

TALLER 6

JUnit

Adriana Sofía Rozo Cepeda

```

v > Taller6-Dpoo [Taller6-Dpoo main]
  > JRE System Library [jre]
  v > TALLER2-DPOO2023-main/TALLER2/TALLER2HAMBURGUESAS/src
    v > CONSOLA
      > Aplicacion.java
    > data
    > Documentos
    v > MODELO
      > Combo.java
      > HamburguesaException.java
      > Ingrediente.java
      > IngredienteRepetidoException.java
      > Pedido.java
      > PedidoValorSuperiorException.java
      > ProductoAjustado.java
      > ProductoMenu.java
      > ProductoRepetidoException.java
    > Procesamiento
    > module-info.java
  v > Test/Source
    v > test
      > ComboTest.java
      > PedidoTest.java
      > ProductoAjustadoTest.java
      > ProductoMenuTest.java
  > JUnit 5
  > TALLER2-DPOO2023-main
  > Test

```

1. Al abrir el proyecto existen dos **grandes source folders**. El TALLER2-DPOO2023-main y el Test/Source.

2. En el package MODELO está la lógica de la aplicación. Ahí están las clases extensión del TALLER 6.

2. En el source folder Test/Source están las clases JUnit de prueba pedidas.

Todas las clases de prueba ejecutan las pruebas correctamente, fueron utilizadas aserciones, pruebas parametrizadas, llamado de otros métodos de prueba, anotaciones tales como BeforeAll. etc.

A continuación, presento la cobertura de las pruebas para cada una de las clases, en el enunciado algunas veces dicen que no se cubra en su totalidad y en clase se manifestó que fueran probados los métodos más importantes:

ProductoAjustado - ProductoAjustadoTest cobertura 96,0%

```

1 package test;
2
3 import static org.junit.jupiter.api.Assertions.*;
15
16 @DisplayName("Pruebas para la clase ProductoAjustado")
17
18 class ProductoAjustadoTest {
19
20     //Atributos
21     private ProductoAjustado producto;
22
23     //Definición de BeforeEach
24
25     @BeforeEach
26     public void setUp() {
27         // Configuración inicial para cada prueba
28         String nombre = "Corral";
29         int precio = 14000;
30         ArrayList<Ingrediente> agregados = new ArrayList<>();
31         ArrayList<Ingrediente> eliminados = new ArrayList<>();
32         int calorías = 0;
33
34         producto = new ProductoAjustado(nombre, precio, calorías, agregados, eliminados);
35     }
36
37     //Pruebas de métodos
38
39     // Métodos 1 y 4
40
41     @Test
42     @DisplayName("Pruebas para el método getPrecio() y generarTextoFactura()")
43     public void testGetPrecioYGenerarTextoFactura() {
44         // Ejecutar las pruebas de modificación
45         assertNotNull(producto, "Error: ProductoAjustado no inicializado");
46         testAgregar();
47     }

```

Element	Coverage	Covered Instru...	Missed Instruct...	Total Instructio...
Taller6-Dpoo	8,4 %	245	2.671	2.916
TALLER2-DPOO2023-main/TALLER2	5,5 %	118	2.047	2.165
Procesamiento	0,0 %	0	820	820
CONSOLA	0,0 %	0	674	674
MODELO	17,6 %	118	553	671
Pedido.java	0,0 %	0	370	370
Combo.java	0,0 %	0	77	77
ProductoMenu.java	0,0 %	0	46	46
PedidoValorSuperiorException.java	0,0 %	0	20	20
IngredienteRepetidoException.java	0,0 %	0	16	16
ProductoRepetidoException.java	0,0 %	0	16	16
HamburguesaException.java	0,0 %	0	4	4
ProductoAjustado.java	96,0 %	97	4	101
Ingrediente.java	100,0 %	21	0	21
Test/Source	16,9 %	127	624	751

Pedido - PedidoTest cobertura 100%

```

1 package test;
2
3 import static org.junit.jupiter.api.Assertions.*;
17
18 @DisplayName("Pruebas para la clase Pedido")
19
20 public class PedidoTest {
21
22     //Atributos
23     private Pedido pedido;
24
25     //Definición de BeforeEach
26
27     @BeforeEach
28     public void setUp() {
29         int numeroPedido = 00001;
30         int id = 204;
31         String cliente = "Sofia";
32         String dir = "Cl 4 NO605";
33         ArrayList<Producto> itemsPedido = new ArrayList<>();
34         int precioneto = 0;
35         int preciototal = 0;
36         int precioiva = 0;
37
38         pedido = new Pedido(numeroPedido, id, cliente, dir, itemsPedido, precioneto, preciototal, precioiva);
39     }
40
41     //Pruebas de métodos
42
43     // Método 1
44

```

Element	Coverage	Covered Instru...	Missed Instruct...	Total Instructio...
Taller6-Dpoo	25,7 %	750	2.166	2.916
TALLER2-DPOO2023-main/TALLER2	18,8 %	407	1.758	2.165
Procesamiento	0,0 %	0	820	820
CONSOLA	0,0 %	0	674	674
MODELO	60,7 %	407	264	671
ProductoAjustado.java	0,0 %	0	101	101
Combo.java	0,0 %	0	77	77
ProductoMenu.java	45,7 %	21	25	46
Ingrediente.java	0,0 %	0	21	21
IngredienteRepetidoException.java	0,0 %	0	16	16
ProductoRepetidoException.java	0,0 %	0	16	16
HamburguesaException.java	0,0 %	0	4	4
PedidoValorSuperiorException.java	80,0 %	16	4	20
Pedido.java	100,0 %	370	0	370
Pedido	100,0 %	370	0	370
Test/Source	45,7 %	343	408	751

Combo- ComboTest cobertura 100%

```

1 package test;
2 import MODELO.Combo;
11
12 @DisplayName("Pruebas para la clase Combo ☺")
13
14 public class ComboTest {
15
16     //Atributos
17     private Combo todoterreno;
18
19     //Definición de BeforeEach
20 @BeforeEach
21     public void setUp() {
22         // Crear algunos productos para el combo
23         ProductoMenu producto1 = new ProductoMenu("Hamburguesa", 1000);
24         ProductoMenu producto2 = new ProductoMenu("Papas", 500);
25         ProductoMenu producto3 = new ProductoMenu("Bebida", 300);
26
27         // Crear un combo
28         todoterreno = new Combo("todoterreno",
29             new ArrayList<ProductoMenu>() {
30                 {
31                     add(producto1);
32                     add(producto2);
33                     add(producto3);
34                 }
35             });
36
37     //Pruebas de métodos
38
39     // Método 1
40 @Test
41 @DisplayName("Pruebas para el método getItems()")

```

ComboTest (20/05/2023 12:03:15 p. m.)

Element	Coverage	Covered Instru...	Missed Instruct...	Total Instructio...
▼ Taller6-Dpoo	8,2 %	240	2.676	2.916
▼ TALLER2-DPOO2023-main/TALLER2	4,4 %	95	2.070	2.165
> Procesamiento	0,0 %	0	820	820
> CONSOLA	0,0 %	0	674	674
▼ MODELO	14,2 %	95	576	671
> Pedido.java	0,0 %	0	370	370
> ProductoAjustado.java	0,0 %	0	101	101
> ProductoMenu.java	39,1 %	18	28	46
> Ingrediente.java	0,0 %	0	21	21
> PedidoValorSuperiorExceptio	0,0 %	0	20	20
> IngredienteRepetidoExceptio	0,0 %	0	16	16
> ProductoRepetidoException.j	0,0 %	0	16	16
> HamburguesaException.java	0,0 %	0	4	4
▼ Combo.java	100,0 %	77	0	77
> Combo	100,0 %	77	0	77
> Test/Source	19,3 %	145	606	751

ProductoMenu - ProductoMenuTest cobertura

```

1 package test;
2
3 import static org.junit.jupiter.api.Assertions.*;
12
13 @DisplayName("Pruebas para la clase ProductoMenu")
14
15 public class ProductoMenuTest {
16
17     //Atributos
18     private ProductoMenu producto;
19
20     //Definición de BeforeEach
21 @BeforeEach
22     public void setUp() {
23         producto = new ProductoMenu("Hamburguesa", 1000);
24     }
25
26     //Pruebas de métodos
27
28     // Método 1
29
30
31 @Test
32 @DisplayName("Pruebas para el método getNombre()")
33     public void testGetNombre() {
34         assertNotNull(producto, "Error: ProductoMenu no se creó correctamente");
35         assertEquals("Hamburguesa", producto.getNombre(), "El nombre del producto no corresponde");
36     }
37
38
39     // Método 2

```

ProductoMenuTest (20/05/2023 12:04:18 p. m.)

Element	Coverage	Covered Instru...	Missed Instruct...	Total Instructio...
▼ Taller6-Dpoo	4,0 %	118	2.798	2.916
▼ TALLER2-DPOO2023-main/TALLER2	1,9 %	42	2.123	2.165
> Procesamiento	0,0 %	0	820	820
> CONSOLA	0,0 %	0	674	674
▼ MODELO	6,3 %	42	629	671
> Pedido.java	0,0 %	0	370	370
> ProductoAjustado.java	0,0 %	0	101	101
> Combo.java	0,0 %	0	77	77
> Ingrediente.java	0,0 %	0	21	21
> PedidoValorSuperiorExceptio	0,0 %	0	20	20
> IngredienteRepetidoExceptio	0,0 %	0	16	16
> ProductoRepetidoException.j	0,0 %	0	16	16
> HamburguesaException.java	0,0 %	0	4	4
▼ ProductoMenu.java	91,3 %	42	4	46
> ProductoMenu	91,3 %	42	4	46
> Test/Source	10,1 %	76	675	751

PRUEBAS DE CLASES JUnit:

Project Explorer JUnit ×

Finished after 0,089 seconds

Runs: 3/3 Errors: 0 Failures: 0

- Pruebas para la clase ProductoAjustado [Runner: JUnit 5] (0,001 s)
 - Pruebas para el método agregar() (0,001 s)
 - Pruebas para el método getPrecio() y generarTextoFactura() (0,000 s)
 - Pruebas para el método quitar() (0,000 s)

Failure Trace

Project Explorer JUnit ×

Finished after 0,119 seconds

Runs: 6/6 Errors: 0 Failures: 0

- Pruebas para la clase ProductoMenu [Runner: JUnit 5] (0,000 s)
 - Pruebas para el método generarTextoFactura() (0,000 s)
 - Pruebas para el método getCalorias() (0,000 s)
 - Pruebas para el método getNombre() (0,000 s)
 - Pruebas para el método getPrecio() (0,000 s)

Failure Trace

Project Explorer JUnit ×

Finished after 0,073 seconds

Runs: 6/6 Errors: 0 Failures: 0

- Pruebas para la clase Combo [Runner: JUnit 5] (0,000 s)
 - Pruebas para el método getDescuento() (0,000 s)
 - Pruebas para el método getItems() (0,000 s)
 - Pruebas para el método getPrecio() y generarTextoFactura() (0,000 s)
 - Pruebas para el método agregarItemACombo() (0,000 s)
 - Pruebas para el método getCalorias() (0,000 s)
 - Pruebas para el método GetNombre() (0,000 s)

Failure Trace

Project Explorer JUnit ×

Finished after 0,132 seconds

Runs: 12/12 Errors: 0 Failures: 0

- Pruebas para la clase Pedido [Runner: JUnit 5] (0,001 s)
 - Primera prueba para el método getPrecioNetoPedido() (0,000 s)
 - Segunda prueba para el método agregarProducto(), incluye PedidoValorSuperior
 - Pruebas para el método getId() (0,000 s)
 - Prueba para el método generarFacturaImprimir() (0,000 s)
 - Prueba para el método guardar() (0,000 s)
 - testGenerarFacturaImprimir() (0,000 s)
 - Prueba para el método getPrecioIvaPedido() (0,000 s)
 - Pruebas para el método getPedido() (0,000 s)
 - Prueba para el método getItemsString() (0,000 s)
 - Primera prueba para el método agregarProducto() (0,000 s)