

# **JUnit y TestNG**

**Laboratorio 05**  
**Ing. Henry Mendoza Puerta**

**JUNIT**

# JUnit

- Se trata de un framework de pruebas orientadas a programadores Java
- Existe dos versiones fuertemente usadas
  - JUnit 3
  - JUnit 4



# JUnit 4

- Utiliza anotaciones a partir del JDK 5 para adelante.
  - Identificar los tests, en vez de usar convenciones de nombres.
  - Tratar excepciones, en vez de capturarlas manualmente.
  - Limitar el tiempo.
  - Ignorar tests.

# Diferencias

- JUnit 3: Los métodos deben empezar por “test”

```
public void testMetodo() {}
```

- JUnit 4: Usa anotaciones

```
@Test public void probarMetodo() {}
```

# Diferencias

- JUnit 4: No es necesario extender TestCase. Podemos importar (los métodos de) la clase Assert.

```
import static org.junit.Assert.*;  
public class PruebaTest {  
    @Test public void probarMetodo() {  
        }  
}
```

# Diferencias

- JUnit 3: Tratar las excepciones manualmente.

```
public void testMetodo() {  
    try {  
        Prueba.calcularMayor(new int[] {});  
        fail("Debe ocurrir un error");  
    } catch (RuntimeException e) {  
        ...  
    }  
}
```

# Diferencias

- JUnit 4: Podemos indicar qué excepción esperamos. Si no se produce, el test fallará.

```
@Test(expected=RuntimeException.class)  
public void probarMetodo() {  
    Prueba.calcularMayor(new int[] {});  
}
```



# Diferencias

- JUnit 4: Podemos anotar un test para que falle si tarda más de x milisegundos:

```
@Test(timeout=500) public void probarMetodo() {}
```

# Diferencias

- JUnit 4: Podemos anotar un test para que no se ejecute de momento. El sistema nos avisará de que ha sido ignorado.:

```
@Ignore public void ignorarMetodo() {}
```

- JUnit 3: Inicializar y limpiar el entorno

```
protected void setUp() {  
    ...  
}  
protected void tearDown() {  
    ...  
}
```

# Diferencias

- JUnit 4: Podemos anotar uno o varios métodos para que se ejecuten antes / después de cada método de la clase test:

```
@Before protected void ejecutarAntesMetodo() {}
```

```
@After protected void ejecutarDespuesMetodo() {}
```

# Diferencias

- JUnit 4: También podemos anotar métodos para que se ejecuten antes / después del conjunto de métodos de la clase test: :

```
@BeforeClass protected void ejecutarInicio() {}
```

```
@AfterClass protected void ejecutarFin() {}
```

TESTNG

# TestNG

- TestNG es un framework para pruebas y testing que trabaja con Java y está basado en JUnit (para Java) y NUnit (para .NET), pero introduciendo nuevas funcionalidades que los hacen más poderosos y fáciles de usar.



# TestNG

- Soporte para pruebas para data-driven testing (with `@DataProvider`).
- Permite distribución de las pruebas en maquinas esclavas.
- Soportado por herramientas y plugins importantes y variados como: (Eclipse, IDEA, Maven, etc.).
- Métodos dependientes para pruebas sobre servidores de aplicación.



# Diferencias

## Functionality - JUnit 4 vs TestNG

	Annotation Support	Exception Test	Ignore Test	Timeout Test	Suite Test	Group Test	Parameterized (primitive value)	Parameterized (object)	Dependency Test
TestNG	✓	✓	✓	✓	✓	✓	✓	✓	✓
JUnit 4	✓	✓	✓	✓	✓	✗	✓	✗	✗