



PRUEBAS DINÁMICAS DE CÓDIGO

Pruebas Unitarias.

Son un tipo de prueba que se centra en evaluar y validar unidades individuales de código en un programa o sistema de software. Estas unidades suelen ser las funciones, métodos o componentes más pequeños y aislados del software, como clases, módulos o incluso líneas de código específicas.

El objetivo principal de las pruebas unitarias es verificar que cada unidad de código funcione correctamente de manera aislada, es decir, que realice su tarea prevista de manera precisa y que produzca los resultados esperados cuando se le proporcionan ciertas entradas. Las pruebas unitarias suelen ser realizadas por los desarrolladores del software como parte de su proceso de desarrollo, antes de que el código se integre en el sistema completo.

Características:

- **Aislamiento:** Las pruebas unitarias se realizan en unidades de código individuales sin depender de otros componentes del sistema. Esto garantiza que los errores se puedan localizar y corregir de manera más efectiva.
- **Automatización:** Las pruebas unitarias suelen ser automatizadas, lo que significa que se pueden ejecutar repetidamente con poco esfuerzo humano. Esto facilita la detección temprana de problemas y la identificación de regresiones después de las modificaciones de código.
- **Reutilización:** Las pruebas unitarias se pueden reutilizar a lo largo del ciclo de desarrollo para garantizar que las unidades de código sigan funcionando correctamente a medida que se realizan cambios y mejoras.
- **Identificación de errores tempranos:** Ayudan a detectar errores y problemas en las unidades de código lo antes posible en el proceso de desarrollo, lo que reduce el costo y la complejidad de corregir errores más adelante en el ciclo de desarrollo.
- **Mejora la calidad del código:** Al garantizar que las unidades de código individuales funcionen correctamente, las pruebas unitarias contribuyen a la calidad general del software y a la confiabilidad del sistema completo.



Ejemplo:

El siguiente código representa un ejemplo simple de cómo realizar pruebas unitarias en Python utilizando el módulo **unittest**.

Se definió una función llamada **suma** que toma dos números como argumentos y devuelve su suma.

Se creó una clase llamada **TestSuma** que hereda de **unittest.TestCase**. Dentro de esta clase, se definieron tres métodos de prueba:

- **test_suma_positiva**
- **test_suma_negativa**
- **test_suma_mezclada**

Cada método de prueba utiliza el método **assertEqual** para verificar si el resultado de la función suma coincide con el valor esperado. Si las afirmaciones son verdaderas, la prueba pasa; de lo contrario, falla.

Código en python:

```
import unittest

# Función que suma dos números
def suma(a, b):
    return a + b

# Definición de una clase para las pruebas unitarias
class TestSuma(unittest.TestCase):

    # Prueba de suma con valores positivos
    def test_suma_positiva(self):
        self.assertEqual(suma(2, 3), 5)

    # Prueba de suma con valores negativos
    def test_suma_negativa(self):
        self.assertEqual(suma(-2, -3), -5)

    # Prueba de suma con un valor positivo y uno negativo
    def test_suma_mezclada(self):
        self.assertEqual(suma(2, -2), 0)

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



Instrucciones:

En parejas, realiza el código para implementar una clase **pila** simple y crea pruebas unitarias en una clase **TestPila**.

Las pruebas por implementar consisten en verificar si los métodos que definan funcionan correctamente. Las funciones que probarán son:

- **push**
- **pop**
- **isEmpty**
- **size**