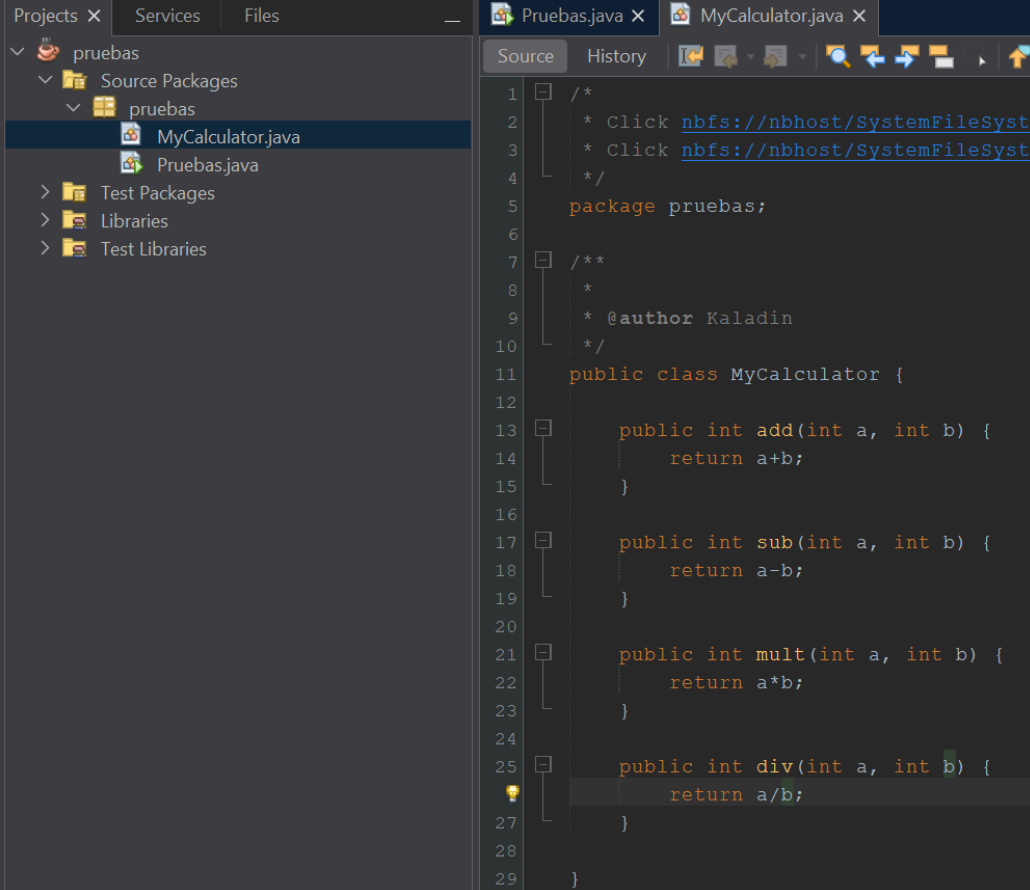


1.



The screenshot shows an IDE with a project named 'pruebas'. The project structure is as follows:

- pruebas
  - Source Packages
    - pruebas
      - MyCalculator.java
      - Pruebas.java
  - Test Packages
  - Libraries
  - Test Libraries

The main editor displays the code for 'MyCalculator.java'.

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem
3   * Click nbfs://nbhost/SystemFileSystem
4   */
5  package pruebas;
6
7  /**
8   *
9   * @author Kaladin
10  */
11 public class MyCalculator {
12
13     public int add(int a, int b) {
14         return a+b;
15     }
16
17     public int sub(int a, int b) {
18         return a-b;
19     }
20
21     public int mult(int a, int b) {
22         return a*b;
23     }
24
25     public int div(int a, int b) {
26         return a/b;
27     }
28
29 }
```

2.

The screenshot shows an IDE with a project named 'pruebas'. The project structure is as follows:

- 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.jar
      - JUnit 5.10.3 - junit-jupiter-engine-5.10.3.jar

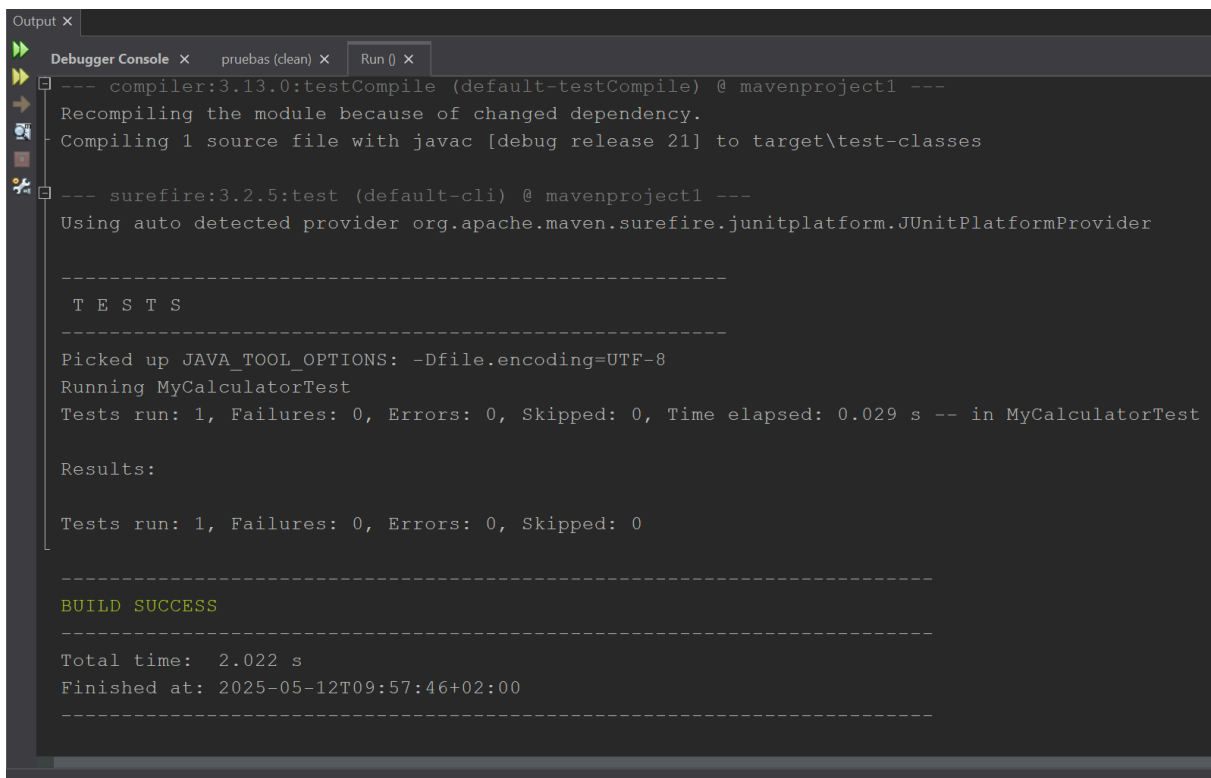
The main editor displays the source code for 'MyCalculatorTest.java'.

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to view the source code.
3   * Click nbfs://nbhost/SystemFileSystem/Templates/UnitTests/JUnit5TestClass.java to create a new test class.
4   */
5
6  import org.junit.jupiter.api.AfterEach;
7  import org.junit.jupiter.api.AfterAll;
8  import org.junit.jupiter.api.BeforeEach;
9  import org.junit.jupiter.api.BeforeAll;
10 import org.junit.jupiter.api.Test;
11 import static org.junit.jupiter.api.Assertions.*;
12
13 /**
14  *
15  * @author Kaladin
16  */
17 public class MyCalculatorTest {
18
19     public MyCalculatorTest() {
20     }
21
22     @Test
23     void add() {
24     }
25
26
27     @BeforeAll
28     public static void setUpClass() {
29     }
30
31
32     @AfterAll
33     public static void tearDownClass() {
34     }
35
36     @BeforeEach
37     public void setUp() {
38     }
39
40     @AfterEach
41     public void tearDown() {
42     }
43
44     // TODO add test methods here.
45     // The methods must be annotated with annotation @Test. For example:
46     //
47     // @Test
48     // public void hello() {}
49 }
```

3.

```
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.



```
Output x  
Debugger Console x pruebas (clean) x Run () x  
--- 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.

```
30         assertEquals(resultadoEsperado, resultadoActual);
31     }
32
33     @Test
34     void sub() {
35         MyCalculadora miCalculadora = new MyCalculadora();
36
37         int resultadoEsperado = 1;
38         int resultadoActual = miCalculadora.sub(6, 5);
39
40         assertEquals(resultadoEsperado, resultadoActual);
41     }
42
43     @Test
44     void mult() {
45         MyCalculadora miCalculadora = new MyCalculadora();
46
47         int resultadoEsperado = 16;
48         int resultadoActual = miCalculadora.mult(4, 4);
49
50         assertEquals(resultadoEsperado, resultadoActual);
51     }
52
53     @Test
54     void div() {
55         MyCalculadora miCalculadora = new MyCalculadora();
56
57         int resultadoEsperado = 2;
58         int resultadoActual = miCalculadora.div(6, 3);
59
60         assertEquals(resultadoEsperado, resultadoActual);
61     }

```

Output X

```
pruebas (clean) X Test (MyCalculadoraTest) X
--- 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 MyCalculadoraTest
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.031 s -- in MyCalculadoraTest

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.

```
24
25     @BeforeEach
26     void setUpMyCalculator() {
27         System.out.println("@BeforeEach => setUpMyCalculator(): Se ha ejecutado una Calculadora ANTES DE CADA prueba");
28         miCalculadora = new MyCalculator();
29     }
30
31     @AfterEach
32     void tearDownMyCalculator() {
33         System.out.println("@AfterEach => tearDownMyCalculator(): Se ha eliminado la Calculadora DESPUÉS DE CADA prueba");
34         miCalculadora = null;
35     }
36
37     @Test
38     void add() {
39
40         int resultadoEsperado = 10;
41         int resultadoActual = miCalculadora.add(5, 5);
42
43         assertEquals(resultadoEsperado, resultadoActual);
44     }
45
46     @Test
47     void sub() {
48
49         int resultadoEsperado = 1;
50         int resultadoActual = miCalculadora.sub(6, 5);
51
52         assertEquals(resultadoEsperado, resultadoActual);
53     }
54 }
```

```
Output X
pruebas (clean) X Test (MyCalculatorTest) X
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.

```
36
37  @Test
38  void divByZero() {
39      int resultadoActual = 0;
40      String mensajeEsperado = "";
41
42      try {
43          resultadoActual = miCalculadora.div(6, 0);
44      } catch (ArithmeticException e) {
45          mensajeEsperado = "División por cero";
46          assertTrue(mensajeEsperado.contains(e.getMessage()));
47      }
48  }
49
```

Output Test Results X

MyCalculatorTest.add X com.mycompany:mavenproject1:jar:1.0-SNAPSHOT (Unit) X

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;
}
```

Output Test Results X

com.mycompany:mavenproject1:jar:1.0-SNAPSHOT (Unit) X

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

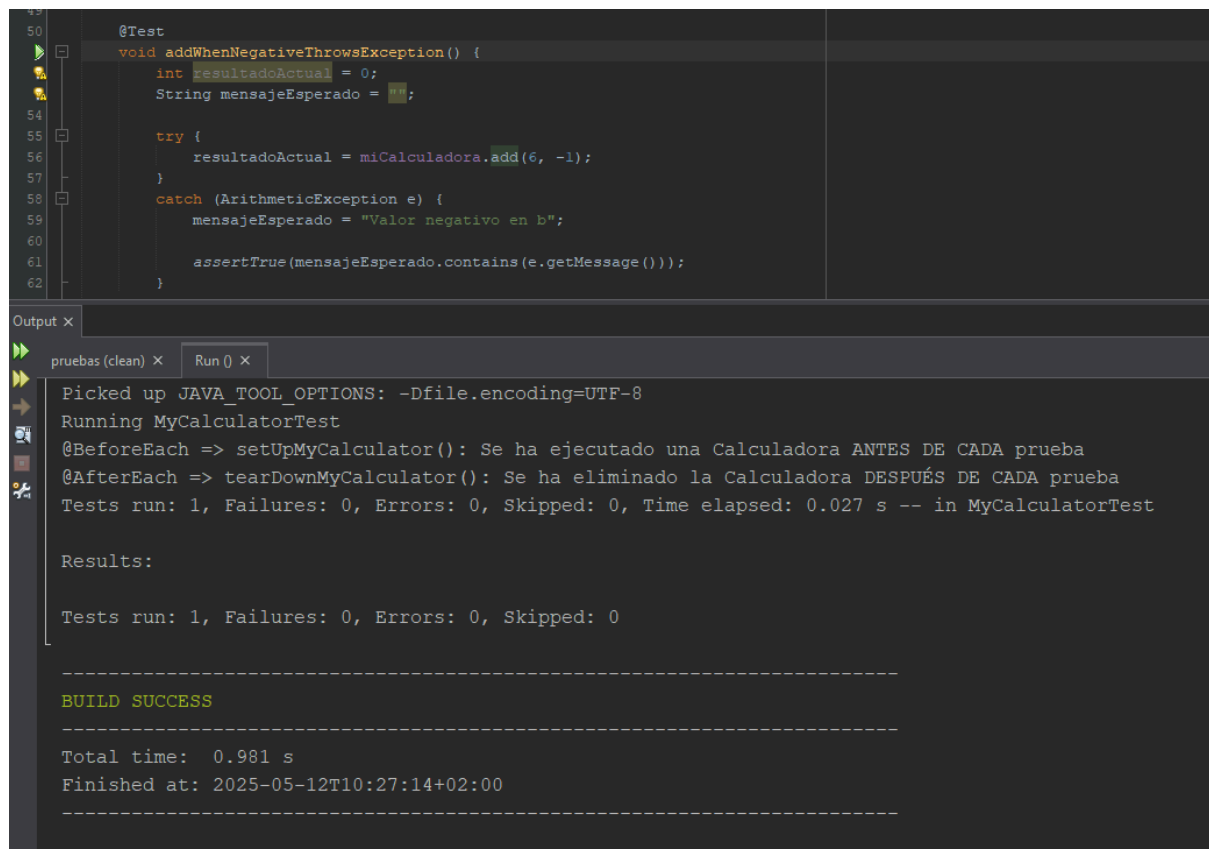
@BeforeEach => setUpMyCalculator(): Se ha ejecutado una Calculadora ANTES DE CADA prueba  
@AfterEach => tearDownMyCalculator(): Se ha eliminado la Calculadora DESPUÉS DE CADA prueba

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

División por cero

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

10.



The screenshot displays a code editor with a Java test method and an output window. The code is as follows:

```
49
50  @Test
51  void addWhenNegativeThrowsException() {
52      int resultadoActual = 0;
53      String mensajeEsperado = "";
54
55      try {
56          resultadoActual = miCalculadora.add(6, -1);
57      }
58      catch (ArithmeticException e) {
59          mensajeEsperado = "Valor negativo en b";
60
61          assertTrue(mensajeEsperado.contains(e.getMessage()));
62      }
63  }
```

The output window shows the following text:

```
pruebas (clean) x Run () x
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)