

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package ed.complejidad;
```

```
import org.junit.AfterClass;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;
```

```
/**
 *
 * @author blackzaforo
 */
```

```
public class ComplejidadTest {
```

```
    private static float points;
    private static float totalPoints;
    private float testValue;
    private Complejidad rec;
```

```
    public ComplejidadTest() {
        rec = new Complejidad();
    }
```

```
    @BeforeClass
    public static void setUpClass() {
        totalPoints = points = 0;
    }
```

```
    @AfterClass
    public static void tearDownClass() {
        System.out.println("=====");
        System.out.println("Calificacion automatica: " + points * 10 /totalPoints);
        System.out.println("=====");
    }
```

```
    /**
     * Test of tPascalRec method, of class Complejidad.
     */
```

```
    @Test
    public void testTPascalRec() {
        System.out.println("tPascalRec");
        testValue = 2;
        totalPoints += testValue;
        IComplejidad instance = new Complejidad();
        assertEquals(10, instance.tPascalRec(5, 2));
        points += testValue / 2;
        assertEquals(3, instance.tPascalRec(3, 2));
        points += testValue / 2;
    }
```

```
    /**
     * Test of tPascalIt method, of class Complejidad.
```

```

*/
@Test
public void testTPascallt() {
    System.out.println("tPascallt");
    testValue = 2;
    totalPoints += testValue;
    IComplejidad instance = new Complejidad();
    assertEquals(10, instance.tPascallt(5, 2));
    points += testValue / 2;
    assertEquals(3, instance.tPascallt(3, 2));
    points += testValue / 2;
}

/**
 * Test of fibonacciRec method, of class Complejidad.
 */
@Test
public void testFibonacciRec() {
    System.out.println("fibonacciRec");
    testValue = 2;
    totalPoints += testValue;
    IComplejidad instance = new Complejidad();
    assertEquals(8, instance.fibonacciRec(6));
    points += testValue / 2;
    assertEquals(21, instance.fibonacciRec(8));
    points += testValue / 2;
}

/**
 * Test of fibonaccilt method, of class Complejidad.
 */
@Test
public void testFibonaccilt() {
    System.out.println("fibonaccilt");
    testValue = 2;
    totalPoints += testValue;
    IComplejidad instance = new Complejidad();
    assertEquals(21, instance.fibonaccilt(8));
    points += testValue / 2;
    assertEquals(144, instance.fibonaccilt(12));
    points += testValue / 2;
}

@Test(expected=IndexOutOfBoundsException.class)
public void testFibltInvalido(){
    System.out.println("Cálculo fibonacci valor invalido");
    testValue = 0.5f;
    totalPoints += testValue;
    try{
        rec.fibonaccilt(-5);
    }catch(IndexOutOfBoundsException e){
        points += testValue;
        throw e;
    }
}

@Test(expected=IndexOutOfBoundsException.class)

```

```

public void testFibRecInvalido(){
    System.out.println("Cálculo fibonacci valor invalido2");
    testValue = 0.5f;
    totalPoints += testValue;
    try{
        rec.fibonacciRec(-10);
    }catch(IndexOutOfBoundsException e){
        points += testValue;
        throw e;
    }
}

@Test(expected=IndexOutOfBoundsException.class)
public void testPascalInvalido(){
    System.out.println("Cálculo pascal valor invalido");
    testValue = 0.5f;
    totalPoints += testValue;
    try{
        rec.tPascallt(-5,1);
    }catch(IndexOutOfBoundsException e){
        points += testValue;
        throw e;
    }
}

@Test(expected=IndexOutOfBoundsException.class)
public void testPascalRecInvalido(){
    System.out.println("Cálculo pascal valor invalido2");
    testValue = 0.5f;
    totalPoints += testValue;
    try{
        rec.tPascalRec(-5,1);
    }catch(IndexOutOfBoundsException e){
        points += testValue;
        throw e;
    }
}
}

```