

Mónica lleva algunos meses dándole vueltas a una idea de negocio que tiene en mente y que cree que puede funcionar muy bien por su zona. El pueblo de Coín es conocido por su producción agricultora, cada vez son más las personas que se acercan a la localidad para adquirir sus frutas y verduras ecológicas. La idea de Mónica es crear un aplicación que una a agricultores con usuarios para la venta y compra de productos ecológicos de la huerta local. Finalmente, decide seguir adelante con el proyecto que se llamará AgroNatura y va a contar contigo para el desarrollo de la aplicación.

Comenzamos desarrollando un programa básico para gestionar los productos de AgroNatura.

En github, en el proyecto [vacostaAF/globalED25: Tarea global DAM 2025](#), encontrarás dos archivos: `Producto.java` y `ProductoCRUD.java`.

`Producto.java` contiene la clase `Producto`, que define los artículos producidos.

`ProductoCRUD.java` contiene un CRUD para la clase `Producto`, sobre una colección `TreeMap`.

Se pide:

- Crear casos de prueba para la clase `ProductoCRUD`, que pruebe que la colección `TreeMap` se gestiona correctamente. Incluir el resultado de la prueba de todos los tests. (4 puntos)
- Documentar todo el código, utilizando `JavaDoc`. Intentar generar la documentación en formato web. (3 puntos)
- Incluir el proyecto generado en una rama con tu nombre, y publicarlo en el repositorio remoto. (3 puntos)

Entregar un documento con todo el código desarrollado, y el resultado de las pruebas.

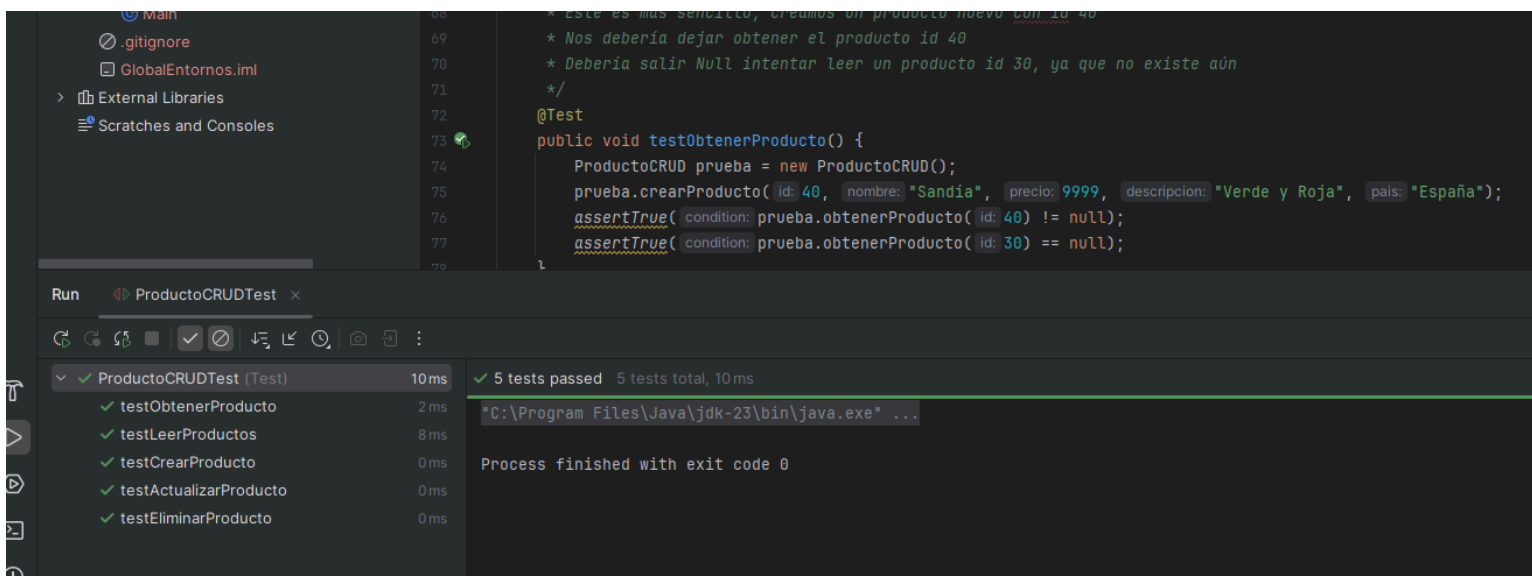
Aunque te comparta el archivo por GitHub, voy a pasarte las capturas del código del test primero:

```
© Producto.java x  © ProductoCRUD.java  © ProductoCRUDTest.java x
1  /**
2   * Vamos a iniciar una clase ProductoCRUDTest para probar los Métodos de la clase ProductoCrud
3   */
4  package Test;
5
6  import Productos.ProductoCRUD;
7  import org.junit.Test;
8  import static org.junit.Assert.*;
9
10 public class ProductoCRUDTest {
11
12     /**
13      * Vamos a crear un producto nuevo con una id, y luego vamos a intentar crear otro con la misma id.
14      * El primero debería ejecutarse sin problemas
15      * El segundo debería dar error ya que existe un producto anterior con esa id
16      */
17     @Test
18     public void testCrearProducto() {
19         ProductoCRUD prueba = new ProductoCRUD();
20         assertTrue(prueba.crearProducto(id: 1, nombre: "Tomate", precio: 1.5, descripcion: "Es rojo", pais: "España"));
21         assertFalse(prueba.crearProducto(id: 1, nombre: "Está repe", precio: 9999, descripcion: "Repetido", pais: "España")); // Mismo ID
22     }
```

```
© Producto.java  © ProductoCRUD.java  © ProductoCRUDTest.java x
10 public class ProductoCRUDTest {
25     /**
26      * En esta vamos a hacer dos test
27      * Primero vamos a leer nuestros productos, sin haber metido ninguno, por lo que debería salir el String "No hay productos."
28      * Despues, metemos la lechuga, ejecutamos el metodo y debería salirlos entre todo el resultado la palabra "Lechuga".
29      */
30     @Test
31     public void testLeerProductos() {
32         ProductoCRUD prueba = new ProductoCRUD();
33         assertEquals(expected: "No hay productos.", prueba.leerProductos());
34
35         prueba.crearProducto(id: 12, nombre: "Lechuga", precio: 999, descripcion: "Es verde", pais: "España");
36         String resultado = prueba.leerProductos();
37         assertTrue(resultado.contains("Lechuga"));
38
39     /**
40      * Aquí vamos a crear un producto nuevo con id 20, y luego vamos actualizarlo.
41      * Al leer el producto debería salir como resultado la actualización "Pimiento Verde"
42      */
43     @Test
44     public void testActualizarProducto() {
45         ProductoCRUD prueba = new ProductoCRUD();
46         prueba.crearProducto(id: 20, nombre: "Pimiento Rojo", precio: 9999, descripcion: "Era rojo", pais: "España");
47         prueba.actualizarProducto(id: 20, nuevoNombre: "Pimiento Verde", nuevoPrecio: 99999, nuevaDescripcion: "Ahora es verde", nuevoPais: "España");
48         String resultado = prueba.leerProductos();
49         assertTrue(resultado.contains("Pimiento Verde"));
50     }
```

```
ct 10 public class ProductoCRUDTest {
    52 /**
    53     Aquí vamos a comprobar 3 cosas
    54     - Que no se puede eliminar un producto id 30 ya que no existe
    55     - Creamos el apio con id 30 y deberíamos poder borrarlo
    56     - Intentamos de nuevo borrar el id 30 que ya fue borrado, por lo que debería salir error
    57     */
    58     @Test
    59     public void testEliminarProducto() {
    60         ProductoCRUD prueba = new ProductoCRUD();
    61         assertFalse(prueba.eliminarProducto(id: 30)); // aún no existe
    62         prueba.crearProducto(id: 30, nombre: "Apio", precio: 9999, descripcion: "Es blanco", pais: "España");
    63         assertTrue(prueba.eliminarProducto(id: 30));
    64         assertFalse(prueba.eliminarProducto(id: 30)); // ya fue eliminado
    65     }
    66
    67     /**
    68     * Este es más sencillo, creamos un producto nuevo con id 40
    69     * Nos debería dejar obtener el producto id 40
    70     * Debería salir Null intentar leer un producto id 30, ya que no existe aún
    71     */
    72     @Test
    73     public void testObtenerProducto() {
    74         ProductoCRUD prueba = new ProductoCRUD();
    75         prueba.crearProducto(id: 40, nombre: "Sandía", precio: 9999, descripcion: "Verde y Roja", pais: "España");
    76         assertTrue(condition: prueba.obtenerProducto(id: 40) != null);
    77         assertTrue(condition: prueba.obtenerProducto(id: 30) == null);
    78     }
}
```

Y por último la prueba de test con un poco del código del test para que puedas comprobarlo



```
Run ProductoCRUDTest x
58 * Este es más sencillo, creamos un producto nuevo con id 40
59 * Nos debería dejar obtener el producto id 40
60 * Debería salir Null intentar leer un producto id 30, ya que no existe aún
61 */
62 @Test
63 public void testObtenerProducto() {
64     ProductoCRUD prueba = new ProductoCRUD();
65     prueba.crearProducto(id: 40, nombre: "Sandía", precio: 9999, descripcion: "Verde y Roja", pais: "España");
66     assertTrue(condition: prueba.obtenerProducto(id: 40) != null);
67     assertTrue(condition: prueba.obtenerProducto(id: 30) == null);
68 }
69
70
71
72
73
74
75
76
77
78
```

Test	Duration	Status
ProductoCRUDTest (Test)	10ms	✓ 5 tests passed 5 tests total, 10ms
testObtenerProducto	2ms	✓
testLeerProductos	8ms	✓
testCrearProducto	0ms	✓
testActualizarProducto	0ms	✓
testEliminarProducto	0ms	✓

Process finished with exit code 0