1.Captura de las funciones comentadas

```java
/**
 * Clase que representa una calculadora con operaciones básicas.
 *
 * @author Adrian
 */
public class Calculadora {

    private float lastResult;
    private String lastOp;

    /**
     * Coge el último resultado de la calculadora.
     *
     * @return El último resultado.
     */
    public float getLastResult() {
        return this.lastResult;
    }

    /**
     * Coge la última operación realizada en la calculadora.
     *
     * @return La última operación.
     */
    public String getLastOp() {
        return this.lastOp;
    }

    /**
     * Hace una suma y actualiza el último resultado y operación.
     *
     * @param op1 Primer número.
     * @param op2 Segundo número.
     * @return Resultado de la suma.
     */
    public float suma(float op1, float op2) {
        float result = op1 + op2;
        this.lastResult = result;
        this.lastOp = "Suma";
        return result;
    }

    /**
     * Hace una resta y actualiza el último resultado y operación.
     *
     * @param op1 Primer número.
     * @param op2 Segundo número.
     * @return Resultado de la resta.
     */
    public float resta(float op1, float op2) {
        float result = op1 - op2;
        this.lastResult = result;
        this.lastOp = "Resta";
        return result;
    }

    /**
     * Hace una multiplicación y actualiza el último resultado y operación.
     *
     * @param op1 Primer número.
     * @param op2 Segundo número.
     * @return Resultado de la multiplicación.
     */
    public float multiplica(float op1, float op2) {
        float result = op1 * op2;
        this.lastResult = result;
        this.lastOp = "Multiplica";
        return result;
    }

    /**
     * Hace una división y actualiza el último resultado y operación.
     *
     * @param op1 Dividendo.
     * @param op2 Divisor.
     * @return Resultado de la división.
     */
    public float divideix(float op1, float op2) {
        float result = op1 / op2;
        this.lastResult = result;
        this.lastOp = "Divideix";
        return result;
    }

    /**
     * Comprueba si un número es mayor que otro.
     *
     * @param op1 Primer número.
     * @param op2 Segundo número.
     * @return true si op1 es mayor que op2, false si op2 es mayor que op1.
     */
    public boolean majorQue(float op1, float op2) {
        if (op1 > op2) {
            return true;
        }
        return false;
    }

    /**
     * Restablece el último resultado a cero y la última operación a Ninguna.
     */
    public void restablecer() {
        this.lastResult = 0;
        this.lastOp = "Ninguna";
    }
```

2.Captura de la clase de proves

```java
public class CalculadoraTest {
    private Calculadora calculadora;
    @BeforeEach
    public void setup(){
        this.calculadora=new Calculadora();
    }
    @Test
    public void Testsuma(){
        assertEquals(expected: 5, actual: calculadora.suma(op1: 3, op2: 2));
    }
    @Test
    public void Testresta(){
        assertEquals(expected: 5, actual: calculadora.resta(op1: 10, op2: 5));

    }
    @Test
    public void Testmult(){
        assertEquals(expected: 6, actual: calculadora.multiplica(op1: 2, op2: 3));

    }
    @Test
    public void Testdividir(){
        assertEquals(expected: 3, actual: calculadora.divideix(op1: 6, op2: 2));

    }
    @Test
    public void Testmajorque(){
        assertEquals(expected: false, actual: calculadora.majorQue(op1: 1, op2: 2));

    }
    @AfterEach
    public void tearDown() {
        calculadora.restablecer();
    }
```

3.Captura de la ejecución del test de JUnit

trabajoeddJUnit (test) × | Test (CalculadoraTest) ×

```
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.063 s -- in CalculadoraTest

Results:

Tests run: 5, Failures: 0, Errors: 0, Skipped: 0

------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  3.532 s
Finished at: 2024-03-08T10:52:41+01:00
------------------------------------------------------------------------
```