Lucca Palhares - 12137
Victor Ribeiro - 46423

Código rodando 100% :

```
with open("test_calculadora.py", "w") as f:
    f.write("""
import pytest
# Funções que serão testadas
def soma(a, b):
    return a + b

def subtracao(a, b):
    return a - b

def multiplicacao(a, b):
    return a * b

def divisao(a, b):
    if b == 0:
        raise ValueError("Não pode dividir por zero!")
    return a / b

# Testes
def test_soma():
    assert soma(3, 2) == 5
    assert soma(-1, 1) == 0

def test_subtracao():
    assert subtracao(5, 3) == 2
    assert subtracao(10, 10) == 0

def test_multiplicacao():
    assert multiplicacao(3, 3) == 9
    assert multiplicacao(4, -2) == -8

def test_divisao():
    assert divisao(10, 2) == 5
    with pytest.raises(ValueError):
        divisao(10, 0)
    """)
!pytest test_calculadora.py
```

```
===================================== test session starts ======================================
platform linux -- Python 3.10.12, pytest-7.4.4, pluggy-1.5.0
rootdir: /content
plugins: anyio-3.7.1, typeguard-4.3.0
collected 4 items

test_calculadora.py ....                                                              [100%]

====================================== 4 passed in 0.02s =======================================
```

Codigo com erro :

```
with open("test_calculadora.py", "w") as f:
    f.write("""
import pytest
# Funções que serão testadas
def soma(a, b):
    return a + b

def subtracao(a, b):
    return a - b

def multiplicacao(a, b):
    return a * b

def divisao(a, b):
    if b == 0:
        raise ValueError("Não pode dividir por zero!")
    return a / b

# Testes
def test_soma():
    assert soma(1, 2) == 5
    assert soma(-1, 1) == 2

def test_subtracao():
    assert subtracao(5, 3) == 9
    assert subtracao(10, 10) == 0

def test_multiplicacao():
    assert multiplicacao(3, 3) == 9
    assert multiplicacao(4, -2) == -7

def test_divisao():
    assert divisao(10, 2) == 5
    with pytest.raises(ValueError):
        divisao(10, 0)
    """)

!pytest test_calculadora.py
```

```
============================= test session starts ==============================
platform linux -- Python 3.10.12, pytest-7.4.4, pluggy-1.5.0
rootdir: /content
plugins: anyio-3.7.1, typeguard-4.3.0
collected 4 items

test_calculadora.py FFF.                                                 [100%]

=================================== FAILURES ===================================
_____________________________________ test_soma _____________________________________

    def test_soma():
        assert soma(1, 2) == 5
>       assert soma(-1, 1) == 2
E       assert 0 == 2
E        +  where 0 = soma(-1, 1)

test_calculadora.py:21: AssertionError
_____________________________________ test_subtracao _____________________________________

    def test_subtracao():
>       assert subtracao(5, 3) == 9
E       assert 2 == 9
E        +  where 2 = subtracao(5, 3)

test_calculadora.py:24: AssertionError
_____________________________________ test_multiplicacao _____________________________________

    def test_multiplicacao():
        assert multiplicacao(3, 3) == 9
>       assert multiplicacao(4, -2) == -7
E       assert -8 == -7
E        +  where -8 = multiplicacao(4, -2)

test_calculadora.py:29: AssertionError
=========================== short test summary info ============================
FAILED test_calculadora.py::test_soma - assert 0 == 2
FAILED test_calculadora.py::test_subtracao - assert 2 == 9
FAILED test_calculadora.py::test_multiplicacao - assert -8 == -7
========================= 3 failed, 1 passed in 0.11s ==========================
```

Nosso código

Erro:

```
with open("test_calculadora_v2.py", "w") as f:
    f.write("""
import pytest

# Novas funções matemáticas que serão testadas
def potencia(a, b):
    return a ** b

def raiz_quadrada(a):
    if a < 0:
        raise ValueError("Não pode calcular a raiz quadrada de número negativo!")
    return a ** 0.5

def logaritmo(a, base=10):
    import math
    if a <= 0:
        raise ValueError("O logaritmo só é definido para números positivos!")
    return math.log(a, base)

def fatorial(n):
    if n < 0:
        raise ValueError("O fatorial não é definido para números negativos!")
    if n == 0:
        return 1
    else:
        resultado = 1
        for i in range(2, n + 1):
            resultado *= i
        return resultado

# Novos testes
def test_potencia():
    assert potencia(2, 3) == 8
    assert potencia(5, 0) == 1

def test_raiz_quadrada():
    assert raiz_quadrada(4) == 2
    assert raiz_quadrada(9) == 3
    with pytest.raises(ValueError):
        raiz_quadrada(-6)

def test_logaritmo():
    assert logaritmo(100) == 2  # Logaritmo base 10 de 100
    assert logaritmo(8, 2) == 3  # Logaritmo base 2 de 8
    with pytest.raises(ValueError):
        logaritmo(-1)

def test_fatorial():
    assert fatorial(5) == 120
    assert fatorial(0) == 1
    with pytest.raises(ValueError):
        fatorial(-3)
    """)
!pytest test_calculadora.py
```

```
============================= test session starts ==============================
platform linux -- Python 3.10.12, pytest-7.4.4, pluggy-1.5.0
rootdir: /content
plugins: anyio-3.7.1, typeguard-4.3.0
collected 4 items

test_calculadora.py F...                                                 [100%]

=================================== FAILURES ===================================
_____________________________________ test_soma _____________________________________

    def test_soma():
>       assert soma(3, 2) == 0
E       assert 5 == 0
E        +  where 5 = soma(3, 2)

test_calculadora.py:30: AssertionError
=========================== short test summary info ============================
FAILED test_calculadora.py::test_soma - assert 5 == 0
========================= 1 failed, 3 passed in 0.16s ==========================
```

Sem erro:

```
with open("test_calculadora_v2.py", "w") as f:
    f.write("""
import pytest

# Funções matemáticas que serão testadas
def potencia(a, b):
    return a ** b

def raiz_quadrada(a):
    if a < 0:
        raise ValueError("Não pode calcular a raiz quadrada de número negativo!")
    return a ** 0.5

def logaritmo(a, base=10):
    import math
    if a <= 0:
        raise ValueError("O logaritmo só é definido para números positivos!")
    return math.log(a, base)

def fatorial(n):
    if n < 0:
        raise ValueError("O fatorial não é definido para números negativos!")
    if n == 0:
        return 1
    else:
        resultado = 1
        for i in range(2, n + 1):
            resultado *= i
        return resultado

# Novos testes
def test_potencia():
    assert potencia(2, 3) == 8
    assert potencia(5, 0) == 1

def test_raiz_quadrada():
    assert raiz_quadrada(4) == 2
    assert raiz_quadrada(9) == 3
    with pytest.raises(ValueError):
        raiz_quadrada(-4)

def test_logaritmo():
    assert logaritmo(100) == 2  # Logaritmo base 10 de 100
    assert logaritmo(8, 2) == 3  # Logaritmo base 2 de 8
    with pytest.raises(ValueError):
        logaritmo(-1)

def test_fatorial():
    assert fatorial(5) == 120
    assert fatorial(0) == 1
    with pytest.raises(ValueError):
        fatorial(-3)
    """)


!pytest test_calculadora_v2.py
```

```
============================= test session starts ==============================
platform linux -- Python 3.10.12, pytest-7.4.4, pluggy-1.5.0
rootdir: /content
plugins: anyio-3.7.1, typeguard-4.3.0
collected 4 items

test_calculadora_v2.py ....                                              [100%]

============================== 4 passed in 0.02s ===============================
```

Codigo com Erro:

```
with open("test_calculadora.py", "w") as f:
    f.write("""
import pytest
# Funções que serão testadas
def soma(a, b):
    return a + b

def subtracao(a, b):
    return a - b

def multiplicacao(a, b):
    return a * b

def divisao(a, b):
    if b == 0:
        raise ValueError("Não pode dividir por zero!")
    return a / b

# Testes
def test_soma():
    assert soma(3, 2) == 0
    assert soma(-1, 1) == 0

def test_subtracao():
    assert subtracao(5, 3) == 2
    assert subtracao(10, 10) == 0

def test_multiplicacao():
    assert multiplicacao(3, 3) == 9
    assert multiplicacao(4, -2) == -8

def test_divisao():
    assert divisao(10, 2) == 5
    with pytest.raises(ValueError):
        divisao(10, 0)
    """)
!pytest test_calculadora.py
```

```
========================================= test session starts ==========================================
platform linux -- Python 3.10.12, pytest-7.4.4, pluggy-1.5.0
rootdir: /content
plugins: anyio-3.7.1, typeguard-4.3.0
collected 4 items

test_calculadora.py F...                                                                        [100%]

================================================ FAILURES ================================================
_____________________________________________ test_soma _____________________________________________

    def test_soma():
>       assert soma(3, 2) == 0
E       assert 5 == 0
E        +  where 5 = soma(3, 2)

test_calculadora.py:20: AssertionError
=================================== short test summary info ====================================
FAILED test_calculadora.py::test_soma - assert 5 == 0
================================== 1 failed, 3 passed in 0.10s ==================================
```

Comece a programar ou gere código com IA.