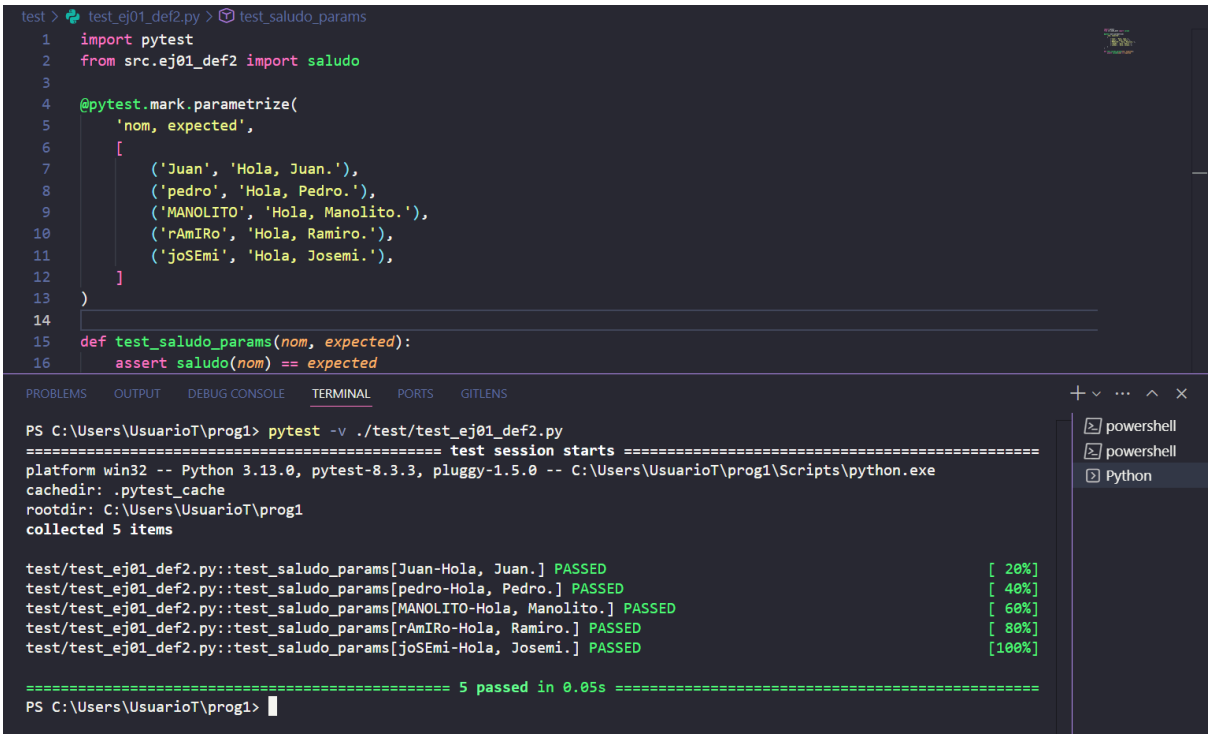
ej01_def2.py

Prueba correcta

En esta primera prueba se comprueba tanto la función saludo como el método .title()

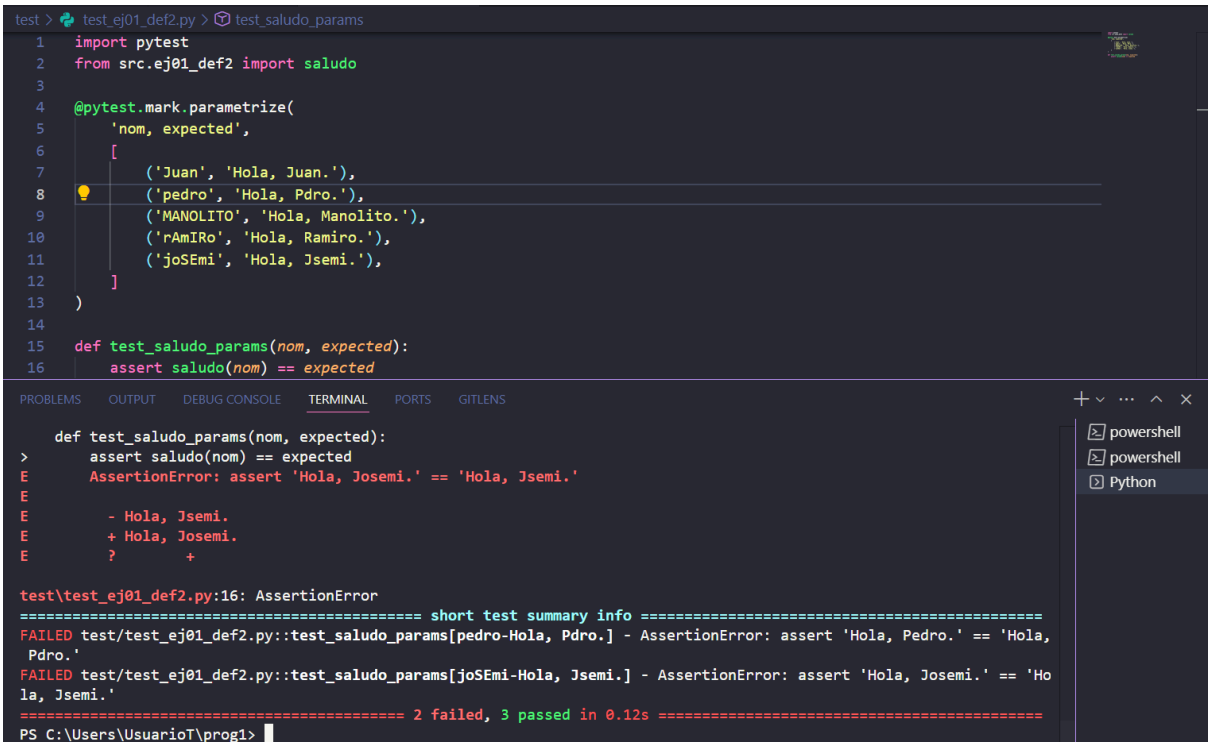


```
test > test_ej01_def2.py > test_saludo_params
import pytest
from src.ej01_def2 import saludo

@pytest.mark.parametrize(
    'nom, expected',
    [
        ('Juan', 'Hola, Juan.'),
        ('pedro', 'Hola, Pedro.'),
        ('MANOLITO', 'Hola, Manolito.'),
        ('rAmIRo', 'Hola, Ramiro.'),
        ('joSEmi', 'Hola, Josemi.'),
    ]
)

def test_saludo_params(nom, expected):
    assert saludo(nom) == expected
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

PS C:\Users\UsuarioT\prog1> pytest -v ./test/test_ej01_def2.py
============================================= test session starts =============================================
platform win32 -- Python 3.13.0, pytest-8.3.3, pluggy-1.5.0 -- C:\Users\UsuarioT\prog1\Scripts\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\UsuarioT\prog1
collected 5 items

test/test_ej01_def2.py::test_saludo_params[Juan-Hola, Juan.] PASSED                    [ 20%]
test/test_ej01_def2.py::test_saludo_params[pedro-Hola, Pedro.] PASSED                  [ 40%]
test/test_ej01_def2.py::test_saludo_params[MANOLITO-Hola, Manolito.] PASSED            [ 60%]
test/test_ej01_def2.py::test_saludo_params[rAmIRo-Hola, Ramiro.] PASSED                [ 80%]
test/test_ej01_def2.py::test_saludo_params[joSEmi-Hola, Josemi.] PASSED                [100%]

============================================== 5 passed in 0.05s ==============================================
PS C:\Users\UsuarioT\prog1>
```

Prueba con error

En esta segunda prueba se fuerza un error eliminando algunas letras de los resultados



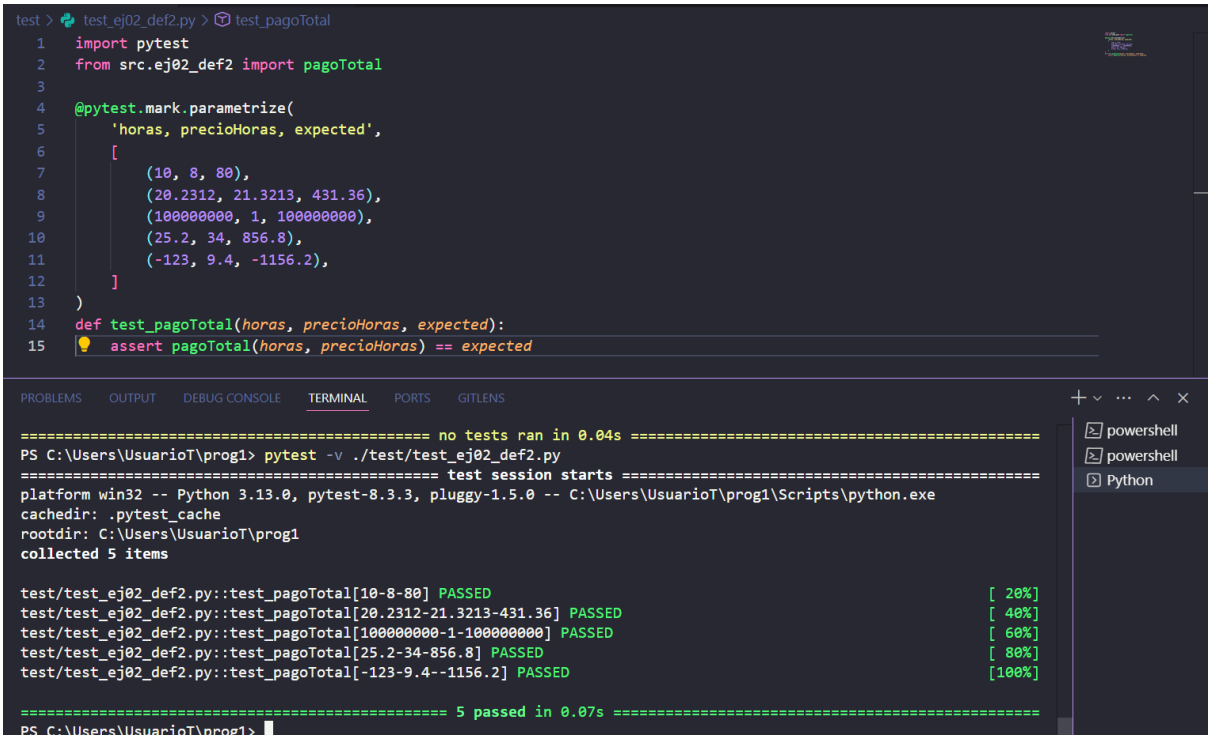
```
test > test_ej01_def2.py > test_saludo_params
import pytest
from src.ej01_def2 import saludo

@pytest.mark.parametrize(
    'nom, expected',
    [
        ('Juan', 'Hola, Juan.'),
        ('pedro', 'Hola, Pdro.'),
        ('MANOLITO', 'Hola, Manolito.'),
        ('rAmIRo', 'Hola, Ramiro.'),
        ('joSEmi', 'Hola, Jsemi.'),
    ]
)

def test_saludo_params(nom, expected):
    assert saludo(nom) == expected
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

    def test_saludo_params(nom, expected):
>       assert saludo(nom) == expected
E       AssertionError: assert 'Hola, Josemi.' == 'Hola, Jsemi.'
E
E         - Hola, Jsemi.
E         + Hola, Josemi.
E         ?        +

test\test_ej01_def2.py:16: AssertionError
========================================== short test summary info ==========================================
FAILED test/test_ej01_def2.py::test_saludo_params[pedro-Hola, Pdro.] - AssertionError: assert 'Hola, Pedro.' == 'Hola,
 Pdro.'
FAILED test/test_ej01_def2.py::test_saludo_params[joSEmi-Hola, Jsemi.] - AssertionError: assert 'Hola, Josemi.' == 'Ho
la, Jsemi.'
======================================== 2 failed, 3 passed in 0.12s ========================================
PS C:\Users\UsuarioT\prog1>
```

ej02_def2.py

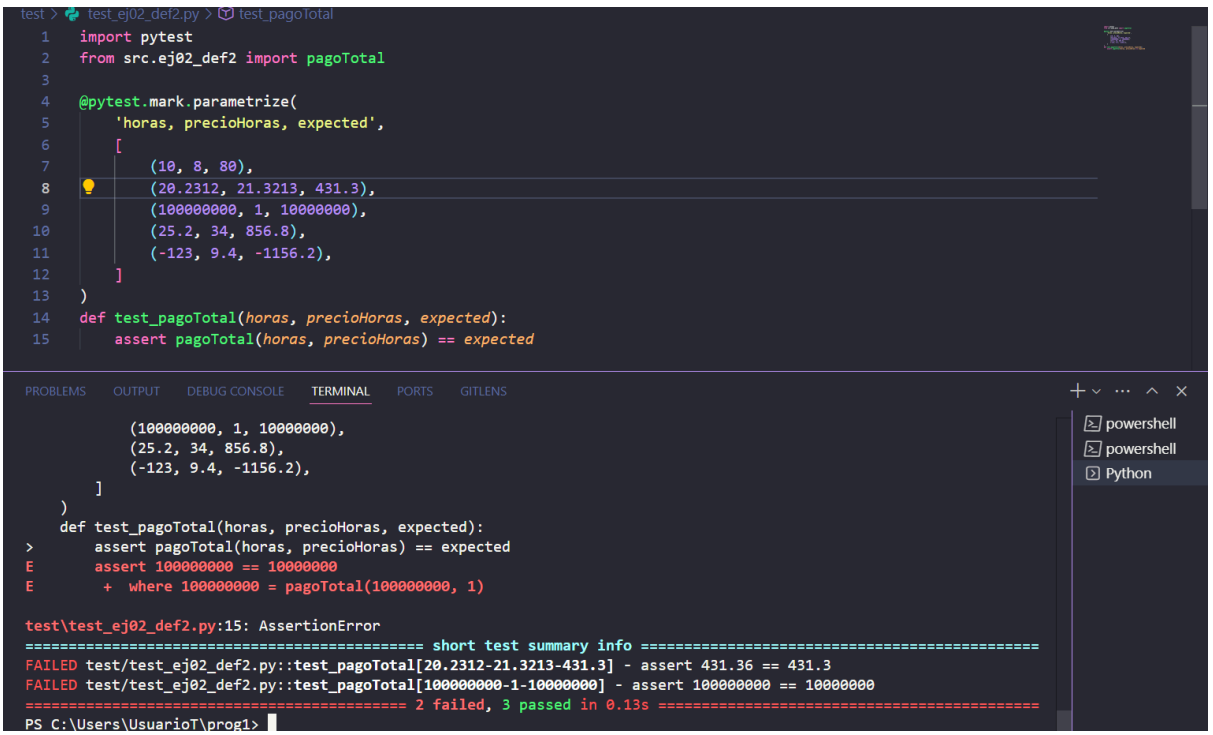
Prueba correcta

En esta primera prueba se comprueba la funcion pagoTotal y que redondee a dos decimales el resultado



Prueba con error

En esta segunda prueba se eliminan tanto dígitos como decimales comprobando ambas.



ej04_def2.py

Prueba correcta

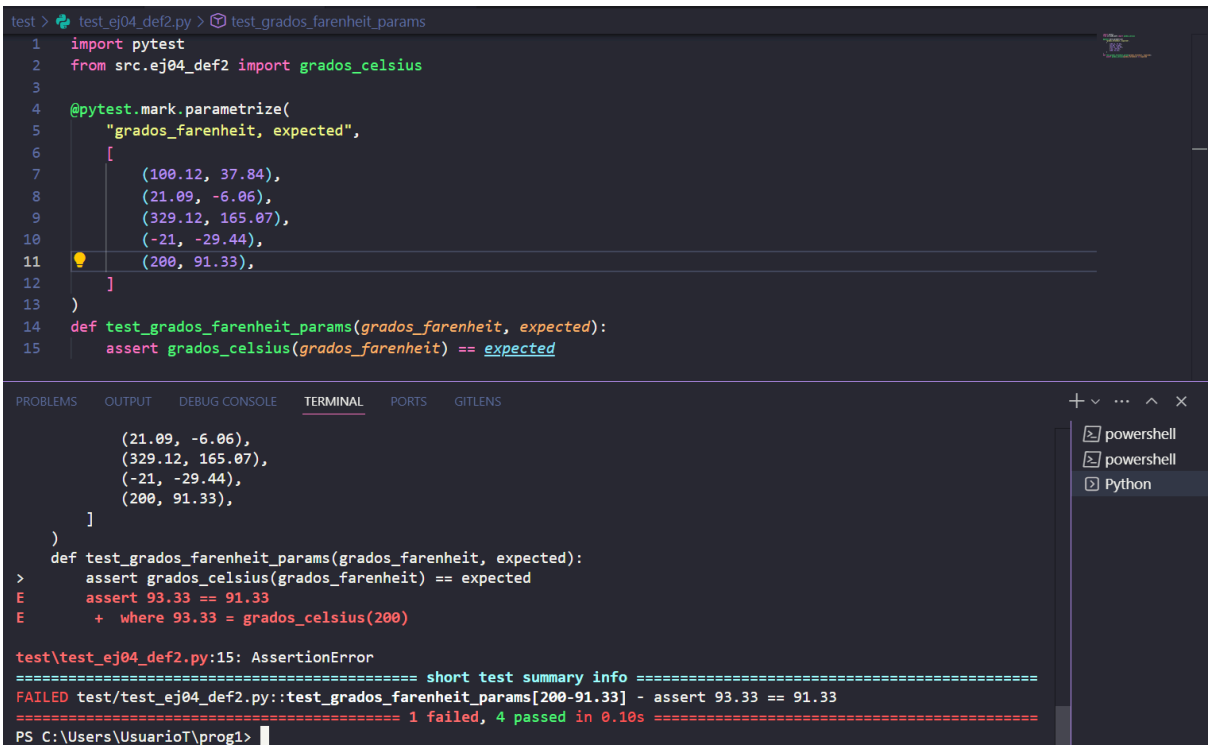
En esta primera prueba se prueba tanto la función grados_celsius como el redondeo.

```
test > test_ej04_def2.py > test_grados_farenheit_params
import pytest
from src.ej04_def2 import grados_celsius

@pytest.mark.parametrize(
    "grados_farenheit, expected",
    [
        (100.12, 37.84),
        (21.09, -6.06),
        (329.12, 165.07),
        (-21, -29.44),
        (200, 93.33),
    ]
)
def test_grados_farenheit_params(grados_farenheit, expected):
    assert grados_celsius(grados_farenheit) == expected
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

PS C:\Users\UsuarioT\prog1> pytest ./test/test_ej04_def2.py
============================================= test session starts =============================================
platform win32 -- Python 3.13.0, pytest-8.3.3, pluggy-1.5.0
rootdir: C:\Users\UsuarioT\prog1
collected 5 items

test\test_ej04_def2.py .....                                                                          [100%]

============================================== 5 passed in 0.03s ==============================================
PS C:\Users\UsuarioT\prog1>
```

Prueba con error

En esta segunda prueba se eliminan tanto dígitos como decimales comprobando ambas.



```
test > test_ej04_def2.py > test_grados_farenheit_params
import pytest
from src.ej04_def2 import grados_celsius

@pytest.mark.parametrize(
    "grados_farenheit, expected",
    [
        (100.12, 37.84),
        (21.09, -6.06),
        (329.12, 165.07),
        (-21, -29.44),
        (200, 91.33),
    ]
)
def test_grados_farenheit_params(grados_farenheit, expected):
    assert grados_celsius(grados_farenheit) == expected
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

            (21.09, -6.06),
            (329.12, 165.07),
            (-21, -29.44),
            (200, 91.33),
        ]
    )
    def test_grados_farenheit_params(grados_farenheit, expected):
>       assert grados_celsius(grados_farenheit) == expected
E       assert 93.33 == 91.33
E        +  where 93.33 = grados_celsius(200)

test\test_ej04_def2.py:15: AssertionError
=========================================== short test summary info ===========================================
FAILED test/test_ej04_def2.py::test_grados_farenheit_params[200-91.33] - assert 93.33 == 91.33
========================================= 1 failed, 4 passed in 0.10s =========================================
PS C:\Users\UsuarioT\prog1>
```

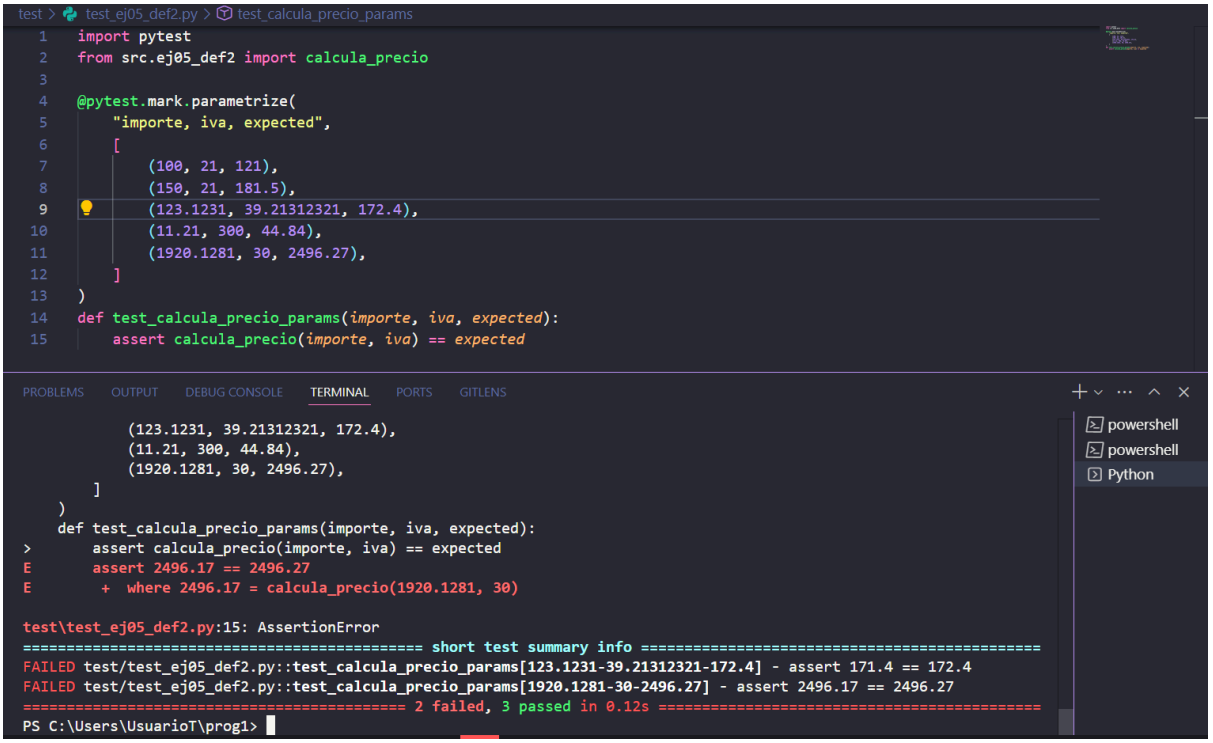
ej05_def2.py

Prueba correcta

En esta primera prueba se prueba tanto la función calcula_precio como el redondeo.

Prueba con error

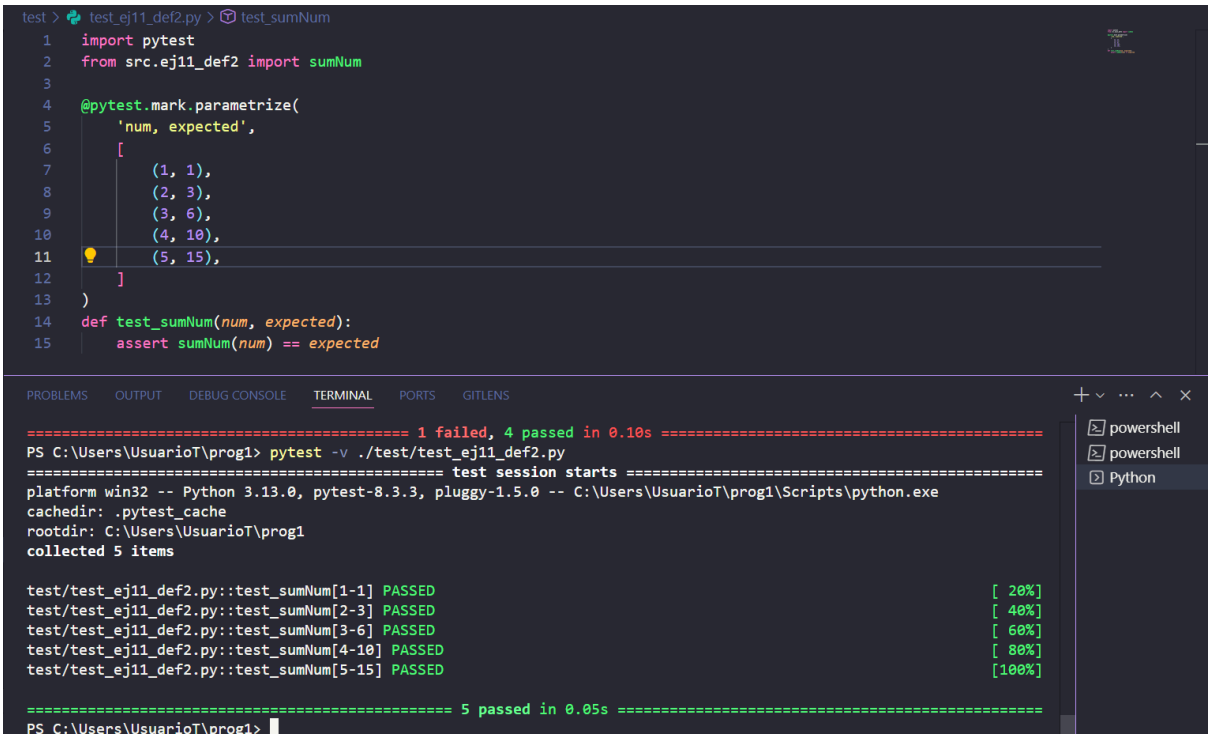
En esta segunda prueba se eliminan tanto dígitos como decimales comprobando ambas.



ej11_def2.py

Prueba correcta

En esta primera prueba se prueba tanto la función sumNum y la conversión a INT



Prueba con error

En esta segunda prueba se eliminan tanto dígitos como INTs comprobando ambas.