

Entrega debidamente documentada las pruebas del EclEmma para dos códigos. Has de adjuntar tanto el código como el pdf debidamente documentado (pantallazos y títulos claros de los pasos que sigues). Has de utilizar el junit

EclEmma

Recuperación

Roberto Carlos Flores Gómez

1. Bisiesto.

Comenzamos creando el archivo Bisiesto.java con el código que se nos ha dado.

```
static void mostrarSiBisiesto(int x) {  
    if ( x % 4 == 0 ) {  
        if ( x % 100 == 0 ) {  
            if ( x % 400 == 0 ) {  
                System.out.println("Es bisiesto");  
            } else {  
                System.out.println("No es bisiesto");  
            }  
        } else {  
            System.out.println("Es bisiesto");  
        }  
    } else {  
        System.out.println("No es bisiesto");  
    }  
}
```

Ahora debemos crear un test para probarlo, el cual debe abarcar todas las posibilidades.

```
public class TestBisiesto {

    @BeforeClass
    public static void setUpBeforeClass() throws Exception {
    }

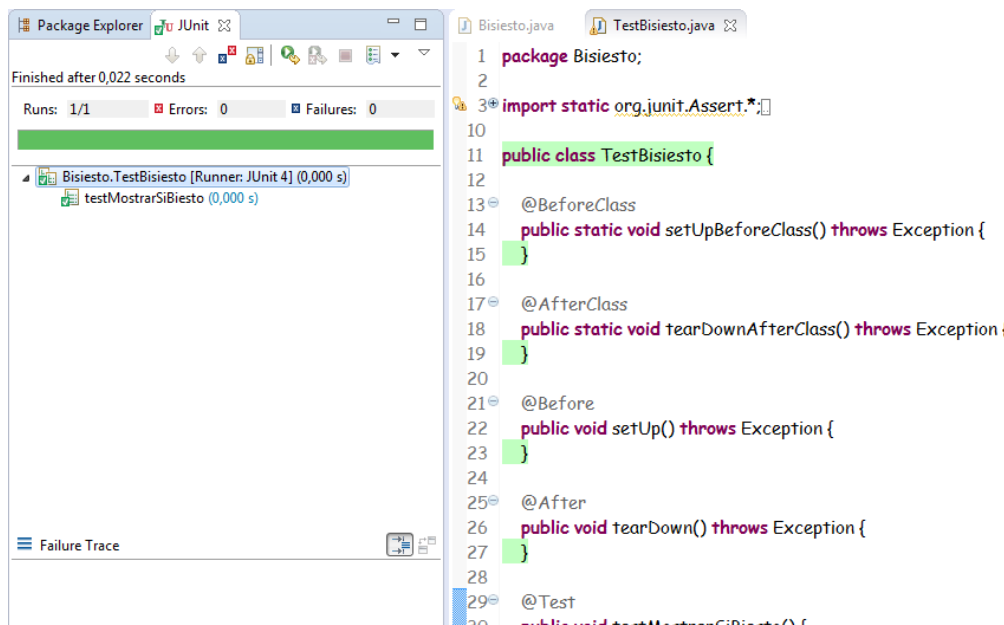
    @AfterClass
    public static void tearDownAfterClass() throws Exception {
    }

    @Before
    public void setUp() throws Exception {
    }

    @After
    public void tearDown() throws Exception {
    }

    @Test
    public void testMostrarSiBisiesto() {
        Bisiesto.mostrarSiBisiesto(1600); //Divisible por 4, por 100 y por 400. Bisiesto
        Bisiesto.mostrarSiBisiesto(1996); //Divisible entre 4 y No divisible por 100. Bisiesto
        Bisiesto.mostrarSiBisiesto(1995); //No divisible por 4. No bisiesto
        Bisiesto.mostrarSiBisiesto(2100); //Divisible por 4 y por 100 pero No entre 400. No bisiesto
        Bisiesto.mostrarSiBisiesto(1670); //No divisible entre 4. No bisiesto
    }
}
```

Una vez tengamos creado el test, lo abrimos con la perspectiva de JUnit.



Perspectiva Eclipse (Eclemma).

The screenshot shows the Eclipse IDE with two panels. The left panel displays the source code of `TestBisiesto.java`. The right panel shows the 'Coverage' view, which provides a summary of test results for the project.

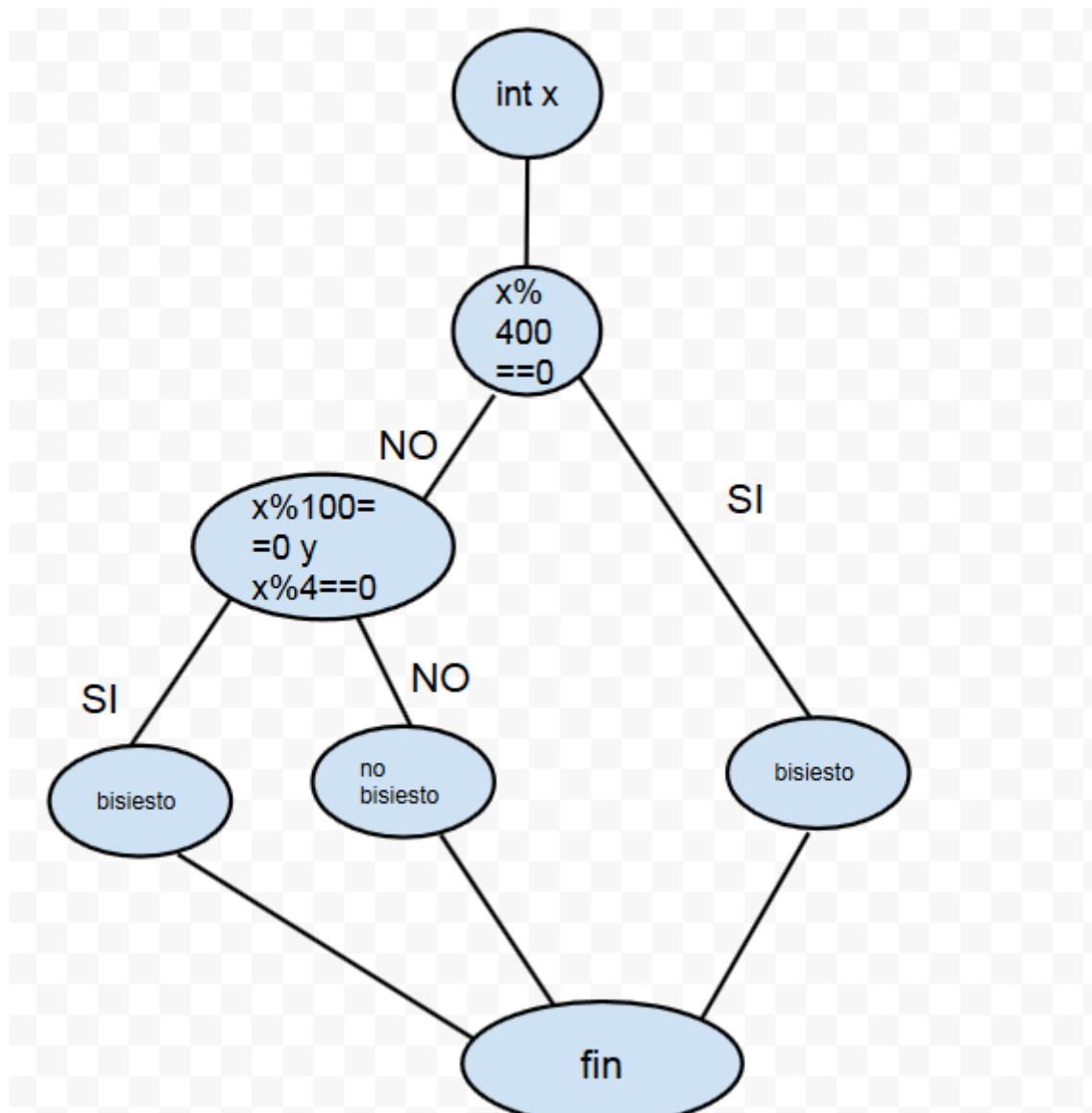
Source Code (TestBisiesto.java):

```
1 package Bisiesto;
2
3 import static org.junit.Assert.*;
4
5 public class TestBisiesto {
6
7     @BeforeClass
8     public static void setUpBeforeClass() throws Exception {
9     }
10
11     @AfterClass
12     public static void tearDownAfterClass() throws Exception {
13     }
14
15     @Before
16     public void setUp() throws Exception {
17     }
18
19     @After
20     public void tearDown() throws Exception {
21     }
22
23     @Test
24     public void testMostrarSiBisiesto() {
25         Bisiesto.mostrarSiBisiesto(1600); //Divisible por 4, por 100 y por 400. Bisiesto
26         Bisiesto.mostrarSiBisiesto(1996); //Divisible entre 4 y No divisible por 100. Bisiesto
27         Bisiesto.mostrarSiBisiesto(1995); //No divisible por 4. No bisiesto
28         Bisiesto.mostrarSiBisiesto(2100); //Divisible por 4 y por 100 pero No entre 400. No bisiesto
29         Bisiesto.mostrarSiBisiesto(1670); //No divisible entre 4. No bisiesto
30     }
31 }
```

Coverage View:

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
Eclemma	93.9 %	46	3	49
src	93.9 %	46	3	49
Bisiesto	93.9 %	46	3	49
Bisiesto.java	90.3 %	28	3	31
TestBisiesto.java	100.0 %	18	0	18

Grafo de flujo del método mostrarSiBisiesto(...).



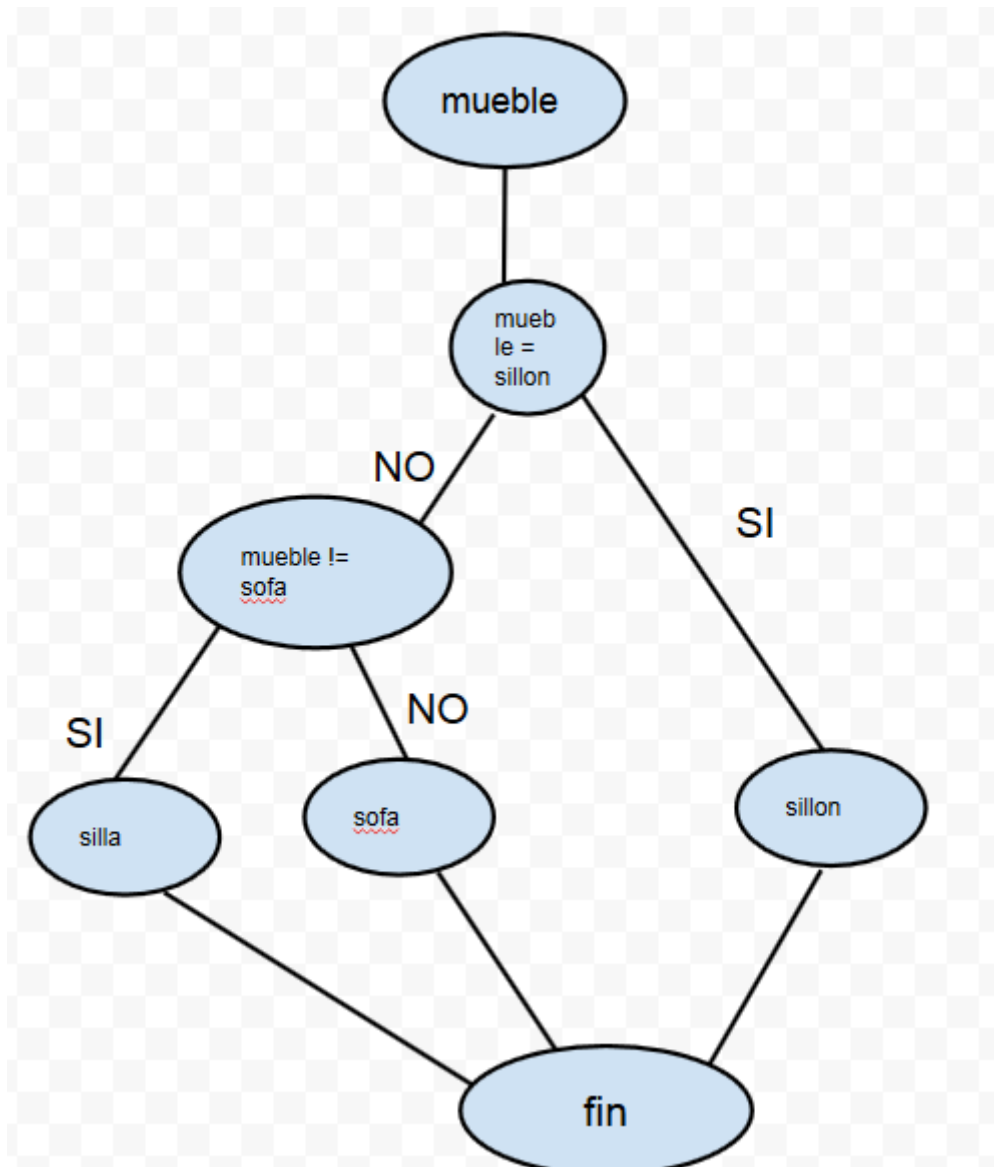
Complejidad ciclomática de mostrarSiBisiesto(...):

$$M = 7 - 6 + 2 \cdot 1 = 3$$

La complejidad ciclomática es 3.

2. Muebles

Grafo de mostrarMuebles(...)



Complejidad ciclomática de mostrarMuebles(...):

$M = 7 - 6 + 2 * 1 = 3$

La complejidad ciclomática es 3.

Perspectiva Eclipse (Eclemma) Clase Muebles

Package Explorer

JUnit 3.8.1

Finished after 0.019 seconds

Run: 1/1 Errors: 0 Failures: 0

Muebles.TestMuebles (Runner: JUnit 4) (0.000 s)

Failure Trace

Bisiesto.java

TestBisiesto.java

Muebles.java

TestMuebles.java

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

@AfterClass

public static void tearDownAfterClass() throws Exception {

}

@Before

public void setUp() throws Exception {

Muebles mueble = new Muebles();

}

@After

public void tearDown() throws Exception {

}

@Test

public void test() {

Muebles.mostrarMuebles("sillon");

Muebles.mostrarMuebles("sofa");

Muebles.mostrarMuebles("silla");

}

}

Problems

JavaDoc

Declaration

Console

Coverage 3.3

TestMuebles (01-jun-2015 21:21:32)

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
Eclemma	44.3 %	39	49	88
src	44.3 %	39	49	88
Bisiesto	0.0 %	0	49	49
Muebles	100.0 %	39	0	39