* **Juan Sebastián Libreros**
* **Johan Daniel Aguirre**

**Requerimientos**

**Requerimientos**

| Cliente | Mercado Libre |
| --- | --- |
| Usuario | Cliente de la plataforma Mercado Libre |
| Requerimientos funcionales | R1. El sistema debe permitir al usuario ingresar la información de cada producto: nombre, descripción, precio, cantidad disponible, categoría y números de veces comprado.  R2. El sistema debe permitir al usuario ingresar información de los pedidos: nombre del comprador, lista de productos, precio total y fecha de la compra.  R3. El sistema debe permitir la búsqueda de productos por nombre, precio, categoría y número de veces comprado.  R4. El sistema debe permitir la búsqueda de pedidos por nombre de comprador, precio total y fecha de compra.  R5. El sistema debe permitir al usuario organizar en el orden que el usuario escoja (ascendentemente o descendentemente) los datos que se muestran el resultado de la búsqueda por las variables de ordenamiento.  R6. El sistema debe permitir almacenar la información de los productos en archivos para que los datos estén disponibles incluso después de que se cierre el programa.  R7. El sistema debe permitir al usuario establecer un rango de búsqueda por atributos numéricos, como el precio, la cantidad disponible o el número de veces comprado. El usuario debe poder especificar un valor mínimo y máximo para buscar los productos dentro de ese rango.  R8. El sistema debe permitir al usuario establecer un rango de búsqueda por atributos de tipo string, como el nombre, donde el usuario pueda definir una letra o prefijo de inicio y una letra o prefijo final para buscar productos que se encuentren en ese rango alfabético. |
| Contexto del problema | La empresa Mercado Libre solicita una aplicación que permita la venta online de su tienda virtual. Se solicita crear un programa que permita ingresar el inventario de la tienda virtual y permita la búsqueda y eliminación de productos. Además, el programa le debe permitir registrar los pedidos que hacen los usuarios de la tienda. |

| Nombre o identificador | **R1:**ingresar la información de cada producto | | |
| --- | --- | --- | --- |
| Resumen | El sistema debe permitir al usuario ingresar la información de cada producto | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | **Condición de selección o repetición** |
| Nombre | String | debe tener un length >=1 |
| descripción | String | debe tener un length >=0 |
| precio | double | debe ser > 0 |
| cantidad disponible | int | debe ser >=0 |
| categoría | Category | debe ser alguno de los parámetros disponibles y debe ser diferente de null |
| número de veces comprado | int | debe ser >=0 |
| Actividades generales necesarias para obtener los resultados | 1 El usuario ingresa al módulo de registro de producto en el sistema.  2. El usuario rellena la información necesaria pedida por el sistema  3. El sistema comprueba la validez de los datos  4. El sistema registra el producto.  5. El sistema muestra un mensaje de confirmación | | |
| Resultado o Postcondición | un nuevo producto a quedado registrado | | |
| Salidas | **Nombre salida** | **Tipo de dato** | **Condición de selección o repetición** |
| msg | String | “el producto se ha registrado correctamente ” |

| Nombre o identificador | **R2:**ingresar la información de cada pedido | | |
| --- | --- | --- | --- |
| Resumen | El sistema debe permitir al usuario ingresar información de los pedidos. | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | **Condición de selección o repetición** |
| nombre del comprador | String | debe contar con un length >=1 |
| lista de productos | Product | debe ser !=null |
| precio total | double | debe ser >0 |
| fecha de la compra | Calendar | debe ser !=null |
| Actividades generales necesarias para obtener los resultados | 1.El usuario ingresa al módulo de registro de pedidos  2.El usuario rellena la información requerida por el sistema  3.El sistema comprueba la validez de la información  4. El sistema modifica el inventario quitando los productos del pedido  5. El sistema muestra un mensaje de confirmación | | |
| Resultado o Postcondición | un nuevo pedido es registrado | | |
| Salidas | **Nombre salida** | **Tipo de dato** | **Condición de selección o repetición** |
| msg | String | “el pedido se registró correctamente” |

| Nombre o identificador | **R3: búsqueda de productos** | | |
| --- | --- | --- | --- |
| Resumen | El sistema debe permitir la búsqueda de productos. | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | **Condición de selección o repetición** |
| Nombre | String | debe tener un length >=1 |
| precio | double | debe ser > 0 |
| categoría | Category | debe ser alguno de los parámetros disponibles y debe ser diferente de null |
| número de veces comprado | int | debe ser >=0 |
| Actividades generales necesarias para obtener los resultados | 1.El usuario ingresa los datos requeridos por el sistema  2.El sistema comprueba la congruencia de los datos  3.El sistema muestra un mensaje con los datos filtrados | | |
| Resultado o Postcondición | El sistema mostrará por pantalla un mensaje con los productos filtrados | | |
| Salidas | **Nombre salida** | **Tipo de dato** | **Condición de selección o repetición** |
| msg | String | debe contar con un length >=0 |

| Nombre o identificador | **R4:**búsqueda de pedidos | | |
| --- | --- | --- | --- |
| Resumen | El sistema debe permitir la búsqueda de pedidos por nombre de comprador, precio total y fecha de compra.  . | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | **Condición de selección o repetición** |
| nombre del comprador | String | debe contar con un length >=1 |
| precio total | double | debe ser >0 |
| fecha de la compra | Calendar | debe ser !=null |
| Actividades generales necesarias para obtener los resultados | 1.El usuario ingresa los datos requeridos por el sistema  2.El sistema comprueba la congruencia de los datos  3.El sistema muestra un mensaje con los datos filtrados | | |
| Resultado o Postcondición | El sistema mostrará por pantalla un mensaje con los pedidos filtrados | | |
| Salidas | **Nombre salida** | **Tipo de dato** | **Condición de selección o repetición** |
| msg | String | debe contar con un length >=0 |

| Nombre o identificador | **R5:**orden de búsqueda | | |
| --- | --- | --- | --- |
| Resumen | El sistema debe permitir al usuario organizar en el orden que el usuario escoja (ascendentemente o descendentemente) . | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | **Condición de selección o repetición** |
| ordtype | int | debe ser >=1 y <=2 |
| Actividades generales necesarias para obtener los resultados | 1. El usuario entra al módulo de búsqueda 2. El usuario rellena la información de la búsqueda que desea realizar: 1 para ascendente y 2 para descendente. 3. El sistema despliega en pantalla las opciones de filtrar ascendentemente o descendentemente 4. El usuario selecciona la opción que desee 5. El sistema ordena según la opción seleccionada y luego muestra en pantalla el resultado | | |
| Resultado o Postcondición | se mostrará en pantalla la búsqueda que haya seleccionado el usuario en el orden que haya seleccionado | | |
| Salidas | **Nombre salida** | **Tipo de dato** | **Condición de selección o repetición** |
| msg | String | lista de búsqueda ordenada segun la seleccion del usuario |

| Nombre o identificador | **R6:**almacenamiento de las operaciones realizadas | | |
| --- | --- | --- | --- |
| Resumen | El sistema debe permitir almacenar la información de los productos en archivos para que los datos estén disponibles incluso después de que se cierre el programa | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | **Condición de selección o repetición** |
| op | String | Debe haber productos registrados en el sistema |
| Actividades generales necesarias para obtener los resultados | Debe haber productos registrados en el sistema | | |
| Resultado o Postcondición | Se guardan los datos en un archivo en formato json | | |
| Salidas | **Nombre salida** | **Tipo de dato** | **Condición de selección o repetición** |
|  |  |  |

| Nombre o identificador | **R7:ingresar** Rango de búsqueda numérica | | |
| --- | --- | --- | --- |
| Resumen | El sistema debe permitir establecer un rango de búsqueda numérica | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | **Condición de selección o repetición** |
| minValue | int | minValue≥0 |
| maxValue | int | maxValue<quantityOfProducts |
| Actividades generales necesarias para obtener los resultados | Debe haber productos registrados en el sistema | | |
| Resultado o Postcondición | Se muestran los resultados que se encuentren dentro de ese rango de busqueda | | |
| Salidas | **Nombre salida** | **Tipo de dato** | **Condición de selección o repetición** |
| listProductsInRange | ArrayList<> |  |

| Nombre o identificador | **R8:**ingresar Rango de búsqueda de Strings | | |
| --- | --- | --- | --- |
| Resumen | El sistema debe permitir establecer un rango de búsqueda por atributos de tipo String | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | **Condicion de selección o repeticion** |
| minValue | String | debe tener un length igual o mayor que 1 |
| maxValue | String | debe tener un length igual o mayor que 1 |
| Actividades generales necesarias para obtener los resultados | Debe haber productos registrados en el sistema | | |
| Resultado o Postcondición | Se muestran los resultados que se encuentren dentro de ese rango de busqueda | | |
| Salidas | **Nombre salida** | **Tipo de dato** |  |
| listProductsInRange | ArrayList<> |  |

**Diseño de pruebas**

**Configuración de los Escenarios**

| **Nombre** | **Clase** | **Escenario** |
| --- | --- | --- |
| setUpStage1 | Product | Un objeto de la clase Product con name = “Bolso Louis Vuitton”, description = “Bolso de alta gama con cuero suizo”, price = 250000, inventory = 250, category = ACCESORIO, sales = 25 |
| setUpStage2 | Product | Un objeto de la clase Product con name = “Tarjeta gráfica NVIDIA GeForce RTX 3060”, description = “Tarjeta grafica de NVIDIA”, price = 2099000, inventory = 150, category = ELECTRONICA, sales = 10 |
| setUpStage3 | Delivery | Un pedido con buyerName = “Juan Sebastian”,listProducts = {objProduct1,objProduct2}, totalPrice = 500400, saleDate = 20/05/2023 |
| setUpStage4 | Store | Una lista de pedidos con 5 elementos:   1. Un pedido con buyerName = “Juan Sebastian L”,listProducts = {objProduct1,objProduct2,objproduct3}, totalPrice = 1200000, saleDate = 20/05/2023 2. Un pedido con buyerName = “Johan Daniel Aguirre”,listProducts = {objProduct1,objprooduct5}, totalPrice = 1000000, saleDate = 10/02/2023 3. Un pedido con buyerName = “Paola Andrea”,listProducts = {objProduct1}, totalPrice = 250000, saleDate = 30/05/2023 . 4. Un pedido con buyerName = “Larry Kapija”,listProducts = {objProduct1,objProduct2}, totalPrice = 500000, saleDate = 24/12/2022 5. Un pedido con buyerName = “Marian Camille Merch”,listProducts = {objProduct1,objProduct2}, totalPrice = 500000, saleDate = 20/05/2023 |
| setUpStage5 | Store | Una lista con 10 elementos de tipo producto:  1.name = “Bolso Louis Vuitton”  description = “Bolso de alta gama con cuero suizo”  price = 250000  inventory = 250  category = ACCESORIO  sales = 25  2.name = “hp pavilion mini 2012 perfecto estado”, description = “computadora mini perfecta para llevar en el bolsillo con características técnicas que por su tamaño no aparenta, en perfecto estado se vende por motivo de viaje NO NEGOCIABLE”  price = 1.500.000  inventory = 1  category =TECNOLOGÍA  sales = 0  3.name = “grafica AMD rx6800xt”  description = “tarjetas gráficas baratas pues se usaron para mineria y ya no les puedo dar uso precio negociable”  price = 2.000.000  inventory = 600  category = TECNOLOGÍA  sales = 30  4.name = “vajilla imusa ceramica”  description =”hermosa vajilla de excelentisima calidad”  price = 40.000  inventory = 50.000  category = HOGAR  sales = 600  5.name = “termo de agua 2l”  description = “termo de agua en diferentes colores y aun excelente precio. Somos una tienda de verificada”  price = 40.000  inventory = 9999  category = ACCESORIOS  sales = 30  6.name = “termo de agua 2l”  description = “termo de agua de excelente calidad somos una importadora seria así que mejor abstenerse de preguntar cosas innecesarias ”  price = 20.000  inventory = 25  category = ACCESORIOS  sales = 2  7.name = “tarjeta de sonido profesional”  description = “tarjeta de sonido sound blaster ultimate la cual cuenta con memoria propia de mas de 1gb y le dara a tus ratos de entretenimiento en casa un toque unico con un sonido profundo y espectacular”  price = 290.000  inventory = 12000  category = TECNOLOGÍA  sales = 1600  8.name = “tarjeta de sonido gamer”  description = “tarjeta de sonido game blaster platinum para escuchar tus juegos favoritos a otro nivel”  price = 175.000  inventory = 25000  category = TECNOLOGÍA  sales = 3600  9.name = “silla de escritorio ergonomica”  description = “silla de oficina marca lo mejor para tu espalda que ayudará a tu columna a liberar toda esa presión acumulada por el trabajo”  price = 360.000  inventory = 5000  category = HOGAR  sales = 350  10.name = “anillo de plata tematica juego Zelda”  description = “anillo de excelente calidad con diferentes logotipos de la temática del videojuego de Zelda”  price = 40.000  inventory = 500  category = ACCESORIOS  sales = 30 |
| setUpStage6 | Store | Una lista vacía de pedidos. |

**Tests**

R1.

| **Objetivo de la Prueba:** Probar que el método setInventory, funcione de la siguiente manera: Si el usuario pide que se le añadan, deberá ingresar un valor no negativo, de lo contrario si desea restar debe ser un número negativo menor a 0 y este a su vez, deberá ser menor también a la cantidad de inventario disponible | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Product | setInventory | setUpStage1 | inventoryToAdd = 25 | inventory = 275 |
| Product | setInventory | setUpStage1 | inventoryToAdd = -251 | throw insufficient inventory Exception |
| Product | setInventory | setUpStage1 | inventoryToAdd = -25 | inventory = 225. |
| Product | setInventory | setUpStage1 | inventoryToAdd = -25a | throw invalid data exception |

| **Objetivo de la Prueba:** Probar que los métodos removeProduct y addProduct de la clase Store funcionan correctamente y borran y añaden los objetos solicitados o en su defecto indique que no existe dicho objeto solicitado | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Store | removeProduct | setUpStage5 | name = “Bolso Louis Vuitton ” | El objeto Bolso Louis Vuitton ya no existe en la tienda |
| Store | removeProduct | setUpStage5 | name = “Juguete de max steel” | throw new ObjectNotFoundException |
| Store | addProduct | setUpStage5 | Un objeto de la clase Product con name = “Vaso de cristal”, description = “Vaso super fino”, price = 250000, inventory = 250, category = ACCESORIO, sales = 25 | object not null in array |
| Store | addProduct | setUpStage5 | Un objeto de la clase Product con name = “Bolso Louis Vuitton”, description = “Bolso de alta gama con cuero suizo”, price = 250000, inventory = 25, category = ACCESORIO, sales = 25 | add to the inventory of name = “Bolso Louis Vuitton” inventory = 275 |

**R2.**

| **Objetivo de la Prueba:** Probar que el metodo de addProduct y removeProduct dentro de la clase delivery añada correctamente el producto y modifique el valor de el total price | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Delivery | addProduct | setUpStage3 | new Product(objProduct3,.., Price = 2000,...) | TotalPrice = 502400 |
| Delivery | removeProduct | setUpStage3 | int 1 | TotalPrice = 500000 |
| Delivery | addProduct | setUpStage3 | new Product(objProduct3,.., Price = -2000,...) | throw new invalidDataException |
| Delivery | removeProduct | setUpStage3 | int = -1 | TotalPrice = 500400 |

| **Objetivo de la Prueba:** Probar que los metodos createDelivery y modifyDelivery de la clase Store funcionan correctamente y añaden | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Store | create Delivery | setUpStage4,  setUpStage5  setUpStage6 | escribir atributos de delivery | se crea el objeto Delivery con los productos solicitados |
| Store | create Delivery | setUpStage4,  setUpStage5  setUpStage6 | delivery, producto = “Yo mismo”, quantityWanted = 999 | no se crea el delivery |
| Store | modifyDelivery | setUpStage4,  setUpStage5  setUpStage6 | quantitWantedPro = 2 | El objeto Delivery se le modifica la cantidad deseada de producto X y cambia el precio total de compra |
| Store | modifyDeliveryt | setUpStage4,  setUpStage5  setUpStage6 | quantitWantedPro = 0 | eliminación del producto X en el pedido |
| Store | modifyDeliveryt | setUpStage4,  setUpStage5  setUpStage6 | quantitWantedPro = -1 | throw invalid data exception |

R3. LOS que estan en azul los estoy haciendo yo JOHAN DANIEL AGUIRRE ARIAS

| **Objetivo de la Prueba:** Probar que las funciones searchProduct funcionan correctamente y encuentra los productos que cumplan con las caracteristicas solicitadas | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Store | searchProduct | setUpStage5 | name = “Termo de agua 2l” | objeto con name termo de agua 2L |
| Store | searchProduct | setUpStage5 | category = TECNOLOGIA | Lista de objetos que tengan como categoría TECNOLOGÍA |
| Store | searchProduct | setUpStage5 | price = 40000 | lista de objetos que tengan precio 40000 |
| Store | searchPorduct | setUpStage5 | numSales = 30 | lista de objetos que hayan sido comprados 30 veces |
| Store | searchPorduct | setUpStage5 | name = “Disco pavlo” | ObjectNotFoundException |

R4.

| **Objetivo de la Prueba:** Probar que los metodos searchDelivery funcionan correctamente y encuentra los pedidos que cumplen con las caracteristicas solicitadas | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Store | searchDelivery | seUpStage4 | nombredelcomprador=”maria delano roosevelt” | pedido que contenga los productos:  objProduct1,objProduct2  un precio de 500000  saleDate = 24/12/2022 |
| Store | searchDelivery | seUpStage4 | precio total: 360000 | nombre del comprador“Juan Sebastian”  productos: objProduct1,objProduct2  precio: 500000 |
| Store | searchDelivery | seUpStage4 | fecha de compra: 20/05/2023 | una lista de dos pedidos:  1.  nombre del comprador: “Larrika pija”  2.  nombre del comprador“Juan Sebastian”  productos: objProduct1,objProduct2  precio: 500000 |
| Store | searchDelivery | seUpStage4 | nombre del comprador= “juanito alachofas” | una excepción de pedido inexistente |

R5.

| **Objetivo de la Prueba:** Probar que las funciones organizeList funcionan correctamente y organiza los productos de la manera solicitada | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Store | organizeList | setUpStage5 | ordtype= 1 | una lista organizada de la siguiente forma:  1.name = “vajilla imusa ceramica”  description =”hermosa vajilla de excelentisima calidad”  price = 40.000  inventory = 50.000  category = HOGAR  sales = 600  2.name = “termo de agua 2l”  description = “termo de agua en diferentes colores y aun excelente precio. Somos una tienda de verificada”  price = 40.000  inventory = 9999  category = ACCESORIOS  sales = 30  3.name = “termo de agua 2l”  description = “termo de agua de excelente calidad somos una importadora seria así que mejor abstenerse de preguntar cosas innecesarias ”  price = 20.000  inventory = 25  category = ACCESORIOS  sales = 2  4.name = “tarjeta de sonido profesional”  description = “tarjeta de sonido sound blaster ultimate la cual cuenta con memoria propia de mas de 1gb y le dara a tus ratos de entretenimiento en casa un toque unico con un sonido profundo y espectacular”  price = 290.000  inventory = 12000  category = TECNOLOGÍA  sales = 1600  5.name = “tarjeta de sonido gamer”  description = “tarjeta de sonido game blaster platinum para escuchar tus juegos favoritos a otro nivel”  price = 175.000  inventory = 25000  category = TECNOLOGÍA  sales = 3600  6.name = “silla de escritorio ergonomica”  description = “silla de oficina marca lo mejor para tu espalda que ayudará a tu columna a liberar toda esa presión acumulada por el trabajo”  price = 360.000  inventory = 5000  category = HOGAR  sales = 350  7.name = “hp pavilion mini 2012 perfecto estado”, description = “computadora mini perfecta para llevar en el bolsillo con características técnicas que por su tamaño no aparenta, en perfecto estado se vende por motivo de viaje NO NEGOCIABLE”  price = 1.500.000  inventory = 1  category =TECNOLOGÍA  sales = 0  8.name = “grafica AMD rx6800xt”  description = “tarjetas gráficas baratas pues se usaron para mineria y ya no les puedo dar uso precio negociable”  price = 2.000.000  inventory = 600  category = TECNOLOGÍA  sales = 30  9.name = “Bolso Louis Vuitton”  description = “Bolso de alta gama con cuero suizo”  price = 250000  inventory = 250  category = ACCESORIO  sales = 25  10.name = “anillo de plata tematica juego Zelda”  description = “anillo de excelente calidad con diferentes logotipos de la temática del videojuego de Zelda”  price = 40.000  inventory = 500  category = ACCESORIOS  sales = 30 |
| Store | organizeList | setUpStage5 | ordtype=2 | una lista organizada de la siguiente forma:  1.name = “anillo de plata tematica juego Zelda”  description = “anillo de excelente calidad con diferentes logotipos de la temática del videojuego de Zelda”  price = 40.000  inventory = 500  category = ACCESORIOS  sales = 30  1.name = “Bolso Louis Vuitton”  description = “Bolso de alta gama con cuero suizo”  price = 250000  inventory = 250  category = ACCESORIO  sales = 25  2.name = “grafica AMD rx6800xt”  description = “tarjetas gráficas baratas pues se usaron para mineria y ya no les puedo dar uso precio negociable”  price = 2.000.000  inventory = 600  category = TECNOLOGÍA  sales = 30  3.name = “grafica AMD rx6800xt”  description = “tarjetas gráficas baratas pues se usaron para mineria y ya no les puedo dar uso precio negociable”  price = 2.000.000  4.name = “hp pavilion mini 2012 perfecto estado”, description = “computadora mini perfecta para llevar en el bolsillo con características técnicas que por su tamaño no aparenta, en perfecto estado se vende por motivo de viaje NO NEGOCIABLE”  price = 1.500.000  inventory = 1  category =TECNOLOGÍA  sales = 0  inventory = 600  category = TECNOLOGÍA  sales = 30  5.name = “silla de escritorio ergonomica”  description = “silla de oficina marca lo mejor para tu espalda que ayudará a tu columna a liberar toda esa presión acumulada por el trabajo”  price = 360.000  inventory = 5000  category = HOGAR  sales = 350  6.name = “tarjeta de sonido gamer”  description = “tarjeta de sonido game blaster platinum para escuchar tus juegos favoritos a otro nivel”  price = 175.000  inventory = 25000  category = TECNOLOGÍA  sales = 3600  7.name = “tarjeta de sonido profesional”  description = “tarjeta de sonido sound blaster ultimate la cual cuenta con memoria propia de mas de 1gb y le dará a tus ratos de entretenimiento en casa un toque único con un sonido profundo y espectacular”  price = 290.000  inventory = 12000  category = TECNOLOGÍA  sales = 1600  8.name = “termo de agua 2l”  description = “termo de agua en diferentes colores y aun excelente precio. Somos una tienda de verificada”  price = 40.000  inventory = 9999  category = ACCESORIOS  sales = 30  9.name = “termo de agua 2l”  description = “termo de agua de excelente calidad somos una importadora seria así que mejor abstenerse de preguntar cosas innecesarias ”  price = 20.000  inventory = 25  category = ACCESORIOS  sales = 2  10.name = “vajilla imusa ceramica”  description =”hermosa vajilla de excelentisima calidad”  price = 40.000  inventory = 50.000  category = HOGAR  sales = 600 |
| Store | organizeList | setUpStage4 | ordtype= 1 | una lista organizada de la siguiente forma:  1.con buyerName = “sebas lib”,listProducts = {objProduct1,objprooduct5}, totalPrice = 1000000, saleDate = 10/02/2023  2.buyerName = “maria rosa del ano”,listProducts = {objProduct1}, totalPrice = 250000, saleDate = 30/05/2023 .  3.buyerName = “maria delano rossvelt”,listProducts = {objProduct1,objProduct2}, totalPrice = 500000, saleDate = 24/12/2022  4.buyerName = “Juan Sebastian”,listProducts = {objProduct1,objProduct2}, totalPrice = 500000, saleDate = 20/05/2023  5.buyerName = “elver gomez torba”,listProducts = {objProduct1,objProduct2,objproduct3}, totalPrice = 1200000, saleDate = 20/05/2023 |
| Store | organizeList | setUpStage4 | ordtype=2 | una lista de la siguiente forma:  1.con buyerName = “elver gomez torba”,listProducts = {objProduct1,objProduct2,objproduct3}, totalPrice = 1200000, saleDate = 20/05/2023  2.buyerName = “Juan Sebastian”,listProducts = {objProduct1,objProduct2}, totalPrice = 500000, saleDate = 20/05/2023  3.buyerName = “maria delano rossvelt”,listProducts = {objProduct1,objProduct2}, totalPrice = 500000, saleDate = 24/12/2022  4.buyerName = “maria rosa del ano”,listProducts = {objProduct1}, totalPrice = 250000, saleDate = 30/05/2023 .  5.buyerName = “sebas lib”,listProducts = {objProduct1,objprooduct5}, totalPrice = 1000000, saleDate = 10/02/2023 |

**R6**

| **Objetivo de la Prueba:** Probar que las funciones save y load funcionan correctamente y guardan y cargan los archivos en formato json | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Store | save | setUpStage5 | una lista con los datos dados en el setUpStage5 | que se creee un archivo en formato json con el nombre ”datos.json” que contiene los datos del escenario 5 |
| Store | save | setIpStage5 | una lista con los datos dados en el setUpStage5 | que sevie una exepcion de datos iguales en el json y se mantega el archivo “datos.json” sin cambios |
| Store | load | setUpStage2 | N/A | que se hallle una lista con 11 productos los 10 guardados en el archivo “datos.json” y el que se encuentra en el escenario 2 |
| Store | load | setUpStage2 | la lista de 11 elementos | que no se cargen arhivos dado que ya los datos del archivo se encuentran cargados. |

**R7**

| **Objetivo de la Prueba:** Probar que las funcion searchInRangeNumeric funcione para la búsqueda dentro del rango solicitado por el usuario | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Store | searchInRangeNumeric | setUpStage5 | minValuePrice = 200  maxValuePrice = 400000 | Todos los objetos de tipo Product que se encuentren dentro del rango de 200 pesos y 400000 pesos |
| Store | searchInRangeNumeric | setUpStage5 | minValueNumSales = -90  maxValueNumSales= 400 | Error, el rango mínimo debe ser mayor o igual a 0 |
| Store | searchInRangeNumeric | setUpStage5 | minValueNumSales = 0  maxValueNumSales= 999999 | Se muestran los objetos dentro del rango, esencialmente todos. |
| Store | searchInRangeNumeric | setUpStage5 | minNumValueAvailable = 0  maxNumValueAvailable = 10 | Se muestran todos los objetos dentro del rango y por atributos número de objetos disponibles |

**R8**

| **Objetivo de la Prueba:** Probar que las funciones searchInRangewords funcione para la búsqueda dentro del rango solicitado por el usuario | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Store | searchInRangewords | setUpStage4 | minValue=”asd”  maxValue=”z” | debe contener una lista vacía pues no hay objetos con estas características |
| Store | searchInRangewords | setUpStage4 | minValue=”Tar”  maxValue=”s” | Muestra los objetos que empiezan por Tar y termina con s |
| Store | searchInRangewords | setIpStage5 | minValue=”1”  maxValue=”a” | debe arrojar la exepcion de invalidez de rango de busqueda |
| Store | searchInRangewords | setUpStage5 | minValue=”a”  maxValue=”z” | debe mostrar todos los objetos que empiezan por a y terminan en z |
| Store | searchInRangewords | setUpStage5 | minValue=”\*”  maxValue=”+” | thrown invalid search exception |
| Store | searchInRangewords | setUpStage5 | minValue=”9”  maxValue=”8” | thrown invalid search exception |