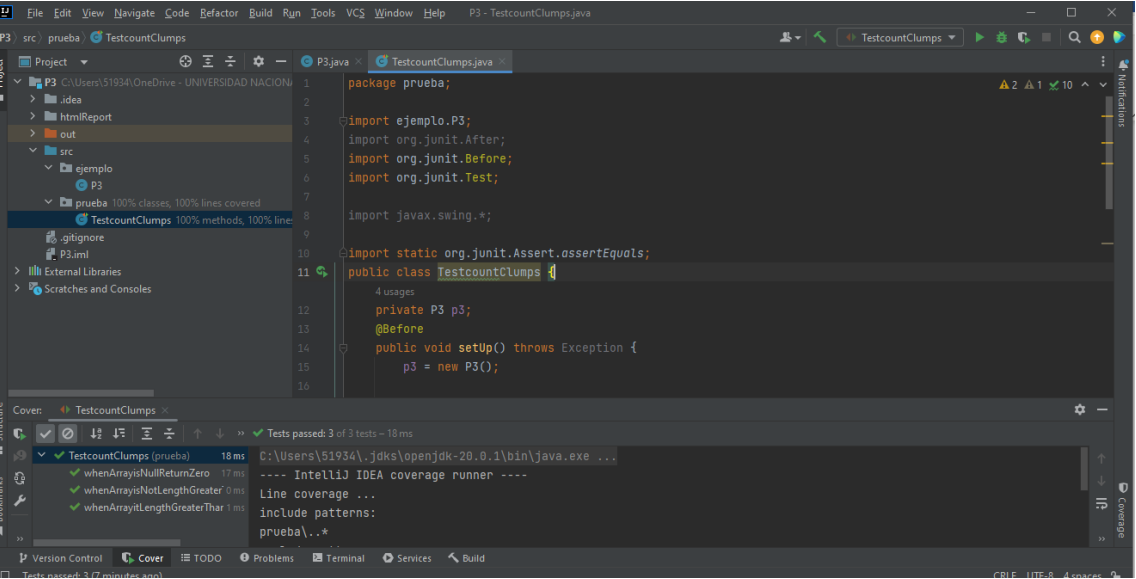


Usando @Before automatizamos el código de tal manera que evitemos escribir código repetido dentro de los métodos de prueba



```
1 package prueba;
2
3 import ejemplo.P3;
4 import org.junit.After;
5 import org.junit.Before;
6 import org.junit.Test;
7
8 import javax.swing.*;
9
10 import static org.junit.Assert.assertEquals;
11 public class TestcountClumps {
12     private P3 p3;
13     @Before
14     public void setUp() throws Exception {
15         p3 = new P3();
16     }
17
18     @Test
19     public void whenArrayIsNullReturnZero() {
20         int[] array = null;
21         assertEquals("El array es null", 0, p3.countClumps(array));
22     }
23
24     @Test
25     public void whenArrayIsNotLengthGreaterThanZeroReturnszero() {
26         int[] array = {1,1,2,2,3,4,5,4,4,4};
27         assertEquals("El array no tiene longitud mayor a 0", 0, p3.countClumps(array));
28     }
29
30     @Test
31     public void whenArrayIsLengthGreaterThanZeroReturnszero() {
32         int[] array = {1,1,2,2,3,4,5,4,4,4};
33         assertEquals("El array tiene longitud mayor a 0", 3, p3.countClumps(array));
34     }
35 }
```

Cover: TestcountClumps x

Tests passed: 3 of 3 tests - 18 ms

TestcountClumps (prueba) 18 ms

whenArrayIsNullReturnZero 17 ms

whenArrayIsNotLengthGreaterThanZero 0 ms

whenArrayIsLengthGreaterThanZero 1 ms

IntelliJ IDEA coverage runner

Line coverage ...

Include patterns:

prueba\..\*

Se uso la herramienta de cobertura Coverage de código favorita.

Current scope: all classes

Overall Coverage Summary

Package	Class, %	Method, %	Line, %
all classes	100% (1/1)	100% (5/5)	100% (8/8)

Coverage Breakdown

Package	Class, %	Method, %	Line, %
prueba	100% (1/1)	100% (5/5)	100% (8/8)

generated on 2023-06-21 11:45

Coverage Report > TestcountClu x Coverage Report > Summary x Coverage Report > TestcountClu x

Archivo | C:\Users\51934\OneDrive\%2520UNIVERSIDAD%2520NACIONAL%2520DE%2520INGENIERIA\Desktop... No sincronizando

Current scope: all classes | prueba

Coverage Summary for Class: TestcountClumps (prueba)

Class	Class, %	Method, %	Line, %
TestcountClumps	100% (1/1)	100% (5/5)	100% (8/8)

```
1 package prueba;
2
3 import ejemplo.P3;
4 import org.junit.After;
5 import org.junit.Before;
6 import org.junit.Test;
7
8 import javax.swing.*;
9
10 import static org.junit.Assert.assertEquals;
11 public class TestcountClumps {
12     private P3 p3;
13     @Before
14     public void setUp() throws Exception {
15         p3 = new P3();
16     }
17
18     @Test
19     public void whenArrayIsNullReturnZero() {
20         int[] array = null;
21         assertEquals("El array es null", 0, p3.countClumps(array));
22     }
23
24     @Test
25     public void whenArrayIsNotLengthGreaterThanZeroReturnszero() {
26         int[] array = {1,1,2,2,3,4,5,4,4,4};
27         assertEquals("El array no tiene longitud mayor a 0", 0, p3.countClumps(array));
28     }
29
30     @Test
31     public void whenArrayIsLengthGreaterThanZeroReturnszero() {
32         int[] array = {1,1,2,2,3,4,5,4,4,4};
33         assertEquals("El array tiene longitud mayor a 0", 3, p3.countClumps(array));
34     }
35 }
```

Veamos para un método de prueba

Current scope: all classes | prueba

Coverage Summary for Package: prueba

Package	Class, %	Method, %	Line, %
prueba	100% (1/1)	60% (3/5)	50% (4/8)

Class	Class, %	Method, %	Line, %
<a href="#">TestcountClumps</a>	100% (1/1)	60% (3/5)	50% (4/8)

generated on 2023-06-21 11:55

The screenshot shows an IDE window with a Java test class named `TestcountClumps` and its coverage summary. The top bar indicates the file path: `C:\Users\S1934\OneDrive%2520UNIVERSIDAD%2520NACIONAL%2520INGENIERIA\Desktop... A`. The top right shows a status bar with `No sincronizando`.

The coverage summary table is as follows:

Class	Class, %	Method, %	Line, %
TestcountClumps	100% (1/1)	60% (3/5)	50% (4/8)

The code editor shows the following Java code:

```
1 package prueba;
2
3 import ejemplo.P3;
4 import org.junit.After;
5 import org.junit.Before;
6 import org.junit.Test;
7
8 import javax.swing.*;
9
10 import static org.junit.Assert.assertEquals;
11 public class TestcountClumps {
12     private P3 p3;
13     @Before
14     public void setUp() throws Exception {
15         p3 = new P3();
16     }
17
18     @Test
19     public void whenArrayIsNullReturnZero() {
20         int[] array = null;
21         assertEquals("El array es null", 0, p3.countClumps(array));
22     }
23
24     @Test
25     public void whenArrayIsNotLengthGreaterThanZeroReturnZero() {
26         int[] array = {};
27         assertEquals("El array no tiene longitud mayor a 0", 0, p3.countClumps(array));
28     }
29
30     @Test
31     public void whenArrayIsLengthGreaterThanZeroReturnZero() {
32         int[] array = {1,1,2,2,3,6,5,4,4,4};
33         assertEquals("El array tiene longitud mayor a 0", 3, p3.countClumps(array));
34     }
35 }
36
```