Informe de Desarrollo de la Calculadora

1. Decisiones de Diseño

Separación en Funciones por Cada Operación: Para hacer que el código sea más fácil de entender y mantener, cada operación matemática (suma, resta, multiplicación, división y raíz cuadrada) se implementó en funciones separadas. Esto hace que cada función tenga una responsabilidad clara, permitiendo que, si queremos agregar una nueva operación en el futuro, se pueda hacer sin afectar las demás.

Validación de Entradas del Usuario con validarInput: Dado que los usuarios pueden introducir cualquier valor, se creó una función especial para revisar si los valores ingresados son correctos. Esta función (validarInput) revisa que las entradas realmente sean números y se asegura, especialmente en el caso de la división, de que el divisor no sea cero. De esta forma, evitamos errores durante las operaciones y damos una experiencia más controlada y segura al usuario.

Uso de Map para Guardar Resultados: Para darle más estructura al almacenamiento de los resultados, en lugar de usar un array simple, se decidió emplear un Map. Este cambio permite asignar a cada resultado un identificador único, proporcionando un contexto adicional que permite saber a qué operación corresponde cada resultado (por ejemplo, "suma_1" o "resta_2"). Esto facilita que el usuario vea un historial de las operaciones con claridad.

2. Errores Detectados y Soluciones

A continuación se presentan algunos de los problemas encontrados durante el desarrollo y las soluciones aplicadas para resolverlos:

Error 1: Entradas Incorrectas y División por Cero

- Problema: Inicialmente, el código no verificaba correctamente si las entradas del usuario eran números y tampoco manejaba adecuadamente el caso de divisiones con divisor cero, lo cual generaba errores.
- Solución: La función validarInput se mejoró para que revisara de manera precisa si los valores ingresados son números y, en caso de división, si el divisor no es cero. Para ello, se utilizó la función isNaN, y se agregó una verificación específica para divisiones. Con esto, logramos que las operaciones solo se ejecuten cuando las entradas son válidas.

Error 2: Uso de Variables Globales (num1 y num2)

- Problema: Al inicio, las variables num1 y num2 se declaraban como variables globales. Esto ocasionaba problemas de sincronización, especialmente al invocar funciones que usaban estas variables sin parámetros, ya que los valores de num1 y num2 no siempre estaban actualizados.
- Solución: Para evitar estos problemas, se cambió la estructura del código para que num1 y num2 se pasen directamente como parámetros a las funciones de operación. Esto ayudó a reducir errores de contexto y

simplificó el código, asegurando que cada operación trabaje con los valores actuales ingresados.

Error 3: Dificultad para Rastrear Operaciones en el Almacenamiento

- Problema: Los resultados se guardaban en un arreglo sin contexto, lo cual dificultaba saber a qué operación correspondía cada resultado.
- Solución: Se optó por utilizar un Map en lugar de un array simple. En el Map, cada operación se almacena con una clave descriptiva (por ejemplo, "suma_1", "resta_2"), que identifica tanto el tipo de operación como su orden. Esta mejora ayuda a los usuarios a entender mejor el historial de resultados.

Error 4: Manejo de Entradas Inválidas en el Menú de Operaciones

- Problema: Cuando el usuario ingresaba una operación no reconocida (como una palabra incorrecta), el programa no reaccionaba de forma adecuada, generando confusión.
- Solución: Se agregó un caso default en el switch de selección de operaciones, que muestra un mensaje de error cuando la entrada no coincide con ninguna operación válida. Esto ayuda a guiar al usuario hacia una opción correcta.

3. Flujo de Interacción del Usuario

Para que la calculadora sea fácil de usar, se diseñó un flujo de interacción claro y directo para el usuario, el cual incluye los siguientes pasos:

1. Seleccionar una Operación: El usuario ve un menú de opciones donde puede elegir entre suma, resta, multiplicación, división, raíz cuadrada o salir del programa.
2. Ingresar Números: Después de seleccionar una operación, el usuario ingresa uno o dos números, dependiendo de la operación elegida. La calculadora valida estas entradas para asegurarse de que sean correctas.
3. Ver Resultados y Almacenamiento: Si los números ingresados son válidos, la calculadora realiza la operación y muestra el resultado. También guarda este resultado en el Map de almacenamiento, permitiendo al usuario revisar un historial de operaciones al final de cada cálculo.
4. Salir de la Calculadora: En cualquier momento, el usuario puede optar por salir de la calculadora eligiendo la opción de “salida”. Esto cierra el programa y muestra un mensaje de finalización.

Este flujo asegura que cada operación sea independiente y que el usuario reciba retroalimentación continua sobre el estado de sus cálculos.