

*Formato de rut rechazado*

Al ingresar el RUT con un formato distinto al solicitado en la expresión regular nos rechaza el envío de datos.

Rut

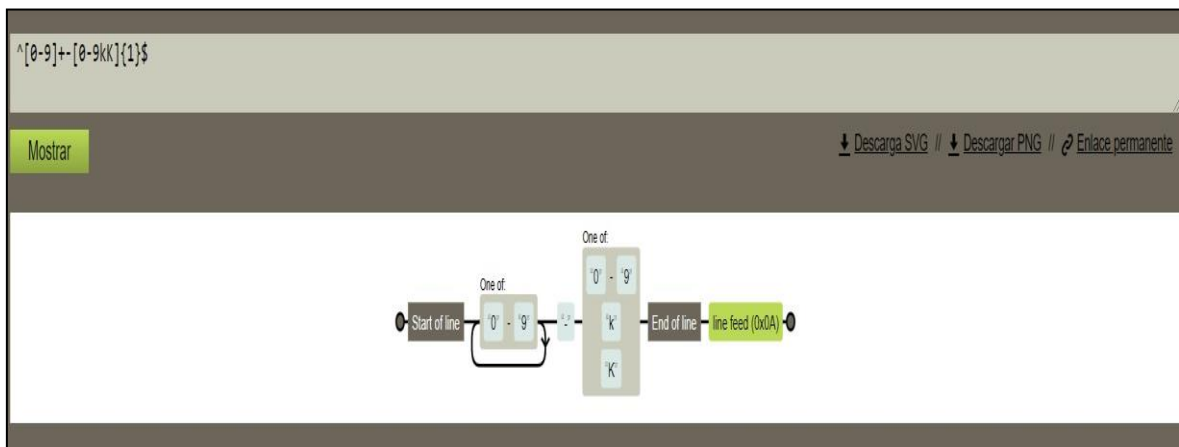
Boton

*Formato aceptado*

Al ingresar el formato aceptado podemos realizar el **submit**.

Es importante destacar que podemos crear cualquier validación de cualquier formato que nosotros deseemos, sin limitaciones. Además, puedes encontrar la validación que desees buscando un poco por internet.

Podemos validar nuestras expresiones regulares en la página [Regexper](#) que nos permite visualizar gráficamente cada elemento del patrón que hayamos confeccionado.



Además, tenemos páginas como [RegExr](#) y [Regexpal](#) para realizar validaciones.

Ahora podemos crear nuestros formularios y validar que el dato que el usuario ingrese corresponda con el formato que nosotros necesitamos.