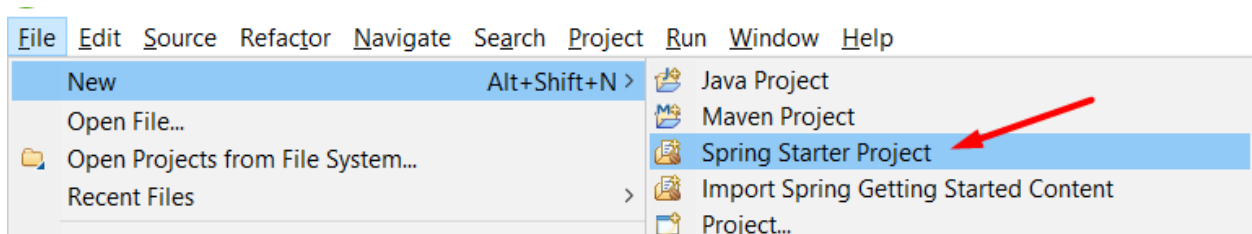


# PROYECTO “JUnit-demo”

## Paso 1: Crear el Proyecto en STS

Abrir Spring Tool Suite (STS).

Ir a **File → New → Spring Starter Project**



Completar los datos del proyecto:

- **Name:** junit-demo
- **Type:** Maven
- **Java Version:** 17
- **Group:** com.example
- **Artifact:** junit-demo
- **Package:** com.example.junitdemo
- **Packaging:** Jar

A screenshot of the 'Spring Starter Project' wizard dialog box. The 'Service URL' is set to 'https://start.spring.io'. The 'Name' field contains 'junit-demo-1'. The 'Use default location' checkbox is checked. The 'Location' field shows 'C:\Users\JR\Documents\workspace\junit-demo-1'. The 'Type' is 'Maven', 'Packaging' is 'Jar', 'Java Version' is '17', and 'Language' is 'Java'. The 'Group' is 'com.example', 'Artifact' is 'junit-demo-1', 'Version' is '0.0.1-SNAPSHOT', 'Description' is 'Demo project for Spring Boot', and 'Package' is 'com.example.junitdemo'.

Hacer clic en **Finish** para generar el proyecto.

## Paso 2: Crear la Clase Calculadora

Creamos la clase Calculadora



```
package com.example.junitdemo.service;

public class Calculadora {

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

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

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

    public double dividir(int a, int b) {
        if (b == 0) {
            throw new IllegalArgumentException("No se puede dividir por cero");
        }
        return (double) a / b;
    }
}
```

Explicación:

- Métodos sumar(), restar(), multiplicar() y dividir().
- dividir() lanza una excepción si `b == 0` para evitar división por cero.

## Paso 3: Escribir Pruebas con JUnit 5

En la sección de la carpeta test en el directorio: **src/test/java/com/example/junitdemo/service/**  
Creamos la lógica del test unitario en el service `CalculadoraTest`



```
package com.example.junitdemo.service;

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

class CalculadoraTest {

    private final Calculadora calculadora = new Calculadora();

    @Test
    void testSuma() {
        int resultado = calculadora.sumar(3, 2);
        assertEquals(5, resultado);
    }

    @Test
    void testResta() {
        int resultado = calculadora.restar(10, 4);
        assertEquals(6, resultado);
    }

    @Test
    void testMultiplicacion() {
        int resultado = calculadora.multiplicar(6, 3);
        assertEquals(18, resultado);
    }

    @Test
    void testDivision() {
        double resultado = calculadora.dividir(9, 3);
        assertEquals(3.0, resultado);
    }

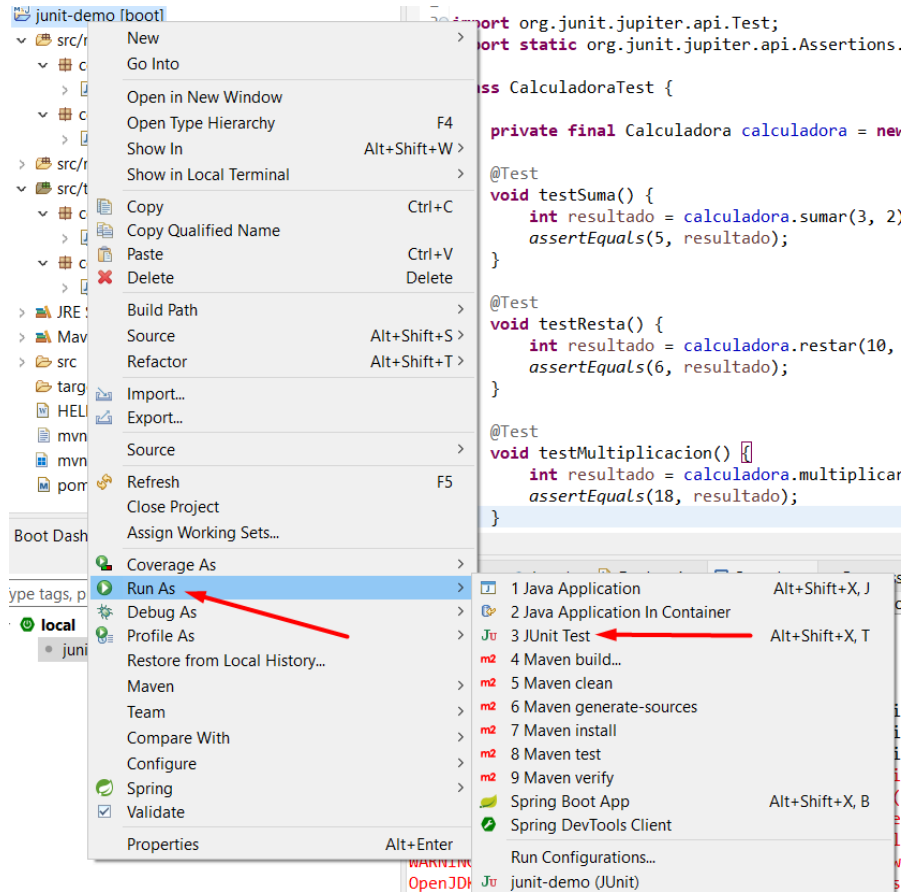
    @Test
    void testDivisionPorCero() {
        Exception exception = assertThrows(IllegalArgumentException.class, () -> {
            calculadora.dividir(5, 0);
        });
        assertEquals("No se puede dividir por cero", exception.getMessage());
    }
}
```

Explicación:

- Cada **@Test** prueba un método de Calculadora.
- **assertEquals(5, resultado)** verifica que la suma sea correcta.
- **assertThrows()** prueba que la excepción se lanza en dividir(5, 0).

## Paso 4: Ejecutar las Pruebas

Ir a CalculadoraTest.java y hacer clic derecho → Run As → JUnit Test.



Deberías ver que todas las pruebas pasan en verde .

