

Estructura de Datos y POO

Semana 13





Contenido

- Pruebas unitarias



PYTHON



Las **pruebas unitarias** o **unit testing** son una forma de comprobar que un fragmento de código funciona correctamente.

Consisten en aislar una parte del código y comprobar que funciona a la perfección. Son pequeños **tests** que validan el comportamiento de un objeto y la lógica.

Unit Testing



Pruebas unitarias



El módulo unittest

Ofrece la posibilidad de crear las pruebas en el propio código implementando una clase llamada **unittest.TestCase**. Cada una de las pruebas puede devolver tres respuestas en función del resultado:

OK: Para indicar que la prueba se ha pasado exitosamente.

FAIL: Para indicar que la prueba no ha pasado exitosamente se lanzará una excepción `AssertionError` (sentencia verdadero-falso)

ERROR: Para indicar que la prueba no ha pasado exitosamente, pero el resultado en lugar de ser una aserción es otro error.

```
test_three_should_return_fizz
result = fizzBuzz.filter(3)
assert result == 'Fizz'

test_five_should_return_buzz
result = fizzBuzz.filter(5)
assert result == 'Buzz'

test_fifteen_should_return_fizz
result = fizzBuzz.filter(15)
```

Pruebas unitarias



import unittest: para indicar que se usará el framework de pruebas

class PromedioTestCase(unittest.TestCase):
indica que la clase hereda de **unittest.TestCase**.

setUp: configuraciones iniciales de las pruebas

testPromedio: es el nombre del test propiamente (siempre debe comenzar con **test**). Hace una llamada al método que se quiere probar

self.assertEqual(result, 5): valida el resultado contra el valor esperado

```
import unittest
from Promedio import promedio

class PromedioTestCase(unittest.TestCase):
    def setUp(self):
        self.lista = [1,2,3,4,5,6,7,8,9]

    def testPromedio(self):
        self.assertEqual(promedio(self.lista), 5)

if __name__ == "__main__":
    unittest.main()
```



Gracias