

Pt3 TestNG

1. Quins altres tipus de test diferents als unitaris permet, i mostrar exemples.

Captura de pantalla de 2019-05-30 16-09-13Priority -> se le indica el orden en que se dese que se ejecuten los test

```
@Test(priority = 1)
public void testEsPrimer() {
    System.out.println("esPrimer");
    int num = 12;
    boolean expResult = false;
    boolean result = Numeros.esPrimer(num);
    assertEquals(expResult, result);
}

/**
 * Test of longitudNumero method, of class Numeros.
 */
@Test(priority = 0)
public void testLongitudNumero() {
    System.out.println("longitudNumero");
    int numero = 486468486;
    int expResult = 9;
    int result = Numeros.longitudNumero(numero);
    assertEquals(expResult, result);
}
```

Se ejecuta primero el que tiene el orden de prioridad 0

The screenshot displays the TestNG test results interface. On the left, a tree view shows the test suite structure: 'All 6 tests passed. (0,03 s)' is expanded, revealing 'Command line test' which has passed. Below it, six individual test methods are listed, each with a green checkmark and a 'passed' status: 'metodes.NumerosNGTest.divideByZero' (0,0 s), 'metodes.NumerosNGTest.method1' (0,015 s), 'metodes.NumerosNGTest.testLongitudNumero' (0,014 s), 'metodes.NumerosNGTest.testMcdRecursiu' (0,001 s), 'metodes.NumerosNGTest.testRevesNumerov1' (0,0 s), and 'metodes.NumerosNGTest.testEsPrimer' (0,0 s). On the right, a text area shows the TestNG execution log, indicating that the 'Command line suite' ran successfully with 'Total tests run: 6, Failures: 0, Skips: 0'.

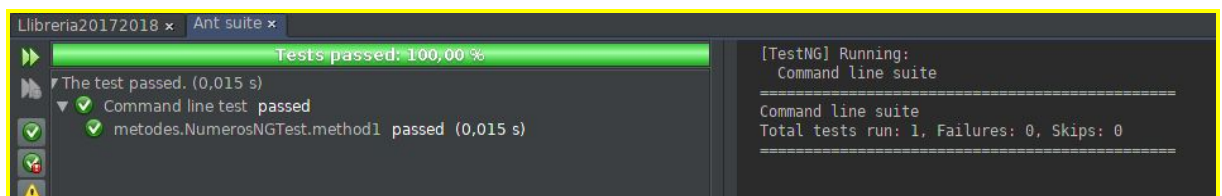
Test Name	Status	Time
metodes.NumerosNGTest.divideByZero	passed	(0,0 s)
metodes.NumerosNGTest.method1	passed	(0,015 s)
metodes.NumerosNGTest.testLongitudNumero	passed	(0,014 s)
metodes.NumerosNGTest.testMcdRecursiu	passed	(0,001 s)
metodes.NumerosNGTest.testRevesNumerov1	passed	(0,0 s)
metodes.NumerosNGTest.testEsPrimer	passed	(0,0 s)

El @Timeout indica el tiempo de espera que tiene a un test y el el @suppressWarnings se le indica para eliminar los posibles errores al ejecutar el test

```
/**
 *
 */
@Test(timeout = 10)
@SuppressWarnings("empty-statement")
public void method1()
{
    for (int i = 0; i < 9999; i++) {

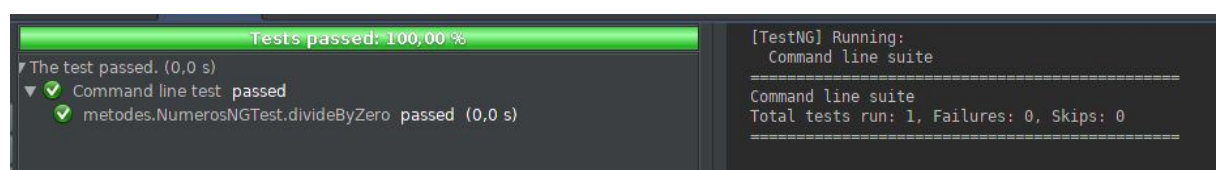
    }
}
```

Se ve que se ejecuta en un tiempo muy corto



Cuando un test propenso a fallar se le agrega la exception que responderá

```
@Test(expectedExceptions = ArithmeticException.class)
public void divideByZero()
{
    int i = 1/0;
}
```



Para ignorar un test se le cambia el estado de enabled a false

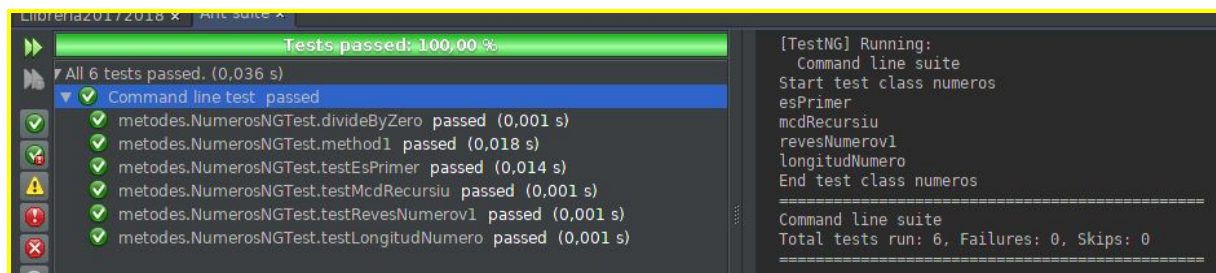
```
@Test(enabled=false)
public void TestWithException()
{
    System.out.println("Soy ignorado :( ");
}
```

2. Exemples d'anotacions suportades.

@BeforeClass y @AfterClass son anotaciones que se ejecutan antes y después de una clase que contiene tests

```
@BeforeClass
public static void setUpClass() throws Exception {
    System.out.println("1");
}

@AfterClass
public static void tearDownClass() throws Exception {
    System.out.println("2");
}
```



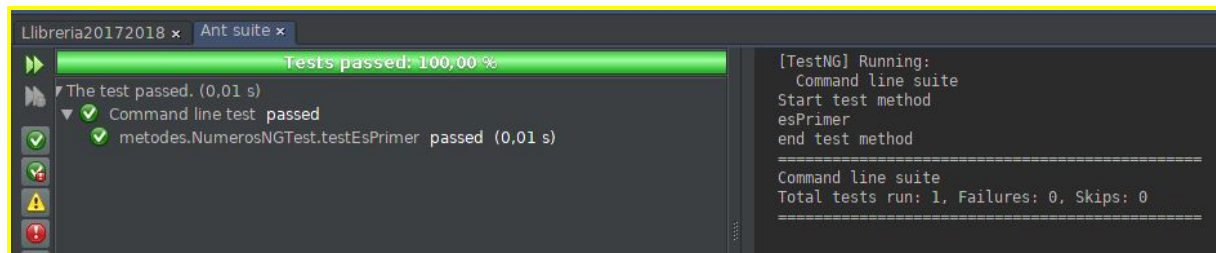
@BeforeMethod y @AfterMethod sons anotaciones que se ejecutan antes y despues de un solo test

```

@BeforeMethod
public void setUpMethod() throws Exception {
    System.out.println("Start test method ");
}

@AfterMethod
public void tearDownMethod() throws Exception {
    System.out.println("end test method");
}

```



3. Per què és més orientat a Objectes?

TestNG està dissenyat per cobrir totes les categories de proves: unitat, funcional, extrem a extrem, integració.

	JUnit4	TestNG
Annotation Support	✓	✓
Suite Test	✓	✓
Ignore Test	✓	✓
Exception Test	✓	✓
Timeout	✓	✓
Parameterized Test	✓	✓
Dependency Test	✗	✓

4. Per què la configuració de les proves és més senzilla?

Simplifica la forma en que se codifiquen les proves. Ya no hay necesidad de un método principal estático en nuestras pruebas. La secuencia de acciones

está regulada por anotaciones fáciles de entender que no requieren que los métodos sean estáticos.

Las excepciones no detectadas son manejadas automáticamente por TestNG sin terminar la prueba prematuramente. Estas excepciones se informan como pasos fallidos en el informe.

5. Exemple d'agrupament dels tests i posada de codi abans/després dels grups de test

Test en grupos

```
@Test(groups={"grupo1"})
public void testRevesNumerov1() {
    System.out.println("revesNumerov1");
    int numero = 151;
    int expectedResult = 151;
    int result = Numeros.revesNumerov1(numero);
    assertThat(expResult, is(equalTo(result)));
}

@Test(groups={"grupo1"})
public void testMcdRecursiu(){
    System.out.println("mcdRecursiu");
    int a =23;
    int b =13;
    int expectedResult = 1;
    int result = Numeros.mcdRecursiu(a, b);
    assertThat(expResult, is(equalTo(result)));
}
```



```

<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >
<suite name="Llibreria20172018">
  <test name="metodes suite">
    <!--
      <packages>
        <package name="metodes"/>
      </packages>-->
    <groups>
      <run>
        <include name="grupol"/>
      </run>
    </groups>
    <classes>
      <class name="metodes.NumerosNGTest"/>
    </classes>
  </test>
</suite>

```

```

tests passed: 100,00 %
Both tests passed. (0,015 s)
  metodes suite passed
    metodes.NumerosNGTest.testMcdRecursiu passed (0,015 s)
    metodes.NumerosNGTest.testRevesNumerov1 passed (0,0 s)

```

6. Exemple de com definir dependències entre tests

```

@Test(dependsOnMethods = { "testEnviroment" })
public void testMostraMissatgeSanti(){
    System.out.println("mostraMissatgeSanti");
    String missatge="Hola";
    String expResult="Hola Santi";
    String result= Cadenes.mostraMissatgeSanti(missatge);

    assertThat(expResult,is(not(result)));
}

@Test
public void testEnviroment(){
    System.out.println("This is enviroment");
}

```

7. Exemple de com passar dades als tests amb fitxers XML, txt, ...

Un test suite para ejecutar todos los métodos que se encuentran en un paquete se crea la etiqueta 'package' y se le pone el nombre donde se encuentra las clases con los test unitarios

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >
<suite name="Llibreria20172018">
  <test name="metodes suite">
    <packages>
      <package name="metodes"/>
    </packages>
  </test>
</suite>
```

The screenshot shows the TestNG results window. The top bar indicates 'Tests passed: 100.00 %'. Below this, a list of tests is shown, all marked as 'passed'. The tests are:

- metodes suite passed
- metodes.NumerosNGTest.testEsPrimer passed (0.002 s)
- metodes.NumerosNGTest.testRevesNumerov1 passed (0.0 s)
- metodes.NumerosNGTest.testLongitudNumero passed (0.0 s)
- metodes.CadenesNGTest.testBuscaCadenaNot passed (0.006 s)
- metodes.CadenesNGTest.testBuscaCadenanull passed (0.001 s)
- metodes.CadenesNGTest.testBuscaCadena passed (0.0 s)

On the right side, the [TestNG] Running path is shown: /home/mirokshi/code/mirokshi/ProjectsJava/Llibreria/esPrimer/revesNumerov1/longitudNumero/buscaCadena/buscaCadena. Below this, a summary for 'Llibreria20172018' is provided: Total tests run: 6, Failures: 0, Skips: 0.

Para ejecutar solo una clases en un test suite se crea la etiqueta 'classes' donde se le indica que clase testear

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >
<suite name="Llibreria20172018">
  <test name="metodes suite">
    <classes>
      <class name="metodes.CadenesNGTest"/>
    </classes>
  </test>
</suite>
```

<https://www.dignitasdigital.com/blog/testng-vs-junit-which-testing-framework-is-best/> JUnit vs TestNg

<https://testng.org/doc/documentation-main.html#testng-xml> -TestNG

<http://hamcrest.org/JavaHamcrest/javadoc/1.3/org/hamcrest/CoreMatchers.html> -List assertions

<https://javacodehouse.com/blog/junit-tutorial/> - Tuto

<https://netbeans.org/kb/docs/java/testng-screencast.html> -Video creación Test NG