

Requerimientos

Cliente	<i>MercadoLibre</i>
Usuario	Usuarios vendedores de mercado libre
Requerimientos funcionales	R1: Registrar Productos R2: Registrar pedidos R3: Buscador R4: Cambiar las cantidades de un producto R5: Eliminar un producto R6: Menu
Requerimientos no funcionales	<ul style="list-style-type: none"> - El buscador debe usar un algoritmo de busqueda binaria - Serializar la informacion usando archivos Json
Contexto del problema	La compañía mercadolibre quiere desarrollar una aplicación que le permita la venta en línea de su tienda virtual, registrando el inventario y permitiendo la búsqueda y eliminación de productos como también registrar los pedidos que los usuario hacen de la tienda

Nombre o identificador	R1: Registrar Productos		
Resumen	la aplicación debe permitir al usuario ingresar información sobre cada producto, incluyendo su nombre, descripción, precio, cantidad disponible, categoría (Libros, Electrónica, Ropa y accesorios, Alimentos y bebidas, Papelería, Deportes, Productos de belleza y cuidado personal, Juguetes y juegos)		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	productName description	String	
	price	double	
	quantity category	int	no puede ser un número negativo

Actividades generales necesarias para obtener los resultados	Crear un objeto producto con todas las características y almacenarlo		
Resultado o Postcondición	Un objeto producto almacenado		
Salidas	Nombre salida	Tipo de dato	Condición de selección o repetición
	msg	String	mensaje de éxito o error

Nombre o identificador	R2: Registrar pedidos		
Resumen	La aplicación debe permitir al usuario ingresar información como nombre del comprador, lista de productos, precio total y fecha de la compra. Al momento de generar un pedido, tenga en cuenta que las cantidades del inventario disminuyen. El usuario puede aumentar la cantidad de cada producto ya registrado.		
Entradas	Nombre de entrada	Tipo de Dato	Condición de selección o repetición
	buyerName listOfProducts	String	deben ser productos registrados
	dateOfSale	String	
Actividades generales necesarias para obtener los resultados	Revisar que los productos esten registrados y calcular el precio total Crear un objeto pedido, almacenarlo y disminuir las cantidades del producto		
Resultado o Postcondición	El pedido almacenado		
Salidas	Nombre salida	Tipo de dato	Condición de selección o repetición

	msg	String	mensaje de éxito o error
--	-----	--------	--------------------------

Nombre o identificador	R3: Buscador		
Resumen	<p>Buscador de productos por nombre, precio, categoría, y número de veces comprado, también un buscador de pedidos por nombre de comprador, precio total y fecha de compra.</p> <p>Los buscadores deben utilizar el algoritmo de búsqueda binaria para encontrar coincidencias en los registros de productos y deben mostrar los resultados por consola.</p> <p>También, el buscador debe permitir al usuario escoger el orden de los datos que se van a mostrar (ascendente o descendente)</p>		
Entradas	Nombre de entrada	Tipo de Dato	Condición de selección o repetición
	productOrOrder opcion	int	
	minProductName maxProductName minPrice maxPrice minCategory maxCategory minNumberOfSales maxNumberOfSales	String String double double String String int int	los minimos y maximos deben ser correctos, no puede ser un maximo menor que un minimos
	minBuyerName maxBuyerName minDateOfSale maxDateOfSale minTotalPrice maxTotalPrice	String String String String double double	los minimos y maximos deben ser correctos, no puede ser un minimo mayor que un maximo
	ascendentOrDescendent	int	
Actividades generales necesarias para obtener los resultados	Después de seleccionar si quiere buscar productos o pedidos se deben buscar segun la seleccion del usuario utilizando y el rango dado, en un algoritmo de busqueda binaria		

Resultado o Postcondición	Imprimir los resultados de la búsqueda por consola según el orden seleccionado por el usuario		
Salidas	Nombre salida	Tipo de dato	Condición de selección o repetición
	msg	String	

Nombre o identificador	R4: Aumentar las cantidades de un producto		
Resumen	El usuario debe poder buscar un producto por nombre y cambiar las cantidades disponibles de ese producto		
Entradas	Nombre de entrada	Tipo de Dato	Condición de selección o repetición
	productName	String	debe ser un nombre de un producto ya registrado
	addAmount	int	Debe ser mayor a 0
Actividades generales necesarias para obtener los resultados	utilizar el buscador para encontrar el producto y aumentar las unidades disponibles por el numero ingresado por el usuario		
Resultado o Postcondición	La cantidad disponible del producto se aumenta por el valor ingresado y se imprime un mensaje de confirmacion		
Salidas	Nombre salida	Tipo de dato	Condición de selección o repetición
	msg	String	

Nombre o identificador	R5: Eliminar un producto
-------------------------------	---------------------------------

Resumen	El usuario puede buscar un producto por su nombre y eliminarlo		
Entradas	Nombre de entrada	Tipo de Dato	Condición de selección o repetición
	productName	String	debe ser un producto ya registrado
Actividades generales necesarias para obtener los resultados	encontrar el producto utilizando el buscador y eliminarlo del arraylist de productos		
Resultado o Postcondición	Mensaje de confirmacion		
Salidas	Nombre salida	Tipo de dato	Condición de selección o repetición
	msg	String	

Nombre o identificador	R6: Menu		
Resumen	Debe haber un menu para que el usuario pueda interactuar con el programa de manera organizada y elegir con facilidad que acciones desea realizar		
Entradas	Nombre de entrada	Tipo de Dato	Condición de selección o repetición
	option	int	
Actividades generales necesarias para obtener los resultados	un switch que dado la opcion ingresada por el usuario llame al método que realiza cada acción		
Resultado o Postcondición	Vuelve a mostrar el menú cuando termine, a menos que elija la opción de terminar.		
Salidas	Nombre salida	Tipo de dato	Condición de selección o repetición

Formato de escenarios y casos de uso

Configuración de los Escenarios

Nombre	Clase	Escenario
setupStage1	<i>ControllerTest</i>	Un objeto de la clase Controller con un arraylist products vacio
setupStage2	<i>ControllerTest</i>	Un objeto de la clase Controller con un arreglo products con 3 objetos de la clase Product: <ul style="list-style-type: none">- productName="cuaderno", description="un cuaderno cuadriculado", price= 23, quantity=3, category="Libros" y numberOfSales=0- productName="Camisa", description="roja", price= 15, quantity=5, category="Ropa y accesorios" y numberOfSales=20- productName="audifonos", description="bluetooth", price= 30, quantity=3, category="Electronica" y numberOfSales=10
setupStage3	<i>ControllerTest</i>	setupStage2 Un objeto de la clase Controller con un arraylist orders vacio
setupStage4	<i>ControllerTest</i>	Un objeto de la clase Order con: <ul style="list-style-type: none">- productName="cuaderno", description="un cuaderno cuadriculado", price= 23, quantity=3, category="Libros" y numberOfSales=0
setupStage5	<i>ControllerTest</i>	setupStage2 3 objetos de la clase Order en el arraylist orders: <ol style="list-style-type: none">1. buyerName="Carlos"<ul style="list-style-type: none">- listOfProducts="cuaderno,audifonos"- dateOfSale="21/2/2023"2. buyerName="Gloria"<ul style="list-style-type: none">- listOfProducts="audifonos"- dateOfSale="25/2/2023"3. buyerName="Simon"<ul style="list-style-type: none">- listOfProducts="camisa, cuaderno"- dateOfSale="21/5/2023"

Diseño de Casos de Prueba

Objetivo de la Prueba: Verificar que el metodo deleteProduct de la clase Controller funcione correctamente				
Clase	Método	Escenario	Valores de Entrada	Resultado esperado
Controller	deleteProduct	setupStage 2	productName="cuaderno"	se eliimina el producto "cuaderno" y quedan camisa y audifonos en el array products
Controller	deleteProduct	setupStage 2	productName="silla"	una excepcion con un mensaje que diga que ese producto no esta registrado

Objetivo de la Prueba: Verificar que el metodo increaseQuantity de la clase Product funcione correctamente				
Clase	Método	Escenario	Valores de Entrada	Resultado esperado
Product	increaseQuantity	setupStage 2	productName="cuaderno" addAmount=2	la cantidad disponible de el producto cuaderno aumenta a 5
Product	increaseQuantity	setupStage 2	productName="silla" addAmount=2	una excepcion con un mensaje que diga que ese producto no esta registrado

Objetivo de la Prueba: Verificar que el metodo registerProduct y registerOrder de la clase Controller funcionen correctamente				
Clase	Método	Escenario	Valores de Entrada	Resultado esperado
Controller	registerProduct	setupStage 1	productName="camiseta", description=" talla m", price= 15, quantity=5, category=1 y numberOfSales=0	el arraylist tiene Un objeto Product con las variables: productName="camiseta", description=" talla m", price= 15, quantity=5, category="Ropa" y numberOfSales=0 en el arreglo
Controller	registerProduct	setupStage 2	productName="camiseta", description=" talla m", price= 15, quantity=5, category=1 y numberOfSales=0	el arraylist tiene 4 objetos products, el ultimo tiene productName="camiseta", description=" talla m", price= 15, quantity=5, category=1 y numberOfSales=0
Controller	registerOrder	setupStage 3	buyerName="Carlos" listOfProducts="cuaderno, audifonos" dateOfSale="21/2/2023"	El arraylist orders tiene un objeto Order con los dos objetos Products "cuaderno" y "audifonos" y el precio total es 53
Controller	registerOrder	setupStage 4	buyerName="Carlos" listOfProducts="cuaderno, megafono" dateOfSale="21/2/2023"	el arreglo sigue vacio y lanza una excepcion con un mensaje que diga que no hay ningun producto megafono registrado

Objetivo de la Prueba: Verificar que los metodos findProducts y findOrders de la clase Controller funcionen correctamente				
Clase	Método	Escenario	Valores de Entrada	Resultado esperado
Controller	findProducts	setupStage2	minProductName="C" maxProductName="M" ascendentOrDescendent=1	Imprime los nombres de los productos con nombres de la C-M en orden ascendente: Camisa y Cuaderno

Controller	findProducts	setupStage2	minPrice=20 maxPrice=30 ascendentOrDescendent=2	<p>Imprime los productos de precio 20-30 en orden descendente:</p> <p>audifonos y cuaderno</p>
Controller	findOrders	setupStage5	minBuyerName="A" maxBuyerName="G" ascendentOrDescendent=2	<p>Imprime las ordenes que el nombre del comprador este entre la A-F en orden descendente:</p> <p>buyerName="Gloria"</p> <ul style="list-style-type: none"> - listOfProducts="audifonos" - dateOfSale="25/2/2023" - totalPrice=30 <p>buyerName="Carlos"</p> <ul style="list-style-type: none"> - listOfProducts="cuaderno,audifonos" - dateOfSale="21/2/2023" - totalPrice=53
Controller	findOrders	setupStage5	minTotalPrice=0 maxTotalPrice=50 ascendentOrDescendent=1	<p>Imprime las ordenes que el precio total este entre 0-50, en orden ascendente:</p> <p>buyerName="Gloria"</p> <ul style="list-style-type: none"> - listOfProducts="audifonos" - dateOfSale="25/2/2023": - totalPrice=30 <p>buyerName="Simon"</p> <ul style="list-style-type: none"> - listOfProducts="camisa, cuaderno" - dateOfSale="21/5/2023" - totalPrice=38