

## Paso a paso para ejecución correcta de la clase validación:

**Realizado por:** Johan Felipe Garcia Salazar

1. Descargar archivo en formato .zip que se encontrara en la bandeja de correo con el asunto "Actividad Expresiones regulares – Johan Garcia)
2. Al descargarlo, descomprimir el archivo donde deberá quedar 1 carpeta llamada "ACTIVIDAD\_III\_TRIMESTRE" y en ella 4 archivos, el formulario (index.html), la clase en JavaScript (validacion.js), el estilo personalizado (style.css) y por último este documento que esta leyendo (Documento paso a paso para ejecución de la clase).
3. Luego se debe proceder a abrir el entorno de código Visual Studio Code, y luego presionar la opción archivo de la barra de herramientas de la parte superior izquierda, que desplegará una serie de opciones donde se deberá elegir la opción abrir carpeta, luego de presionar esta opción deberá abrirse el explorador de archivos, donde debe seleccionar la carpeta ya descomprimida previamente llamada "ACTIVIDAD\_III\_TRIMESTRE".
4. Una vez abierto el documento se puede proceder a la implementación de la clase, en caso de que el formulario donde se quiera implementar este en otra ventana de Visual Studio Code, se debe primero abrir el archivo llamado "validacion.js" y copiar todas sus líneas de código (415) o también por supuesto se podrá crear un nuevo archivo dentro de la misma carpeta abierta en esa ventana de Visual Studio Code, llamarlo de la forma que desee a excepción de "index.html" ya que de esta manera ya se encontrara llamado el formulario que se encuentra en la carpeta.
5. Luego de decidir de que manera se quiere trabajar si en la misma ventana en que se abre la carpeta descomprimida o entra aparte, se debe proceder a hacer algunos ajustes en el formulario de HTML en el que se desea implementar la clase (Todos estos cambios tomando como base del archivo "index.html" que se encuentra en la carpeta), como:

- Copiar dentro de las etiquetas <head> el CDN de CSS de Bootstrap que controla la mayoría de estilos del formulario. **(OPCIONAL)**

- Copiar link de CSS personalizado dentro de las etiquetas <head> el atributo "href" del link deberá llamarse "style.css" en caso de trabajar en la misma carpeta, sino hacer el mismo proceso del paso 4 en caso de querer trabajar en otra ventana. **(OPCIONAL)**

```
10 <!--Link CSS Personalizado-->
11 <link rel="stylesheet" href="style.css">
```

- Anidar el contenido de las etiquetas <body> en <div> de la misma manera que el formulario que ya se encuentra en la carpeta y cambiar los nombres de las clases de estos y de los demás elementos del HTML, esto con el fin de reflejar mejor el contenido visual. **(OPCIONAL)**

- Agregar la propiedad “novalidate” a la etiqueta <form>, esto con el fin de asegurarse que solo se realicen las validaciones de la clase de JavaScript. **(OBLIGATORIO)**

```
22      <form id="formulario" class="row g-3 needs-validation" novalidate>
```

- Copiar la línea de código (162) del HTML base que contiene en las etiquetas <script> en el enlace al archivo “validacion.js” para que se ejecute la clase. **(OBLIGATORIO)**

```
161      <!--Script de Validación Utilizando Clase-->
162      <script src="validacion.js"></script>
```

- Copiar CDN de JavaScript de Bootstrap. **(OPCIONAL)**

- Agregar <span> debajo del input tipo “range” que corresponde a la edad, esto con el fin de ver el valor exacto del rango en años. **(OPCIONAL)**

```
60      <input type="range" class="form-range" id="edad" min="1" max="100" value="0" required>
61      <span id="num_edad">0</span> años
```

- Ahora lo más importante que es cambiar la propiedad ID de los inputs del formulario, para la correcta obtención posteriormente de esos ID en el código de validación de JavaScript, los ID a modificar serían los siguientes:

- ID de la etiqueta <form>, que se tomara en el JavaScript para realizar un escuchador tipo “submit”, el ID a asignar debe ser “formulario”, como se ve en la imagen:

```
22      <form id="formulario" class="row g-3 needs-validation" novalidate>
```

- ID del input tipo “text” que corresponde al campo nombre, el ID a asignar debe ser “nombre”, como se ve en la imagen:

```
24      <label for="nombre" class="form-label">Nombres</label>
25      <input type="text" class="form-control" id="nombre" mi
```

- ID del input tipo “text” que corresponde al campo de apellidos, el ID a asignar debe ser “apellido”, como se ve en la imagen:

```
28      <label for="apellido" class="form-label">Apellidos</label>
29      <input type="text" class="form-control" id="apellido">
```

- ID del <select> de tipo de documento, el ID a asignar debe ser “tipo\_documento”, como se ve en la imagen:

```
32      <label for="tipo_documento" class="form-label">Tipo de Documento</label>
33      <select id="tipo_documento" class="form-select" required>
```

- En las opciones de este <select> de tipo de documento, más específicamente en la opción de “tarjeta de identidad” es importante

también agregar un ID que luego será manipulado por una validación de la clase, el ID a asignar debe ser “TI”, como se ve en la imagen:

```
35      <option id="TI" value="1">(TI) Tarteja de Identidad</option>
```

- ID del input tipo “number” que corresponde al campo de numero de documento, el ID a asignar debe ser “num\_documento”, como se ve en la imagen:

```
42      <label for="num_documento" class="form-label">Numero de Documento
43      <input type="number" class="form-control" id="num_documento" min=
```

- ID del input tipo “date” que corresponde al campo de fecha de nacimiento, el ID a asignar debe ser “fecha”, como se ve en la imagen:

```
46      <label for="fecha" class="form-label">Fecha de Nacimiento</label>
47      <input type="date" class="form-control" id="fecha" required>
```

- ID del <select> de género, el ID a asignar debe ser “genero”, como se ve en la imagen:

```
50      <label for="genero" class="form-label">Genero</label>
51      <select id="genero" class="form-select" required>
```

- ID del input tipo “range” que corresponde al campo edad, el ID a asignar debe ser “edad”, como se ve en la imagen:

```
59      <label for="edad" class="form-label">Edad</label>
60      <input type="range" class="form-range" id="edad"
```

- ID de la etiqueta <span> que se mencionó anteriormente para mostrar el valor exacto del input tipo “range”, ósea la edad, el ID a asignar debe ser “num\_edad”, como se ve en la imagen:

```
61      <span id="num_edad">0</span> años
```

- ID del input tipo “tel” que corresponde al campo de número de teléfono, el ID asignar debe ser “num\_telefono”, como se ve en la imagen:

```
68      <label for="num_telefono" class="form-label">Numero de Telefono
69      <input type="tel" id="num_telefono" class="form-control" patter
```

- ID del input tipo “email” que corresponde al campo de correo, el ID a asignar debe ser “correo”, como se ve en la imagen:

```
139     <label for="correo" class="form-label">Correo Electronico
140     <input type="email" id="correo" class="form-control" patt
```

- ID del input tipo “password” que corresponde al campo de contraseña, el ID a asignar debe ser “contraseña”, como se ve en la imagen:

```

143 <label for="contraseña" class="form-label">Contraseña</label>
144 <input type="password" id="contraseña" class="form-control"

```

- ID del input tipo “submit” que se encarga del envío del formulario, el ID a asignar debe ser “enviar”, como se ve en la imagen:

```

153 <button class="btn btn-success" id="enviar" type="submit">Enviar</button>

```

- ID del input tipo “reset” que corresponde al botón de limpiar los datos del formulario, el ID a asignar debe ser “limpiar”, como se ve en la imagen:

```

154 <button class="btn btn-danger" id="limpiar" type="reset">Limpiar</button>

```

- Ahora en dos casos puntuales de dos campos del formulario, más específicamente en las habilidades y hobbies, lo que deberemos cambiar no es el ID si no el atributo “name”, donde como veremos a continuación es importante ya que al ser varios input tanto de habilidades como hobbies, el ID no nos funcionara para iterar sobre cada uno de ellos ya que esto es un selector único, por eso se opta con la opción de trabajar con el selector de JavaScript “document . getElementsByName” en donde si tomaremos todos los checkbox o input tipo radio, que tengan el nombre que le vamos a asignar:

- Agregar el atributo “name” a todos los inputs tipo checkbox que corresponden a las habilidades, y asignarle el valor de “status”, como se ve en la imagen:

```

73 <label for="habilidades" class="form-label">Habilidades</label>
74 <div class="form-check">
75 <input class="form-check-input" name="status" type="checkbox">
76 <label class="form-check-label" for="otra">

```

- Agregar el atributo “name” a todos los inputs tipo radio que corresponden al campo de hobbies, y asignarle el valor de “flexRadioDefault”, como se ve en la imagen:

```

106 <label for="hobbies" class="form-label">Hobbies</label>
107 <div class="form-check">
108 <input class="form-check-input" type="radio" name="flexRadioDefault">
109 <label class="form-check-label" for="programar">

```

- Ya para finalizar con las modificaciones que se deben realizar en el HTML base en el que se quiere probar la clase de validación es cambiar la propiedad “value” de los <select> (tipo de documento y genero) y el input tipo range (edad), esto con el fin de que salgan satisfactoriamente todas las validaciones ya que estos valores son utilizados en las respectivas condicionales de los métodos de la clase, hacer los cambios a continuación:

- En el las opciones del <select> de tipo de documento hacer las siguientes asignaciones de “value”:
  - Al <option> de “seleccionar” agregar el value=”0”
  - Al <option> de “tarjeta de identidad” agregar el value=”1”
 Las demos values de las demás opciones no serán relevantes en la ejecución de la validación por ende no se especifican, se deja una imagen para ver como irán estas dos opciones del <select>:

```

33 <select id="tipo_documento" class="form-select" required>
34 <option value="0" selected disabled>Seleccionar</option>
35 <option id="TI" value="1">(TI) Tarteja de Identidad</option>

```

- En las opciones del <select> de genero hacer las siguientes asignaciones de “value”

- Al <option> de “seleccionar” agregar el value=”0”

Las demos values de las demás opciones no serán relevantes en la ejecución de la validación por ende no se especifican, se deja una imagen para ver como irá esta opción del <select>:

```

51 <select id="genero" class="form-select" required>
52 <option value="0" selected disabled>Seleccionar</option>

```

- Por último, en el input tipo “range” de edad hacer las siguientes asignaciones de “value”

- Asignar la propiedad value=”0” con el fin de que cuando se cargue el formulario, el range este en su nivel mínimo, y no en la mitad como normalmente esta por defecto, adjunto imagen:

```

59 <label for="edad" class="form-label">Edad</label>
60 <input type="range" class="form-range" id="edad" min="1" max="100" value="0"

```

Con este paso a paso se realizará la ejecución correcta de la validación realizada en JavaScript por medio de una clase.

**IMPORTANTE:** Sera también de mucha ayuda en dado caso de que se tengas dudas adicionales a las que se especifican en este documento, el leer los comentarios realizados en el código con el fin de explicarlo.