

TAREFA JUNIT. PROXECTO CALCULADORA

1. Modifica o método **testDividir()** de la clase **CalculadoraTest** para que probe $\frac{0}{1}=0$. Que ocorre ao executar novamente o test? E si a

operación a probar é $\frac{1}{0}=0$?

```
58 double numero2 = 0.0;
59 CalculadoraTest instance = new CalculadoraTest();
60 double expectedResult = 0.0;
61 double result = instance.multiplicar(numero1, numero2);
62 assertEquals(actual: result, expected: expectedResult, delta: 0.0);
63 // TODO review the generated test code and remove the default call to fail.
64 fail(message: "The test case is a prototype.");
65 }
66
67 /**
68  * Test of dividir method, of class CalculadoraTest.
69  */
70 @Test
71 public void testDividir() {
72     System.out.println("dividir");
73     double numero1 = 0.0;
74     double numero2 = 1.0;
75     CalculadoraTest instance = new CalculadoraTest();
76     double expectedResult = 0.0;
77     double result = instance.dividir(numero1, numero2);
78     assertEquals(actual: result, expected: expectedResult, delta: 0.0);
79     // TODO review the generated test code and remove the default call to fail.
80     fail(message: "The test case is a prototype.");
81 }
```

Test Results x

CalculadoraTest x

Tests passed: 0.00 %

No test passed, 5 tests failed. (0.005 s)

TestNG tests Failed

- calculadortest.CalculadoraTestNG
- calculadortest.CalculadoraTestNG
- calculadortest.CalculadoraTestNG
- calculadortest.CalculadoraTestNG
- calculadortest.CalculadoraTestNG

dividir
main
multiplicar
restar
sumar
=====

CalculadoraTest
Total tests run: 5, Failures: 5, Skips: 0
=====

The tests failed.

```
57 double numero1 = 0.0;
58 double numero2 = 0.0;
59 CalculadoraTest instance = new CalculadoraTest();
60 double expectedResult = 0.0;
61 double result = instance.multiplicar(numero1, numero2);
62 assertEquals(actual: result, expected: expectedResult, delta: 0.0);
63 // TODO review the generated test code and remove the default call to fail.
64 fail(message: "The test case is a prototype.");
65 }
66
67 /**
68  * Test of dividir method, of class CalculadoraTest.
69  */
70 @Test
71 public void testDividir() {
72     System.out.println("dividir");
73     double numero1 = 1.0;
74     double numero2 = 0.0;
75     CalculadoraTest instance = new CalculadoraTest();
76     double expectedResult = 0.0;
77     double result = instance.dividir(numero1, numero2);
78     assertEquals(actual: result, expected: expectedResult, delta: 0.0);
79     // TODO review the generated test code and remove the default call to fail.
80     fail(message: "The test case is a prototype.");
81 }
```

Test Results x

CalculadoraTest x

Tests passed: 0.00 %

No test passed, 5 tests failed. (0.006 s)

TestNG tests Failed

- calculadortest.CalculadoraTestNG
- calculadortest.CalculadoraTestNG
- calculadortest.CalculadoraTestNG
- calculadortest.CalculadoraTestNG
- calculadortest.CalculadoraTestNG

dividir
main
multiplicar
restar
sumar
=====

CalculadoraTest
Total tests run: 5, Failures: 5, Skips: 0
=====

The tests failed.

2. Modifica o código da clase **CalculadoraTest** incluíndo un método **@BeforeMethod** e realizando as modificacións necesarias nos métodos para utilizar unha instancia da clase **Calculadora** en lugar de creala en cada método.

Executa de novo o test e analiza a mellora realizada neste novo código.

```
import org.junit.jupiter.api.*;
import static org.junit.jupiter.api.Assertions.*;

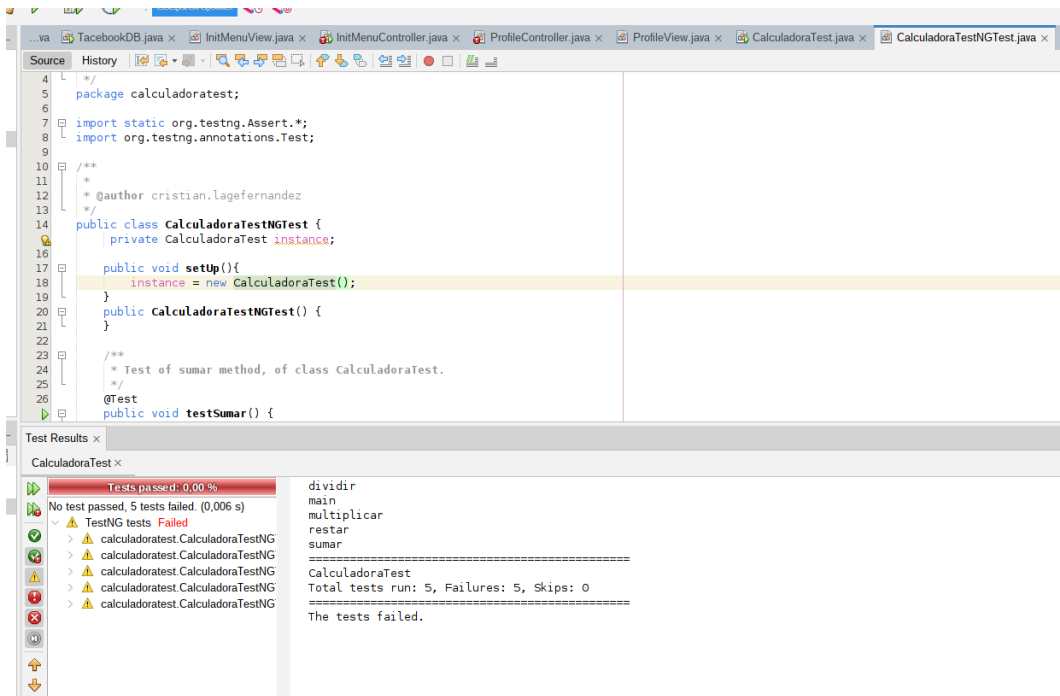
public class CalculadoraTest {

    private Calculadora instance;

    @BeforeMethod
    public void setUp(){
        instance = new Calculadora();
    }

    /**
     * Test of sumar method, of class Calculadora.
     */
    @Test
    public void testSumar() {
        System.out.println("sumar");
        double numero1 = 5;
        double numero2 = 4;
        double expectedResult = 9;
        double result = instance.sumar(numero1, numero2);
        assertEquals(expResult, result, 0.0);
    }
}

.....Continúa...
```



3. Engade un método á clase **Calculadora** que comprobe se un número é par (recibe un enteiro e devolve un valor *booleano*). Agrega un método **@Test** para facer as proba pertinentes utilizando o método *assert* máis adecuado.