**Formato de escenarios y casos de uso**

**Configuración de los Escenarios**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setupStage1 | ProductTest | Un objeto de la clase Product con productName=”PS5”, description=”VideoGame”, price=2000000.0, quantityAvailable=4, category=”Toys and Games”, purchasedNumber=2 |
| setupStage1 | OrderTest |  |
| setupStage1 | ProductListTest | Un objeto de la clase ProductList sin productos |
| setupStage2 | ProductListTest | Un objeto de la clase ProductListo con tres objetos de la clase Product:   * Primero con: productName=”PS5”, description=”VideoGame”, price=2000000.0, quantityAvailable=4, category=”Toys and Games”, purchasedNumber=2 * Segundo con: productName=”Hamburguesa”, description=”Pan con carner”, price=25000.0, quantityAvailable=50, category=”Foods and Drinks”, purchasedNumber=3 * Tercero con: productName=”Camiseta”, description=”Camiseta oversize”, price=70000.0, quantityAvailable=20, category=”Clothes and\_accesories”, purchasedNumber=10 |
| setupStage1 | OrderListTest | Un objeto de la clase OrderList sin pedidos |
| setupStage2 | OrderListTest |  |

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
|  |  |  |
|  |  |  |
|  |  |  |

**Diseño de Casos de Prueba**

**RF2 – El programa debe permitir al usuario ingresar información sobre cada producto.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar que los métodos addProduct de la clase Controller funcionen correctamente, creando con éxito el objeto Product. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**RF3 – El programa debe permitir al usuario eliminar un producto del inventario.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar que el método deleteProduct de la clase ProductList funcionen correctamente, removiendo de la lista el objecto Product deseado. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| ProductList | deleteProduct | setupStage2 | nameProduct=”Hamburguesa” | La lista de productos ahora tiene 2 productos: PS5 y Camiseta. |
| Product | deleteProduct | setupStage2 | nameProduct=”Coca cola” | La lista de productos conserva sus 3 productos. Además, lanza una excepción y un mensaje de error. |

**RF4 – El programa debe permitir al usuario aumentar la cantidad de cada producto ya registrado.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar que el método addQuantityAvailable de la clase Product funcione correctamente, adicionando la cantidad pasada por parámetro a la cantidad actual del producto. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Product | addQuantityAvailable | setupStage1 | nameProducto=”PS5”, quantity2Add=4 | La cantidad del producto se ha actualizado a 8. |
| Product | addQuantityAvailable | setupStage1 | nameProducto=”Xbox”, quantity2Add=3 | Lanza una Excepción y el mensaje dice que el producto no ha sido registrado. |
| Product | addQuantityAvailable | setupStage1 | nameProducto=”PS5”, quantity2Add=-4 | La cantidad del producto se mantiene en 4. Adicionalmente, lanza una Excepción y el mensaje dice que no se pueden usar números negativos. |
| Product | addQuantityAvailable | setupStage1 | nameProducto=”PS5”, quantity2Add=2.5 | La cantidad del producto se mantiene en 4. Adicionalmente, lanza una Excepción y el mensaje dice que no se pueden usar números decimales. |

**RF5 – El programa debe permitir al usuario ingresar información sobre cada pedido.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar que los métodos addOrder de la clase Controller funcionen correctamente, creando con éxito el objeto Order. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**RF6 – El programa debe permitir disminuir las cantidades del inventario al momento de generar un pedido.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar que el método subtractQuantityAvailable de la clase Product funcione correctamente, disminuyendo la cantidad pasada por parámetro a la cantidad actual del producto. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Product | subtractQuantityAvailable | setupStage1 | nameProducto=”PS5”, quantity2Subtract=4 | La cantidad del producto se ha actualizado a 0. |
| Product | subtractQuantityAvailable | setupStage1 | nameProducto=”PS5”, quantity2Subtract=7 | La cantidad del producto se mantiene en 4. Adicionalmente, lanza una Excepción y el mensaje dice que la cantidad a comprar es mayor que las unidades disponibles. |
| Product | subtractQuantityAvailable | setupStage1 | nameProducto=”Xbox”, quantity2Subtract=3 | Lanza una Excepción y el mensaje dice que el producto no ha sido registrado. |
| Product | subtractQuantityAvailable | setupStage1 | nameProducto=”PS5”, quantity2Subtract=-4 | La cantidad del producto se mantiene en 4. Adicionalmente, lanza una Excepción y el mensaje dice que no se pueden usar números negativos. |
| Product | subtractQuantityAvailable | setupStage1 | nameProducto=”PS5”, quantity2Subtract=2.5 | La cantidad del producto se mantiene en 4. Adicionalmente, lanza una Excepción y el mensaje dice que no se pueden usar números decimales. |

**RF7 – El programa debe permitir al usuario eliminar un pedido.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar que el método deleteOrder de la clase OrderList funcionen correctamente, removiendo de la lista el objecto Order deseado. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| ProductList | deleteProduct | setupStage2 | nameProduct=”Hamburguesa” | La lista de productos ahora tiene 2 productos: PS5 y Camiseta. |
| Product | deleteProduct | setupStage2 | nameProduct=”Coca cola” | La lista de productos conserva sus 3 productos. Además, lanza una excepción y un mensaje de error. |