**Formato de escenarios y casos de uso**

**Configuración de los Escenarios – Inventory**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| SetUp1 | InventoryTest | Inicializa el objecto de Inventory |
| SetUp2 | InventoryTest | Crea el objeto de Inventory y añade 5 productos.  new Product ("Jabon", "Para pieles limpias", 2500, 10, 8)  new Product ("Shampoo", "Para cabellos sedosos", 12500, 10, 8)  new Product ("Mascarilla", "Para caras sedosas", 10000, 10, 8)  new Product ("Exfoliante", "Para pieles sedosas", 2500, 10, 8)  new Product ("Crema", "Para caras sedosas", 15000, 10, 8) |

**Configuración de los Escenarios – SearchEngine**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| SetUp1 | ProductSearchEngineTest | Crea el objeto de Inventory y añade 5 productos.  new Product ("Jabon", "Para pieles limpias", 2500, 10, 8)  new Product ("Shampoo", "Para cabellos sedosos", 12500, 10, 8)  new Product ("Mascarilla", "Para caras sedosas", 10000, 10, 8)  new Product ("Exfoliante", "Para pieles sedosas", 2500, 10, 8)  new Product ("Crema", "Para caras sedosas", 15000, 10, 8) |
| SetUp2 | ProductSearchEngineTest | Añade 6 productos extra al escenario #1.  new Product ("Mascarilla Arroz", "Para pieles sedosas", 2500, 10, 8)  new Product ("Masa de Arepas", "Para ricas arepas", 12500, 10, 8)  new Product ("Mascara Halloween", "Para asustar a todos", 10000, 10, 8)  new Product ("Crema coreana", "Para pieles sedosas", 2500, 10, 8)  new Product ("Pan Mariana", "Para ricas arepas", 12500, 10, 8)  new Product ("Cruz Cristiana", "Para asustar a todos", 10000, 10, 8) |
| SetUp2 | ProductSearchEngineTest | Inicializa el objeto de inventario y añade los siguientes productos:  new Product ("Jabon", "Para pieles sedosas", 100, 10, 8)  new Product ("Shampoo", "Para cabellos sedosos", 500, 10, 8)  new Product ("Mascarilla", "Para caras sedosas", 510, 10, 8)  new Product ("Exfoliante Arcilla", "Para pieles sedosas", 600, 10, 8)  new Product ("Crema", "Para pieles sedosas", 750, 10, 8)  new Product ("Crema Arawak", "Para pieles sedosas", 800, 10, 8)  new Product ("Crema Bonita", "Para pieles sedosas", 832, 10, 8)  new Product ("Crema Cicatrizante", "Para pieles sedosas", 910, 10, 8)  new Product ("Exfoliante Bakano", "Para pieles sedosas", 1024, 10, 8)  new Product ("Exfoliante Coco", "Para pieles sedosas", 2000, 10, 8) |

**Configuración de los Escenarios – Shop**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| SetUp1 | ShopTest | Inicializa el objeto Shop y le añade 3 productos.  new Product("Jabon", "Para pieles sedosas", 2500, 10, 8)  new Product ("Shampoo", "Para cabellos sedosos", 12500, 10, 8)  new Product ("Mascarilla", "Para caras sedosas", 10000, 10, 8) |
| setUp2 | ShopTest | Usa el setUp1 para inicializar el objeto Order con dos productos (Jabon y Shampoo) y la fecha actual. |

**Configuración de los Escenarios – OrderSearchEngine**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| SetUp1 | OrderSearchEngineTest | Crea el objeto de Shop, una lista de productos y añade 3 pedidos  new Product("Jabon", "Para pieles sedosas", 2500, 10, 8)  new Product ("Shampoo", "Para cabellos sedosos", 12500, 10, 8)  new Product ("Mascarilla", "Para caras sedosas", 10000, 10, 8)  new Order("Camilo De La Pava", LocalDateTime.now())  new Order("Juan Lopez", LocalDateTime.now())  new Order("Camilo Diaz", LocalDateTime.now()) |

**Diseño de Casos de Prueba - Inventory**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Probar que una serie de productos se añadan correctamente al inventario | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Inventory | addProduct | setUp1 | new Product ("Jabon", "Para pieles sedosas", 100, 10, 8)); | - Se agrega el objeto Product con nombre “Jabon” en la posicion 0 del ArrayList. |
| Inventory | addProduct | setUp1 | new Product ("Shampoo", "Para cabellos sedosos", 500, 10, 8)); | - Se agrega el objeto Product con nombre “Shampoo” en la posicion 1 del ArrayList. |
| Inventory | addProduct | setUp1 | new Product ("Mascarilla", "Para caras sedosas", 510, 10, 8)); | - Se agrega el objeto Product con nombre “Mascarilla” en la posicion 2 del ArrayList. |
| Inventory | getProductList | setUp1 |  | - Debe retornar 3 debido a que se agregaron 3 elementos. |
| Inventory | addProduct | setUp1 | new Product ("Mascarilla", "Para caras sedosas", 510, 10, 8). | - No debe agregar el producto debido a que ya existe. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que al comprar un producto, las unidades disponibles de este se decrementen. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Inventory | pruchaseProduct | SetUp1 | Id: "Shampoo", amountToDecrease: 5 | - El resulado debe ser 5, debido a que se restan 5 unidades del producto Shampoo. |
| Inventory | pruchaseProduct | SetUp1 | Id: "Jabon", amountToDecrease: 3 | - El resulado debe ser 7, debido a que se restan 3 unidades del producto Jabon. |
| Inventory | pruchaseProduct | SetUp1 | Id: "Exfoliante", amountToDecrease: 8 | - El resulado debe ser 2, debido a que se restan 8 unidades del producto Jabon. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que un producto se elimina correctamente del inventario. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Inventory | deleteProduct | SetUp1 | Id: "Shampoo" | - El objeto de resultado debe tener de nombre Shampoo, debido a que el producto a eliminar fue ese.  - El inventario debe quedar con 4 elementos. |
| Inventory | pruchaseProduct | SetUp1 | Id: “Exfoliante” | - El objeto de resultado debe tener de nombre Exfoliante, debido a que el producto a eliminar fue ese.  - El inventario debe quedar con 3 elementos. |
| Inventory | pruchaseProduct | SetUp1 | Id: "Jabon" | - El objeto de resultado debe tener de nombre Jabon, debido a que el producto a eliminar fue ese.  - El inventario debe quedar con 2 elementos. |
| Inventory | pruchaseProduct | SetUp1 | Id: "Crema" | - El objeto de resultado debe tener de nombre Crema, debido a que el producto a eliminar fue ese.  - El inventario debe quedar con 1 elemento. |

**Diseño de Casos de Prueba - Shop**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que una serie de productos se añadan correctamente al inventario del objeto Shop | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Shop | getInventory().getProductsList().size(): | setUp1 |  | - El inventario debe tener 3 productos. |
| Shop | getInventory().getByIndex(0).getName() | setUp1 |  | - El objeto en la posicion 0 del ArrayList de Product, debe tener como nombre “Jabon” |
| Shop | getInventory().getByIndex(1).getPrice() | setUp1 |  | - El objeto en la posicion 1 del ArrayList de Product, debe tener como precio $12500. |
| Shop | getInventory().getByIndex(2).getAmount() | setUp1 |  | - El producto agregado en la posicion 2 debe tener 10 unidades. |
| Shop | addProduct | setUp1 | new Product("Shampoo", "Para una cabello sedosa", 10000,10, 4) | - No debe agregar el producto debido a que ya existe un producto con ese mismo nombre. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que se crea un pedido exitosamente y que se decrementan las unidades disponibles de un producto al ser comprado. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Shop | createOrder | setUp2 | - amountProduct: 4  - amountProduct: 2 | - Crea una orden y se añade al ArrayList en la posicion 0  - La orden en la posicion 0 debe tener el nombre de “Camilo” |
| Shop | createOrder | setUp2 | - El producto “Jabon” debe tener 6 unidades disponibles (Se compraron 4)  - El producto “Shampoo” debe tener 8 unidades disponibles (Se compraron 2) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que se incrementan las unidades de un producto exitosamente. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Shop | increaseAmountOfProduct | setUp1 | - productIndex: 1  - amount: 15 | - Incrementa al objeto Product en la posicion 0, la cantidad de 15 unidades extra (25 en total) |
| Shop | increaseAmountOfProduct | setUp1 | - productIndex: 2  - amount: 8 | - Incrementa al objeto Product en la posicion 1, la cantidad de 8 unidades extra (18 en total) |
| Shop | increaseAmountOfProduct | setUp1 | - productIndex: 3  - amount: 3 | - Incrementa al objeto Product en la posicion 2, la cantidad de 3 unidades extra (13 en total) |

**Diseño de Casos de Prueba – ProductSearchEngine**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que el algoritmo de ordenamiento ordena correctamente. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| ProductSearchEngine | sortUsingAVariable | setUp1 | - sortVariable: “name” | - Ordena el arreglo de productos de forma ascendente y usando como variable de ordenamiento el nombre de los productos. |
| - order: true |
| ProductSearchEngine | getIndex | setUp1 | - productIndex: 0 | - El primer producto debe tener de nombre “Crema” |
| ProductSearchEngine | getIndex | setUp1 | - productIndex: 1 | - El segundo producto debe tener de nombre “Exfoliante” |
| ProductSearchEngine | getIndex | setUp1 | - productIndex: 2 | - El tercer producto debe tener de nombre “Jabon” |
| ProductSearchEngine | getIndex | setUp1 | - productIndex: 3 | - El cuarto producto debe tener de nombre “Mascarilla” |
| ProductSearchEngine | getIndex | setUp1 | - productIndex: 4 | - El quinto producto debe tener de nombre “Shampoo” |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que el algoritmo de ordenamiento ordena correctamente. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| ProductSearchEngine | sortUsingAVariable | setUp1 | - sortVariable: “price”  - order: true | - Ordena el arreglo de productos de forma ascendente y usando como variable de ordenamiento el precio de los productos. |
| ProductSearchEngine | getIndex | setUp1 | - productIndex: 0 | - El primer producto debe tener de precio “2500” |
| ProductSearchEngine | getIndex | setUp1 | - productIndex: 1 | - El segundo producto debe tener de precio “Exfoliante” |
| ProductSearchEngine | getIndex | setUp1 | - productIndex: 2 | - El tercer producto debe tener de precio “Jabon” |
| ProductSearchEngine | getIndex | setUp1 | - productIndex: 3 | - El cuarto producto debe tener de precio “Mascarilla” |
| ProductSearchEngine | getIndex | setUp1 | - productIndex: 4 | - El quinto producto debe tener de precio “Shampoo” |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que el algoritmo de busqueda por prefijo y sufijo funcione correctamente. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| ProductSearchEngine | bSOfCoincidencesUsing  SuffixOrPrefix | setUp2 | - criteria: true  - value: “Mas” | - El resultado es un ArrayList de Product que debe contener productos que empiecen con el prefijo “Mas” |
|  |
| ProductSearchEngine | bSOfCoincidencesUsing  SuffixOrPrefix | setUp2 | - criteria: false  -value: “ana” | - El resultado es un ArrayList de Product que debe contener productos que terminen con el sufijo “ana |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que el algoritmo de busqueda por rango busque correctamente. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| ProductSearch | bsProductUsingRangeOf | setUp3 | - min: 550 - max: 1000 - typeVariable: “price” | - El resultado es un ArrayList de Product que debe contener productos que tengan precios dentro del rango especificado. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que el algoritmo de ordenamiento ordena correctamente. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| ProductSearchEngine | binarySearchOfProduct  UsingStringValue | setUp1 | - value: “Crema dental” | - Retorna nulo debido a que este producto no existe en el inventario |
| - typeVariable: “name” |
| ProductSearchEngine | binarySearchOfProduct  UsingStringValue | setUp1 | - value: “Shampoo” | - El resultado debe tener por nombre “Shampoo” |
| - typeVariable: “name” |
| ProductSearchEngine | binarySearchOfProduct  UsingStringValue | setUp1 | - value: “Jabon” | - El resultado debe tener por nombre “Jabon” |
| - typeVariable: “name” |
| ProductSearchEngine | binarySearchOfProduct  UsingStringValue | setUp1 | - value: “Exfoliante” | - El resultado debe tener por nombre “Exfoliante” |
| - typeVariable: “name” |