

**INSTITUTO POLITECNICO NACIONAL**

Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias Sociales y Administrativas

**PLAN DE PRUEBAS V1**

INGENIERIA DE PRUEBAS

PROF. CRUZ MARTINEZ RAMON

6NM60

**INTEGRANTES:**

ALMARAZ PAULIN LISSET AMEYALLI (Diseñadora)

CRUZ BOBADILLA SAMANTHA MONSERRAT (Analista y tester)

DIAZ HIDALGO FERNANDO (Desarrollador)

**PLAN DE PRUEBAS V1**

**Objetivo:**

El plan de pruebas tiene como objetivo validar el correcto funcionamiento de la calculadora asegurando que cumple con los requisitos establecidos. Se realizarán pruebas funcionales, de interfaz de usuario y de manejo de errores.

**Alcance:**

El alcance de las pruebas incluirá las siguientes áreas funcionales:

1. Operaciones matemáticas básicas
2. Manejo de la división entre cero
3. Interfaz de usuario
4. Historial de operaciones
5. Funcionalidad de los botones de borrado
6. Manejo de operadores consecutivos
7. Operaciones con números negativos

**Criterios de Aceptación:**

1. La calculadora debe permitir realizar suma, resta, multiplicación y división correctamente.
2. Debe aceptar números positivos y negativos como operandos.
3. Si el usuario intenta dividir por cero, el sistema debe mostrar el mensaje "infinite".
4. La disposición de los botones debe seguir la estructura de un teclado numérico estándar.
5. La pantalla debe ubicarse en la parte superior y mostrar correctamente las operaciones ingresadas.
6. Los botones deben ser accesibles y funcionar correctamente al hacer clic en ellos.
7. La calculadora debe almacenar y mostrar un historial de operaciones realizadas en la sesión actual.
8. Borrar último dígito**:** Al presionar este botón, solo se debe eliminar el último número ingresado sin afectar el resto de la operación.
9. Borrar todo: Al presionar este botón, la pantalla debe limpiarse completamente, permitiendo una nueva operación desde cero.
10. Borrar hasta el último signo de operación: Al presionar este botón, la calculadora debe eliminar todo lo que está después del último operador ingresado.
11. Si el usuario ingresa operadores consecutivos, la calculadora debe interpretar correctamente la operación sin generar errores.
12. Los usuarios deben poder ingresar números negativos sin errores.

**Recursos:**

**1. Personal Requerido**

* **Tester:** Ejecutará los casos de prueba, reportarán defectos y validarán correcciones.
* **Desarrollador:** Resolverán los defectos encontrados y harán ajustes según los reportes de pruebas.
* **Analista:** Coordinará la planificación y ejecución de pruebas, priorizará pruebas críticas y revisará la documentación.

**2. Herramientas de Prueba**

Ordenador Portatil

**3. Entornos de Prueba**

Navegador Brave

**4. Datos de Prueba**

**Operaciones matemáticas de prueba:**

* Números positivos y negativos.
* Operaciones con cero (incluyendo división por cero).
* Secuencias de operadores (por ejemplo, 72++).
* Números grandes y decimales.

**Casos límite:**

* Operaciones muy largas.
* Eliminación de caracteres con los botones de borrado en distintos escenarios.

**Datos para pruebas de interfaz:**

* Comprobación de la correcta disposición de botones.
* Validación de la pantalla de resultados e historial.

**Riesgos:**

* **Error en los cálculos matemáticos**

**Posible impacto:** Resultados incorrectos pueden afectar la confiabilidad de la calculadora.  
**Mitigación:**

* Diseñar casos de prueba con valores límite, números negativos y decimales.
* Validar los resultados con cálculos manuales y herramientas externas.
* **Manejo incorrecto de la división entre cero**

**Posible impacto:** Si el sistema no muestra correctamente el mensaje "infinite", los usuarios pueden recibir valores inesperados o errores.  
**Mitigación:**

* Probar múltiples escenarios de división por cero.
* Asegurar que la aplicación no se cierre ni bloquee después de este error.
* **Fallos en la funcionalidad de botones de borrado**

**Posible impacto:** Si los botones de borrado no funcionan correctamente, los usuarios no podrán corregir errores en sus cálculos.  
**Mitigación:**

* Crear pruebas específicas para cada tipo de borrado.
* Evaluar su comportamiento en operaciones largas o con operadores múltiples.

**Qué se va a probar**

**1. Operaciones matemáticas básicas (REQ-01)**

* Verificar que la calculadora realiza correctamente la suma, resta, multiplicación y división.
* Asegurar que acepta y procesa correctamente números positivos y negativos.
* Validar los resultados con diferentes combinaciones de operandos (por ejemplo, números grandes, fracciones, enteros, negativos).

**2. Manejo de la división entre cero (REQ-02)**

* Probar que si el usuario intenta dividir por cero, se muestra el mensaje "infinite".
* Asegurar que la calculadora no se bloquea ni cierra después de una división por cero.

**3. Interfaz de usuario (REQ-03)**

* Verificar que la disposición de los botones sigue el diseño de los teclados numéricos tradicionales.
* Comprobar que la pantalla muestra correctamente los números y operaciones.

**4. Historial de operaciones (REQ-04)**

* Verificar que todas las operaciones realizadas se registran correctamente en el historial.
* Comprobar que el historial se borra automáticamente cuando se cierra la calculadora.

**5. Funcionalidad de los botones de borrado (REQ-05)**

* Asegurar que el botón de "Borrar último dígito" elimina solo el último número ingresado.
* Validar que el botón de "Borrar todo" limpia completamente la pantalla.
* Comprobar que el botón "Borrar hasta el último signo de operación" elimina correctamente los números después del último operador ingresado.

**6. Manejo de operadores consecutivos (REQ-06)**

* Validar que si el usuario ingresa operadores consecutivos (como 72++), la calculadora los interpreta correctamente y realiza la operación deseada.
* Asegurar que no se generan errores o bloqueos al ingresar operadores seguidos.

**7. Operaciones con números negativos (REQ-07)**

* Comprobar que los usuarios pueden ingresar números negativos correctamente.
* Validar que la calculadora maneja correctamente las reglas de los signos en todas las operaciones matemáticas.

**Estrategia:**

Tener presente los requerimientos que necesita tener el sistema.

Una vez que hayamos reconocido los requerimientos empezaremos con las pruebas unitarias en el sistema para confirmar que cumplen correctamente con la funcionalidad requerida del sistema. Probaremos modulo por modulo para encontrar errores, fallas y defecto.

Terminado estas pruebas unitarias procederemos con las pruebas de integración donde probaremos que los módulos funcionen juntos adecuadamente.

Para que finalmente hagamos las pruebas del sistema y verifiquemos que los requerimientos funcionales y no funcionales no tengan defectos, es decir que todo el programa funcione correctamente.

Con la información recabada haremos un informe de pruebas para su posterior corrección.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID del requerimiento | Descripción | Tipo de requerimiento | Caso de prueba | Estado de prueba |
| REQ-01 | Realizar operaciones básicas | Funcional | CP-01-01 | Aprobado |
| REQ-02 | Manejo de la división entre cero | Funcional | CP-01-02 | Aprobado |
| REQ-03 | Interfaz de usuario | No funcional | CP-01-03 | Aprobado |
| REQ-04 | Operaciones con números negativos | Funcional | CP-01-04 | Aprobado |
| REQ-05 | Visualización del historial de operaciones | Funcional | CP-01-05 | Aprobado |
| REQ-06 | Manejo de operadores consecutivos | Funcional | CP-01-06 | No aprobado |
| REQ-07 | Operaciones con números negativos | Funcional | CP-01-07 | Aprobado |