


Evaluación del módulo

Consigna del proyecto 

Proyecto: Aplicación de consola

Módulo: Fundamentos de programación en Javascript

Situación inicial

 Unidad solicitante: Equipo de desarrollo de una plataforma de aprendizaje interactivo.

Un equipo de desarrollo ha sido encargado de crear una aplicación en JavaScript que permita a los usuarios **realizar operaciones básicas y trabajar con estructuras fundamentales del lenguaje**. Se busca desarrollar habilidades en el lenguaje JavaScript a través de una práctica aplicada, implementando los conceptos fundamentales del lenguaje.

Nuestro objetivo

El objetivo del proyecto es desarrollar una aplicación en consola que permita a los usuarios:

- Realizar operaciones matemáticas básicas.
- Implementar estructuras condicionales y de bucles.
- Usar funciones para modularizar el código.
- Trabajar con arreglos y objetos.

El código deberá estar estructurado de manera clara según las mejores prácticas de desarrollo en JavaScript.

Requerimientos

La aplicación deberá cumplir con los siguientes requisitos:

→ **Requerimientos generales:**

- ↪ Debe ejecutarse en la **consola del navegador**.
- ↪ Debe permitir al usuario ingresar datos mediante **prompt o variables** predefinidas.
- ↪ Debe realizar **al menos tres operaciones diferentes** usando funciones.
- ↪ Debe utilizar **condicionales y estructuras de repetición**.

⇒ Debe incluir **arreglos y objetos** para almacenar y manipular datos.



→ **Requerimientos técnicos:**

- ⇒ **Funciones y modularización:** Separación del código en funciones reutilizables.
- ⇒ **Uso de estructuras de control:** Implementación de if, switch, for, while, según corresponda.
- ⇒ **Uso de arreglos y objetos:** Manipulación de datos utilizando estos conceptos.
- ⇒ **Validaciones:** Control de entradas del usuario para evitar errores.



Paso a paso

Este proyecto será avanzado paso a paso en cada clase y podrás completarlo progresivamente a través de los espacios **asincrónicos** en la herramienta de VISUAL STUDIO CODE. También tendrás espacios de consulta en las clases sincrónicas para despejar tus dudas.

1. Introducción al lenguaje JavaScript (Lección #1)



- ⇒  **Objetivo:** Familiarizarse con la sintaxis básica de JavaScript y ejecutar código en la consola.
- ⇒  **Tareas a desarrollar:**
 - Escribir un script simple que muestre mensajes en la consola con `console.log()`.
 - Usar `prompt()` para recibir datos del usuario.
 - Mostrar mensajes personalizados con `alert()`.

2. Variables, expresiones y sentencias condicionales (Lección #2)



- ⇒  **Objetivo:** Aprender a manejar variables, operadores y estructuras condicionales.
- ⇒  **Tareas a desarrollar:**
 - Definir variables utilizando `let` y `const`.
 - Pedir al usuario que ingrese dos números y almacenarlos en variables.

- Implementar operaciones matemáticas (suma, resta, multiplicación, división).
- Usar estructuras condicionales (`if`, `else`, `switch`) para evaluar diferentes situaciones.



3. Arreglos y ciclos (Lección #3)

- ⇒  **Objetivo:** Introducir el uso de arreglos y estructuras de repetición.
- ⇒  **Tareas a desarrollar:**
 - Crear un arreglo con una lista de elementos.
 - Usar `for` y `while` para recorrer arreglos.
 - Implementar una función que filtre elementos según una condición.

4. Funciones en JavaScript (Lección #4)

- ⇒  **Objetivo:** Modularizar el código utilizando funciones.
- ⇒  **Tareas a desarrollar:**
 - Crear funciones para cada operación matemática.
 - Implementar una función que reciba parámetros y retorne un resultado.
 - Llamar funciones dentro de otras funciones para optimizar el código.

5. Conceptos básicos de objetos en JavaScript (Lección #5)

- ⇒  **Objetivo:** Utilizar objetos para organizar datos de manera estructurada.
- ⇒  **Tareas a desarrollar:**
 - Crear un objeto con propiedades y valores.
 - Implementar métodos dentro de un objeto.
 - Usar un arreglo de objetos y recorrerlo con `map()` o `forEach()`.

¿Qué vamos a validar?

Para evaluar el trabajo, se tendrán en cuenta los siguientes aspectos:

→ **Aspectos técnicos:**

- ↪ Correcta aplicación de los fundamentos de JavaScript.
- ↪ Uso adecuado de funciones para modularizar el código.
- ↪ Implementación de estructuras de control correctamente.
- ↪ Manejo adecuado de arreglos y objetos.

→ **Aspectos estructurales:**

- ↪ Legibilidad y organización del código.
- ↪ Uso de buenas prácticas en nombres de variables y funciones.
- ↪ Comentarios explicativos en el código.

→ **Aspectos funcionales:**

- ↪ La aplicación debe ejecutarse sin errores en la consola.
- ↪ Todas las funcionalidades solicitadas deben estar implementadas.
- ↪ Se debe proporcionar una breve documentación con una explicación del código.

Referencias

- [MDN Web Docs - JavaScript](#)
- [W3Schools - JavaScript Tutorial](#)
- [JavaScript.info](#)

Recursos

Para reforzar los conocimientos y facilitar la resolución del proyecto, te recomendamos revisar los siguientes recursos:

→ Artículos técnicos y blogs

- [FreeCodeCamp](#)
- [CSS Tricks](#)
- [Dev.to](#) - Comunidad de Desarrolladores

→ Ejemplos y repositorios en GitHub

- [Ejercicios de JavaScript en GitHub](#)
- [Repositorio con ejemplos prácticos de JavaScript](#)

Entregables

Al finalizar el proyecto, se esperan los siguientes elementos, como evidencia concreta del trabajo realizado:

- Código fuente de la aplicación de ser posible en un repositorio (GitHub, ZIP, o similar).
- Un archivo README.md dentro de la carpeta del código fuente explicando la funcionalidad.
- Capturas de pantalla mostrando la aplicación en funcionamiento.
- Breve documentación y análisis del proyecto.

Todo esto podrás compartirlo a través de la plataforma de aprendizaje.

Portafolio

Se recomienda que los participantes suban este proyecto a GitHub y lo incluyan en su portafolio de desarrollo web. Una buena presentación del trabajo en plataformas como GitHub Pages, CodePen o similar puede ayudar a demostrar sus habilidades en JavaScript de manera efectiva.

Resolución

A continuación, dejo una posible implementación en JavaScript de la aplicación en consola que cumple con todos los requisitos del trabajo práctico. Se estima que el alumno pueda completar el proyecto en un alrededor de 2.5 horas. Es importante tener en cuenta que el trabajo práctico en sí está abierto a que los alumnos utilicen otras funciones y estructuras distintas a las de esta resolución, pero es solo una guía de lo esperado y las condiciones de validación deben ser comparadas con los criterios preestablecidos.

// 1. Función para realizar operaciones matemáticas básicas

```
function calcular(operacion, num1, num2) {  
  switch (operacion) {  
    case "sumar":  
      return num1 + num2;  
    case "restar":  
      return num1 - num2;  
    case "multiplicar":  
      return num1 * num2;  
    case "dividir":  
      return num2 !== 0 ? num1 / num2 : "Error: No se puede dividir por cero.";  
    default:  
      return "Operación no válida.";  
  }  
}
```

// 2. Función para recorrer un arreglo y devolver valores filtrados

```
function filtrarNumerosMayores(arr, limite) {  
  return arr.filter(num => num > limite);  
}
```

// 3. Función para generar un objeto con datos de usuario

```
function crearUsuario(nombre, edad, ciudad) {  
  return {  
    nombre,  
    edad,  
    ciudad,  
    descripcion: function () {  
      return `${this.nombre} tiene ${this.edad} años y vive en ${this.ciudad}.`;  
    }  
  }  
}
```

```

    };
  }
// 4. Función que usa un bucle para generar una tabla de multiplicar
function tablaMultiplicar(numero) {
  let resultado = [];
  for (let i = 1; i <= 10; i++) {
    resultado.push(`${numero} x ${i} = ${numero * i}`);
  }
  return resultado;
}
// 5. Simulación de ingreso de datos mediante prompt (si se ejecuta en un
navegador)
// Para Node.js, se podría usar 'readline-sync' en su lugar.
let operacion = "sumar"; // Se puede cambiar a "restar", "multiplicar" o "dividir"
let resultadoOperacion = calcular(operacion, 10, 5);
console.log(`Resultado de la operación (${operacion}):`, resultadoOperacion);
let numeros = [10, 25, 30, 5, 15, 40];
console.log("Números mayores que 20:", filtrarNumerosMayores(numeros, 20));
let usuario = crearUsuario("Sofía", 28, "Rosario");
console.log(usuario.descripcion());
console.log("Tabla de multiplicar del 7:");
console.log(tablaMultiplicar(7).join("\n"));

```


¡Éxitos!

Nos vemos más adelante

