



15-8-2025

Calculadora en Python mejorada



Ana Laura Díaz Becerril
CH03 SOPORTE TI

Calculadora mejorada

En esta practica me enfrente a retos de sintaxis, estructura de ideas, razonamiento lógico, manejo de funciones.

Como primera estructura de mi código me di cuenta de que las funciones estaban juntas con la petición al usuario, por lo cual cuando revise lo que pedía el test.py me percate que no iba a funcionar el testeo. Comparto el ejemplo de como estaba estructurado mi programa en una primera versión:

```
def addmultiplenumbers (numeros):  
    return sum(numeros)  
ingresa = input("Por favor ingresa los valores que quieras sumar, separados por un espacio:")  
#en esta parte como pueden ser valores negativos o algun error de usuario  
try:  
    lista_numeritos = [float(numero) for numero in ingresa.split()]  
    resultado = addmultiplenumbers(lista_numeritos)  
    print("La suma de los valores es:", resultado)  
except ValueError:  
    print("Error: Ingresa solo numeros validos separados por espacios")
```

Modifique el programa y continúe con errores, pero fueron de sintaxis completamente, sin hacer el test si funcionaba bien el programa, pero lo hice muy complicado y eso fue lo que ocasiono que no cumpliera con lo que pedía el test. Anexo evidencia de mi código complejo:

```
elif name == "par":  
    def isiteven (numeros):  
        #para validar si es par o no  
        if not numeros.is_integer():  
            print("El numero no es entero")  
            return  
        numeros = int(numeros)  
        if numeros % 2 == 0:  
            print("TRUE")  
            return True  
        else:  
            print("FALSE")  
            return False  
    ingresa = input("Por favor ingresa el valor que deseas saber si es par o impar ")
```

Instale la herramienta de pytest para poder hacer bien las pruebas. Me apoye con el uso de ChatGPT para comprender como mejorar la sintaxis por lo cual lo modifique y realice las pruebas del test, las cuales salieron bien después de la modificación de mi código complejo.

Instalación de pytest:

```
PS C:\Users\GENERATION MEXICO\Desktop\GITana\lab-calculadora> pip install pytest
Collecting pytest
  Downloading pytest-8.4.1-py3-none-any.whl.metadata (7.7 kB)
Collecting colorama>=0.4 (from pytest)
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Collecting iniconfig>=1 (from pytest)
  Downloading iniconfig-2.1.0-py3-none-any.whl.metadata (2.7 kB)
Collecting packaging>=20 (from pytest)
  Downloading packaging-25.0-py3-none-any.whl.metadata (3.3 kB)
Collecting pluggy<2,>=1.5 (from pytest)
  Downloading pluggy-1.6.0-py3-none-any.whl.metadata (4.8 kB)
Collecting pygments>=2.7.2 (from pytest)
  Downloading pygments-2.19.2-py3-none-any.whl.metadata (2.5 kB)
Downloading pytest-8.4.1-py3-none-any.whl (365 kB)
Downloading pluggy-1.6.0-py3-none-any.whl (20 kB)
Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Downloading iniconfig-2.1.0-py3-none-any.whl (6.0 kB)
Downloading packaging-25.0-py3-none-any.whl (66 kB)
Downloading pygments-2.19.2-py3-none-any.whl (1.2 MB)
1.2/1.2 MB 1.2 MB/s eta 0:00:00
Successfully installed colorama-0.4.6 iniconfig-2.1.0 packaging-25.0 pluggy-1.6.0 pygments-2.19.2 pytest-8.4.1

[notice] A new release of pip is available: 25.1.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\GENERATION MEXICO\Desktop\GITana\lab-calculadora> pytest -v
===== test session starts =====
platform win32 -- Python 3.13.5, pytest-8.4.1, pluggy-1.6.0 -- C:\Users\GENERATION MEXICO\AppData\Local\Programs\Python\Python313\python.exe
```

Primera prueba de mi programa:

```
PS C:\Users\GENERATION MEXICO\Desktop\GITana\lab-calculadora> pytest -v
===== test session starts =====
platform win32 -- Python 3.13.5, pytest-8.4.1, pluggy-1.6.0 -- C:\Users\GENERATION MEXICO\AppData\Local\Programs\Python\Python313\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\GENERATION MEXICO\Desktop\GITana\lab-calculadora
collected 8 items

test_main.py::test_add_multiple_positive_integers PASSED [ 12%]
test_main.py::test_add_multiple_complicated_numbers PASSED [ 25%]
test_main.py::test_multiply_multiple_positive_integers PASSED [ 37%]
test_main.py::test_multiply_multiple_complicated_numbers PASSED [ 50%]
test_main.py::test_is_six_even PASSED [ 62%]
test_main.py::test_is_minus_three_point_eight_even FAILED [ 75%]
test_main.py::test_is_three_an_integer PASSED [ 87%]
test_main.py::test_is_seven_point_three_an_integer PASSED [100%]

===== FAILURES =====
_____ test_is_minus_three_point_eight_even _____

    def test_is_minus_three_point_eight_even():
        response = isiteven(-3.8)
>       assert response == False
E       assert None == False

test_main.py:34: AssertionError
----- Captured stdout call -----
El numero no es entero
===== short test summary info =====
FAILED test_main.py::test_is_minus_three_point_eight_even - assert None == False
===== 1 failed, 7 passed in 0.47s =====
PS C:\Users\GENERATION MEXICO\Desktop\GITana\lab-calculadora>
```

Ultima prueba positiva:

```
PS C:\Users\GENERATION MEXICO\Desktop\GITana\lab-calculadora> pytest -v
===== test session starts =====
platform win32 -- Python 3.13.5, pytest-8.4.1, pluggy-1.6.0 -- C:\Users\GENERATION MEXICO\AppData\Local\Programs\Python\Python313\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\GENERATION MEXICO\Desktop\GITana\lab-calculadora
collected 8 items

test_main.py::test_add_multiple_positive_integers PASSED [ 12%]
test_main.py::test_add_multiple_complicated_numbers PASSED [ 25%]
test_main.py::test_multiply_multiple_positive_integers PASSED [ 37%]
test_main.py::test_multiply_multiple_complicated_numbers PASSED [ 50%]
test_main.py::test_is_six_even PASSED [ 62%]
test_main.py::test_is_minus_three_point_eight_even PASSED [ 75%]
test_main.py::test_is_three_an_integer PASSED [ 87%]
test_main.py::test_is_seven_point_three_an_integer PASSED [100%]

===== 8 passed in 0.06s =====
PS C:\Users\GENERATION MEXICO\Desktop\GITana\lab-calculadora>
```

Aprendí bastante, porque no sabía como hacer operaciones muy largas o complejas en una sola, fue bastante gratificante darme cuenta que si tenia una buena base solo tenia que refinar y simplificar más mi programación.