**GRUPO DE TRABAJO**

* **Juan Felipe Jojoa Crespo A00382042**
* **Felipe Rojas Prado A00393918**
* **Juan Sebastian Gonzalez A00371810**

**Formato de escenarios y casos de uso**

**Configuración de los Escenarios**

| **Nombre** | **Clase** | **Escenario** |
| --- | --- | --- |
| setupPList | productTest | Product productList = new productList<>; |
| setupFilledProductList | searchTest | Product productList = new productList<>;  Product p1 = new Product(orange, orange citrus fruit, 1000, 5, Fruit);  Product p1 = new Product(apple, red fruit, 1000, 5, Fruit);  Product p1 = new Product(beef, cow meat, 10000, 5, Meat); |
| setupSaleList | saleTest | Sale sale = new Sale();  Product product = new Product("orange", "orange citrus fruit", 1000, 10, "Fruit");  sale.addProduct(product, 5);  int expectedStock = 5;  int actualStock = product.getStock(); |

| **Nombre** | **Clase** | **Escenario** |
| --- | --- | --- |
| setupOrderList | newOrderTest | PurchaseOrder purchaseOrder = new PurchaseOrder();  Product product1 = new Product("orange", "orange citrus fruit", 1000, 10, "Fruit");  Product product2 = new Product("apple", "red fruit", 1000, 10, "Fruit");  purchaseOrder.addProduct(product1, 5);  purchaseOrder.addProduct(product2, 5);  int expectedTotalPrice = 10000;  int actualTotalPrice = purchaseOrder.getTotalPrice(); |
| setupSearchOrderList | searchOrderTest | PurchaseOrder purchaseOrder = new PurchaseOrder();  purchaseOrder.setNumber("1234");  Product product1 = new Product("orange", "orange citrus fruit", 1000, 10, "Fruit");  Product product2 = new Product("apple", "red fruit", 1000, 10, "Fruit");  purchaseOrder.addProduct(product1, 5);  purchaseOrder.addProduct(product2, 5);  PurchaseOrder expectedPurchaseOrder = purchaseOrder;  PurchaseOrder actualPurchaseOrder = system.searchPurchaseOrder("1234"); |
| setupJSONTest | jsonTest | Inventory inventory = new Inventory();  Product product1 = new Product("orange", "orange citrus fruit", 1000, 10, "Fruit");  Product product2 = new Product("apple", "red fruit", 1000, 10, "Fruit");  inventory.addProduct(product1);  inventory.addProduct(product2);  String expectedJsonString = "{\"products\":[{\"name\":\"orange\",\"description\":\"orange citrus fruit\",\"price\":1000,\"stock\":10,\"category\":\"Fruit\"},{\"name\":\"apple\",\"description\":\"red fruit\",\"price\":1000,\"stock\":10,\"category\":\"Fruit\"}]}";  String actualJsonString = inventory.exportToJsonString(); |

| **Nombre** | **Clase** | **Escenario** |
| --- | --- | --- |
| setupListOfProducts | createListTest | ProductList productList = new ProductList();  Product product1 = new Product("orange", "orange citrus fruit", 1000, 10, "Fruit");  Product product2 = new Product("apple", "red fruit", 1000, 10, "Fruit");  productList.addProduct(product1);  productList.addProduct(product2);  int expectedSize = 2;  int actualSize = productList.getSize(); |

…

\* El nombre de los escenarios puede ser setupStage1, setupStage2, etc.

\* La clase es la clase de testing correspondiente al modelo donde acontece el escenario. Por ejemplo si usted está probando User, clase será UserTest.

\* El escenario es la descripción de las condiciones iniciales del escenario.

**Diseño de Casos de Prueba**

| **Objetivo de la Prueba:** Verify that the system allows adding a new product to the store. | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Controller | addStoreProduct | setupPList | User inputs (productName,productDescription, productPrice, productQuantity, productCategory) without failing conditions. | Created product, with inputs as attributes. |
| Controller | addStoreProduct | setupPList | User inputs (productName,productDescription, productPrice, productQuantity, productCategory) with failing conditions | Error uncompleted info, Error wrong info, Error values can´t be null, Error value cant be less than 1. |
|  |  |  |  |  |

| **Objetivo de la Prueba:** Verify that the system allows you to search for a product in the list of products in the store. | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Controller | searchProduct | setupFilledProductList | User inputs(productName, productID, productCategory, productPrice, productSales) Without failing conditions. | Product information is displayed with its name, ID, price, sales, category, and quantity available. |
| Controller | searchProduct | setupFilledProductList | User inputs  (productName, productID, productCategory, productPrice, productSales) With failing conditions. | A message is displayed indicating that the product does not exist in the store. |
|  |  |  |  |  |

| **Objetivo de la Prueba:** Verify that the system updates the stock of a product after a sale. | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Controller | sellProduct | setupSaleList | User inputs(productName, productID, productCategory, productPrice) Without failing conditions. | The stock of the product is reduced in units. |
| Controller | sellProduct | setupSaleList | User inputs(productName, productID, productCategory, productPrice) With failing conditions. | An error message is displayed indicating that the product does not exist and the stock is not updated. |
|  |  |  |  |  |

| **Objetivo de la Prueba:** Verify that the system allows you to create a new purchase order. | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Controller | addPurchaseOrder | setupOrderList | User inputs(clientName, productList) Without failing conditions. | The purchase order is successfully created in the system and the inventory is updated with the corresponding quantities. |
| Controller | addPurchaseOrder | setupOrderList | User inputs(clientName, productList) With failing conditions. | An error message is displayed indicating that the product does not exist in inventory and the purchase order is not created. |
|  |  |  |  |  |

| **Objetivo de la Prueba:** Verify that the system can search for a purchase order. | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Controller | searchOrderByNumber | setupSearchOrderList | User inputs(orderID, clientName, totalPrice, orderDate) Without failing conditions. | The system returns the information of the purchase order corresponding to the number entered. |
| Controller | searchOrderByNumber | setupSearchOrderList | User inputs(orderID, clientName, totalPrice, orderDate) With failing conditions. | The system returns a message indicating that the purchase order was not found in the system. |
|  |  |  |  |  |

| **Objetivo de la Prueba:** Verify that the system can export inventory data in JSON format. | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Controller | exportToJson | setupJSONTest | User inputs(listProduct) Without failing conditions. | The JSON file successfully created at specified location. |
| Controller | exportToJson | setupJSONTest | User inputs(listProduct) With failing conditions. | The JSON file cannot be created correctly because the list is empty. |
|  |  |  |  |  |

…

| **Objetivo de la Prueba:** Verify that the system can create a list of products. | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Controller | createProductList | setupListOfProducts | User inputs(productID, quantity) Without failing conditions. | The system creates a product list successfully. |
| Controller | createProductList | setupListOfProducts | User inputs(productID, quantity) With failing conditions. | The system does not create a list of products due to errors in the inputs. |
|  |  |  |  |  |

…

\* Una prueba se compone de un conjunto de casos de prueba.

\* Cada fila representa un ***caso de prueba*** diferente

\* En el objetivo de la prueba debe escribir una descripción sobre qué es lo que específicamente está probando del modelo del programa.

\* La clase es la clase del modelo que está siendo puesto a prueba.

\* El método es específicamente el método de la clase que está siendo puesto a prueba.

\* El escenario se refiere al nombre del escenario que usted definió. Todos los casos de prueba corresponden a escenarios.

\* Los valores de entrada son valores que entran al método puesto a prueba.

\* El resultado esperado es lo que se espera que suceda luego de ejecutar el método.