



# Unit Testing

# Definición

Método de pruebas en donde se prueba una sola unidad de código, para determinar si esta bien construida para ser usada

# Ventajas

Hace el proceso más ágil

Mejora la calidad del código

Detección temprana de bugs

Facilita cambios

Simplifica la integración de código

Ayuda a la documentación

Pensamiento orientado al diseño

Reducción de costos



# JUnit

## JUnit 5: org.junit.jupiter BEFORE

```
import static org.junit.jupiter.api.Assertions.fail;
import static org.junit.jupiter.api.Assumptions.assumeTrue;

import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Disabled;
import org.junit.jupiter.api.Test;

class StandardTests {

    @BeforeAll
    static void initAll() {
    }

    @BeforeEach
    void init() {
    }
}
```

## JUnit 5: org.junit.jupiter TEST

```
@Test
void succeedingTest() {
}

@Test
void failingTest() {
    fail("a failing test");
}

@Test
@Disabled("for demonstration purposes")
void skippedTest() {
    // not executed
}
```

## JUnit 5: org.junit.jupiter

### ASSERTIONS

```
class AssertionsDemo {

    private final Calculator calculator = new Calculator();

    private final Person person = new Person("Jane", "Doe");

    @Test
    void standardAssertions() {
        assertEquals(2, calculator.add(1, 1));
        assertEquals(4, calculator.multiply(2, 2),
            "The optional failure message is now the last parameter");
        assertTrue('a' < 'b', () -> "Assertion messages can be lazily evaluated -- "
            + "to avoid constructing complex messages unnecessarily.");
    }

    @Test
    void groupedAssertions() {
        // In a grouped assertion all assertions are executed, and all
        // failures will be reported together.
        assertAll("person",
            () -> assertEquals("Jane", person.getFirstName()),
            () -> assertEquals("Doe", person.getLastName())
        );
    }
}
```

## JUnit 5: org.junit.jupiter EXCEPTIONS

```
@Test
void exceptionTesting() {
    Exception exception = assertThrows(ArithmeticException.class, () ->
        calculator.divide(1, 0));
    assertEquals("/ by zero", exception.getMessage());
}
```



## JUnit 5: org.junit.jupiter AFTER

```
@AfterEach
void tearDown() {
}

@AfterAll
static void tearDownAll() {
}
```



# Mockito

# Mockito

```
public class PersonServiceImplTest {

    @Mock
    private PersonsRepository repository;

    @InjectMocks
    private PersonServiceImpl service;

    @BeforeEach
    public void setUp() { MockitoAnnotations.openMocks( testClass: this); }

    @Test
    public void testGetAllEmpty() {
        when(repository.findAll()).thenReturn(this.mockList( empty: true));

        List<PersonDTO> result = service.getAll();

        assertNotNull(result);
        assertEquals( expected: 0, result.size());
    }

    @Test
    public void testGetAll() {
        when(repository.findAll()).thenReturn(this.mockList( empty: false));

        List<PersonDTO> result = service.getAll();

        assertNotNull(result);
        assertEquals( expected: 3, result.size());
        assertEquals( expected: 2, result.get(1).getId());
    }
}
```



# Mock VS Stub

# Mock VS Stub

Tanto los mocks como los stubs son considerados dobles de pruebas, los cuales pueden adoptar el comportamiento que se desea con el fin de satisfacer la prueba.

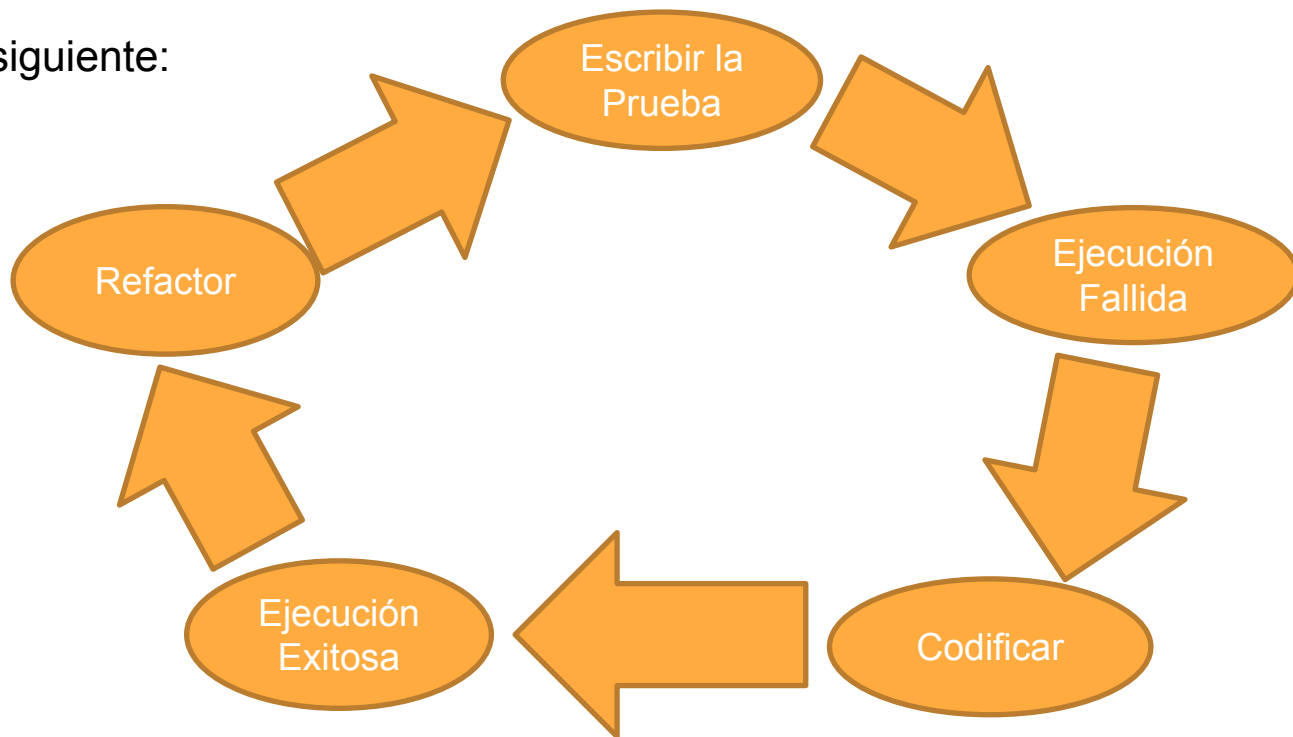
La diferencia radical es que **los stubs son usualmente programados manualmente**, mientras que los mocks son manejados por una framework determinado

# Test Driven Development

# TDD

Proceso basado en repeticiones cortas de desarrollo enfocado en el concepto de probar primero.

El ciclo es el siguiente:





# Ejemplo



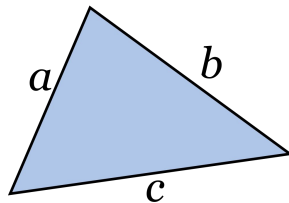
## Ejemplo

Se requiere crear una clase que calcule el area y el perimetro de unos triángulos generados por un servicio externo. Adicionalmente se debe validar que los triangulos sean válidos.

Un triángulo es válido si:

1. Sus lados son mayores a 0
2. Cumple con el teorema de desigualdad triangular

Desigualdad  
del triángulo



$$a + b > c$$

$$b + c > a$$

$$c + a > b$$

# Algunas Referencias

<https://junit.org/junit5/docs/current/user-guide/>

<https://danielme.com/2017/07/24/tutorial-test-doubles-mockito/>

<https://www.javatpoint.com/mockito>