

## Taller 6

La representación Given-When-Then es un enfoque estructurado para diseñar escenarios de prueba y especificar las acciones y los resultados esperados de esas pruebas. El "Given" describe el contexto en el que se lleva a cabo la prueba, el "When" describe la acción que se está probando y el "Then" describe el resultado esperado de la acción.

### ProductoMenuTest

Escenario: El usuario solicita la factura utilizando el método generarTextoFactura():

Given: Dado que se ha creado un objeto ProductoMenu con nombre "costeña", precio 20000 y 90 calorías.

When: Cuando se llama al método generarTextoFactura() en el objeto productoMenu.

Then: Entonces el resultado esperado del método generarTextoFactura() debe ser "Costeña \$20000 90".

Método: assertEquals

Escenario: El usuario solicita la conocer la cantidad de calorías utilizando el método getCalorias():

Given: Dado que se pide una hamburguesa "costeña", de precio 20000 y con 90 calorías.

When: Cuando se llama al método getCalorias() en el objeto productoMenu.

Then: Entonces el resultado esperado del método getCalorias() debe ser 90.

Método: assertEquals

### ProductoAjustado Test

Escenario: El usuario desea conocer el precio del producto ajustado

Given:

- Dado que se ha creado un objeto ProductoMenu base con nombre "corral queso", precio 16000 y 47 calorías.
- Dado que se ha creado un objeto ProductoAjustado utilizando el objeto ProductoMenu base.

When:

- Cuando se crea un objeto Ingrediente con nombre "lechuga", precio 1000 y 20 calorías.
- Cuando se agrega el Ingrediente al objeto productoAjustado.
- Cuando se llama al método getPrecio() en el objeto productoAjustado.

Then: Entonces el resultado esperado del método getPrecio() debe ser la suma del precio base y el precio del ingrediente: 16000 + 1000.

Método: assertEquals

Escenario: Agregar un ingrediente al producto ajustado

Given:

- Dado que se ha creado un objeto ProductoMenu base con nombre "corral queso", precio 16000 y 47 calorías.
- Dado que se ha creado un objeto ProductoAjustado utilizando el objeto ProductoMenu base.

When:

- Cuando se crea un objeto Ingrediente con nombre "lechuga", precio 1000 y 20 calorías.
- Cuando se agrega el Ingrediente al objeto productoAjustado.

Then: Entonces se espera que el objeto productoAjustado contenga el Ingrediente agregado.

Método: assertTrue

Escenario: Eliminar un ingrediente del producto ajustado

Given:

- Dado que se ha creado un objeto ProductoMenu base con nombre "corral queso", precio 16000 y 47 calorías.
- Dado que se ha creado un objeto ProductoAjustado utilizando el objeto ProductoMenu base.
- Dado que se ha agregado un objeto Ingrediente con nombre "lechuga", precio 1000 y 20 calorías al objeto productoAjustado.

When: Cuando se elimina el Ingrediente en la posición 0 del objeto productoAjustado.

Then: Entonces se espera que el objeto productoAjustado no contenga el Ingrediente eliminado.

Método: assertFalse

Escenario: Generar la factura

Given:

- Dado que se ha creado un objeto ProductoMenu base con nombre "corral queso", precio 16000 y 47 calorías.
- Dado que se ha creado un objeto ProductoAjustado utilizando el objeto ProductoMenu base.
- Dado que se ha agregado un objeto Ingrediente con nombre "lechuga", precio 1000 y 20 calorías al objeto productoAjustado.

When: Cuando se llama al método generarTextoFactura() en el objeto productoAjustado.

Then: Entonces el resultado esperado del método generarTextoFactura() debe ser "Corral queso \$17000 67".

Método: assertEquals

Escenario: El usuario consulta la cantidad de calorías

Given:

- Dado que se ha creado un objeto ProductoMenu base con nombre "corral queso", precio 16000 y 47 calorías.
- Dado que se ha creado un objeto ProductoAjustado utilizando el objeto ProductoMenu base.
- Dado que se ha agregado un objeto Ingrediente con nombre "lechuga", precio 1000 y 20 calorías al objeto productoAjustado.

When: Cuando se llama al método getCalorias() en el objeto productoAjustado.

Then: Entonces el resultado esperado del método getCalorias() debe ser la suma de las calorías base y las calorías de los ingredientes:  $47 + 20 = 67$ .

Método: assertEquals

### **PedidoTest**

Escenario: El usuario quiere conocer la cantidad de pedidos creados

Given: Dado que no se han creado pedidos previamente.

When: Cuando se llama al método getNumeroPedidos().

Then: Entonces el resultado esperado del método getNumeroPedidos() debe ser 0, indicando que no hay pedidos registrados.

Método: assertEquals

Escenario: El usuario quiere conocer el id del pedido

Given: Dado que se ha creado un objeto Pedido con nombre de cliente "Carol Florido" y dirección "Calle 138".

When: Cuando se llama al método getIdPedido() en el objeto pedido.

Then: Entonces el resultado esperado del método getIdPedido() debe ser un número mayor o igual a 100, indicando que se ha asignado correctamente un identificador único al pedido.

Método: assertEquals

Escenario: El usuario quiere agregar un producto

Given:

- Dado un objeto Pedido creado.
- Dado un objeto ProductoMenu llamado producto con nombre "wrap de lomo", precio 22000 y 97 calorías.

When: Cuando se llama al método agregarProducto(producto) en el objeto pedido.

Then:

- Entonces se espera que el producto sea agregado correctamente al pedido.
- Se espera que al llamar al método getProductospedido(pedido), la lista de productos del pedido contenga el producto agregado.

Método: assertTrue

Escenario: Se quiere agregar un producto, dado que hay una restricción de precio

Given:

- Dado un objeto Pedido creado.
- Dado un objeto ProductoMenu llamado producto con nombre "wrap de lomo", precio 22000000 y 97 calorías.

When: Cuando se llama al método agregarProducto(producto) en el objeto pedido.

Then: Entonces se espera que se lance una excepción del tipo PrecioException, indicando que el precio del producto supera el límite permitido.

Método: (expected = PrecioException.class)

Escenario: El usuario solicita conocer el precio del pedido

Given:

- Dado un objeto Pedido creado.
- Dado un objeto ProductoMenu llamado producto1 con precio 22000 y 97 calorías.
- Dado un objeto ProductoMenu llamado producto2 con precio 20900 y 53 calorías.

When:

- Cuando se llama al método agregarProducto(producto1) y luego al método agregarProducto(producto2) en el objeto pedido.
- Cuando se llama al método getPrecioNetoPedido() en el objeto pedido.

Then: Entonces el resultado esperado del método getPrecioNetoPedido() debe ser la suma de los precios de todos los productos agregados al pedido:  $22000 + 20900 = 42900$ .

Método: assertEquals

Escenario: El usuario solicita conocer las calorías

Given:

- Dado un objeto Pedido creado.
- Dado un objeto ProductoMenu llamado producto1 con precio 22000 y 97 calorías.
- Dado un objeto ProductoMenu llamado producto2 con precio 20900 y 53 calorías.

When:

- Cuando se llama al método `agregarProducto(producto1)` y luego al método `agregarProducto(producto2)` en el objeto pedido.
- Cuando se llama al método `getCalorias()` en el objeto pedido.

Then: Entonces el resultado esperado del método `getCalorias()` debe ser la suma de las calorías de todos los productos agregados al pedido:  $97 + 53 = 150$ .

Método: `assertEquals`

Escenario: El usuario solicita conocer el precio total del pedido

Given:

- Dado un objeto Pedido creado.
- Dado un objeto ProductoMenu llamado `producto1` con precio 22000 y 97 calorías.
- Dado un objeto ProductoMenu llamado `producto2` con precio 20900 y 53 calorías.

When:

- Cuando se llama al método `agregarProducto(producto1)` y luego al método `agregarProducto(producto2)` en el objeto pedido.
- Cuando se llama al método `getPrecioTotalPedido()` en el objeto pedido.

Then:

Entonces el resultado esperado del método `getPrecioTotalPedido()` debe ser la suma del precio neto y el precio IVA del pedido.

El cálculo del precio neto es:  $22000 + 20900 = 42900$ .

El cálculo del precio IVA es:  $42900 * 0.19 = 8151$ .

Por lo tanto, el resultado esperado del método `getPrecioTotalPedido()` debe ser:  $42900 + 8151 = 51051$ .

Método: `assertEquals`

Escenario: El usuario solicita conocer el precio del IVA

Given:

- Dado un objeto Pedido creado.
- Dado un objeto ProductoMenu llamado `producto1` con precio 22000 y 97 calorías.
- Dado un objeto ProductoMenu llamado `producto2` con precio 20900 y 53 calorías.

When:

- Cuando se llama al método `agregarProducto(producto1)` y luego al método `agregarProducto(producto2)` en el objeto pedido.
- Cuando se llama al método `getPrecioIVAPedido()` en el objeto pedido.

Then:

Entonces el resultado esperado del método `getPrecioIVAPedido()` debe ser el cálculo del precio neto multiplicado por el porcentaje de IVA.

El cálculo del precio neto es:  $22000 + 20900 = 42900$ .

El cálculo del precio IVA es:  $42900 * 0.19 = 8151$ .

Por lo tanto, el resultado esperado del método `getPrecioIVAPedido()` debe ser 8151.

Método: `assertEquals`

Escenario: El usuario quiere generar la factura

Given:

- Dado un objeto Pedido creado.
- Dado un objeto ProductoMenu llamado `producto1` con nombre "wrap de lomo", precio 22000 y 97 calorías.

When:

- Cuando se llama al método `agregarProducto(producto1)` en el objeto pedido.
- Cuando se llama al método `generarTextoFactura()` en el objeto pedido.

Then:

Entonces se espera que el método `generarTextoFactura()` retorne un texto formateado que representa la factura del pedido.

El texto de la factura debe contener los detalles del cliente, los precios, las calorías y los productos comprados.

Se espera que el texto de la factura sea igual al texto esperado.

Método: `assertEquals`

Escenario: El usuario quiere guardar la factura

Given:

- Dado un objeto Pedido creado.
- Dado un objeto ProductoMenu llamado `producto1` con nombre "wrap de lomo", precio 22000 y 97 calorías.
- Dado un archivo `factura.txt` existente.

When:

- Cuando se llama al método `agregarProducto(producto1)` en el objeto pedido.
- Cuando se llama al método `guardarFactura("factura.txt")` en el objeto pedido.

Then:

Entonces se espera que el método `guardarFactura("factura.txt")` guarde correctamente la factura del pedido en el archivo `factura.txt`.

El archivo factura.txt debe contener los detalles del cliente, los precios, las calorías y los productos comprados en un formato legible.

Se espera que el archivo factura.txt sea creado y contenga la información esperada.

\*Nota: Para verificar el resultado esperado, se debe inspeccionar el contenido del archivo factura.txt y compararlo con el formato y contenido esperado de la factura generada.

Método: assertEquals

### **ComboTest**

Escenario: El usuario quiere agregar un item al combo

Given:

- Dado un objeto Combo creado.
- Dado un objeto ProductoMenu llamado productoMenu con nombre "corral queso", precio 150 y 500 calorías.

When:

Cuando se llama al método agregarItemCombo(productoMenu) en el objeto combo.

Then:

Entonces se espera que el método agregarItemCombo(productoMenu) agregue correctamente el productoMenu al combo.

Se espera que el tamaño de la lista itemsCombo del combo sea igual a 1.

Se espera que el primer elemento de la lista itemsCombo sea igual a productoMenu.

Método: assertEquals

Escenario: El usuario quiere generar el texto de la factura

Given:

- Dado un objeto Combo creado.
- Dado un objeto ProductoMenu llamado productoMenu1 con nombre "mexicana", precio 22000 y 98 calorías.
- Dado un objeto ProductoMenu llamado productoMenu2 con nombre "criolla", precio 22000 y 56 calorías.

When:

- Cuando se llama al método agregarItemCombo(productoMenu1) en el objeto combo.
- Cuando se llama al método agregarItemCombo(productoMenu2) en el objeto combo.
- Cuando se llama al método generarTextoFactura() en el objeto combo.

Then:

Entonces se espera que el método generarTextoFactura() genere correctamente el texto de la factura del combo.

Se espera que el texto de la factura contenga el nombre del combo, el precio total y las calorías totales en el formato esperado.

Método: `assertEquals`

Escenario: El usuario quiere conocer las calorías

Given:

- Dado un objeto Combo creado.
- Dado un objeto ProductoMenu llamado `productoMenu1` con nombre "mexicana", precio 22000 y 98 calorías.
- Dado un objeto ProductoMenu llamado `productoMenu2` con nombre "criolla", precio 22000 y 56 calorías.

When:

- Cuando se llama al método `agregarItemCombo(productoMenu1)` en el objeto combo.
- Cuando se llama al método `agregarItemCombo(productoMenu2)` en el objeto combo.
- Cuando se llama al método `getCalorias()` en el objeto combo.

Then:

Entonces se espera que el método `getCalorias()` calcule correctamente las calorías totales del combo.

Se espera que el resultado de `getCalorias()` sea igual a la suma de las calorías de los productos del combo.

Nota: Para verificar el resultado esperado, se deben comparar los valores calculados con los valores esperados utilizando la aserción adecuada.

Método: `assertEquals`