

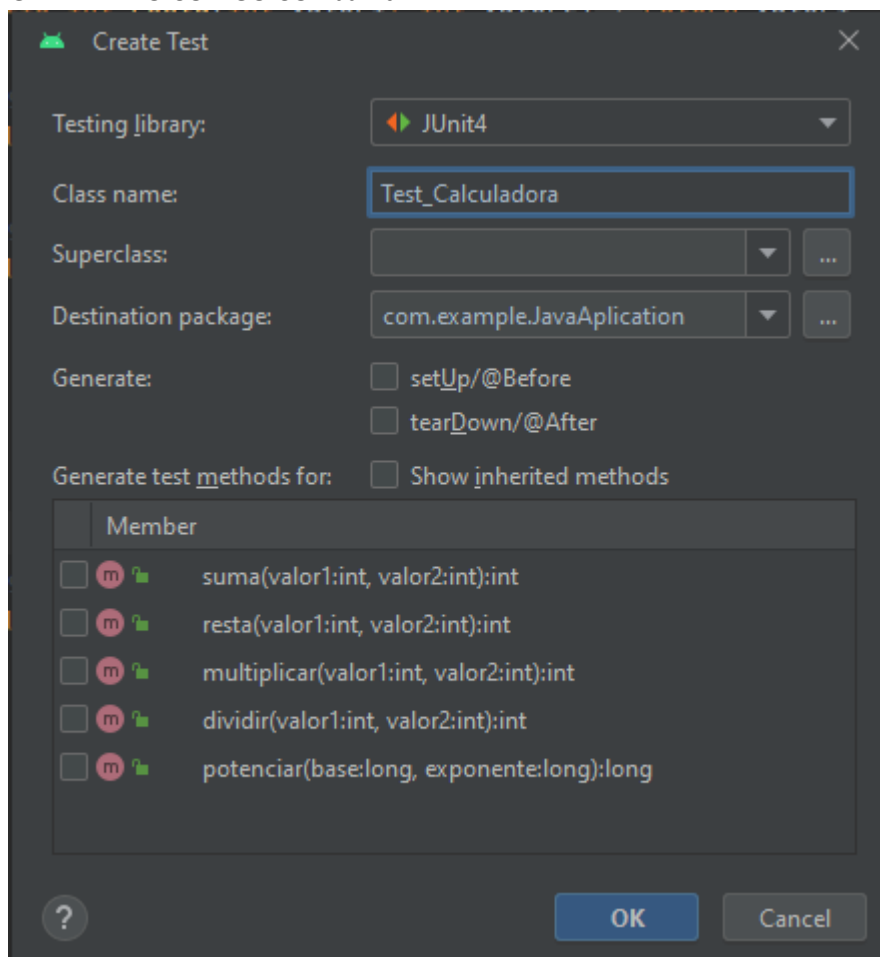
INFORME JAVA TEST CALCULADORA

ESTEBAN BUSTAMANTE: 2220221053

JUAN ESTEBAN GARZON: 2220221046

1. Código:

GENERANDO CODIGO CON Junit 4:



- MAINACTIVITY_TEST.JAVA

```
package com.example.JavaAplication;

import org.junit.Test;

import static org.junit.Assert.assertEquals;

|
no usages
public class MainActivityTest {

    no usages
    @Test
    public void Suma() {
        Test_Calculadora test = new Test_Calculadora();
        int resultado = test.suma(2, 3);
        assertEquals( expected: 5, resultado);
    }

    no usages
    @Test
    public void Resta() {
        Test_Calculadora test = new Test_Calculadora();
        int resultado = test.resta(5, 3);
        assertEquals( expected: 2, resultado);
    }

    no usages
}
```

```
no usages
@Test
public void Multiplicar() {
    Test_Calculadora test = new Test_Calculadora();
    int resultado = test.multiplicar(4, 6);
    assertEquals( expected: 24, resultado);
}

no usages
@Test
public void Dividir() {
    Test_Calculadora test = new Test_Calculadora();
    int resultado = test.dividir(8, 2);
    assertEquals( expected: 4, resultado);
}

no usages
@Test
public void Potenciar() {
    Test_Calculadora test = new Test_Calculadora();
    long resultado = test.potenciar( base: 2, exponente: 3);
    assertEquals( expected: 8, resultado);
}
}
```

- TEST_CALCULADORA.JAVA:

The screenshot shows an IDE with a project structure on the left and the source code of `Test_Calculadora.java` on the right. The project structure includes:

- app
 - manifests
 - java
 - com.example.JavaAplication
 - MainActivity
 - Test_Calculadora
 - com.example.JavaAplication (androidTest)
 - com.example.JavaAplication (test)
 - ExampleUnitTest
 - MainActivityTest
 - java (generated)
 - res
 - res (generated)
 - Gradle Scripts

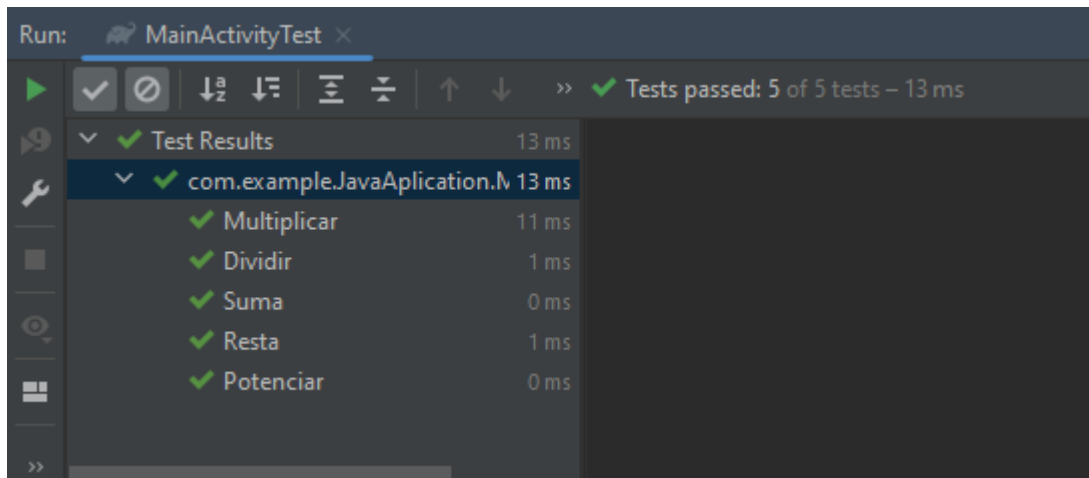
The source code of `Test_Calculadora.java` is as follows:

```
1 package com.example.JavaAplication;
2
3 10 usages
4 public class Test_Calculadora {
5     1 usage
6     public int suma(int valor1, int valor2) { return valor1 + valor2; }
7
8     1 usage
9     public int resta(int valor1, int valor2) { return valor1 - valor2; }
10
11     1 usage
12     public int multiplicar(int valor1, int valor2) { return valor1 * valor2; }
13
14     1 usage
15     public int dividir(int valor1, int valor2) {
16         if (valor2 == 0) {
17             throw new IllegalArgumentException("No se puede dividir por cero");
18         }
19         return valor1 / valor2;
20     }
21
22     1 usage
23     public long potenciar(long base, long exponente) {
24         if (exponente < 0) {
25             throw new IllegalArgumentException("El exponente debe ser positivo o cero");
26         }
27     }
28 }
```

This is a close-up view of the `potenciar` method in `Test_Calculadora.java`. The code is as follows:

```
1 usage
2 public long potenciar(long base, long exponente) {
3     if (exponente < 0) {
4         throw new IllegalArgumentException("El exponente debe ser positivo o cero");
5     }
6
7     long resultado = 1;
8     for (int i = 0; i < exponente; i++) {
9         resultado *= base;
10    }
11    return resultado;
12 }
```

2. EJECUCION:



- EJECUCION COON ERROR EN VALOR ESPERADO:

