

## Laboratorio 3 – Introducción de JavaScript

### Utilizando JavaScript

**Título del Laboratorio:** Calculadora de Operaciones Básicas en JavaScript.

**Duración:** 4 horas.

**Objetivos del Laboratorio.**

1. *Entender y aplicar la declaración de variables en JavaScript.*
2. *Utilizar operadores matemáticos y lógicos para realizar operaciones.*
3. *Implementar estructuras de control como condicionales y bucles para gestionar la lógica de la aplicación.*
4. *Definir y utilizar funciones para modularizar el código.*

**Materiales Necesarios:**

1. *Editor de texto (Recomendado: Visual Studio Code)*
2. *Navegador web (Recomendado: Google Chrome o Mozilla Firefox)*
3. *Editor de código online (Recomendado: mycompiler.io)*
4. *Conexión a Internet para acceder a recursos y tutoriales*

### Introducción

En este laboratorio, aprenderás a trabajar con variables, operadores matemáticos, condicionales, bucles y funciones en JavaScript. Crearás una calculadora sencilla que permita realizar operaciones básicas como suma, resta, multiplicación y división. Este proyecto tiene como objetivo practicar el uso de conceptos fundamentales de programación en JavaScript.

## Resultados Esperados

### 1. Actividad 1. Creación de la lógica en JavaScript.

- **Paso 1. Declaración de Variables**

Para comenzar con el desarrollo de este laboratorio, comience declarando un total de 2 variables con un valor numérico asignado por usted. Estas variables serán los numero con los cuales usted realizará las operaciones matemáticas.

Adicionalmente cree una variable para almacenar la operación matemática que desea realizar. Esta va a tomar valores de "suma", "resta", "multiplicación" y "división"

- **Paso 2. Función para realizar operaciones**

Crea una función llamada `realizarOperacion` que reciba tres parámetros: num1, num2 y operación

Dentro de la función, utiliza una estructura condicional (if...else if...else) para realizar la operación indicada en el parámetro operacion:

- Si operacion es "suma", devuelve la suma de num1 y num2.
- Si operacion es "resta", devuelve la resta de num1 y num2.
- Si operacion es "multiplicacion", devuelve la multiplicación de num1 y num2.
- Si operacion es "division", devuelve la división de num1 y num2.

- **Paso 3. Realizar validaciones de datos y operaciones.**

Utiliza una estructura condicional para comprobar que la operación ingresada es válida. Si la operación no es una de las cuatro opciones mencionadas, devuelve un mensaje de error.

Asegúrate de manejar correctamente la división por cero en la operación de división

Puedes realizar esto con ayuda de la estructura condicional if.

- **Paso 4. Bucle para realizar múltiples operaciones.**

Crea un bucle while que permita al usuario realizar varias operaciones de manera continua. El bucle debe continuar hasta que el usuario ingrese la palabra "salir" como la operación.

Dentro del bucle, pide al usuario que ingrese los dos números y la operación que desea realizar (puedes simular esto con prompt o valores fijos para simplificar).

Después de cada operación, muestra el resultado de la operación y pregunta al usuario si desea realizar otra operación.

- **Paso 5. Prueba de la calculadora**

En el bucle, si el usuario ingresa "salir", muestra un mensaje de despedida y termina el bucle. Realiza las pruebas necesarias para validar que la calculadora esta funcionando correctamente.

- **Paso 6. Copia tu código en el repositorio y súbelo.**

Finalmente, si ya terminaste el desarrollo de tu calculadora y esta funcionando correctamente, copia tu código JavaScript en un nuevo archivo JS en la carpeta de tu repositorio creado en el desarrollo del Laboratorio 2.

Luego añade los cambios, realiza un commit y súbelo a tu repositorio remoto.

---

- **Conclusiones.**

Al finalizar este laboratorio, habrás creado una calculadora de operaciones matemáticas simples utilizando JavaScript. Este ejercicio te permitirá afianzar tu comprensión de las variables, los operadores, las condicionales, los bucles y las funciones en JavaScript. Con estos conceptos, tendrás una base sólida para desarrollar aplicaciones más complejas en el futuro.