

Capacitación Pruebas Unitarias (xUnit)

2 de Diciembre 2022 – Sesión 10

**FUTURE
AT HEART**

Contenido



Modulo I – Conceptos Básicos

- ¿Qué es una prueba unitaria?
- Otros niveles de prueba
- Conociendo las pruebas unitarias
- Técnicas de diseño de casos de prueba
- Principios FIRST

Modulo II – Test Driven Development

- Definición
- Desarrollo Ágil – Características
- Ciclo de desarrollo TDD
- Como escribir código que se pueda probar
- Cobertura de código
- Ventajas/Desventajas

Modulo III – xUnit

- ¿Qué es xUnit?
- ¿Por qué xUnit?
- ¿Cómo funciona xUnit?
- Conceptos básicos
- Escenarios con xUnit
- Patrones de nombramiento de los escenarios
- Simulaciones
- Aserciones

Modulo IV – Taller

- Preparación del Entorno
- Ejemplo práctico
- Creando casos de prueba
- Ejecución de casos
- Mocks
- Mutation Testing
- Conclusiones/Recomendaciones



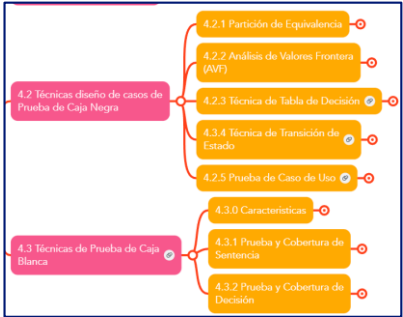
Frameworks de Pruebas unitarias - JUnit

JUnit 5, TestNG, PHPUnit, JUnit, NUnit, MSTest, Jest

JUnit

```
@Test
@DisplayName("Prueba 1: 1 + 1=2 ")
void addTwoNumbersV2() {
    Calculator calculator = new Calculator();
    int result=calculator.add(1, 1);
    assertEquals( expected: 2,result , message: "1 + 1 should equal 2");//Assert
}

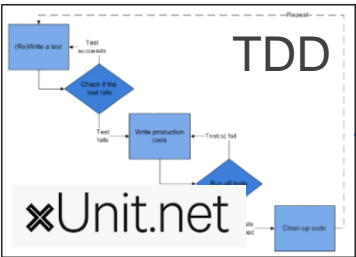
@Test
@DisplayName("Prueba 1: -1 + (-1)= -2 ")
void addTwoNumbersV3() {
    Calculator calculator = new Calculator();
    int result=calculator.add(-1, -1);
    assertEquals( expected: -2,result , message: "1 + 1 should equal 2");//Assert
}
```



Principios FIRST

Fast (rápido)
Independent (independiente)
Repeatable (repetible)
Self-validating (auto e)
Timely (oportuno)

JUnit



Taller TDD

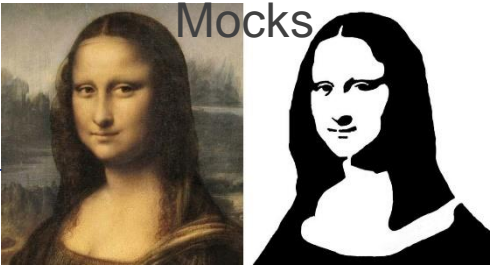
```
public class StringCalculator {
    public String calculate(String input) {
        // ...
    }
}
```

@Anotaciones JUnit

```
@Test
@DisplayName("Prueba 1: 1 + 1=2 ")
void addTwoNumbersV2() {
    Calculator calculator = new Calculator();
    int result=calculator.add(1, 1);
    assertEquals( expected: 2,result , message: "1 + 1 should equal 2");//Assert
}
```

Cobertura

```
package org.example;
import java.lang.Integer;
public class PicoPlaca {
    public boolean isValidPlaca(String ciudad, String dia, String placa) {
        // ...
    }
}
```



Complementos

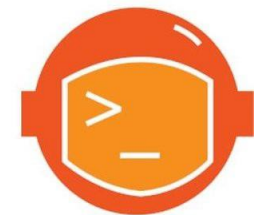
- Emular API
 - Emular API Rest para pruebas de integración (ok)
- Uso básico de Postman para pruebas
 - Uso de variables de prueba en Postman (ok)
 - Asignar datos de prueba vía archivo (ok)
 - Ejecutar desde línea de comandos con Newman (ok)
 - Soap UI
- Uso de Junit para ejecutar pruebas funcionales
 - Junit y Selenium



POSTMAN



Selenium





NTT DATA

GRACIAS

**FUTURE
AT HEART**