

# Testing para Java EE

---

Víctor Orozco

25 de abril de 2018

Nabenik

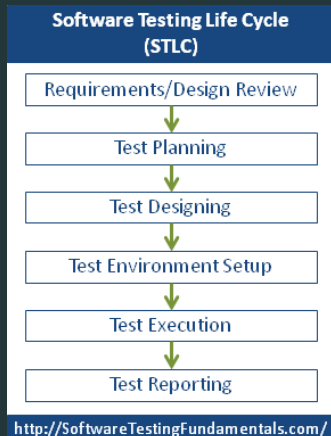
# Víctor Orozco

- Developer (JVM/Open Source Advocate)
- JUG Leader
- Consultor independiente
- Profesor universitario
- @tuxtor
- <http://vorozco.com>
- <http://tuxtor.shekalug.org>



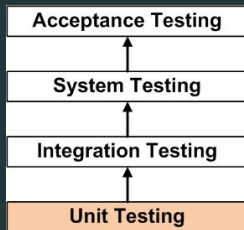
NABENIK





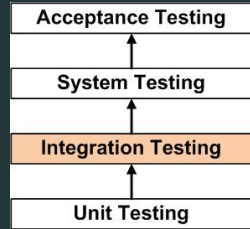
# Unit Testing - Tipos

- Funcionamiento de cada unidad
- Estructurado: Programa, función, procedimiento
- POO: Clase y derivados (servicio, endpoint)
- Frameworks, drivers, stubs, mocks
- Developers
- Tip: Probar lo que tiene impacto



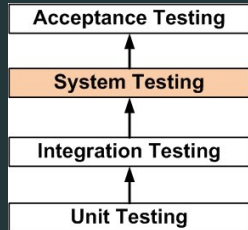
# Integration Testing - Tipos

- Unidades combinadas
- Interfaces e integraciones
- Sistemas externos
- Frameworks, White Box, Black Box
- Developers y testers
- Tip: Documentar integraciones



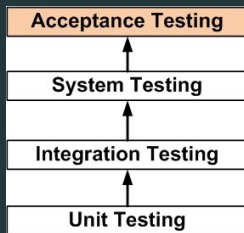
# System Testing - Tipos

- Sistema vs. requerimientos
- Black Box
- Testers
- Tip: Requerimientos actualizados



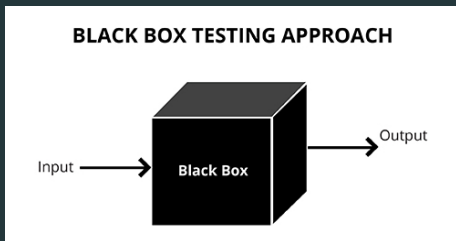
# Acceptance Testing - Tipos

- Requerimientos vs. necesidades reales
- Ad-hoc
- Pruebas internas: Alpha, Testers
- Pruebas externas: Beta, Usuarios y clientes



# Black box - Técnicas

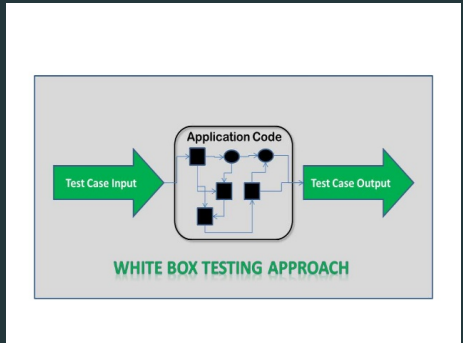
- Integration
- System
- Acceptance





# White box - Técnicas

- Unit
- Integration
- System

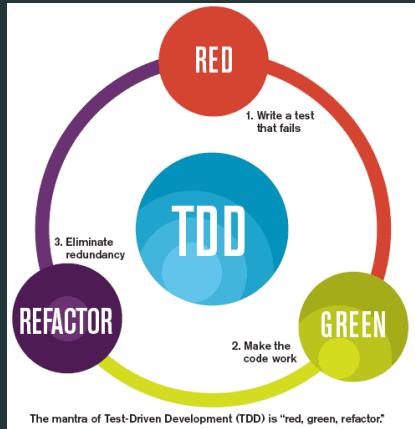


- Acceptance
- Monkey Testing



# TDD - Técnicas

- Unit
- Integration

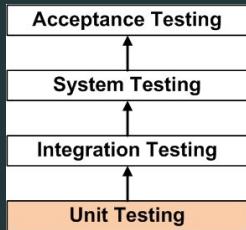


# Unit testing

---

# Unit Testing - Java

- JUnit, TestNG
- Mockito, JMock, JMockit
- Arquillian (parcialmente)



## JUnit 4

---

- Primera prueba
- @Before @After
- Asserts
- @BeforeClass @AfterClass
- Pruebas parametrizadas
- Suites

## JUnit - BeforeClass - AfterClass

Se ejecuta una vez por *testing class*, como primera acción

```
@BeforeClass
```

```
public static void setUpClass() {  
    System.out.println("First_on_every_class");  
}
```

Se ejecuta una vez por *testing class*, como ultima acción

```
@AfterClass
```

```
public static void tearDownClass() {  
    System.out.println("Last_on_every_class");  
}
```



## JUnit - Before - After

Se ejecuta una vez por *@Test*, como primera acción

@Before

```
public void testSetup(){  
    System.out.println("First_on_every_test");  
}
```

Se ejecuta una vez por *@Test*, como ultima acción

@After

```
public void destroyTest(){  
    System.out.println("Test_ending");  
}
```

## JUnit - Test

Ejecuta una prueba completa a partir de la comprobación de *asserts*

```
@Test
```

```
public void testProduct(){  
    System.out.println("Product");  
    assertEquals(50, calc.product(10, 5));  
}
```

Comprueba excepciones

```
@Test(expected = ArithmeticException.class)  
public void testDivideByZero(){  
    calc.divide(10, 0);  
}
```

Comprueba timeout

```
@Test(timeout = 2000)
public void testSuperMultiplication()
    throws InterruptedException{
    assertEquals(10,
        calc.superMultiplication(10, 1));
}
```

# Mockito

---

- Stub
- Mockito.mock
- when().then()
- @Mock
- @Rule

# Mockito - Stub

Creación de stubs mediante decoración

```
CalculatorService calcService =  
    Mockito.mock(CalculatorService.class);
```

Decoración con metadatos

```
@Mock  
CalculatorService calcService;
```

```
@Rule  
public MockitoRule rule = MockitoJUnit.rule();
```

Escritura de reglas

```
when(calcService.add(3, 2)).thenReturn(5);
```

Verificación de uso

```
verify(calcService).add(3, 2);
```

# Integration testing

---



## J2EE

- Pruebas unitarias imposibles.<sup>en</sup> todo el stack
- Pruebas de integración difíciles
- Contenedores lentos

## JavaEE

- Pruebas unitarias imposibles.<sup>en</sup> todo el stack
- Pruebas de integración con arquillian y enablers
- Microcontenedores y micro despliegues

# JavaEE Enablers

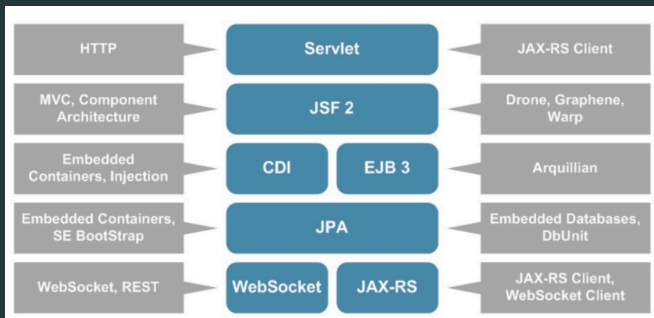


Figura 1: Testing Java EE - Créditos: Reza Rahman

# Arquillian



Figura 2: Estructura prueba Arquillian - Créditos: Proyecto Arquillian

- Embedded: Misma JVM que el runner
- Managed: Contenedor externo, lifecycle con arquillian
- Remote: Contenedor externo

```
@Deployment
public static WebArchive createDeployment() {
    return ShrinkWrap
        .create(WebArchive.class,
            "todo-list-servlet-test.war")
        .addClass(ItemServlet.class);
}
```

- vorozco@nabenik.com
- <http://nabenik.com>



This work is licensed under a Creative Commons  
Attribution-ShareAlike 3.0 Guatemala License.