

## Plan de Pruebas: Calculadora 2.0

### Objetivo:

El objetivo de este plan de pruebas es verificar la funcionalidad de la Calculadora 2.0 para asegurar que cumple con los requisitos especificados y que es capaz de manejar adecuadamente los escenarios de uso previstos, garantizando precisión en las operaciones, estabilidad de la aplicación y facilidad de uso.

### Alcance:

El alcance de las pruebas incluirá las siguientes áreas funcionales:

1. Operaciones básicas (suma, resta, multiplicación y división).
2. Funcionamiento de los 19 botones básicos.
3. Funciones de memoria (M+, M-, MC y MR).
4. Visualización correcta de resultados en pantalla.
5. Manejo de entradas inválidas (división entre cero, caracteres no numéricos).

### Criterios de Aceptación:

1. Las operaciones básicas deben arrojar resultados correctos.
2. Los 19 botones deben estar presentes y funcionar de acuerdo a su propósito.
3. Las funciones de memoria deben operar correctamente:
  - M+: sumar valores acumulativamente.
  - M-: restar valores de la memoria.
  - MC: limpiar memoria.
  - MR: mostrar el valor almacenado.
4. La aplicación debe manejar adecuadamente entradas inválidas (ejemplo: división entre cero).
5. Los resultados deben visualizarse en pantalla de forma clara y precisa.

### **Recursos:**

1. Personal de pruebas asignado (desarrollador y tester).
2. Entorno de prueba: computadora de escritorio con sistema operativo Windows/Linux.
3. Datos de prueba numéricos (enteros, decimales, positivos, negativos).
4. Herramientas de prueba: pruebas manuales y, de ser posible, unitarias automatizadas.

### **Pruebas de Unidad:**

- Verificar individualmente cada botón y operación de la calculadora.

### **Pruebas de Integración:**

- Validar la interacción entre botones (ejemplo: combinación de operaciones y memoria).

### **Pruebas de Sistema:**

- Evaluar la calculadora como un todo para garantizar el correcto funcionamiento bajo distintos escenarios.

### **Pruebas de Aceptación del Usuario (UAT):**

- Validar la usabilidad de la calculadora con usuarios finales (facilidad de uso, distribución de botones).

### **Pruebas de Seguridad:**

- Confirmar que la aplicación maneje adecuadamente entradas inválidas sin fallar ni cerrar inesperadamente.

### **Casos de Prueba:**

### **Operaciones básicas:**

1. Verificar que suma, resta, multiplicación y división funcionen correctamente.
2. Validar resultados con números enteros y decimales.

### **Botones básicos (19):**

- Verificar que todos los botones existen y funcionan (0–9, +, -, ×, ÷, =, ., C, CA).

### **Memoria (M+, M-, MC, MR):**

1. Verificar que M+ acumule valores.
2. Verificar que M- reste valores acumulados.
3. Verificar que MC limpie memoria (regrese a 0).
4. Verificar que MR muestre el valor guardado en memoria.

### **Visualización en pantalla:**

1. Confirmar que todos los resultados se muestran claramente.
2. Validar la prioridad de operaciones (ejemplo:  $2+3\times 4$  debe dar 14).

### **Manejo de entradas inválidas:**

1. Dividir entre cero y verificar que no se genere error crítico.
2. Ingresar caracteres no numéricos (si se permite) y validar la respuesta del sistema.

### **Cronograma:**

1. Pruebas de Unidad: Semana 1.
2. Pruebas de Integración: Semana 2.
3. Pruebas de Sistema: Semana 3.

4. Pruebas de Aceptación del Usuario: Semana 4.
5. Pruebas de Seguridad: Semana 4 (en paralelo).

### **Informes**

Se generarán informes de estado semanales, destacando los casos de prueba ejecutados, defectos encontrados y el avance del proceso de validación.