**Definición de Requisitos:**

* **Proyecto Sencillo:** Calculadora básica.
* **Funcionalidades:**
  + Realizar operaciones aritméticas básicas: suma, resta, multiplicación, división.
  + Interfaz gráfica simple con botones para cada operación.
  + Campo de texto para mostrar el resultado.
  + Validación de entradas para evitar errores (como división por cero).

**Requisitos Funcionales:**

* **Operaciones Aritméticas:**
  + El usuario podrá realizar las operaciones de suma, resta, multiplicación y división entre dos números.
* **Interfaz de Usuario (UI):**
  + La calculadora tendrá botones para cada operación, un campo de entrada para los números, y un campo de salida para mostrar el resultado.
* **Validación:**
  + Si el usuario intenta dividir por cero, el sistema mostrará un mensaje de error.

**Requisitos No Funcionales:**

* **Usabilidad:**
  + La interfaz debe ser fácil de usar, con una disposición intuitiva de botones y campos.
* **Rendimiento:**
  + La calculadora debe responder instantáneamente a las entradas del usuario.
* **Compatibilidad:**
  + El programa debe funcionar en cualquier sistema operativo que soporte Python.

**2. Diseño:**

**Arquitectura del Software:**

* **Módulos:**
  + main.py: Contiene la lógica principal del programa.
  + calculator.py: Define las funciones aritméticas (suma, resta, multiplicación, división).
  + gui.py: Maneja la interfaz gráfica de usuario (GUI).

**Diagrama de Flujo:**

* El diagrama muestra el flujo de datos desde la entrada del usuario hasta la salida, pasando por la selección de la operación y el cálculo.

**Maqueta (Mockup):**

* Diseño básico de la interfaz:
  + Un campo de entrada para los números.
  + Botones para cada operación (+, -, \*, /).
  + Un botón de "Calcular" que muestra el resultado en un campo de salida.
  + Un botón de "Limpiar" para reiniciar los campos de entrada y salida.

**3. Implementación:**

**3.1. Código de la Calculadora:**

* **calculator.py:**

python

Copiar código

def suma(a, b):

return a + b

def resta(a, b):

return a - b

def multiplicacion(a, b):

return a \* b

def division(a, b):

if b == 0:

return "Error: División por cero"

return a / b

* **gui.py:** (Usando Tkinter para la interfaz gráfica)

python

Copiar código

import tkinter as tk

from calculator import suma, resta, multiplicacion, division

def calcular():

num1 = float(entry1.get())

num2 = float(entry2.get())

operacion = variable.get()

if operacion == "+":

resultado = suma(num1, num2)

elif operacion == "-":

resultado = resta(num1, num2)

elif operacion == "\*":

resultado = multiplicacion(num1, num2)

elif operacion == "/":

resultado = division(num1, num2)

label\_resultado.config(text=f"Resultado: {resultado}")

root = tk.Tk()

root.title("Calculadora Básica")

entry1 = tk.Entry(root)

entry1.pack()

entry2 = tk.Entry(root)

entry2.pack()

variable = tk.StringVar(root)

variable.set("+") # valor por defecto

operaciones = ["+", "-", "\*", "/"]

option\_menu = tk.OptionMenu(root, variable, \*operaciones)

option\_menu.pack()

button\_calcular = tk.Button(root, text="Calcular", command=calcular)

button\_calcular.pack()

label\_resultado = tk.Label(root, text="Resultado: ")

label\_resultado.pack()

root.mainloop()

* **main.py:**

python

Copiar código

import gui # Este archivo iniciará la GUI de la calculadora

**4. Pruebas:**

**Pruebas Unitarias:**

* Crear un archivo test\_calculator.py y usar unittest para probar las funciones aritméticas.

python

Copiar código

import unittest

from calculator import suma, resta, multiplicacion, division

class TestCalculator(unittest.TestCase):

def test\_suma(self):

self.assertEqual(suma(2, 3), 5)

def test\_resta(self):

self.assertEqual(resta(5, 3), 2)

def test\_multiplicacion(self):

self.assertEqual(multiplicacion(3, 4), 12)

def test\_division(self):

self.assertEqual(division(10, 2), 5)

self.assertEqual(division(5, 0), "Error: División por cero")

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

**Pruebas de Integración:**

* Probar la interacción entre la interfaz gráfica y las funciones de cálculo.

**5. Documentación y Presentación:**

* Preparar una presentación que incluya:
  + Descripción del proyecto y sus funcionalidades.
  + Explicación de la arquitectura y el diseño.
  + Demostración del código y de la calculadora en funcionamiento.
  + Discusión sobre las pruebas realizadas y los resultados obtenidos.

Con estos pasos cubres todos los aspectos del proyecto desde la definición de requisitos hasta la implementación y pruebas. Si necesitas más detalles en algún paso o ayuda adicional, ¡no dudes en pedírmelo!