10/02/2026

TAREA 5.2B - DEFINICIÓN Y EJECUCIÓN DE TESTS UNITARIOS CON PYTEST

Nombre y apellidos: Cristina Sandoval Laborde

Curso: 2ºDAM

Asignatura: Desarrollo de interfaces

Índice

Qué debes comprobar (mínimo 1 test por punto)

Crea **al menos un test unitario** para cada uno de estos casos:

Constructor (Cartera(...))

1.Saldo inicial por defecto

Comprueba que, si creas una cartera sin indicar saldo, el saldo inicial es **0**.

```
def test_constructor_saldo_defecto():
    c = Cartera()
    assert c.saldo == 0
```

2.Tipo incorrecto en el constructor

Comprueba que, si al constructor le pasas un tipo incorrecto (por ejemplo, un texto), el saldo se queda en **0**.

```
def test_constructor_tipo_incorrecto():

    c = Cartera("dinero")
    assert c.saldo == 0
```

3.Saldo negativo en el constructor

Comprueba que, si al constructor le pasas un saldo negativo, el saldo se pone en **0**.

```
def test_constructor_saldo_negativo():

    c = Cartera(-100)
    assert c.saldo == 0
```

4.Saldo inicial válido (entero positivo)

Comprueba que, si al constructor le pasas un entero positivo, el saldo inicial se asigna correctamente.

```
def test_constructor_saldo_valido():

    c = Cartera(50)
    assert c.saldo == 50
```

Método ingresar(cantidad)

1.Ingresar dinero suma correctamente

Comprueba que, al ingresar dinero, el método devuelve el **nuevo saldo** (saldo anterior + ingreso) y el saldo interno queda actualizado.

Método gastar(cantidad)

```
def test_ingresar_suma_correctamente():

    c = Cartera(50)
    nuevo_saldo = c.ingresar(50)
    assert nuevo_saldo == 100
    assert c.saldo == 100
```

Test de ingresar cantidad no válida, devuelve None y no cambia el saldo

```
def test_ingresar_no_valido():
    c = Cartera(50)
    resultado = c.ingresar(-10)
    assert resultado is None
    assert c.saldo == 50
```

1.Gastar dinero resta correctamente

Comprueba que, al gastar dinero, el método devuelve el **nuevo saldo** (saldo anterior − gasto) y el saldo interno queda actualizado.

```
def test_gastar_resta_correctamente():

    c = Cartera(100)
    nuevo_saldo = c.gastar(40)
    assert nuevo_saldo == 60
    assert c.saldo == 60
```

2.Gastar más del saldo disponible

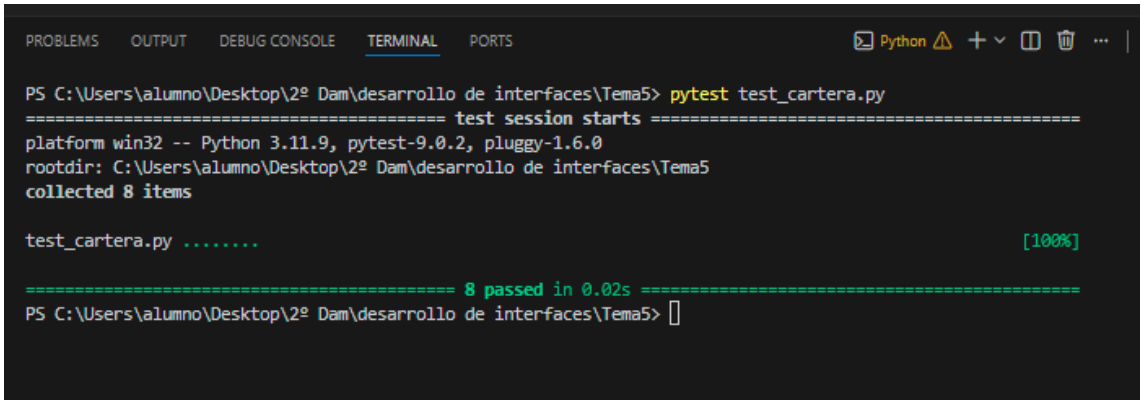
Comprueba que, si se intenta gastar más dinero del que hay, el método devuelve **None** y el saldo no cambia.

```
def test_gastar_mas_del_disponible():
    c = Cartera(50)
    resultado = c.gastar(100)
    assert resultado is None
    assert c.saldo == 50
```

Test para comprobar si se ingresa un tipo incorrecto, el saldo no cambia

```python
def test_ingresar_tipo_incorrecto():
    c = Cartera(50)
    resultado = c.ingresar("mucho")
    assert resultado is None
    assert c.saldo == 50
```

Comprobación final de la ejecución de los test con el comando pytest test_cartera.py



Webgrafía

https://experts-deny-b9a.craft.me/y3eKOUPLrtEM6g