



## Entrenamiento – Módulo 3 Semana 1

### Hoja de trabajo

El objetivo de esta hoja de trabajo es guiarte a desarrollar un programa interactivo en **JavaScript** para consolidar tus conocimientos sobre **captura de datos, validación y condicionales**. Crearás un sistema que valide la edad ingresada por un usuario y muestre mensajes dinámicos basados en esta información. Aprenderás a gestionar errores, implementar validaciones y mostrar resultados claros y organizados.

#### **Problema a resolver:** Sistema Interactivo de Mensajes

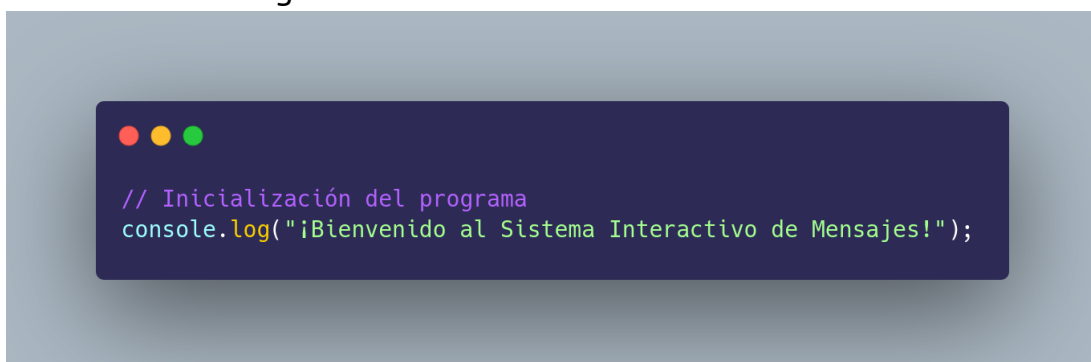
Tu tarea es crear un programa que:

1. Solicite al usuario su nombre y su edad.
2. Valide que la edad ingresada sea un número válido.
3. Muestre un mensaje personalizado dependiendo de la edad ingresada.
  - a. Si el usuario es menor de 18, animarlo a seguir aprendiendo.
  - b. Si el usuario es 18 o mayor, motivarlo para explorar nuevas oportunidades.
4. Muestre un error si el usuario ingresa un valor no numérico para la edad.

#### **Pasos sugeridos para solucionar el problema:**

##### **Paso 1: Configuración del proyecto**

1. Crea un archivo JavaScript llamado **sistema\_interactivo.js**.
2. Configura un entorno de desarrollo:
  - Usa un editor de código como **VS Code**.
  - Asegúrate de tener un navegador instalado para ejecutar el programa desde la consola del navegador.



Sebastián Agudelo. (2024) – Captura de pantalla del código, paso 1 entrenamiento – módulo 3 – semana 1

##### **Paso 2: Captura de datos**

1. Utiliza **prompt()** para solicitar los datos necesarios al usuario:
  - Nombre
  - Edad

```
// Capturar datos del usuario
let nombre = prompt("Por favor, ingresa tu nombre:");
let edad = prompt("Por favor, ingresa tu edad:");
```

Sebastián Agudelo. (2024) – Captura de pantalla del código, paso 2.1 entrenamiento – módulo 3 – semana 1

## 2. Convierte la edad a número para validarla:

```
// Convertir edad a número
edad = parseInt(edad);
```

Sebastián Agudelo. (2024) – Captura de pantalla del código, paso 2.2 entrenamiento – módulo 3 – semana 1

## Paso 3: Validación y mensajes personalizados

### 1. Implementa las siguientes validaciones con condicionales:

- Si la edad es menor de 18, muestra un mensaje animando al usuario.
- Si la edad es 18 o mayor, motiva al usuario con un mensaje positivo.
- Si la entrada no es un número, muestra un error en la consola.

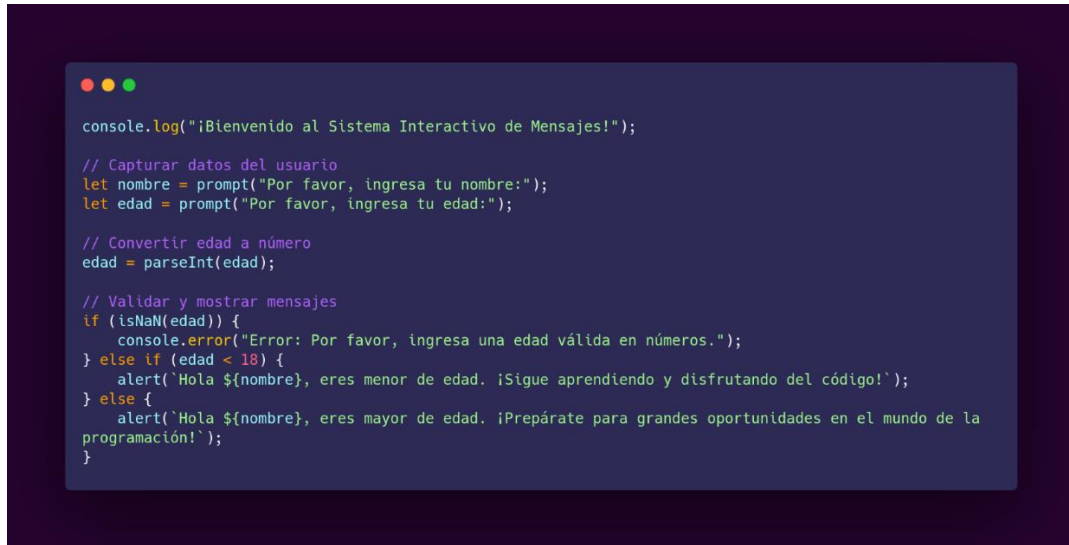
```
// Código para Validación y Mensajes:
if (isNaN(edad)) {
  console.error("Error: Por favor, ingresa una edad válida en números.");
} else if (edad < 18) {
  alert(`Hola ${nombre}, eres menor de edad. ¡Sigue aprendiendo y disfrutando del código!`);
} else {
  alert(`Hola ${nombre}, eres mayor de edad. ¡Prepárate para grandes oportunidades en el mundo de la programación!`);
}
```

Sebastián Agudelo. (2024) – Captura de pantalla del código, paso 3 entrenamiento – módulo 3 – semana 1



## Paso 4: Pruebas del programa

1. Asegúrate de probar el programa con diferentes entradas:
  - Edad válida menor de 18 (ej.: 15).
  - Edad válida igual o mayor a 18 (ej.: 25).
  - Entrada no numérica (ej.: "texto").



Sebastián Agudelo. (2024) – Captura de pantalla del código, paso 4 entrenamiento – módulo 3 – semana 1

## Paso 5: Ejecución del programa

1. Guarda el archivo **sistema\_interactivo.js** en tu computadora.
2. Abre tu navegador web (por ejemplo, Chrome o Firefox).
3. Haz lo siguiente para ejecutarlo:
  - Presiona F12 para abrir las herramientas de desarrollo del navegador.
  - Dirígete a la pestaña Consola.
  - Copia y pega el código completo en la consola y presiona Enter para ejecutarlo.

## Resultado esperado:

1. Solicitud de nombre y edad al usuario.
2. Muestra del mensaje adecuado basado en la edad ingresada.
3. Mensaje de error si el usuario no ingresa un número como edad.

## Ejemplo de interacción:

- **Entrada del usuario:**
  - Nombre: María
  - Edad: 17
- **Salida:**
  - "Hola María, eres menor de edad. ¡Sigue aprendiendo y disfrutando del código!"

## Recomendaciones finales:

- **Documenta tu código:** Usa comentarios para describir cada sección.
- **Sigue buenas prácticas:** Usa nombres de variables descriptivos y el formato camelCase.
- **Explora más:** Intenta mejorar el programa con validaciones adicionales o mensajes personalizados.