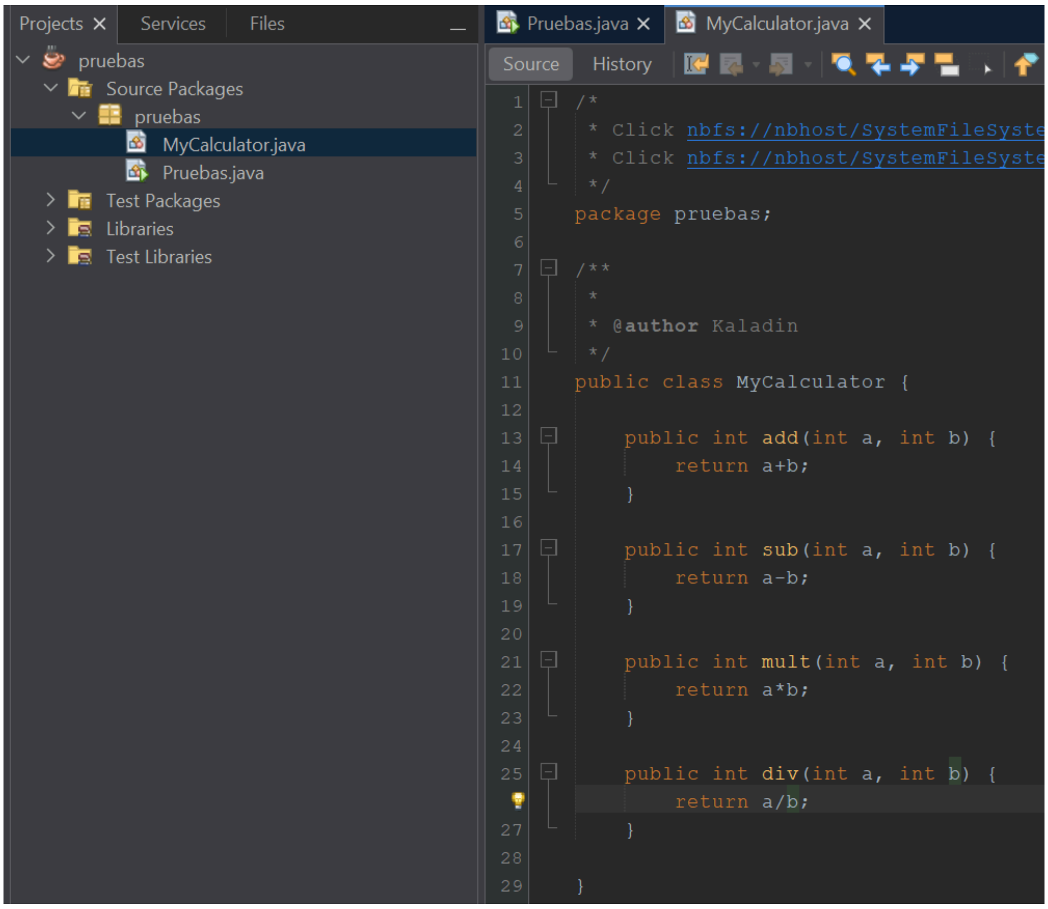
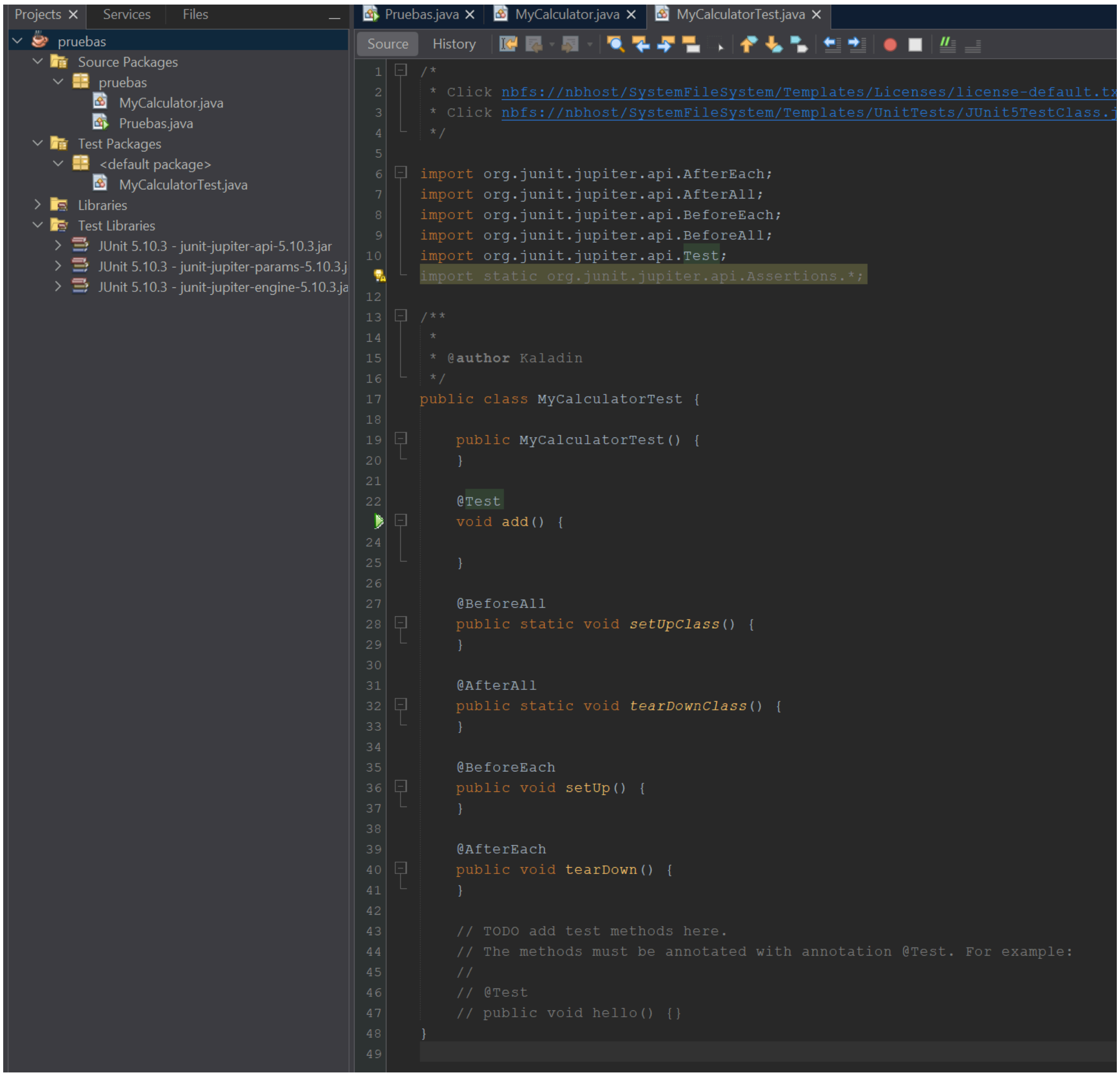
1.



```java
/*
 * Click nbfs://nbhost/SystemFileSyste
 * Click nbfs://nbhost/SystemFileSyste
 */
package pruebas;

/**
 *
 * @author Kaladin
 */
public class MyCalculator {

    public int add(int a, int b) {
        return a+b;
    }

    public int sub(int a, int b) {
        return a-b;
    }

    public int mult(int a, int b) {
        return a*b;
    }

    public int div(int a, int b) {
        return a/b;
    }

}
```

2.



Projects Services Files

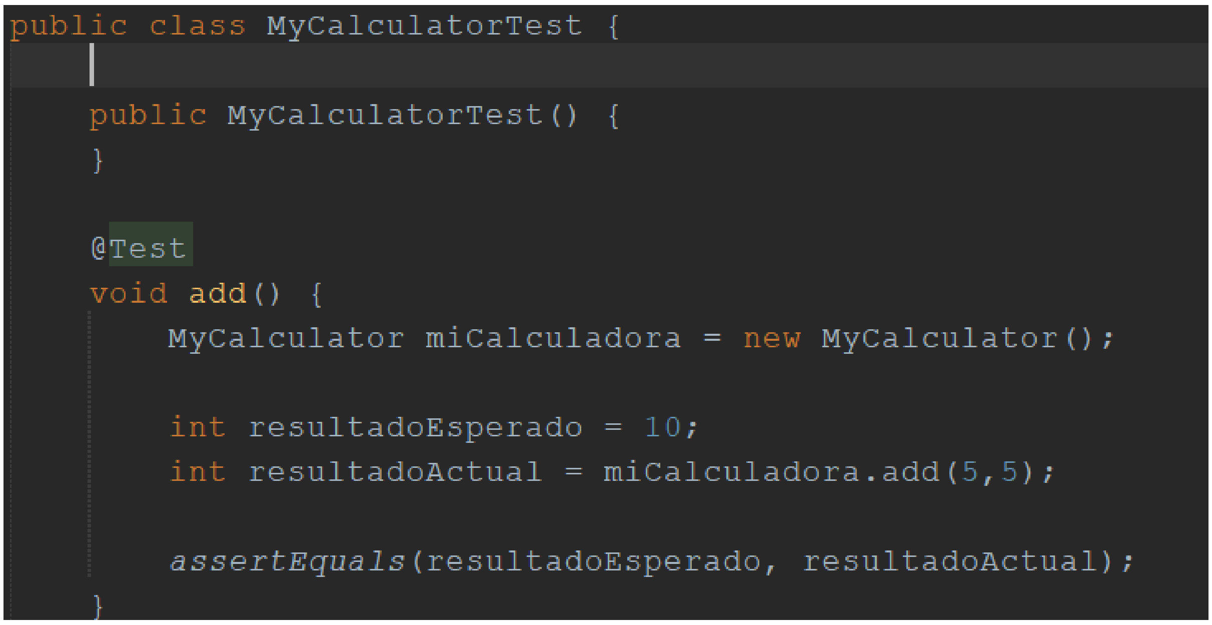
- pruebas
  - Source Packages
    - pruebas
      - MyCalculator.java
      - Pruebas.java
  - Test Packages
    - <default package>
      - MyCalculatorTest.java
  - Libraries
  - Test Libraries
    - JUnit 5.10.3 - junit-jupiter-api-5.10.3.jar
    - JUnit 5.10.3 - junit-jupiter-params-5.10.3.j
    - JUnit 5.10.3 - junit-jupiter-engine-5.10.3.ja

Pruebas.java MyCalculator.java MyCalculatorTest.java

Source History

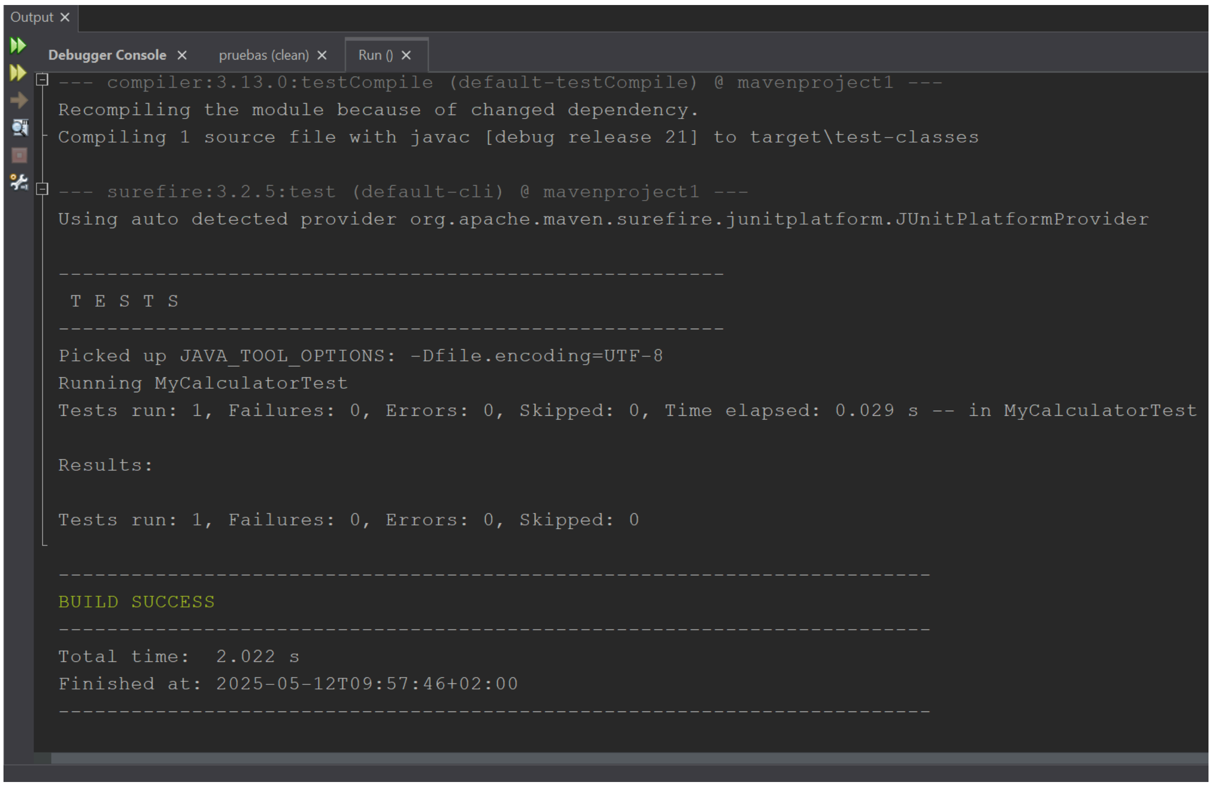
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.tx
 * Click nbfs://nbhost/SystemFileSystem/Templates/UnitTests/JUnit5TestClass.j
 */

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

/**
 *
 * @author Kaladin
 */
public class MyCalculatorTest {

    public MyCalculatorTest() {
    }

    @Test
    void add() {

    }

    @BeforeAll
    public static void setUpClass() {
    }

    @AfterAll
    public static void tearDownClass() {
    }

    @BeforeEach
    public void setUp() {
    }

    @AfterEach
    public void tearDown() {
    }

    // TODO add test methods here.
    // The methods must be annotated with annotation @Test. For example:
    //
    // @Test
    // public void hello() {}
}
```

3.



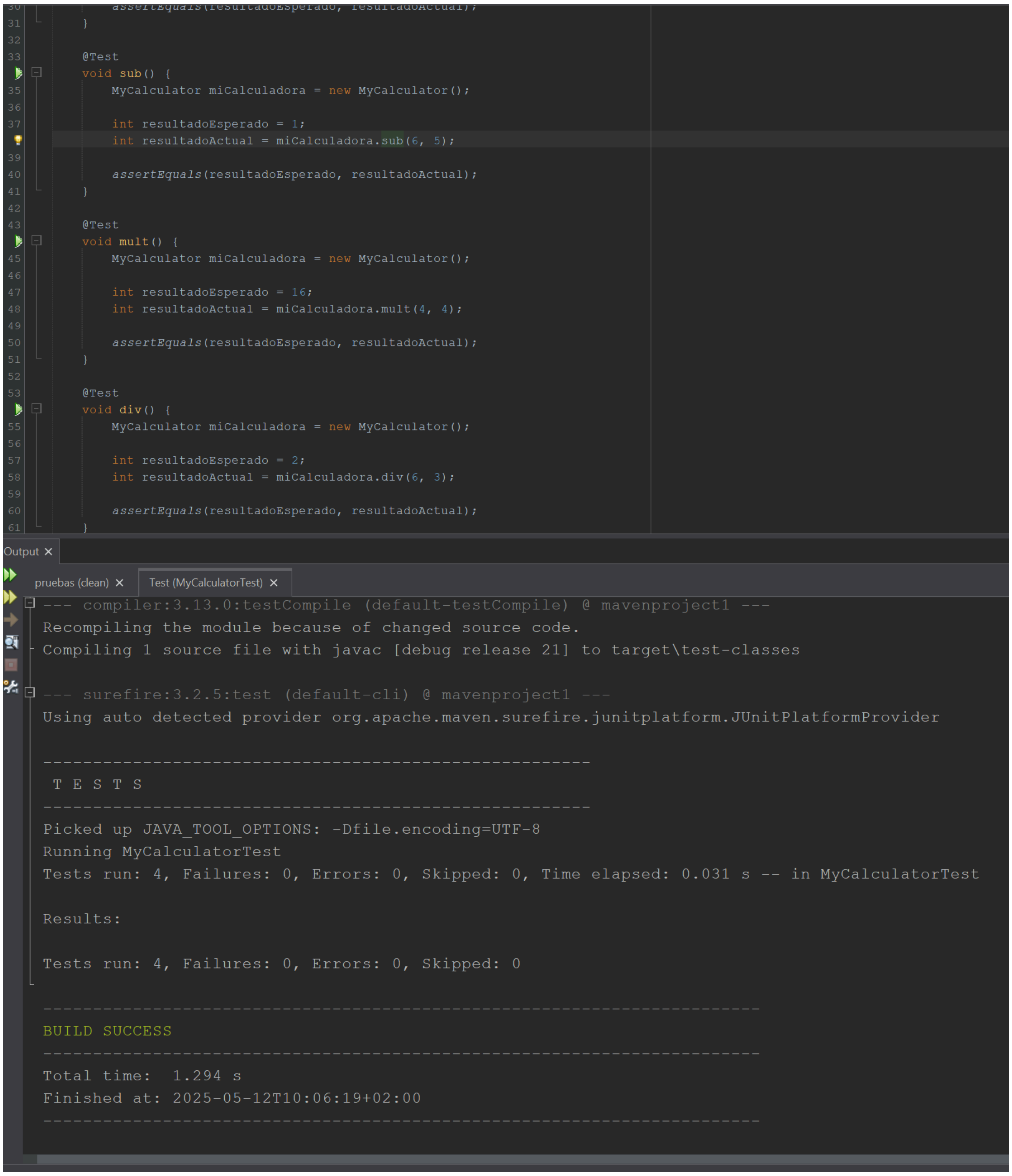
```java
public class MyCalculatorTest {

    public MyCalculatorTest() {
    }

    @Test
    void add() {
        MyCalculator miCalculadora = new MyCalculator();

        int resultadoEsperado = 10;
        int resultadoActual = miCalculadora.add(5,5);

        assertEquals(resultadoEsperado, resultadoActual);
    }
```

4.



```
--- compiler:3.13.0:testCompile (default-testCompile) @ mavenproject1 ---
Recompiling the module because of changed dependency.
Compiling 1 source file with javac [debug release 21] to target\test-classes

--- surefire:3.2.5:test (default-cli) @ mavenproject1 ---
Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Picked up JAVA_TOOL_OPTIONS: -Dfile.encoding=UTF-8
Running MyCalculatorTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.029 s -- in MyCalculatorTest

Results:

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  2.022 s
Finished at: 2025-05-12T09:57:46+02:00
------------------------------------------------------------------------
```

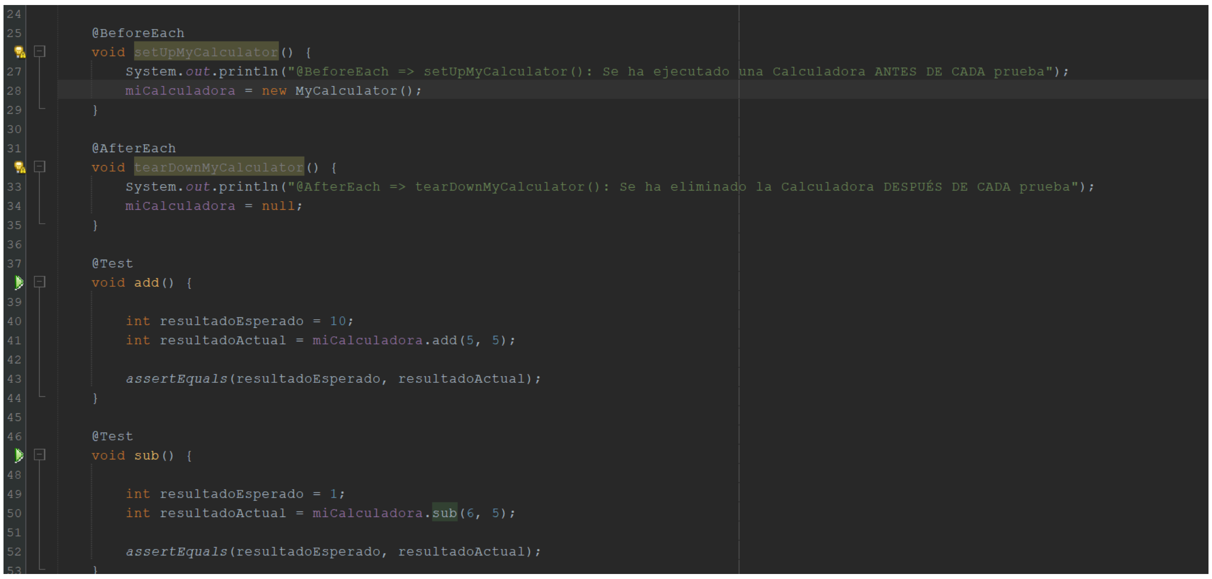
5.



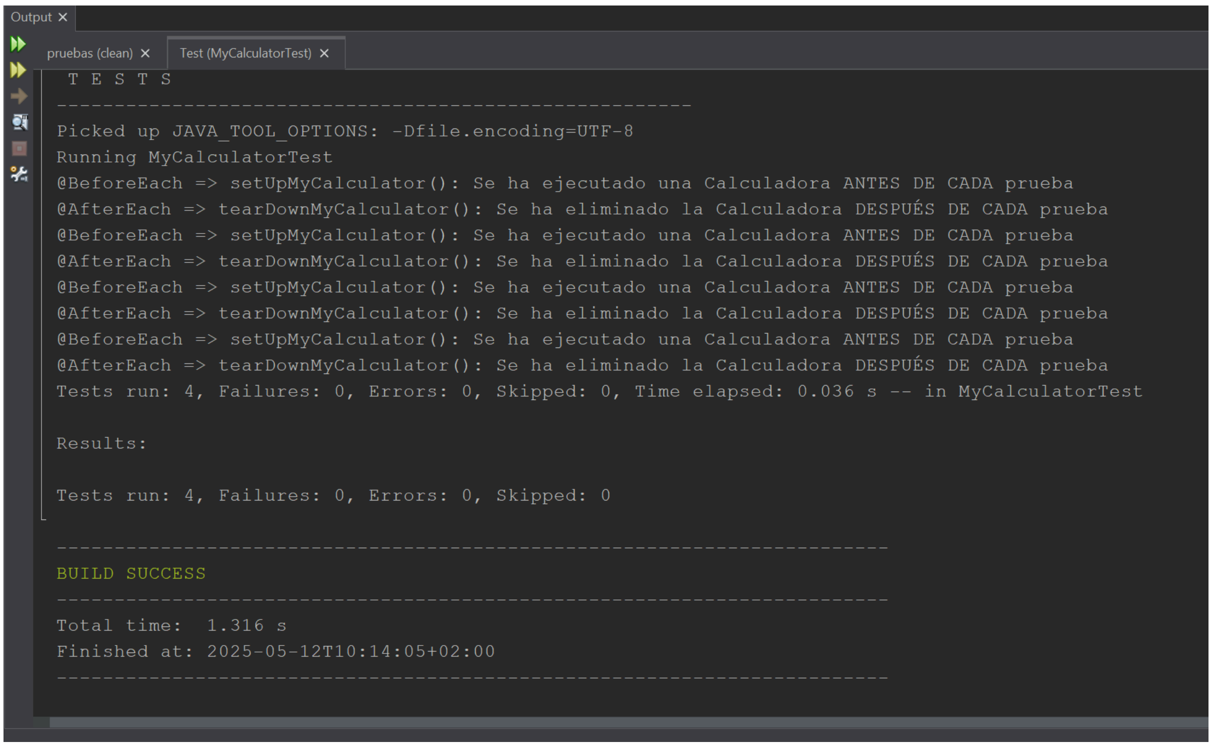
```java
        assertEquals(resultadoEsperado, resultadoActual);
    }

    @Test
    void sub() {
        MyCalculator miCalculadora = new MyCalculator();

        int resultadoEsperado = 1;
        int resultadoActual = miCalculadora.sub(6, 5);

        assertEquals(resultadoEsperado, resultadoActual);
    }

    @Test
    void mult() {
        MyCalculator miCalculadora = new MyCalculator();

        int resultadoEsperado = 16;
        int resultadoActual = miCalculadora.mult(4, 4);

        assertEquals(resultadoEsperado, resultadoActual);
    }

    @Test
    void div() {
        MyCalculator miCalculadora = new MyCalculator();

        int resultadoEsperado = 2;
        int resultadoActual = miCalculadora.div(6, 3);

        assertEquals(resultadoEsperado, resultadoActual);
    }
```

```
--- compiler:3.13.0:testCompile (default-testCompile) @ mavenproject1 ---
Recompiling the module because of changed source code.
Compiling 1 source file with javac [debug release 21] to target\test-classes

--- surefire:3.2.5:test (default-cli) @ mavenproject1 ---
Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Picked up JAVA_TOOL_OPTIONS: -Dfile.encoding=UTF-8
Running MyCalculatorTest
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.031 s -- in MyCalculatorTest

Results:

Tests run: 4, Failures: 0, Errors: 0, Skipped: 0

------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.294 s
Finished at: 2025-05-12T10:06:19+02:00
------------------------------------------------------------------------
```

6.
7.

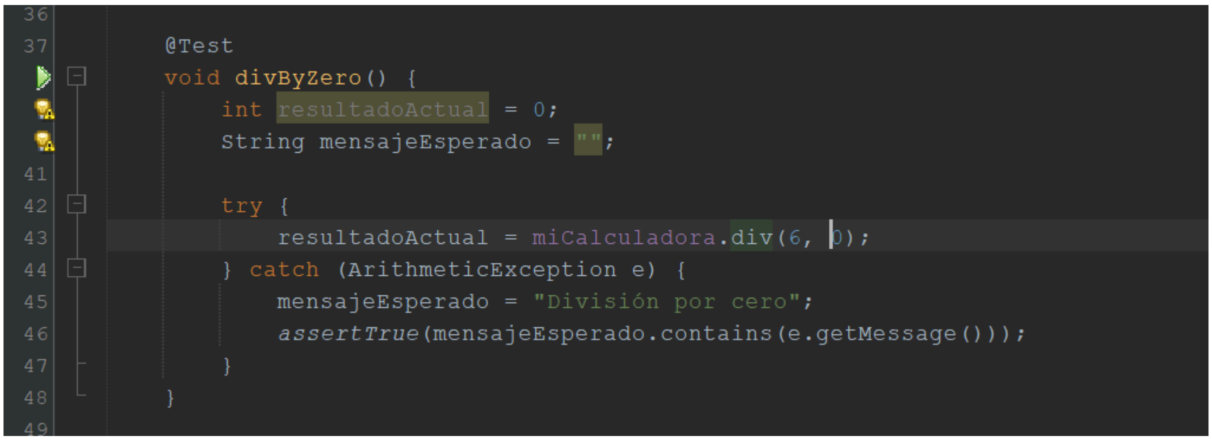


```java

    @BeforeEach
    void setUpMyCalculator() {
        System.out.println("@BeforeEach => setUpMyCalculator(): Se ha ejecutado una Calculadora ANTES DE CADA prueba");
        miCalculadora = new MyCalculator();
    }

    @AfterEach
    void tearDownMyCalculator() {
        System.out.println("@AfterEach => tearDownMyCalculator(): Se ha eliminado la Calculadora DESPUÉS DE CADA prueba");
        miCalculadora = null;
    }

    @Test
    void add() {

        int resultadoEsperado = 10;
        int resultadoActual = miCalculadora.add(5, 5);

        assertEquals(resultadoEsperado, resultadoActual);
    }

    @Test
    void sub() {

        int resultadoEsperado = 1;
        int resultadoActual = miCalculadora.sub(6, 5);

        assertEquals(resultadoEsperado, resultadoActual);
    }
```

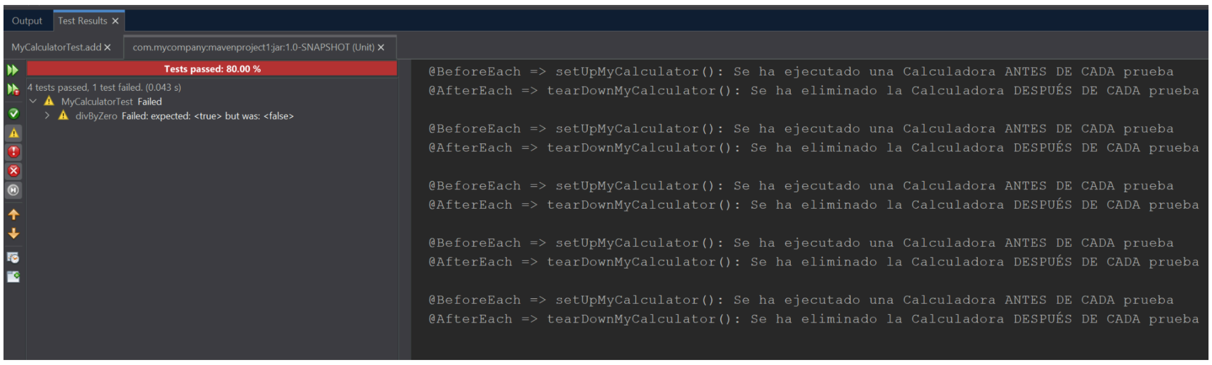


```
 T E S T S
-------------------------------------------------------
Picked up JAVA_TOOL_OPTIONS: -Dfile.encoding=UTF-8
Running MyCalculatorTest
@BeforeEach => setUpMyCalculator(): Se ha ejecutado una Calculadora ANTES DE CADA prueba
@AfterEach => tearDownMyCalculator(): Se ha eliminado la Calculadora DESPUÉS DE CADA prueba
@BeforeEach => setUpMyCalculator(): Se ha ejecutado una Calculadora ANTES DE CADA prueba
@AfterEach => tearDownMyCalculator(): Se ha eliminado la Calculadora DESPUÉS DE CADA prueba
@BeforeEach => setUpMyCalculator(): Se ha ejecutado una Calculadora ANTES DE CADA prueba
@AfterEach => tearDownMyCalculator(): Se ha eliminado la Calculadora DESPUÉS DE CADA prueba
@BeforeEach => setUpMyCalculator(): Se ha ejecutado una Calculadora ANTES DE CADA prueba
@AfterEach => tearDownMyCalculator(): Se ha eliminado la Calculadora DESPUÉS DE CADA prueba
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.036 s -- in MyCalculatorTest

Results:

Tests run: 4, Failures: 0, Errors: 0, Skipped: 0

------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.316 s
Finished at: 2025-05-12T10:14:05+02:00
------------------------------------------------------------------------
```

8.



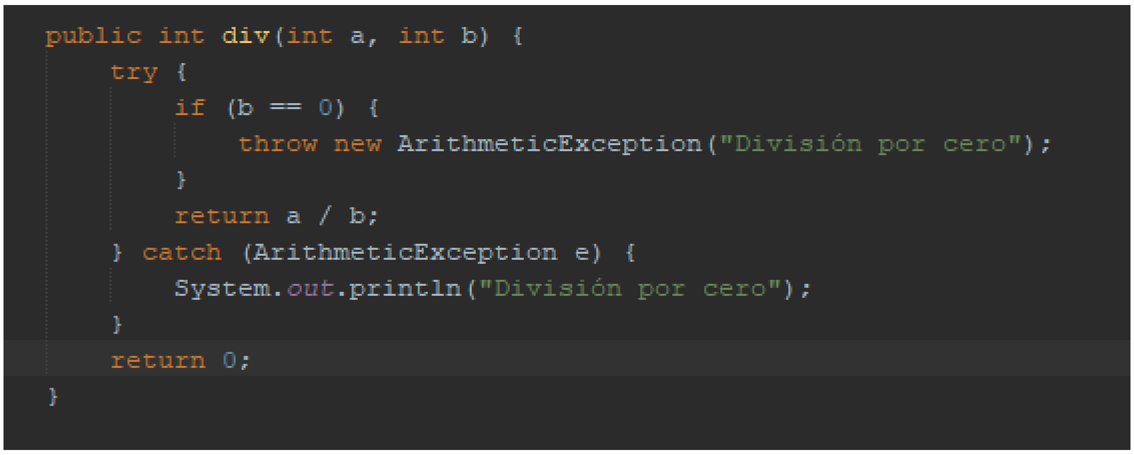
```
    @Test
    void divByZero() {
        int resultadoActual = 0;
        String mensajeEsperado = "";

        try {
            resultadoActual = miCalculadora.div(6, 0);
        } catch (ArithmeticException e) {
            mensajeEsperado = "División por cero";
            assertTrue(mensajeEsperado.contains(e.getMessage()));
        }
    }
```



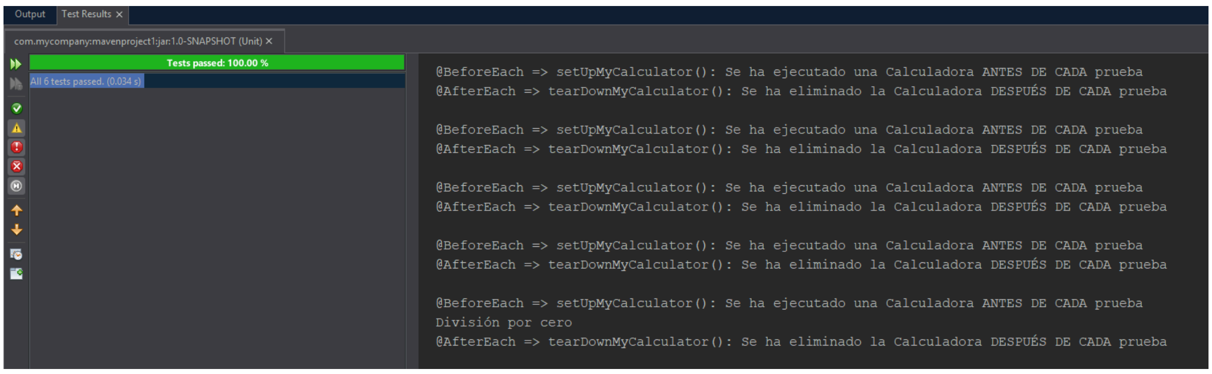
```
Tests passed: 80.00 %
4 tests passed, 1 test failed. (0.043 s)
  MyCalculatorTest Failed
    divByZero Failed: expected: <true> but was: <false>
```

```
@BeforeEach => setUpMyCalculator(): Se ha ejecutado una Calculadora ANTES DE CADA prueba
@AfterEach => tearDownMyCalculator(): Se ha eliminado la Calculadora DESPUÉS DE CADA prueba

@BeforeEach => setUpMyCalculator(): Se ha ejecutado una Calculadora ANTES DE CADA prueba
@AfterEach => tearDownMyCalculator(): Se ha eliminado la Calculadora DESPUÉS DE CADA prueba

@BeforeEach => setUpMyCalculator(): Se ha ejecutado una Calculadora ANTES DE CADA prueba
@AfterEach => tearDownMyCalculator(): Se ha eliminado la Calculadora DESPUÉS DE CADA prueba

@BeforeEach => setUpMyCalculator(): Se ha ejecutado una Calculadora ANTES DE CADA prueba
@AfterEach => tearDownMyCalculator(): Se ha eliminado la Calculadora DESPUÉS DE CADA prueba

@BeforeEach => setUpMyCalculator(): Se ha ejecutado una Calculadora ANTES DE CADA prueba
@AfterEach => tearDownMyCalculator(): Se ha eliminado la Calculadora DESPUÉS DE CADA prueba
```

9.



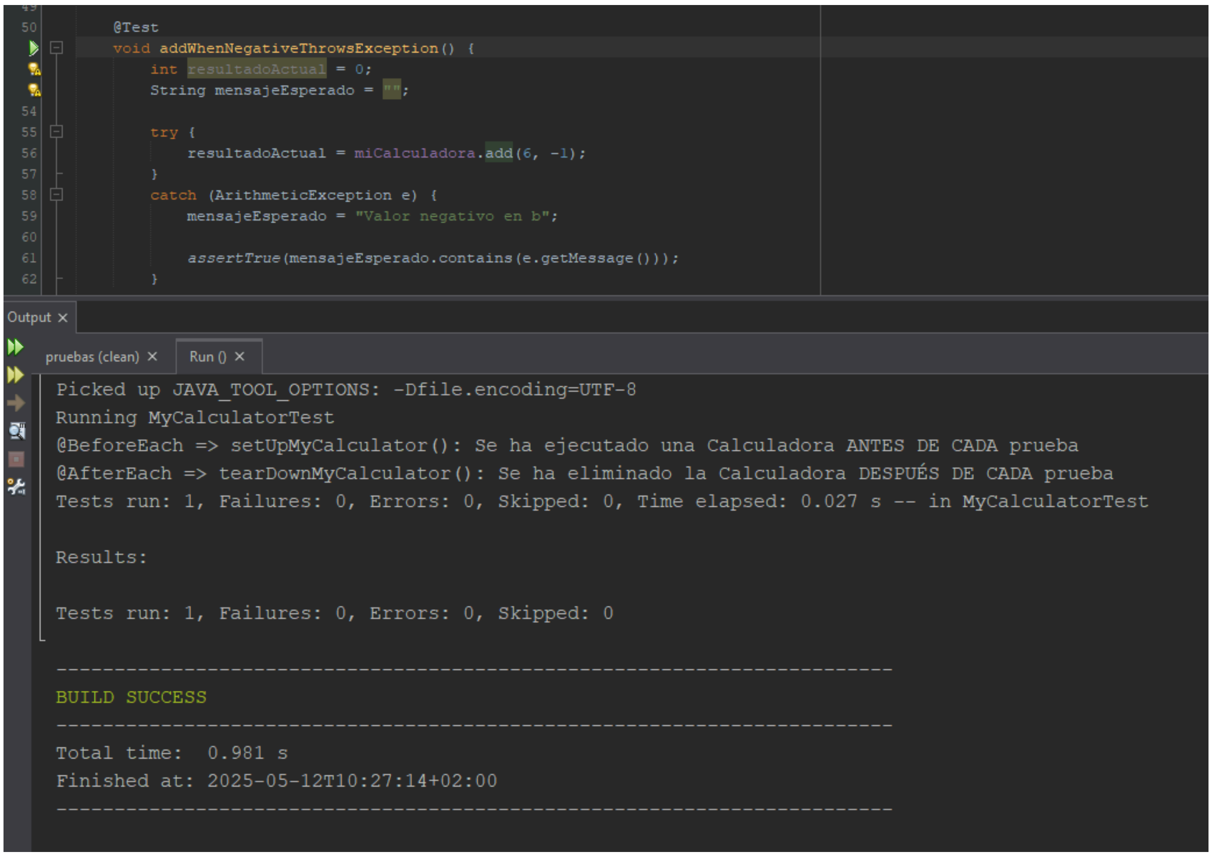
```
public int div(int a, int b) {
    try {
        if (b == 0) {
            throw new ArithmeticException("División por cero");
        }
        return a / b;
    } catch (ArithmeticException e) {
        System.out.println("División por cero");
    }
    return 0;
}
```



```
Tests passed: 100.00 %
All 6 tests passed. (0.034 s)
```

```
@BeforeEach => setUpMyCalculator(): Se ha ejecutado una Calculadora ANTES DE CADA prueba
@AfterEach => tearDownMyCalculator(): Se ha eliminado la Calculadora DESPUÉS DE CADA prueba

@BeforeEach => setUpMyCalculator(): Se ha ejecutado una Calculadora ANTES DE CADA prueba
@AfterEach => tearDownMyCalculator(): Se ha eliminado la Calculadora DESPUÉS DE CADA prueba

@BeforeEach => setUpMyCalculator(): Se ha ejecutado una Calculadora ANTES DE CADA prueba
@AfterEach => tearDownMyCalculator(): Se ha eliminado la Calculadora DESPUÉS DE CADA prueba

@BeforeEach => setUpMyCalculator(): Se ha ejecutado una Calculadora ANTES DE CADA prueba
@AfterEach => tearDownMyCalculator(): Se ha eliminado la Calculadora DESPUÉS DE CADA prueba

@BeforeEach => setUpMyCalculator(): Se ha ejecutado una Calculadora ANTES DE CADA prueba
División por cero
@AfterEach => tearDownMyCalculator(): Se ha eliminado la Calculadora DESPUÉS DE CADA prueba
```

10.



```java
    @Test
    void addWhenNegativeThrowsException() {
        int resultadoActual = 0;
        String mensajeEsperado = "";

        try {
            resultadoActual = miCalculadora.add(6, -1);
        }
        catch (ArithmeticException e) {
            mensajeEsperado = "Valor negativo en b";

            assertTrue(mensajeEsperado.contains(e.getMessage()));
        }
```

```
Picked up JAVA_TOOL_OPTIONS: -Dfile.encoding=UTF-8
Running MyCalculatorTest
@BeforeEach => setUpMyCalculator(): Se ha ejecutado una Calculadora ANTES DE CADA prueba
@AfterEach => tearDownMyCalculator(): Se ha eliminado la Calculadora DESPUÉS DE CADA prueba
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.027 s -- in MyCalculatorTest

Results:

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.981 s
Finished at: 2025-05-12T10:27:14+02:00
------------------------------------------------------------------------
```

11.

(No se como testear con coverage en NetBeans)