



INSTITUTO POLITÉCNICO NACIONAL



**Unidad Profesional Interdisciplinaria de
Ingeniería y Ciencias Sociales y Administrativas**

**Ingeniería de Pruebas
Plan de Pruebas**

Proyecto: Calculadora en Python con Tkinter

Versión: 2.0

Fecha: [22/09/2025]

Autores:

- Contreras Mosco Cristobal
- Díaz Pérez Diego
- Hernández Aguirre Ricardo
- Monroy Muñoz Angel Yael
- Salazar Rocha Any Jennifer

Índice

1. Introducción

2. Objetivo de las pruebas
3. Alcance de las pruebas
4. Estrategia de Pruebas
5. Entorno de Pruebas
6. Casos de Prueba
7. Criterios de Aceptación
8. Registro de Resultados

Introducción

El propósito de este plan de pruebas es garantizar que la aplicación Calculadora 2.0 cumpla con los requerimientos funcionales y no funcionales. La nueva versión incluye botones de memoria (M+, M-, M, MR), además de operaciones básicas, historial y control de errores.

Objetivo de las pruebas

Validar que la calculadora realice operaciones aritméticas, maneje errores, gestione el historial, ejecute correctamente las funciones de memoria y presente una interfaz clara y usable.

Los objetivos principales son:

- Confirmar que todas las operaciones aritméticas (suma, resta, multiplicación, división) funcionan sin errores.
- Validar la funcionalidad completa de los botones de memoria (M+, M-, M, MR).
- Asegurar que el historial de operaciones se actualice y muestre de manera precisa.
- Verificar que el sistema maneje correctamente los errores, como la división por cero y las entradas inválidas.
- Evaluar la facilidad de uso de la interfaz de usuario.
- Documentar y rastrear todos los defectos encontrados.

Alcance de las pruebas

Se probarán las operaciones básicas, manejo de errores, historial, funciones de memoria y la interfaz gráfica. No se incluyen pruebas de compatibilidad con sistemas distintos a Windows/Linux con Python 3.8+.

El alcance incluye las siguientes áreas:

- **Pruebas de Funcionalidad:**
 - Operaciones Básicas: Validar suma, resta, multiplicación y división.
 - Funcionalidad de Memoria: Probar el almacenamiento (M+), la sustracción (M-), la recuperación (MR) y la limpieza (M) de valores.

- **Funcionalidad de Historial:** Verificar que las operaciones y los resultados se registren correctamente.
- **Manejo de Errores:** Probar el comportamiento de la aplicación con entradas incorrectas y operaciones inválidas.
- **Pruebas de Usabilidad:** Evaluar la facilidad de navegación y la claridad de la interfaz para el usuario final.
- **Pruebas de Regresión:** Asegurar que las nuevas características no hayan introducido errores en la funcionalidad existente.

Estrategia de Pruebas

Pruebas unitarias, integración, sistema y aceptación. Se utilizará una estrategia de pruebas de caja negra, centrándose en la funcionalidad sin necesidad de conocer la estructura interna del código. La secuencia de pruebas será:

1. **Pruebas de Unidad:** Verificación de cada módulo del código de forma aislada.
2. **Pruebas de Integración:** Validación de la interacción entre los diferentes módulos, como la pantalla, los botones de operación y las funciones de memoria.
3. **Pruebas de Sistema:** Evaluación del sistema completo para asegurar que todos los componentes trabajen juntos sin problemas.
4. **Pruebas de Aceptación del Usuario (UAT):** El usuario final probará la aplicación para validar que cumple con sus expectativas.







Entorno de Pruebas



Hardware: PC/Laptop con mínimo 4GB RAM
Software:

- **Sistema Operativo:** Windows 10/11, macOS, o una distribución de Linux compatible.
- **Lenguaje de Programación:** Python 3.8 o superior.
- **Librería de GUI:** Tkinter (incluida en la instalación estándar de Python).

- Editor de Código: VS Code, PyCharm, o cualquier editor de texto compatible.

Casos de Prueba

| ID | Caso | Entrada | Resultado Esperado | Tipo | Captura |
|-------|------------------------|----------------------|------------------------|-------------|---|
| CP-01 | Suma simple | 5+3= | 8 | Unidad |  |
| CP-02 | División por cero | 5÷0 | ERROR: ÷0 | Sistema |  |
| CP-03 | Uso del historial | 2+2=, 3*3= | Historial actualizado | Integración |  |
| CP-04 | Memoria suma (M+) | Pantalla=5 5 → M+ | Memoria=5, MR verde | Sistema |  |
| CP-05 | Memoria resta (M-) | Pantalla=2 2 → M- | Memoria=3 | Sistema |  |
| CP-06 | Memoria recuperar (MR) | MR | Valor en pantalla | Sistema |  |

| | | | | | |
|-------|------------------------|------|----------------------|---------|---|
| CP-07 | Memoria limpiar (M) | M | Memoria=0, MR normal | Sistema |  |
| CP-08 | Error entrada inválida | ++-- | Error mostrado | Sistema |  |

Criterios de Aceptación

El 100% de los casos de prueba críticos para operaciones básicas y la funcionalidad de memoria deben aprobarse.

La aplicación no debe mostrar errores inesperados, como fallos o bloqueos.

El 98% de los casos de prueba totales debe ser exitoso antes de la liberación.