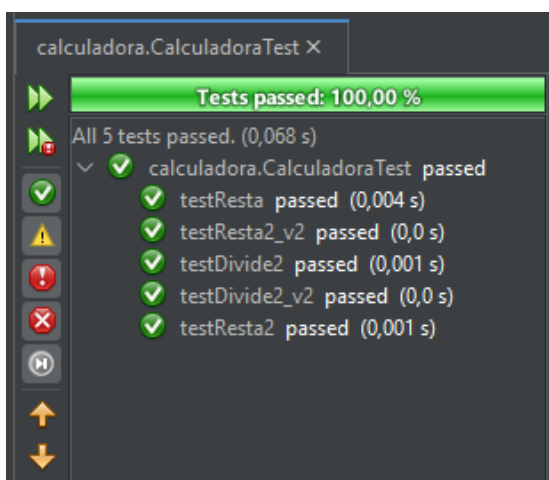


Entrega proactividad UD5 - Entornos de Desarrollo

Ejercicio 1:

```
public int resta() {  
    int result;  
    if (resta2()) {  
        result = num1 - num2;  
    } else {  
        result = num2 - num1;  
    }  
    return result;  
}  
  
public boolean resta2() {  
    if (num1 > num2) {  
        return true;  
    } else {  
        return false;  
    }  
}  
  
public String divide2() {  
    if (num2 == 0) {  
        return null;  
    }  
    int result = num1 / num2;  
    return String.valueOf(result);  
}
```

Código Calculadora.java



Resultados tests - Calculadora.

```

@Test
public void testResta() {
    Calculadora calculadoraTest = new Calculadora(10, 2);
    int expectedResult = 8;
    int result = calculadoraTest.resta();
    assertEquals(expectedResult, result);
}

@Test
public void testResta2() {
    Calculadora calculadoraTest = new Calculadora(10, 2);
    boolean result = calculadoraTest.resta2();

    assertTrue(result);
}

@Test
public void testResta2_v2() {
    Calculadora calculadoraTest = new Calculadora(2, 10);
    boolean result = calculadoraTest.resta2();

    assertFalse(result);
}

@Test
public void testDivide2() {
    Calculadora calculadoraTest = new Calculadora(10, 1);
    String result = calculadoraTest.divide2();

    assertEquals(result, "10");
}

@Test
public void testDivide2_v2() {
    Calculadora calculadoraTest = new Calculadora(10, 0);
    String result = calculadoraTest.divide2();

    assertNull(result);
}

```

Ejercicio 2:

```
public class Fecha {

    SimpleDateFormat formato;
    Date hoy;

    public Fecha() {
        hoy = new Date();
    }

    public static void main(String[] args) {
        Fecha unodenero = new Fecha();

        System.out.println(unodenero.devuelveFecha(1));
    }
    public String devuelveFecha(int tipo) {
        String cad = "";
        switch (tipo) {
            case 1: {
                formato = new SimpleDateFormat("yyyy/MM");
                cad = formato.format(hoy);
                break;
            }
            case 2: {
                formato = new SimpleDateFormat("MM/yyyy");
                cad = formato.format(hoy);
                break;
            }
            case 3: {
                formato = new SimpleDateFormat("MM/yy");
                cad = formato.format(hoy);
                break;
            }
            default: {
                cad = "ERROR";
            }
        }
        System.out.println(cad);
    }
    return cad;
}
```

Código Fecha.java

```

@Test
public void testDevuelveFecha_error() {
    int tipo = 0;
    Fecha instance = new Fecha();
    String expectedResult = "ERROR";
    String result = instance.devuelveFecha(tipo);
    assertEquals(expectedResult, result);
}

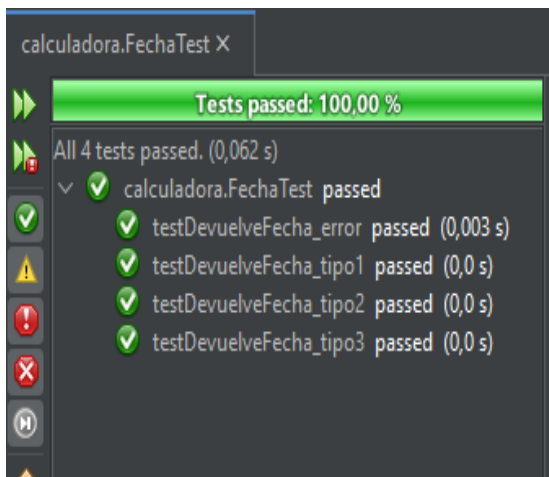
@Test
public void testDevuelveFecha_tipo1() {
    int tipo = 1;
    Fecha instance = new Fecha();
    String expectedResult = "2024/02";
    String result = instance.devuelveFecha(tipo);
    assertEquals(expectedResult, result);
}

@Test
public void testDevuelveFecha_tipo2() {
    int tipo = 2;
    Fecha instance = new Fecha();
    String expectedResult = "02/2024";
    String result = instance.devuelveFecha(tipo);
    assertEquals(expectedResult, result);
}

@Test
public void testDevuelveFecha_tipo3() {
    int tipo = 3;
    Fecha instance = new Fecha();
    String expectedResult = "02/24";
    String result = instance.devuelveFecha(tipo);
    assertEquals(expectedResult, result);
}

```

Código de FechaTest.java



Resultados tests - Fecha.

Ejercicio 3:

```
public class Factorial {  
  
    int num;  
  
    public Factorial(int num) {  
        this.num = num;  
    }  
  
    public long calculo() {  
        if(num < 0) {  
            throw new IllegalArgumentException(  
                "Número " + num + " no puede ser menor que 0."  
            );  
        }  
        long fact = 1;  
  
        for(int i = 2; i <= num; i++) {  
            fact *= i;  
        }  
  
        return fact;  
    }  
}
```

Código Factorial.java

```

@Test
public void testCalculo() {
    Factorial instance = new Factorial(3);
    long expResult = 6;
    long result = instance.calculo();
    assertEquals(expResult, result);
}

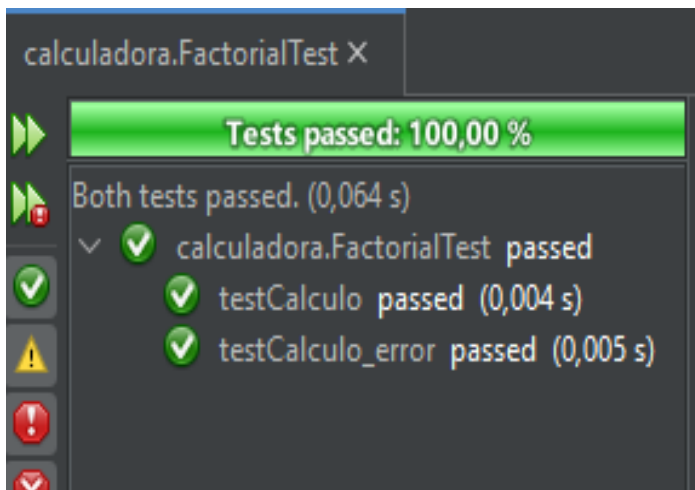
@Test
public void testCalculo_error() {
    Factorial instance = new Factorial(-2);

    Exception exception = assertThrows(
        IllegalArgumentException.class, () -> {
            instance.calculo();
        });
    String expectedMessage = "Número -2 no puede ser menor que 0.";
    String resultMessage = exception.getMessage();

    assertTrue(resultMessage.contains(expectedMessage));
}

```

Código FactorialTest.java



Resultados tests – Factorial.