INSTITUTO POLITÉCNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA Y CIENCIAS SOCIALES Y ADMINISTRATIVAS

INGENIERÍA DE PRUEBAS

6NV61

***Análisis de Requerimientos***

ALUMNOS:

* FIGUEROA HERNÁNDEZ DIANA PAOLA
* MIGUEL ALARCÓN ADRIÁN MANUEL
* RUIZ SEGURA MICHELLE
* VEGA ALONSO BRENDA

FECHA: 23 DE SEPTIEMBRE DE 2025

PROFESOR: RAMON CRUZ MARTINEZ

**Contenido**

[Objetivo del proyecto 2](#_heading=h.pv1hhzicb0ee)

[Alcance 3](#_heading=h.5ciclqdjddge)

[Justificación 4](#_heading=h.3s3fc6txws98)

[Marco Teórico 5](#_heading=h.aw18o9b9pb0)

[Planteamiento de la solución 6](#_heading=h.bwzdypuahuex)

[Casos de uso 7](#_heading=h.grepoyeyuix9)

[Referencias 9](#_heading=h.tim3bq65cafk)

# **Objetivo del proyecto.**

El desarrollo de una calculadora tiene como finalidad aplicar los principios de análisis, diseño, programación y pruebas en un proyecto de software estructurado. Su implementación permite comprender la relevancia del manejo de casos especiales que pueden provocar errores en la ejecución, como las operaciones con cero o el ingreso de datos no válidos.

Este enfoque contribuye a la formación de competencias en validación de entradas, detección y gestión de errores, así como en el diseño de soluciones robustas y confiables. De igual forma, favorece la práctica del trabajo colaborativo bajo roles definidos, acercando a los estudiantes a la dinámica de proyectos reales en el ámbito profesional.

# **Alcance.**

El proyecto contempla el desarrollo de una calculadora digital que permita realizar operaciones aritméticas básicas, tales como suma, resta, multiplicación y división. Se incluirá la gestión de casos especiales, como divisiones entre cero o la introducción de datos no válidos, con el objetivo de garantizar un comportamiento controlado y confiable del sistema. El sistema contará con una interfaz sencilla que facilite la interacción del usuario, así como con la validación de entradas para evitar errores durante la ejecución. El trabajo se llevará a cabo siguiendo un proceso estructurado de análisis, diseño, implementación y pruebas, con la asignación de responsabilidades específicas a cada rol del equipo.

# **Justificación.**

La implementación de una calculadora con manejo de casos especiales (como divisiones entre cero o entradas inválidas) permite poner en práctica competencias esenciales en ingeniería de software: la validación de datos, el manejo de errores y la creación de sistemas robustos. Esto es clave para evitar defectos que pueden derivar en fallas costosas en contextos reales.

* Los estudios demuestran que la falta de calidad en el software tiene consecuencias económicas significativas: en 2020, las fallas de software costaron a la economía estadounidense cerca de 2.08 billones de dólares.
* Asimismo, un sistema inadecuado de pruebas podría implicar pérdidas de hasta 59.5 mil millones de dólares, de los cuales más de 22 mil millones serían evitables con mejores infraestructuras de testeo .

Este proyecto promueve una cultura de prevención y calidad desde etapas tempranas del ciclo de desarrollo, así como el trabajo colaborativo bajo roles definidos, conceptos fundamentales para futuros entornos profesionales.

# **Marco Teórico**

1. Calculadora (software o dispositivo)

Una calculadora es un dispositivo o aplicación cuyo propósito principal es realizar cálculos matemáticos. Puede variar en complejidad: desde una calculadora básica que opera con cifras y funciones aritméticas elementales hasta variantes científicas, gráficas o programables que permiten operaciones avanzadas como trigonometría, logaritmos o manejo de matrices .

Las primeras calculadoras eran manuales, como el ábaco, pero con el tiempo evolucionaron a dispositivos electrónicos portátiles que hoy podemos emular en software .

1. Ingeniería de Pruebas de Software

La ingeniería de pruebas se define como una disciplina que aplica un enfoque sistemático y cuantificable al proceso de prueba de software dentro del ciclo de vida del desarrollo. Su objetivo es garantizar la calidad, validez y confiabilidad del sistema .

Las pruebas permiten:

* Verificación, es decir, verificar que el software cumple con las especificaciones técnicas.
* Validación, asegurar que cumple con las necesidades del usuario .

Según esta disciplina, una prueba se considera exitosa si detecta un defecto; si no lo hace, hay una falla en la prueba .

# **Planteamiento de la solución.**

Se propone el desarrollo de una calculadora digital que permita realizar operaciones aritméticas básicas de manera confiable y controlada. El sistema integrará mecanismos de validación de entradas y manejo de errores, asegurando una experiencia de uso estable y evitando fallos comunes como divisiones entre cero o el ingreso de caracteres inválidos.

La solución será implementada bajo un enfoque modular, siguiendo las etapas de análisis, diseño, desarrollo y pruebas, con roles claramente definidos en el equipo de trabajo (analista, diseñador, desarrollador y tester). De esta manera, se garantiza un proceso ordenado que fomente la calidad del software y la colaboración en equipo.

**Requisitos funcionales.**

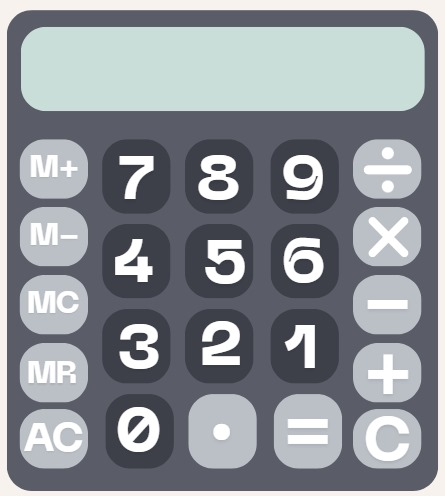
1. El sistema deberá permitir la ejecución de operaciones aritméticas básicas: suma, resta, multiplicación y división.
2. El sistema deberá identificar y manejar intentos de división entre cero, mostrando un mensaje de error adecuado.
3. El sistema deberá validar que los datos ingresados corresponden a valores numéricos antes de ejecutar la operación.
4. El sistema deberá mostrar de forma clara el resultado de la operación realizada.
5. El sistema deberá hacer uso adecuado del punto decimal, evitando que sea ingresado de forma no lógica/ incorrecta
6. El sistema deberá permitir la limpieza de los campos de entrada para realizar nuevos cálculos.
7. Gestión de Memoria de la Calculadora
   1. El sistema deberá permitir almacenar un valor en memoria mediante el botón M+ (Memory Add), sumando el valor actual o el resultado mostrado a la memoria existente.
   2. El sistema deberá permitir restar un valor de la memoria mediante el botón M- (Memory Subtract), restando el valor actual o el resultado mostrado de la memoria existente.
   3. El sistema deberá permitir recuperar el valor almacenado en memoria mediante el botón MR (Memory Recall), mostrando dicho valor en pantalla para su uso en operaciones posteriores.
   4. El sistema deberá permitir limpiar el valor almacenado en memoria mediante el botón MC (Memory Clear), estableciendo el valor de la memoria a cero.

**Requisitos no funcionales.**

1. El sistema deberá contar con una interfaz gráfica sencilla e intuitiva para el usuario.
2. El tiempo de respuesta del sistema al realizar una operación no deberá superar un segundo.
3. El sistema deberá estar implementado en un lenguaje de programación adecuado para su fácil comprensión y mantenimiento por parte del equipo.
4. El sistema deberá ser portable y ejecutarse en equipos de escritorio bajo un entorno previamente definido.
5. El sistema deberá contemplar pruebas unitarias y funcionales para garantizar su correcto funcionamiento.
6. La interfaz gráfica deberá integrar los nuevos botones de memoria (M+, M-, MR, MC) de forma coherente con el diseño existente, manteniendo la sencillez e intuitividad.
7. El acceso a las funciones de memoria deberá tener un tiempo de respuesta similar al de las operaciones aritméticas básicas (no superior a un segundo).

# **Diagrama de flujo**

# **Diseño V2.**



# **Referencias Bibliográficas.**

1. Simonova, M. (2023, 26 diciembre). Council Post: Costly Code: The price of software errors. Forbes. <https://www.forbes.com/councils/forbestechcouncil/2023/12/26/costly-code-the-price-of-software-errors/>
2. American Society of Home Inspectors, Inc. (2002, 1 octubre). Software Errors Cost U.S. Economy $59.5 Billion Annually – American Society of Home Inspectors, Inc. American Society Of Home Inspectors, Inc. <https://www.homeinspector.org/reporter-articles/software-errors-cost-u-s-economy-59-5-billion-annually/>
3. Calculadora: ¿qué es y cómo funciona? | Lenovo México. (s. f.). <https://www.lenovo.com/mx/es/glosario/calculadora/>