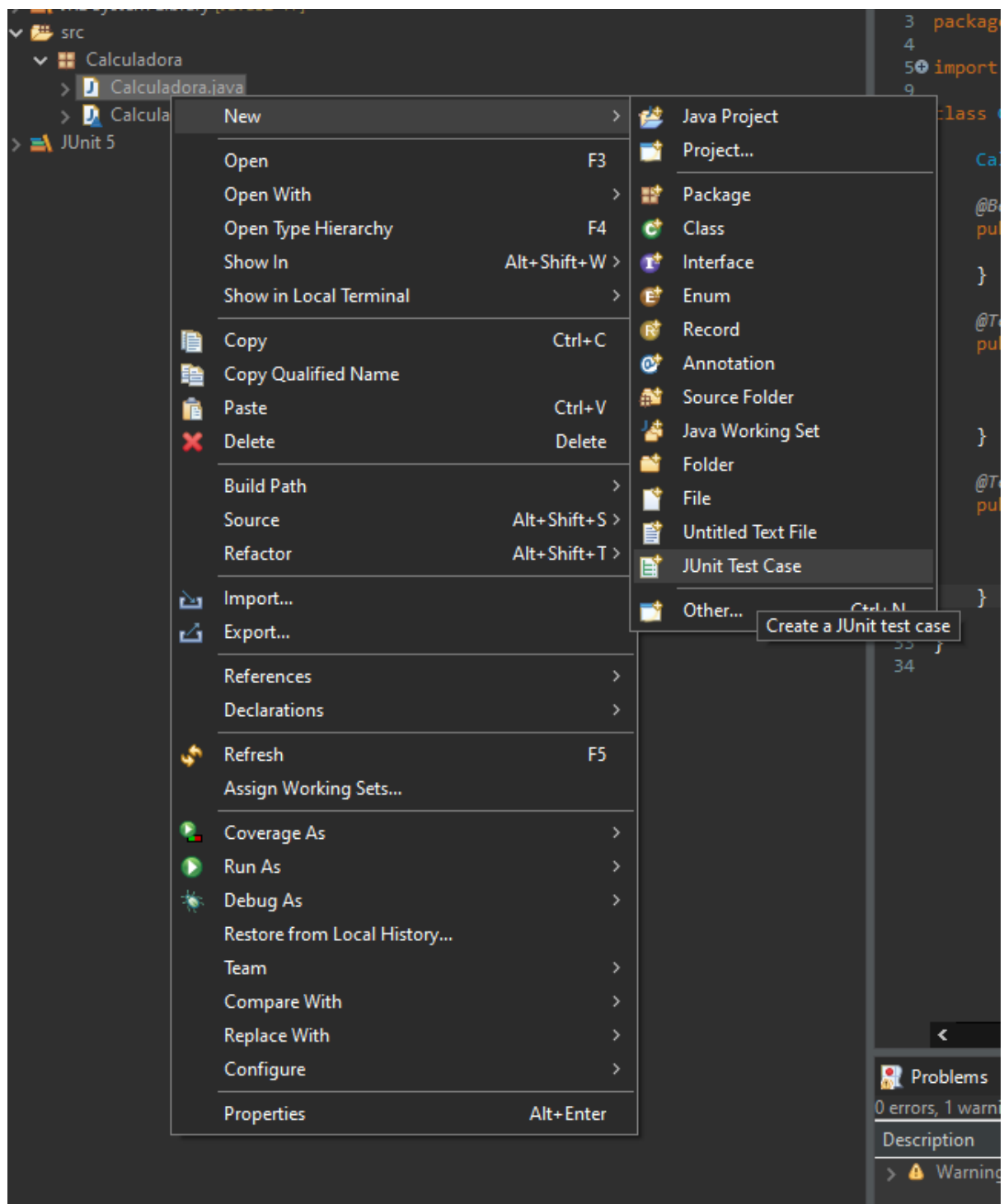
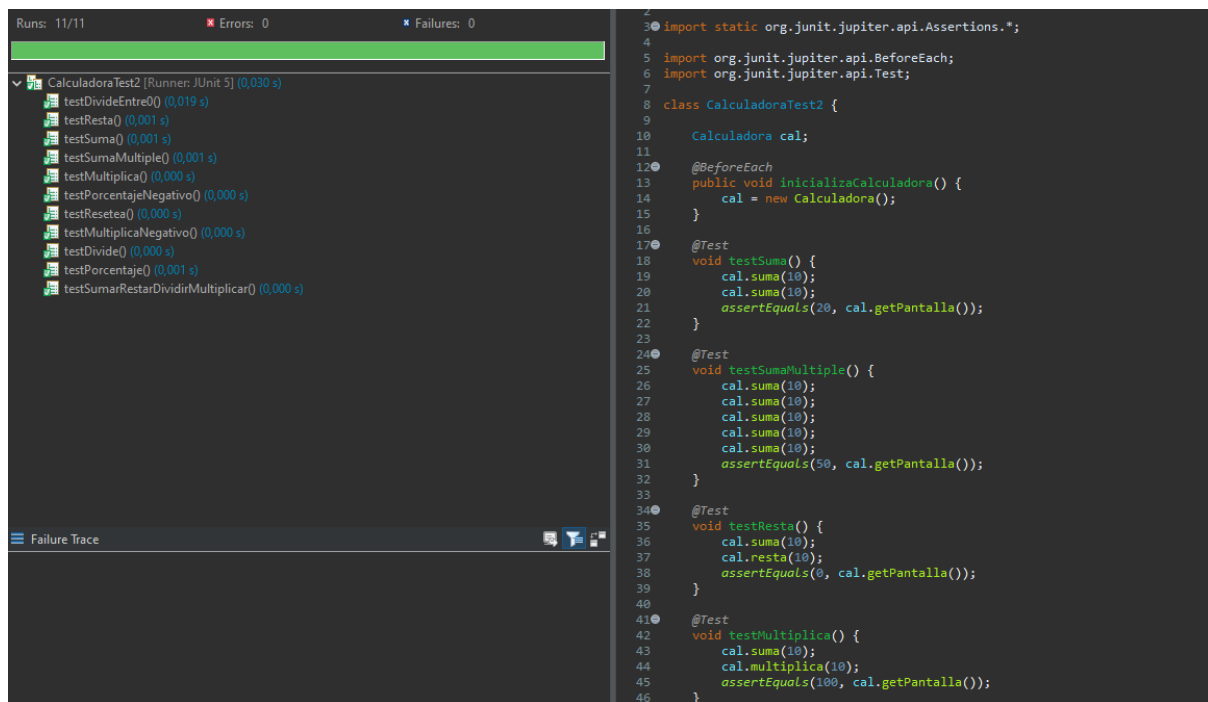


Primero creo una nueva clase de test, CalculadoraTest2.



Luego, creo los métodos de testeo y ejecuto el JUnit Test Case:



The screenshot displays an IDE interface. On the left, a 'Runs' tab shows the execution of 'CalculadoraTest2' with 0 errors and 0 failures. Below this, a list of test methods is shown, each with a green icon indicating success and a duration of 0.000 s. The methods include: testDivideEntre0(), testResta(), testSuma(), testSumaMultiple(), testMultiplica(), testPorcentajeNegativo(), testResetea(), testMultiplicaNegativo(), testDivide(), testPorcentaje(), and testSumarRestarDividirMultiplicar(). At the bottom left, a 'Failure Trace' tab is visible. On the right, the source code for 'CalculadoraTest2' is shown. It includes imports for JUnit assertions and test annotations, and defines the 'Calculadora' class and several test methods: 'inicializaCalculadora()', 'testSuma()', 'testSumaMultiple()', 'testResta()', and 'testMultiplica()'. Each test method calls the corresponding method on the 'Calculadora' object and uses 'assertEquals' to verify the result.

```
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 import org.junit.jupiter.api.BeforeEach;
6 import org.junit.jupiter.api.Test;
7
8 class CalculadoraTest2 {
9
10     Calculadora cal;
11
12     @BeforeEach
13     public void inicializaCalculadora() {
14         cal = new Calculadora();
15     }
16
17     @Test
18     void testSuma() {
19         cal.suma(10);
20         cal.suma(10);
21         assertEquals(20, cal.getPantalla());
22     }
23
24     @Test
25     void testSumaMultiple() {
26         cal.suma(10);
27         cal.suma(10);
28         cal.suma(10);
29         cal.suma(10);
30         cal.suma(10);
31         assertEquals(50, cal.getPantalla());
32     }
33
34     @Test
35     void testResta() {
36         cal.suma(10);
37         cal.resta(10);
38         assertEquals(0, cal.getPantalla());
39     }
40
41     @Test
42     void testMultiplica() {
43         cal.suma(10);
44         cal.multiplica(10);
45         assertEquals(100, cal.getPantalla());
46     }
47 }
```

He creado métodos básicos para probar los operandos principales (sumar, restar, dividir, multiplicar), así como los métodos de resetear y obtener el porcentaje.

Luego, creé otros que combinan los métodos mencionados.

Todos pasan el JUnit sin fallos.