1. MÉTODO SUMA CON PYTHON Y PRUEBAS UNITARIAS CON PYTEST

PASO 1.1: Crear archivo calc.py

Comando

```
mkdir pipeline
cd pipeline
sudo nano calc.py
```

Contenido calc.py

```
def suma(a, b):
    try:
        # Convertir a float para ver si el dato es y un numero valido
        a = float(a)
        b = float(b)
        return a + b
    except ValueError:
        return "error"
```

PASO 1.2: Crear archivo de pruebas unitarias test.py

Comando

```
sudo nano test.py
```

Contenido test.py

```
import pytest
from calc import suma

def test_suma_numeros():
    # Prueba 1: suma de dos numeros enteros
    assert suma(2, 3) == 5

    # Prueba 2: suma de dos numeros decimales
    assert suma(2.5, 3.5) == 6.0

    # Prueba 3: suma de numero entero y numero decimal
    assert suma(2, 3.5) == 5.5

    # Prueba 4: manejo de error al recibir un texto
    assert suma(2, "texto") == "error"
```

PASO 1.3: Requerimientos para realizar pruebas unitarias en Ubuntu

Para hacer las pruebas en la terminal, necesitamos instalar pytest, que un framework de pruebas unitarias de Python para probar el archivo test.py. **pip** es el gestor de paquetes de Python, y permite instalar, actualizar y gestionar paquetes adicionales de Python como pytest.

NOTA IMPORTANTE: Cuando instales paquetes para Python lo puedes hacer de forma global o en un entorno virtual. La instalación global de paquetes en Python los hace accesibles para todos los proyectos en el sistema, pero puede generar conflictos de versiones si diferentes proyectos necesitan versiones distintas de un mismo paquete. En cambio, la instalación en un entorno virtual crea un espacio aislado de dependencias para cada proyecto, evitando conflictos y facilitando la gestión de versiones específicas. Para descontinuar un entorno virtual, basta con eliminar su carpeta, mientras que los paquetes instalados globalmente permanecen en el sistema hasta ser desinstalados manualmente.

Comandos (OPCION INSTALACIÓN GLOBAL)

```
sudo apt update
sudo apt install python3-pip
pip3 install pytest
```

Comandos (OPCION INSTALACIÓN EN UN ENTORNO VIRTUAL)

```
sudo apt update
sudo apt install python3 python3-venv
python3 -m venv venv
source venv/bin/activate
pip install pytest
```

[python3 -m venv venv] Crea un entorno virtual llamado "venv" en la carpeta actual

[source venv/bin/activate] Activa el entorno virtual

PASO 1.4: Ejecucion de pruebas unitarias con pytest

Teniendo los requerimientos instalados , ahora si puedes ejecutar las pruebas unitarias registradas en test.py

Comando

```
pytest test.py
```

Asi luce una prueba unitaria exitosa

```
root@servidor:/home/galindo/pipeline# pytest test.py
============================================================ test session starts ==========
platform linux -- Python 3.12.3, pytest-7.4.4, pluggy-1.4.0
rootdir: /home/galindo/pipeline
collected 1 item

test.py .

============================================================ 1 passed in 0.00s ===========
root@servidor:/home/galindo/pipeline#
```