**Format scenarios and use cases.**

**Scenario Configuration**

|  |  |  |
| --- | --- | --- |
| **Name** | **Class** | **Scenario** |
| SetupStage1 | StoreDataTest | An StoreData object with no products or orders |
| SetupStage2 | StoreDataTest | An StoreData object class that contains two Objects of the Product class and an Object of the Order class.   * First Product:   + name= Sword   + description = It is a sword with a sharp edge   + price = 110.99   + quantity = 100   + category = Weapons   + timesPurchased = 8 * Second Product:   + name= Rice   + description = 1kg of rice   + price = 7.30   + quantity = 10   + category = Supermarket   + timesPurchased = 1 * Order:   + buyerName= Ana Maria   + productList = Sword, Sword, Sword, Sword, Sword, Sword, Sword, Rice   + totalPrice = 810.227   + date = Calendar.getInstance() |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Class** | **Scenario** | |
| SetupStage1 | ProductTest | An empty StoreManager object | |
| SetupStage2 | ProductTest | An StoreManager object that contains two Objects of the Product class. | |
| First Product:  name= Sword  description = It is a sword with a sharp edge  price = 110.99  quantity = 100  category = Weapons  timesPurchased = 8 | Second Product:  name= Rice  description = 1kg of rice  price = 7.30  quantity = 10  category = Supermarket  timesPurchased = 1 |

|  |  |  |
| --- | --- | --- |
| **Name** | **Class** | **Scenario** |
| SetupStage1 | OrderTest | An empty StoreManager object with one product with:   * name= Sword * description = It is a sword with a sharp edge * price = 110.99 * quantity = 100 * category = Weapons * timesPurchased = 8 |
| SetupStage2 | OrderTest | An StoreManager object that contains two Objects of the Product class and an Object of the Order class.   * First Product:   + name= Sword   + description = It is a sword with a sharp edge   + price = 110.99   + quantity = 100   + category = Weapons   + timesPurchased = 1 * Second Product:   + name= Rice   + description = 1kg of rice   + price = 7.30   + quantity = 10   + category = Supermarket   + timesPurchased = 1 * Order:   + buyerName= Ana Maria   + productList = Sword, Rice   + totalPrice = 118.29   + date = Calendar.getInstance() |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Class** | **Scenario** | |
| SetupStage1 | PersitenceTest | An empty StoreManager object | |
| SetupStage3 | PersitenceTest | An StoreManager object that contains two Objects of the Product class. | |
| First Product:  name= Sword  description = It is a sword with a sharp edge  price = 110.99  quantity = 90  category = Weapons  timesPurchased = 10 | Second Product:  name= Rice  description = 1kg of rice  price = 7.30  quantity = 20  category = Supermarket  timesPurchased = 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Class** | **Scenario** | |
| SetupStage1 | SearchEngineTest | An empty StoreManager object | |
| SetupStage2 | SearchEngineTest | An StoreManager object with 20 imported objects of the Product class and 20 imported objects of the Order class: The information of each Object is imported in JSON format from the following address “…/data/example\_data.json” | |
| SetupStage3 | SearchEngineTest | An StoreManager object that contains two Objects of the Product class. | |
| First Product:  name= Sword  description = It is a sword with a sharp edge  price = 110.99  quantity = 90  category = Weapons  timesPurchased = 10 | Second Product:  name= Rice  description = 1kg of rice  price = 7.30  quantity = 20  category = Supermarket  timesPurchased = 1 |

**Test Case Design**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objective of the Test:** Verify that the addProduct and removeProduct methods of the StoreManager class are working correctly, adding two Product objects | | | | |
| **Class** | **Method** | **Scenario** | **Input Values** | **Expected result** |
| ProductTest | addProduct | setupStage1 | Products attributes | The product is created correctly |
| ProductTest | addProduct | SetupStage2 | Product attributes,  Product2 price below zero | When trying to add a product throws **UnvalidPriceException** |
| ProductTest | addProduct | SetupStage2 | Product attributes,  Product1 name repeated | When trying to add a product throws **AlreadyAddedException** |
| ProductTest | removeProduct | SetupStage1 | “Rice” | The product was successfully removed |
| ProductTest | removeProduct | SetupStage2 | “Ring” | When trying to remove the product throws **InvalidReferenceException** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objective of the Test:** Verify that the addProduct and removeProduct methods of the StoreManager class are working correctly, adding two Product objects | | | | |
| **Class** | **Method** | **Scenario** | **Input Values** | **Expected result** |
| OrderTest | addOrder | SetupStage1 | Order attributes | The order is created correctly |
| OrderTest | addOrder | SetupStage1 | Order attributes  Order has no products | When trying to order without any product throws **EmptyOrderException, the order is not created** |
| OrderTest | removeOrder | SetupStage2 | buyerName and orderSelection | The order was successfully removed |
| OrderTest | removeOrder | SetupStage2 | buyerName and orderSelection | When trying to remove the order throws **InvalidReferenceException** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objective of the Test:** Verify that the updateProductQuantities and calculateTotalPrice methods of the StoreManager class are working correctly, adding two Product objects | | | | |
| **Class** | **Method** | **Scenario** | **Input Values** | **Expected result** |
| ProductTest | updateProductQuantities | SetupStage2 | Select Product  Nwquant = 1, option 2 | New product Quantity must be quantity+2 |
| ProductTest | updateProductQuantities | SetupStage2 | Select Product  Nwquant = 10, option 3 | New product Quantity must be quantity+10 |
| OrderTest | calculateTotalPrice | SetupStage1 | ProductList of order 1 | The total price should be the sum of the prices of each product |
| OrderTest | calculateTotalPrice | SetupStage1 | None | Total price must be greater than 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objective of the Test:** Verify that the data management in JSON format of the program works correctly | | | | |
| **Class** | **Method** | **Scenario** | **Input Values** | **Expected result** |
| PersistenceTest | exportData | SetupStage2 |  |  |
| PersistenceTest | importData | SetupStage2 |  |  |
| PersistenceTest | exportData | SetupStage1 | Products ArrayList product = pollo …  Order = Pedro Name of the file = ExampleDataTest | The ExampleDataTest File must be created in …/data/ExampleDataTest with all previous information |
| PersistenceTest | importData | SetupStage1 | fileName= JanethFiles | The JanethFiles.json does not exist in …/data/, program must.  Return false.  It must check that storeManagerProducts are 0 |