

Paso a paso para ejecución correcta de la clase validación:

Realizado por: Johan Felipe Garcia Salazar

1. Abrir enlace del repositorio donde se encuentra montado el micro-proyecto (https://github.com/FGFERNAN/ADSO_2919581_Johan_Garcia), este enlace se encontrará en la bandeja de entrada del correo con el asunto "Micro-Proyecto II", al abrir el enlace se deberá descargar el repositorio, este repositorio se descargará en un archivo.zip. O también se puede descargar directamente el archivo en formato.zip que se adjuntará en el mismo correo y que contendrá también el micro-proyecto.
2. Al descargarlo (ya sea del repositorio o del correo directamente), descomprimir el archivo donde deberá quedar 1 carpeta llamada "ADSO_2919581_Johan_Garcia" y en ella 2 carpetas, la que corresponde a este entregable se llama "Actividad_III_Trimestre" en ella se encontraran 6 archivos, el formulario (index.html), la clase en JavaScript (validacion.js), el controlador de eventos (controller.js), el estilo personalizado (style.css), el formulario con el que se realizara la prueba ya adaptado para la validación (Form.html) y por último este documento que está 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 subcarpeta que corresponde a este entregable ya descomprimida previamente, llamada "Actividad_III_Trimestre".
4. Una vez abierto el documento se puede proceder a la implementación de la clase, en dado 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 qué 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:

IMPORTANTE: Las siguientes modificaciones ya fueron realizadas al formulario en el que se probara la validación, este formulario modificado se encuentra en la carpeta listo para ejecutar con el nombre de "Form.html"

- Copiar dentro de las etiquetas <head> el CDN de CSS de Bootstrap que controla la mayoría de estilos del formulario (línea de código 8 y 9 del formulario base), incluyendo los mensajes de validación.

```

7      <!--CDN CSS de Bootstrap-->
8      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
9      integrity="sha384-QwTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhY6hW+ALEwIH" crossorigin="a

```

- 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 Perosnalizado-->
11     <link rel="stylesheet" href="style.css">

```

- Anidar los inputs y/o select del HTML en el que se quiere probar la validación dentro de un <div> para cada uno (adjunto imagen de ejemplo), esto con el fin de que se muestren de manera adecuada los mensajes de error o validación correcta de Bootstrap.

```

16     <div>
17     |   <input type="date" id="fecha">
18     </div>

```

-Una vez anidados los inputs y selects del formulario de la manera correcta dicha en el paso anterior, se debe proceder a agregar los mensajes de error o validación correcta en el HTML por medio de las etiquetas <div> también, estos mensajes de se deben agregar dentro de los <div> de cada input o select, así como se muestra en la imagen.

```

16     <div>
17     |   <input type="date" id="fecha">
18     |   <div class="invalid-feedback">
19     |     Edad Invalida (Debes ser mayor de edad).
20     |   </div>
21     |   <div class="valid-feedback">
22     |     Edad Validada Exitosamente.
23     |   </div>
24     </div>

```

Cabe aclarar que cada uno de estos <div> debe conservar el nombre de su clase, con el fin de que tome los estilos de Bootstrap, y de que estos mensajes cambian dependiendo el input, por ende, se debe copiar el que corresponda tomando como base el formulario “index.html”

- 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.

```

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

```

- Copiar la línea de código (232) del HTML base que contiene en las etiquetas <script> el enlace al archivo “validacion.js” para que se ejecute la clase.

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

-Luego se debe copiar ahora la línea de código (235) del HTML base que contiene las etiquetas <script> que enlazan el HTML al archivo que controla los eventos “controller.js”

```
234      <!--Script de controlador de eventos-->
235      <script src="controller.js"></script>
```

- Copiar CDN de JavaScript de Bootstrap (línea de código 238 y 239 del formulario base).
(OPCIONAL)

```
238      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity=
239      crossorigin="anonymous"></script>
```

-Agregar el input tipo “range” al formulario en el que se quiere probar la validación, esto para que se ejecute de manera correcta la validación que valida si la edad ingresada es menor de 18, se adjunta imagen de las propiedades que deberá tener este input (se encuentra en la línea 96 del formulario base).

```
96      <input type="range" class="form-range" id="edad" min="1" max="100" value="0" disabled required>
```

- Agregar etiqueta debajo del input tipo “range” que corresponde a la edad, esto con el fin de ver el valor exacto del rango en años (se encuentra en la línea 97 del formulario base).

```
97      <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:

```
14      <form id="formulario" novalidate>
```

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

```
28      <input type="text" id="nombre">
```

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

```
60      <select id="tipo_documento">
```

- 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:

```
63      <option id="TI" value="1">TI</option>
```

- 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:

```
17      <input type="date" id="fecha">
```

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

```
73      <select id="genero">
```

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

```
25      <input type="range" id="edad" min="1" max="100" value="0" disabled>
```

- ID de la etiqueta 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:

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

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

```
51      <input type="email" id="correo">
```

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

```
37      <input type="password" id="contraseña">
```

Dentro del <div> del input contraseña añadir después del input un <div> que contendrá un mensaje que se mostrara en el HTML con las especificaciones que debe tener la contraseña, este <div> contiene estilos de bootstrap, como se ve en la imagen (OPCIONAL):

```

35 </div>
36 <input type="password" id="contraseña">
37 <div class="form-text">
38   Tu contraseña debe tener mínimo 8 caracteres, una letra mayúscula, una letra minúscula
39   y un carácter especial (#,$,*).
40 </div>

```

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

```

100 <input type="color" id="color">

```

-
- En el botón de enviar el formulario se debe comprobar que sea type “submit” esto para el correcto funcionamiento del escuchador.

```

109 <button type="submit">Send data</button>

```

- Se debe agregar el botón “Editar” que hará la acción de ejecutar el método solicitado llamado formDisableEdit(), este botón deberá ser type “button” y deberá tener el ID “disabledEdit”, así como se ve en la imagen:

```

110 <button id="disabledEdit" type="button">Editar</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:

```

86 <input type="checkbox" name="status">

```

- 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:

```

95 <input type="radio" name="flexRadioDefault">
96 <input type="radio" name="flexRadioDefault">
97 <input type="radio" name="flexRadioDefault">
98 <input type="radio" name="flexRadioDefault">
99 <input type="radio" name="flexRadioDefault">
00 <input type="radio" name="flexRadioDefault">

```

- Ya para finalizar con las modificaciones que se deben realizar en el HTML 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 él 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>:

```
60      <select id="tipo_documento">
61          <option value="0" disabled readonly selected>SELECT</option>
62          <option value="2">CC</option>
63          <option id="TI" value="1">TI</option>
64      </select>
```

- 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>:

```
73      <select id="genero">
74          <option value="0" disabled readonly selected>SEX</option>
75          <option value="1">Female</option>
76          <option value="2">Male</option>
77      </select>
```

- 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:

```
25      <input type="range" id="edad" min="1" max="100" value="0" disabled>
```

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

IMPORTANTE: Será 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.