



# **Informe de pruebas 2.0: Calculadora básica**

## **Ingeniería de pruebas**

### **Nombres:**

**Granados Martínez Oscar Alejandro (Tester)**

**Hernández cabrera Aaron (Programador)**

**González Velázquez Ángel Omar  
(Diseñador)**

**Santillán Balmaceda Dante (Analista)**

**Mares Romero Ma. Eugenia**

**6NM62**

**27/03/2025**



## Informe de Pruebas:

Proyecto: CALCULADORA BÁSICA

## Información General:

Fecha del Informe: 18 de marzo de 2025

Versión del Software: 2.0

## Tecnologías Utilizadas:

Lenguaje de Programación: TypeScript

Framework: React

Estilos: Tailwind CSS

## Resumen Ejecutivo:

El equipo de pruebas ha concluido las actividades de prueba para la Calculadora Básica v2.0, que incluye una nueva funcionalidad: cálculo de porcentajes. Se realizaron pruebas exhaustivas para validar la correcta implementación del botón de porcentaje (%), así como la integración con las operaciones existentes. A continuación, se detallan los resultados y hallazgos clave.

## Actividades Realizadas:

### Pruebas de Funcionalidad

#### Nueva funcionalidad:

- Cálculo de porcentajes:
  1. Se verificó que el botón % realice cálculos correctos en expresiones como  $50\%100$  (debe devolver 50).
  2. Se validó la integración del porcentaje con otras operaciones (ejemplo:  $50\%100 + 10$  debe devolver 60).
- Defectos identificados:
  1. Expresiones complejas con %:  
En casos como  $50\%100\%10$ , la calculadora no maneja correctamente la precedencia de operaciones (resultado esperado: 5, pero puede variar).



- Uso de % sin operandos:  
Al presionar % sin números ingresados (ejemplo: %50), la calculadora muestra "Error", pero no proporciona retroalimentación clara al usuario.

### Pruebas de Usabilidad

Botón %:

- Se confirmó que el botón está bien integrado en la interfaz y sigue el esquema de colores de operaciones especiales.
- Feedback visual: No hay indicación visual de que el % sea una operación compuesta (podría añadirse un tooltip o animación).

### Pruebas de Rendimiento

Expresiones con %:

- No se observaron problemas de rendimiento al calcular porcentajes en números grandes (ejemplo: 999999%1000000).
- Carga de la función replace: El uso de `input.replace(/(\d+)%(\d+)/g, "($1/100)*$2")` no afecta la velocidad de respuesta.

### Pruebas de Manejo de Errores

Casos críticos:

- División por cero con porcentajes (ejemplo: 50%0) muestra "Error" correctamente.
- Entradas inválidas:
  1. 50% (sin segundo operando) muestra "Error", pero podría mejorarse con un mensaje descriptivo.
  2. % solo no genera un crash, pero tampoco una respuesta útil.

## Cobertura de Pruebas:

Cobertura de Pruebas

Tipo de Prueba	Cobertura
Pruebas Unitarias	40%
Pruebas de Integración	30%
Pruebas de Sistema	60%

Defectos Encontrados: 3

Defectos Resueltos: 1 (Error en 50%0)

Defectos Pendientes: 2 (Precedencia en 50%100%10 y feedback visual).



## Recomendaciones:

- Mejoras en la lógica de %:  
Implementar paréntesis automáticos para garantizar precedencia correcta (ejemplo:  $50\%100\%10 \rightarrow (50\%100)\%10$ ).
- Feedback al usuario:  
Añadir un mensaje claro cuando % se usa sin operandos (ejemplo: "Faltan números").
- Pruebas adicionales: Validar expresiones combinadas como  $(50\%100) + (20\%30)$ .

## Agradecimientos:

Agradecemos al equipo de desarrollo por la rápida implementación del botón % y su colaboración en la resolución de defectos.