

### Parte 1: Verificación de que todas hagan lo mismo

Cada una de las calculadoras realiza las mismas operaciones (suma, multiplicación y encontrar el número mayor) y tiene la misma interfaz con dos entradas de números y un botón para realizar el cálculo. Los resultados de las operaciones se muestran en el mismo formato.

- **Calculadora Clásica**: Usa una función tradicional en JavaScript para realizar las operaciones y actualizar los resultados en el DOM con `getElementById`.
- **Calculadora Moderna**: Utiliza un enfoque más moderno con funciones de flecha y el uso de `addEventListener` para manejar el evento del botón.
- **Calculadora Funcional**: Se enfoca en un estilo más funcional con funciones puras para obtener el número, establecer el texto y realizar las operaciones.
- **Calculadora con Clases**: Implementa una clase `Calculadora` para estructurar el código y manejar las operaciones.

En resumen, todas las calculadoras realizan las mismas funciones de suma, multiplicación y encontrar el mayor número, aunque usan diferentes enfoques de programación.

## Parte 2: Análisis del código con IA

A continuación se presenta el código comentado de cada una de las versiones de la calculadora.

### Calculadora Clásica

```
// Esta función se activa cuando se hace clic en el botón 'Calcular'
function calcular() {
    // Se obtienen los valores de los números ingresados
    const numero1 = Number(document.getElementById("numero1").value);
    const numero2 = Number(document.getElementById("numero2").value);

    // Se calculan la suma y el producto
    const suma = numero1 + numero2;
    const producto = numero1 * numero2;
    let mayor;

    // Se determina cuál es el número mayor
    if (numero1 > numero2) {
        mayor = numero1;
    } else if (numero2 > numero1) {
        mayor = numero2;
    } else {
        mayor = "Los números son iguales";
    }

    // Se muestran los resultados en el HTML
    document.getElementById("suma").textContent = suma;
    document.getElementById("producto").textContent = producto;
    document.getElementById("mayor").textContent = mayor;
}
```

## Calculadora Moderna

```
// Se agrega un event listener al botón para ejecutar el cálculo cuando se hace clic
document.getElementById("calcularBtn").addEventListener("click", () => {
  // Se obtienen los números ingresados como valores numéricos
  const numero1 = Number(document.getElementById("numero1").value);
  const numero2 = Number(document.getElementById("numero2").value);

  // Se calculan la suma, el producto y el número mayor
  const suma = numero1 + numero2;
  const producto = numero1 * numero2;
  const mayor = numero1 === numero2 ? "Los números son iguales" :
Math.max(numero1, numero2);

  // Se muestran los resultados en el HTML
  document.getElementById("suma").textContent = suma;
  document.getElementById("producto").textContent = producto;
  document.getElementById("mayor").textContent = mayor;
});
```

## Calculadora Funcional

```
// Función para obtener el valor numérico de un input por su id
const getNumber = (id) => Number(document.getElementById(id).value);

// Función para establecer el texto de un elemento por su id
const setText = (id, value) => document.getElementById(id).textContent = value;

// Función principal para realizar los cálculos
const calcular = () => {
  // Se obtienen los dos números ingresados
  const [numero1, numero2] = [getNumber("numero1"), getNumber("numero2")];

  // Se definen las operaciones de suma, producto y mayor usando funciones puras
  const operaciones = {
    suma: (a, b) => a + b,
    producto: (a, b) => a * b,
    mayor: (a, b) => (a === b ? "Los números son iguales" : Math.max(a, b))
  };

  // Se establecen los resultados en el HTML
  setText("suma", operaciones.suma(numero1, numero2));
  setText("producto", operaciones.producto(numero1, numero2));
  setText("mayor", operaciones.mayor(numero1, numero2));
};

// Se asocia la función calcular al botón de clic
document.getElementById("calcularBtn").addEventListener("click", calcular);
```

## Calculadora con Clases

// Definición de la clase Calculadora

```
class Calculadora {
```

```
    // Constructor que recibe los ids de los elementos HTML
```

```
    constructor(numero1Id, numero2Id, sumald, productold, mayorId) {
```

```
        this.numero1Input = document.getElementById(numero1Id);
```

```
        this.numero2Input = document.getElementById(numero2Id);
```

```
        this.sumaOutput = document.getElementById(sumald);
```

```
        this.productoOutput = document.getElementById(productold);
```

```
        this.mayorOutput = document.getElementById(mayorId);
```

```
    }
```

```
    // Método para obtener el número ingresado
```

```
    obtenerNumero(input) {
```

```
        return Number(input.value);
```

```
    }
```

```
    // Método principal que ejecuta los cálculos
```

```
    calcular() {
```

```
        const numero1 = this.obtenerNumero(this.numero1Input);
```

```
        const numero2 = this.obtenerNumero(this.numero2Input);
```

```
        // Se llaman a los métodos para mostrar los resultados
```

```
        this.mostrarResultados(numero1, numero2);
```

```
    }
```

```
    // Método para mostrar los resultados de las operaciones
```

```
    mostrarResultados(numero1, numero2) {
```

```
        this.sumaOutput.textContent = numero1 + numero2;
```

```
        this.productoOutput.textContent = numero1 * numero2;
```

```
        this.mayorOutput.textContent = numero1 === numero2 ? "Los números son
```

```
iguales" : Math.max(numero1, numero2);
```

```
    }
```

```
}
```

```
// Se crea una instancia de la clase y se asocia el evento al botón
const calculadora = new Calculadora("numero1", "numero2", "suma", "producto",
"mayor");
document.getElementById("calcularBtn").addEventListener("click", () =>
calculadora.calcular());
```

### Comparativa de las Versiones

Aquí se presenta una tabla comparativa entre las diferentes versiones de la calculadora en términos de facilidad de entender, limpieza del código y facilidad de mantenimiento/expansión.

Característica	Calculadora Clásica	Calculadora Moderna	Calculadora Funcional	Calculadora con Clases
Facilidad de entender	Moderada	Alta	Alta	Alta
Código limpio	Moderado	Alto	Alto	Medio
Mantenimiento/Futuro	Moderado	Alto	Alto	Alto

### Sugerencia de Otros Enfoques

La IA sugiere que se podrían explorar más enfoques de implementación como:

- **Programación reactiva** usando bibliotecas como React.js, que permite un manejo más eficiente de las interfaces de usuario.
- **Uso de frameworks** como Vue.js o Angular para estructurar más fácilmente el código.
- **Uso de Web Workers** para hacer los cálculos en segundo plano si se tiene una gran cantidad de operaciones.

## Parte 3: Conclusiones

### 1. ¿Qué estilo te parece más fácil y más difícil?

Para mí, la *Calculadora Moderna* es la más fácil. El código es más sencillo y se entiende mejor, sobre todo porque no tengo mucha experiencia con programación. Es más directo y fácil de seguir.

En cambio, la *Calculadora con Clases* me resulta más difícil. Aunque tiene sus ventajas, el código es más complicado y no me resulta tan fácil de entender, especialmente porque no tengo mucha idea de cómo funcionan las clases y los objetos en programación.

### 2. ¿Es todo JavaScript? ¿Por qué tantas formas?

Sí, todas las versiones están escritas en JavaScript, pero cada una emplea un enfoque diferente. JavaScript por lo que he podido investigar es muy flexible y permite varios estilos para resolver un mismo problema. Por ejemplo:

- *Calculadora Clásica*: Usa funciones globales, un enfoque tradicional.
- *Calculadora Moderna*: Emplea funciones flecha y eventos, haciendo el código más limpio y modular.
- *Calculadora Funcional*: Se enfoca en funciones puras, promoviendo un estilo matemático.
- *Calculadora con Clases*: Usa programación orientada a objetos, ideal para proyectos más grandes y estructurados.

### 3. ¿Qué versión de la calculadora es más adecuada para nosotros?

Para proyectos simples, la Calculadora Moderna es la mejor opción: es fácil de entender, eficiente y suficiente para tareas básicas. Si el objetivo es aprender programación orientada a objetos o desarrollar aplicaciones más complejas, la Calculadora con Clases es la mejor elección.

### 4. ¿Qué versión de la calculadora es más adecuada para aprender?

Después de investigar las cuatro versiones, creo que la Calculadora Moderna es la mejor para aprender. Es fácil de entender y permite experimentar con el código mientras aprendes. Introduce conceptos modernos de forma clara, lo que me parece ideal para alguien que está empezando y quiere practicar sin complicarse demasiado.

### 5. ¿Qué ventajas y desventajas encuentras en cada enfoque?

- **Calculadora Clásica:**
  - Ventajas: Fácil de entender para principiantes, código sencillo.
  - Desventajas: Difícil de mantener en proyectos grandes, ya que las funciones globales pueden generar conflictos.
- **Calculadora Moderna:**
  - Ventajas: Código limpio, modular y fácil de mantener.
  - Desventajas: Requiere algo de práctica con funciones flecha y eventos.



- **Calculadora Funcional:**
  - Ventajas: Promueve código con funciones puras.
  - Desventajas: Puede ser difícil de entender sin experiencia en programación.
- **Calculadora con Clases:**
  - Ventajas: Facilita el trabajo con aplicaciones grandes gracias a la programación orientada a objetos.
  - Desventajas: Requiere comprensión de clases y objetos, lo que puede ser complicado para principiantes.

## 6. ¿Qué versión o propuesta te parece más interesante?

Desde mi punto de vista, la Calculadora Moderna es la más interesante. Como no tengo experiencia, me parece que el planteamiento lógico es más sencillo y fácil de entender. Es directo y claro, lo que me ayuda a seguir el código sin complicaciones. Creo que es una buena forma de empezar a familiarizarme con la programación sin que sea complicado para mí.