

JUnit Test

Montes Cetina Juan Nicolás

Universidad de San Buenaventura, Sede Bogotá.

Facultad de Ingeniería.

Programa de Sistemas

Bogotá, Colombia

2020

## Tabla de contenido

Tests de conexión a la base de datos Cassandra .....	3
ConexionTest .....	3
ConexionAfterClass.....	4
ConexionParameterized .....	5
ConexionIgnore .....	6
ConexionRuleTimeOut.....	7
Conexion FixMethodOrder .....	8
Tests de creación de productos de la clase BO .....	9
CrearProductosBOTest .....	9
CrearProductosBORuleTimeOut.....	9
CrearProductosBOIgnore.....	10
CrearProductosBOFixMethodOrder.....	11
CrearProductosBOAfterClass .....	12
CrearProductosBOBeforeClass .....	13
Tests de actualización de productos de la clase BO .....	14
EditarProductosBOTest .....	14
EditarProductosBORuleTimeOut.....	14
EditarProductosBOIgnore.....	15
EditarProductosBOFixMethodOrder .....	16
EditarProductosBOBeforeClass .....	17
EditarProductosBOAfterClass .....	18
Tests de eliminación de productos de la clase BO .....	19
EliminarProductosBOTest .....	19
EliminarProductosBORuleTimeOut.....	20
EliminarProductosBOIgnore.....	21
EliminarProductosBOFixMethodOrder .....	22
EliminarProductosBOBeforeClass .....	23
EliminarProductosBOAfterClass .....	24

# Tests de conexión a la base de datos Cassandra

## ConexionTest

```
package Test;

import bd.Conexion;
import bo.Productos_BO;
import org.junit.After;
import org.junit.Test;
import static org.junit.Assert.*;
import org.junit.Before;

/**
 *
 * @author Juan Nicolas
 */
public class ConexionTest {

    Conexion co;
    Productos_BO prod_bo;

    @Before
    public void before() {
        System.out.println("before()");
        co = new Conexion();
        prod_bo = new Productos_BO();
    }

    @Test
    public void testConexion() {
        System.out.println("testConexion()");
        Conexion.getConnection();
    }

    @After
    public void after() {
        System.out.println("after()");
        assertTrue("Se ha conectado a la base de datos Cassandra", true);
    }

}
```



## ConexionAfterClass

```
/**
 *
 * @author Juan Nicolas
 */
public class ConexionAfterClass {

    Conexion co;
    Productos_BO prod_bo;

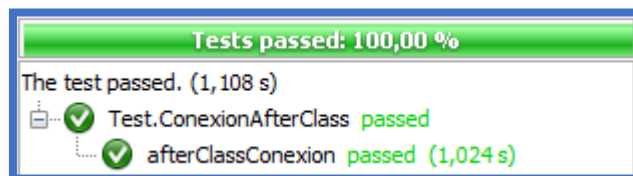
    @Before
    public void before() {
        co = new Conexion();
        prod_bo = new Productos_BO();
    }

    @BeforeClass
    public static void beforeClass() {
        System.out.println("beforeClass()");
    }

    @Test
    public void afterClassConexion() {
        System.out.println("afterClassConexion()");
        Conexion.getConnection();
    }

    @After
    public void after() {
    }

    @AfterClass
    public static void afterClass() {
        System.out.println("afterClass()");
        assertTrue("Se ha conectado a la base de datos Cassandra", true);
    }
}
```



## ConexionParameterized

```
/**
 *
 * @author Juan Nicolas
 */
public class ConexionParameterized {

    private Conexion co;

    @Before
    public void before() {
        co = new Conexion();
    }

    @BeforeClass
    public static void BeforeClass() {
    }

    @AfterClass
    public static void AfterClass() {
    }

    @After
    public void after() {
    }

    @Parameterized.Parameters
    public static Object[][] datos() {
        return new Object[8][0];
    }

    @Test
    public void testConexion() {
        Conexion.getConnection();
    }
}
```

Tests passed: 100,00 %

The test passed. (1,108 s)

Test.ConexionParameterized passed  
testConexion passed (1,03 s)

## ConexionIgnore

```
/**
 *
 * @author Juan Nicolas
 */
public class ConexionIgnore {

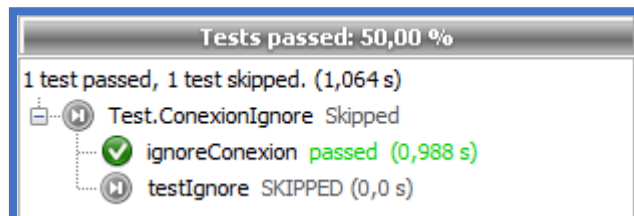
    Conexion co;

    @Before
    public void before() {
        co = new Conexion();
    }

    @Test
    public void ignoreConexion() {
        Conexion.getConnection();
    }

    @After
    public void after() {
        assertTrue("Se ha conectado a la base de datos Cassandra", true);
    }

    @Test
    @Ignore
    public void testIgnore() {
        System.out.println(5 * 4);
    }
}
```



## ConexionRuleTimeOut

```
/**
 *
 * @author Juan Nicolas
 */
public class ConexionRuleTimeOut {

    Conexion co;

    @Before
    public void before() {
        co = new Conexion();
    }



    @Test
    public void ruleTimeOutConexion() {
        Conexion.getConnection();
    }

    @After
    public void after() {
        assertTrue("Se ha conectado a la base de datos Cassandra", true);
    }

    @Rule
    public Timeout timeout = Timeout.millis(5000);
}
```

Tests passed: 100,00 %

The test passed. (1,07 s)

  Test.ConexionRuleTimeOut passed

 ruleTimeOutConexion passed (0,989 s)

## Conexion FixMethodOrder

```
/**
 *
 * @author Juan Nicolas
 */
@FixMethodOrder(MethodSorters.NAME_ASCENDING)
public class ConexionFixMethodOrder {

    Conexion co;
    Productos_BO prod_bo;

    public ConexionFixMethodOrder() {
        co = new Conexion();
        prod_bo = new Productos_BO();
    }

    @BeforeClass
    public static void BeforeClass() {
    }

    @AfterClass
    public static void AfterClass() {
    }

    @Before
    public void Before() {
    }

    @After
    public void After() {
    }

    @Test
    public void t1_getConexion() {
        Conexion.getConnection();
    }

    @Test
    public void t2_Crear() {
        prod_bo.crear(new Productos(1, "Pan de semillas", "Marinelas", 20, "Libras", 500000, "3/07/2021", "2/3/2020"));
    }
}
```

Tests passed: 100,00 %

Both tests passed. (6,41 s)

Test.ConexionFixMethodOrder passed

t1\_getConexion passed (0,991 s)

t2\_Crear passed (5,338 s)



## Tests de creación de productos de la clase BO

### CrearProductosBOTest

```
/**
 *
 * @author Juan Nicolas
 */
public class CrearProductosBOTest {

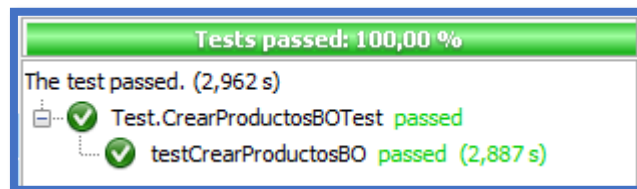
    Productos_BO prod_bo;

    @Before
    public void before() {
        System.out.println("before()");
        prod_bo = new Productos_BO();
    }

    @Test
    public void testCrearProductosBO() {
        System.out.println("testCrear()");
        prod_bo.crear(new Productos(1, "Pan de semillas", "Marinela", 20, "Libras", 20000, "3/07/2021", "2/3/2020"));
    }

    @After
    public void after() {
        System.out.println("after()");
        assertTrue("Se agrego el producto", true);
    }

}
```



### CrearProductosBORuleTimeOut

```
/**
 *
 * @author Juan Nicolas
 */
public class CrearProductosBORuleTimeOut {

    Productos_BO prod_bo;

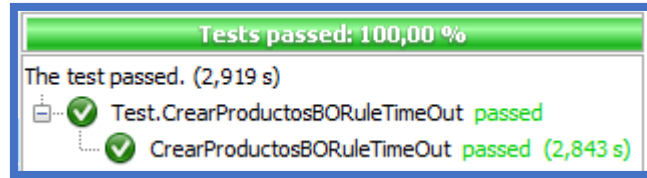
    @Before
    public void before() {
        prod_bo = new Productos_BO();
    }

    @Test
    public void CrearProductosBORuleTimeOut() {
        prod_bo.crear(new Productos(2, "Pan integral", "Comapans", 15, "Libras", 200000, "2/10/2021", "1/1/2020"));
    }

    @After
    public void after() {
        assertTrue("Se agrego el producto", true);
    }

    @Rule
    public Timeout timeout = Timeout.millis(5000);

}
```



## CrearProductosBOIgnore

```
/**
 *
 * @author Juan Nicolas
 */
public class CrearProductosBOIgnore {

    Productos_BO prod_bo;

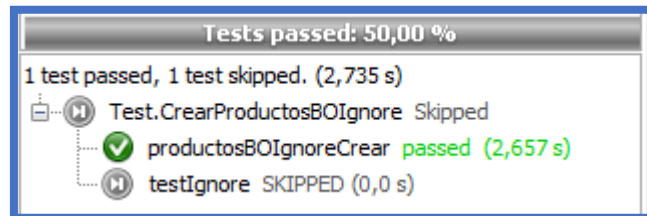
    @Before
    public void before() {

        prod_bo = new Productos_BO();
    }

    @Test
    public void productosBOIgnoreCrear() {
        prod_bo.crear(new Productos(1, "Pan integral", "Comapan", 15, "Libras", 30000, "2/10/2021", "1/1/2020"));
    }

    @After
    public void after() {
        assertTrue("Se agrego el producto", true);
    }

    @Test
    @Ignore
    public void testIgnore() {
        System.out.println(34 - 9);
    }
}
```



## CrearProductosBOFixMethodOrder

```
/**
 *
 * @author Juan Nicolas
 */
@FixMethodOrder(MethodSorters.NAME_ASCENDING)
public class CrearProductosBOFixMethodOrder {
    Conexion co;
    Productos_BO prod_bo;
    public CrearProductosBOFixMethodOrder() {
        co = new Conexion();
        prod_bo = new Productos_BO();
    }

    @BeforeClass
    public static void BeforeClass() {
    }

    @AfterClass
    public static void AfterClass() {
    }

    @Before
    public void Before() {
    }

    @After
    public void After() {
    }

    @Test
    public void test1_getConexion() {
        Conexion.getConnection();
    }

    @Test
    public void test2_CrearProductosBO() {
        prod_bo.crear(new Productos(1, "Pan de semillas", "Marinelas", 20, "Libras", 500000, "3/07/2021", "2/3/2020"));
    }
}
```

Tests passed: 100,00 %

Both tests passed. (2,983 s)

✓

Test.CrearProductosBOFixMethodOrder passed

✓

test1\_getConexion passed (1,018 s)

✓

test2\_CrearProductosBO passed (1,881 s)

## CrearProductosBOAfterClass

```
/**
 *
 * @author Juan Nicolas
 */
public class CrearProductosBOAfterClass {

    Productos_BO prod_bo;

    @Before
    public void before() {
        prod_bo = new Productos_BO();
    }

    @BeforeClass
    public static void beforeClass() {
        System.out.println("beforeClass()");
    }

    @Test
    public void afterClassCrear() {
        System.out.println("afterClassCrear()");
        prod_bo.crear(new Productos(1, "Pan de semillas", "Marinela", 20, "Libras", 20000, "3/07/2021", "2/3/2020"));
    }

    @After
    public void after() {
    }

    @AfterClass
    public static void afterClass() {
        System.out.println("afterClass()");
        assertTrue("Se agrego el producto", true);
    }
}
```

Tests passed: 100,00 %

The test passed. (4,553 s)



Test.CrearProductosBOAfterClass passed



afterClassCrear passed (4,477 s)

## CrearProductosBOBeforeClass

```
/**
 *
 * @author Juan Nicolas
 */
public class CrearProductosBOBeforeClass {

    Conexion co;
    Productos_BO prod_bo;

    public CrearProductosBOBeforeClass() {
        co = new Conexion();
        prod_bo = new Productos_BO();
    }

    @BeforeClass
    public static void BeforeClass() {
        Conexion.getConnection();
    }

    @AfterClass
    public static void AfterClass() {
    }

    @Before
    public void Before() {
    }

    @After
    public void After() {
    }

    @Test
    public void testBeforeClassCrearProductosBO() {
        prod_bo.crear(new Productos(1, "Pan de semillas", "Marinela", 20, "Libras", 20000, "3/07/2021", "2/3/2020"));
    }
}
```

Tests passed: 100,00 %

The test passed. (2,878 s)

Test.CrearProductosBOBeforeClass passed

testBeforeClassCrearProductosBO passed (1,789 s)

## Tests de actualización de productos de la clase BO

### EditarProductosBOTest

```
/**
 *
 * @author Juan Nicolas
 */
public class EditarProductosBOTest {

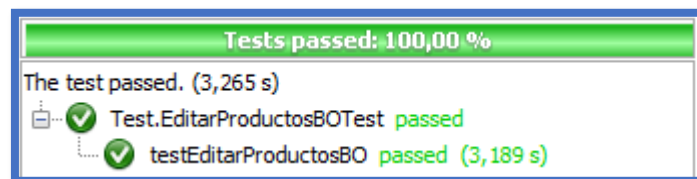
    Productos_BO prod_bo;

    @Before
    public void before() {
        System.out.println("before()");
        prod_bo = new Productos_BO();
    }

    @Test
    public void testEditarProductosBO() {
        System.out.println("testEditar()");
        prod_bo.editar(new Productos(1, "Pan de semillas", "Marinela", 20, "Libras", 20000, "3/07/2021", "2/3/2020"));
    }

    @After
    public void after() {
        System.out.println("after()");
        assertTrue("Se editado el producto", true);
    }

}
```



### EditarProductosBORuleTimeout

```
/**
 *
 * @author Juan Nicolas
 */
public class EditarProductosBORuleTimeout {

    Productos_BO prod_bo;

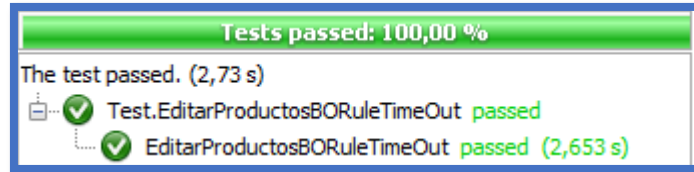
    @Before
    public void before() {
        prod_bo = new Productos_BO();
    }

    @Test
    public void EditarProductosBORuleTimeout() {
        prod_bo.editar(new Productos(2, "Pan integral", "Comapans", 15, "Libras", 200000, "2/10/2021", "1/1/2020"));
    }

    @After
    public void after() {
        assertTrue("Se edito el producto", true);
    }

    @Rule
    public Timeout timeout = Timeout.millis(5000);

}
```



## EditorProductosBOIgnore

```
/**
 *
 * @author Juan Nicolas
 */
public class EditorProductosBOIgnore {

    Productos_BO prod_bo;

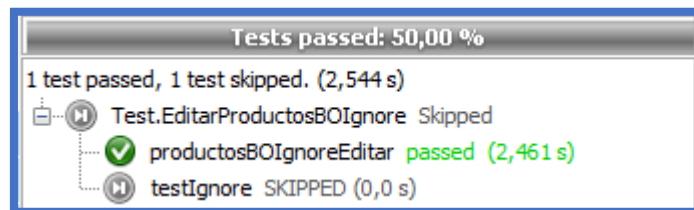
    @Before
    public void before() {

        prod_bo = new Productos_BO();
    }

    @Test
    public void productosBOIgnoreEditar() {
        prod_bo.editar(new Productos(1, "Pan integral", "Comapan", 15, "Libras", 30000, "2/10/2021", "1/1/2020"));
    }

    @After
    public void after() {
        assertTrue("Se edito el producto", true);
    }

    @Test
    @Ignore
    public void testIgnore() {
        System.out.println(5*5);
    }
}
```



## EditarProductosBOFixMethodOrder

```
/**
 *
 * @author Juan Nicolas
 */
@FixMethodOrder(MethodSorters.NAME_ASCENDING)
public class EditarProductosBOFixMethodOrder {
    Conexion co;
    Productos_BO prod_bo;

    public EditarProductosBOFixMethodOrder() {
        co = new Conexion();
        prod_bo = new Productos_BO();
    }

    @BeforeClass
    public static void BeforeClass() {
    }

    @AfterClass
    public static void AfterClass() {
    }

    @Before
    public void Before() {
    }

    @After
    public void After() {
    }

    @Test
    public void test1_getConexion() {
        Conexion.getConnection();
    }

    @Test
    public void test2_EditarProductosBO() {
        prod_bo.editar(new Productos(1, "Pan de semillas", "Marinelas", 20, "Libras", 500000, "3/07/2021", "2/3/2020"));
    }
}
```

Tests passed: 100,00 %

Both tests passed. (2,732 s)

- ✓ Test.EditarProductosBOFixMethodOrder passed
  - ✓ test1\_getConexion passed (0,952 s)
  - ✓ test2\_EditarProductosBO passed (1,704 s)



## EditarProductosBOBeforeClass

```
/**
 *
 * @author Juan Nicolas
 */
public class EditarProductosBOBeforeClass {

    Conexion co;
    Productos_BO prod_bo;

    public EditarProductosBOBeforeClass() {
        co = new Conexion();
        prod_bo = new Productos_BO();
    }

    @BeforeClass
    public static void BeforeClass() {
        Conexion.getConnection();
    }

    @AfterClass
    public static void AfterClass() {
    }

    @Before
    public void Before() {
    }

    @After
    public void After() {
    }

    @Test
    public void testBeforeClassEditarProductosBO() {
        prod_bo.editar(new Productos(1, "Pan de semillas", "Marinela", 20, "Libras", 20000, "3/07/2021", "2/3/2020"));
    }
}
```

Tests passed: 100,00 %

The test passed. (2,475 s)



Test.EditarProductosBOBeforeClass passed



testBeforeClassEditarProductosBO passed (1,451 s)

## EditorProductosBOAfterClass

```
/**
 *
 * @author Juan Nicolas
 */
public class EditorProductosBOAfterClass {

    Productos_BO prod_bo;

    @Before
    public void before() {
        prod_bo = new Productos_BO();
    }

    @BeforeClass
    public static void beforeClass() {
        System.out.println("beforeClass()");
    }

    @Test
    public void afterClassEditor() {
        System.out.println("afterClassEditor()");
        prod_bo.editar(new Productos(1, "Pan de semillas", "Marinela", 20, "Libras", 20000, "3/07/2021", "2/3/2020"));
    }

    @After
    public void after() {
    }

    @AfterClass
    public static void afterClass() {
        System.out.println("afterClass()");
        assertTrue("Se edito el producto", true);
    }
}
```

Tests passed: 100,00 %

The test passed. (2,354 s)



Test.EditorProductosBOAfterClass passed



afterClassEditor passed (2,281 s)

## Tests de eliminación de productos de la clase BO

### EliminarProductosBOTest

```
/**
 *
 * @author Juan Nicolas
 */
public class EliminarProductosBOTest {

    Productos_BO prod_bo;

    @Before
    public void before() {
        System.out.println("before()");
        prod_bo = new Productos_BO();
    }

    @Test
    public void testEliminarProductosBO() {
        System.out.println("testEliminar()");
        prod_bo.eliminar(1);
    }

    @After
    public void after() {
        System.out.println("after()");
        assertTrue("Se elimino el producto", true);
    }

}
```

Tests passed: 100,00 %

The test passed. (4,001 s)

- Test.EliminarProductosBOTest passed
- testEliminarProductosBO passed (3,918 s)

## EliminarProductosBORuleTimeOut

```
/**
 *
 * @author Juan Nicolas
 */
public class EliminarProductosBORuleTimeOut {

    Productos_BO prod_bo;

    @Before
    public void before() {
        prod_bo = new Productos_BO();
    }



    @Test
    public void EliminarProductosBORuleTimeOut() {
        prod_bo.eliminar(1);
    }


    @After
    public void after() {
        assertTrue("Se elimino el producto", true);
    }

    @Rule
    public Timeout timeout = Timeout.millis(5000);
}
```

Tests passed: 100,00 %

The test passed. (2,792 s)

  Test.EliminarProductosBORuleTimeOut passed

 EliminarProductosBORuleTimeOut passed (2,708 s)

## EliminarProductosBOIgnore

```
/**
 *
 * @author Juan Nicolas
 */
public class EliminarProductosBOIgnore {

    Productos_BO prod_bo;

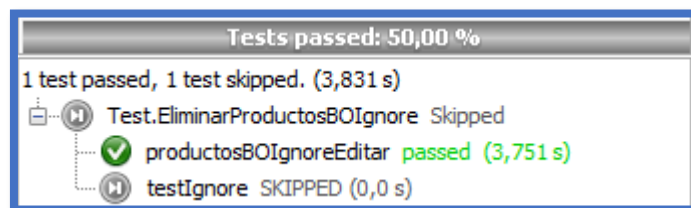
    @Before
    public void before() {

        prod_bo = new Productos_BO();
    }

    @Test
    public void productosBOIgnoreEditar() {
        prod_bo.eliminar(1);
    }

    @After
    public void after() {
        assertTrue("Se elimino el producto", true);
    }

    @Test
    @Ignore
    public void testIgnore() {
        System.out.println(5 * 5);
    }
}
```



## EliminarProductosBOFixMethodOrder

```
/**
 *
 * @author Juan Nicolas
 */
@FixMethodOrder(MethodSorters.NAME_ASCENDING)
public class EliminarProductosBOFixMethodOrder {

    Conexion co;
    Productos_BO prod_bo;

    public EliminarProductosBOFixMethodOrder() {
        co = new Conexion();
        prod_bo = new Productos_BO();
    }

    @BeforeClass
    public static void BeforeClass() {
    }

    @AfterClass
    public static void AfterClass() {
    }

    @Before
    public void Before() {
    }



    @After
    public void After() {
    }


    @Test
    public void test1_getConexion() {
        Conexion.getConnection();
    }

    @Test
    public void test2_EliminarProductosBO() {
        prod_bo.eliminar(1);
    }
}
```

Tests passed: 100,00 %

The test passed. (0,0 s) Running...

  Test.EliminarProductosBOFixMethodOrder passed

 test1\_getConexion passed (0,0 s)

## EliminarProductosBOBeforeClass

```
/**
 *
 * @author Juan Nicolas
 */
public class EliminarProductosBOBeforeClass {

    Conexion co;
    Productos_BO prod_bo;

    public EliminarProductosBOBeforeClass() {
        co = new Conexion();
        prod_bo = new Productos_BO();
    }

    @BeforeClass
    public static void BeforeClass() {
        Conexion.getConnection();
    }

    @AfterClass
    public static void AfterClass() {
    }

    @Before
    public void Before() {
    }

    @After
    public void After() {
    }

    @Test
    public void testBeforeClassEliminarProductosBO() {
        prod_bo.eliminar(1);
    }
}
```

Tests passed: 100,00 %

The test passed. (6,245 s)



Test.EliminarProductosBOBeforeClass passed



testBeforeClassEliminarProductosBO passed (3,615 s)

## EliminarProductosBOAfterClass

```
/**
 *
 * @author Juan Nicolas
 */
public class EliminarProductosBOAfterClass {

    Productos_BO prod_bo;

    @Before
    public void before() {
        prod_bo = new Productos_BO();
    }

    @BeforeClass
    public static void beforeClass() {
        System.out.println("beforeClass()");
    }

    @Test
    public void afterClassEliminar() {
        System.out.println("afterClassEliminar()");
        prod_bo.eliminar(1);
    }

    @After
    public void after() {
    }

    @AfterClass
    public static void afterClass() {
        System.out.println("afterClass()");
        assertTrue("Se elimino el producto", true);
    }
}
```

Tests passed: 100,00 %

The test passed. (4,455 s)

- Test.EliminarProductosBOAfterClass passed
- afterClassEliminar passed (4,347 s)