

Client	Mercadolibre
User	User seller, User buyer
REQUERIMIENTOS FUNCIONALES	<p>R1: Register a product</p> <p>R2: Make an order</p> <p>R3: Increase the inventory of a product</p> <p>R4: Search product by unique attribute</p> <p>R5: Search product by a range</p> <p>R6: Search order by unique attribute</p> <p>R7: Search order by range</p>
CONTEXT OF THE PROBLEM	<p>MercadoLibre has contracted your services to develop an application that allows the online sale of its virtual shop. Your task is to create a program that allows you to enter the inventory of the online shop and allows the s</p>
NON-FUNCTIONAL REQUIREMENTS	<p>RNF 1: Use of persistence with JSON serialization is required.</p> <p>RNF 2: Use of exceptions is required to comply with software quality.</p> <p>RNF 3: Project must be tested with TDD</p> <p>RNF 4: 10 equi-temporal deliverables are required in GitHub repository</p> <p>RNF 5: Quality indicator tracking is required and will be represented in the repository README.</p>

Nombre o identificador	R1: Register a product		
Resumen	The application must allow the user to enter information about the products, including name, description, price, quantity available, category and number of times purchased. Categories such as books, electronics, clothing and accessories, food and beverages, stationery, sports, beauty and personal care products, toys and games should be considered.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	Name	String	
	Description	String	
	Price	double	The prices must be equals or greater than 0
	Amount Available	int	The amount must be equals or greater than 0
	Category	String	
	Amount purchased	int	The amount has to be equals or greater than 0
Actividades generales necesarias para obtener los resultados	<ul style="list-style-type: none"> - The user is allowed to enter the name, description, price, and quantity available. - Existing categories are displayed - user is allowed to enter the category and units sold - Required and formatting fields are validated to ensure data integrity. 		
Resultado o postcondición	The result is a product with the correct data added in the system.		
Salidas	Nombre Salida	Tipo de dato	Condición de selección o repetición
	Register status	boolean	

Nombre o identificador	R2: Make an order		
Resumen	The application should allow the user to enter order information, including the name of the buyer, the list of products, but the total price is calculated with respect to the products and the date is calculated at the time of purchase. When an order is generated, the inventory quantity of the purchased products must be updated.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	Buyer name	String	
	Products list	String	
Actividades generales necesarias para obtener los resultados	<ul style="list-style-type: none"> - The user is allowed to enter the personal name - The user is allowed to select a product and quantity, until the user decides not to add more products. - Validation of required fields and formatting to ensure the integrity of the data entered by the user - The system calculates the total order price - The total price is displayed on the screen. - The user is allowed to accept the purchase of the order - The date of the order is saved. - The system updates the available quantity of purchased products in the application's product database. 		
Resultado o postcondición	<ul style="list-style-type: none"> - Order information must be stored in the application's order database. - The inventory quantity of the purchased products must be updated in the product database of the application. 		
Salidas	Nombre Salida	Tipo de dato	Condición de selección o repetición
	Order status	boolean	No exceptions have been lodged

Nombre o identificador	R3: Increase the inventory of a product		
Resumen	Allow the user to increase the quantity of a product already registered.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	Product	String	
	Quantity to be added	int	The amount must be greater than 0
Actividades generales necesarias para obtener los resultados	<ul style="list-style-type: none"> - Search for the product in the application's product database. - Update the product quantity by adding the quantity provided by the user. 		
Resultado o postcondición	The quantity of the product is updated by adding the quantity provided by the user.		
Salidas	Nombre Salida	Tipo de dato	Condición de selección o repetición
	Process status	boolean	

Nombre o identificador	R4: Search a product by unique attribute		
Resumen	The program should allow the user to look for a product using a single attribute.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	search criteria	String	
	search value	String	
Actividades generales necesarias para obtener los resultados	<ol style="list-style-type: none"> 1.The user is allowed to enter the search criteria 2.The user is allowed to enter the search Value 3.If the search criteria is numeric, the system will parse the search value to an Integer. 4.The system will search the product in the product container structure using the binary search algorithm according to the search criteria. 		
Resultado o postcondición	The system will show the product if it was found or will show an alert if the product was not found.		
Salidas	Nombre Salida	Tipo de dato	Condición de selección o repetición

	Product Found Info	String	If the product was found
	Warning message	String	If the product wasn't found

Nombre o identificador	R5: Search product by a range		
Resumen	The system must allow the search for products in a numeric or alphanumeric range. To do so, a search criterion must be entered, such as name, price, category, among others, followed by a lower limit and an upper limit, which must belong to the type of criterion selected. Then, the system will search and display the products within that range.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	Search criteria	String	
	Lower limit	String	
	Upper limit	String	The upper limit must be the greater of the lower limit.
Actividades generales necesarias para obtener los resultados	1.The user is allowed to enter the search criteria 2.The user is allowed to enter the search lower limit 3.The user is allowed to enter the search upper limit 4.If the search criteria is numeric, the system will parse the search values to Integers. 5.The system will search the product in the product container structure using the binary search algorithm according to the search criteria.		
Resultado o postcondición	The system will show the product if it was found or will show an alert if the product was not found.		
Salidas	Nombre Salida	Tipo de dato	Condición de selección o repetición
	Product Found Info	String	If the product was found
	Warning message	String	If the product wasn't found

Nombre o identificador	R6: Search order by unique attribute		
Resumen	The program should allow the user to look for an order using a single attribute.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	search criteria	String	
	search value	String	
Actividades generales necesarias para obtener los resultados	1.The user is allowed to enter the search criteria 2.The user is allowed to enter the search Value 3.If the search criteria is numeric, the system will parse the search value to an Integer. 4.The system will search the order in the order container structure using the binary search algorithm according to the search criteria.		
Resultado o postcondición	The system will show the order if it was found or will show an alert if the order was not found.		
Salidas	Nombre Salida	Tipo de dato	Condición de selección o repetición
	Order found Info	String	If the order was found
	Warning message	String	If the order wasn't found

Nombre o identificador	R7: Search order by range		
Resumen	The system must allow the search of orders in a numeric or alphanumeric range. To do so, a search criterion must be entered, such as buyer's name, price, and others, followed by a lower limit and an upper limit, which must belong to the type of criterion selected. Then, the system will search and display the products within that range.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	Search criteria	String	
	Lower limit	String	
	Upper limit	String	The upper limit must be the greater of the lower limit.
Actividades generales necesarias para obtener los resultados	1.The user is allowed to enter the search criteria 2.The user is allowed to enter the search lower limit 3.The user is allowed to enter the search upper limit 4.If the search criteria is numeric, the system will parse the search values to Integers. 5.The system will search the order in the order container structure using the binary search algorithm according to the search criteria.		
Resultado o postcondición	The system will show the order if it was found or will show an alert if the product was not found.		
Salidas	Nombre Salida	Tipo de dato	Condición de selección o repetición
	Order found Info	String	If the order was found
	Warning message	String	If the order wasn't found

Escenarios

Nombre	Clase	Escenario
setupStage2	OrderTest	<p>Se crea una Order = NameClient = “Sánchez”, priceTotal = 0.2, date = 11/04/2022</p> <p>Un objeto de la clase Order con 1 producto:</p> <ul style="list-style-type: none"> - Producto 1: Name=Diaz, description= “Roncador”, price = 0.2“, amount = 1000, timesPurchased = 0, category: TOYS AND GAMES
setupStage3	ProductTest	Se hace un escenario vacio para la clase Product
setupStage4	Controllertest	<p>Se tiene un objeto contenedor de order con 3 order las cuales son:</p> <ul style="list-style-type: none"> - Producto 1: name= “Guante”, description= “Buenardo, buen relieve”, price = 12.50“, amount = 10, timesPurchased = 5, category: CLOTHES - Producto 2: name= “Barnie”, description= “Pequeño y suave”, price = 15“, amount = 0, timesPurchased = 5, category: ENTERTAINMENT. - Producto 3: name= “Camisa”, description= “Grande”, price = 10“, amount = 11, timesPurchased = 55, category: CLOTHES - Producto 4: name= “Movie”, description= “Aspera”, price = 10“, amount = 0, timesPurchased = 55, category: ENTERTAINMENT. - Product 5: name= “Tablet”, description = “Electronic device”, price = 250, availableAmount = 15, category = ELECTRONIC, purchasedTimesAmount = 3.
setupStage5	Controllertest	<p>Se tiene un objeto contenedor de productos con 4 productos los cuales son:</p> <ul style="list-style-type: none"> - Atributos de la Order 1: clientName = “Diaz”, priceTotal = 23.50, date = 1/10/2022 <ul style="list-style-type: none"> - Producto 1: Name= “Guante”, description= “Buenardo, buen relieve”, price = “23.50“, amount = 10, timesPurchased = 5, category: CLOTHES - Atributos de la Order 2: clientName = “Diaz”, priceTotal = 23.50, date = 1/12/2022 <ul style="list-style-type: none"> - Producto 1: Name= “Camisa”, description= “Grande”, price = 23.50, amount = 11, timesPurchased = 55, category: CLOTHES - Atributos de la Order 3: clientName = “Ñerby Hernandez 🍆”, priceTotal = 243.50, date = 4/10/2022 <ul style="list-style-type: none"> - Producto 1: Name= “Tablet”, description= “”, price = 243.50, amount = 11, timesPurchased = 55, category:

		TECHNOLOGY
setupStage6	OrderTest	Se hace un escenario vacío para la clase Order

Diseño de casos de prueba

R1: Register a product

Objetivo de la Prueba: Verificar que el registro de un producto se realice correctamente, teniendo en cuenta que no debe admitir valores negativos para el precio, cantidad disponible y el número de veces comprado.

Clase	Método	Escenario	Valores de Entrada	Resultado esperado
Product	registerProduct	setupStage3	Name= "Barnie", Description= "Pequeño y suave", price = -10", amount = 0, timesPurchased = 5	No se crea el producto y se lanza una exception que indica que el precio no puede ser negativo
Product	registerProduct	setupStage3	Name= "Barnie", Description= "Pequeño y suave", price = 10", amount = -15, timesPurchased = 5	No se crea el producto y se lanza una exception que indica que la cantidad no puede corresponder a un valor negativo.
Product	registerProduct	setupStage3	Name= "Barnie", Description= "Pequeño y suave", price = 10", amount = 15, timesPurchased = -1	No se crea el producto y se lanza una exception que indica que las veces vendidas no puede corresponder a un valor negativo.
Controller	registerProduct	setupStage4	Name= "Barnie", Description= "Pequeño y suave", price =	Se logra crear el producto, lo cual se evidencia al aumentar el tamaño del contenedor en 1, pasando de 4 a 5

			10“, amount = 15, timesPurchased = 1	
--	--	--	---	--

R2: Make an order

Objetivo de la Prueba: Verificar que el registro de un pedido se realice correctamente.				
Clase	Método	Escenario	Valores de Entrada	Resultado esperado
Order	registerOrder	setupStage4	Name= “Ricardo Urbina”, ListProducts= ,	No se crea el pedido y se lanza una exception que indica que la lista de pedidos no puede ser negativa
Order	registerOrder	setupStage4	Name= “Ricardo Urbina”, ListProducts= Producto 1, Producto 3,	Se espera que el producto calcule el precio respecto a los pedidos y cree un pedido con los siguientes atributos Name= “Ricardo Urbina”, ListProducts= Producto 1, Producto 3, price =22.50, fecha = \$actual en formato DD/MM/AA\$
Order	registerOrder	setupStage4	Name= “Ricardo Urbina”, ListProducts= Producto 1, Producto 3,	Se espera que el pedido sea creado y el los atributos del producto 1 sea actualiza en “amount = 9” y “timesPurchased = 6” “y producto 3 sea “amount = 10“ y “timesPurchased = 56”
Order	registerOrder	setupStage4	Name= “Ricardo Urbina”, ListProducts= Producto 1, Producto 2,	No se crea el pedido y se lanza una exception que indica que la cantidad del producto dos no es suficiente

R3: Increase the inventory of a product.

Objetivo de la Prueba: Verificar que los métodos increaseStock de la clase Product funcionen correctamente, adicionando el stock pasado por parámetro al stock actual del producto.

Clase	Método	Escenario	Valores de Entrada	Resultado esperado
Product	increaseStock	setupStage4	Producto a modificar: Producto 5 stockToAdd = 5	El stock del producto se ha actualizado a 20
Product	increaseStock	setupStage4	Producto a modificar: Producto 5 stockToAdd = -5	El stock del producto queda en 15. Lanza Exception y el mensaje dice que no se pueden usar números negativos.
Product	increaseStock	setupStage4	Producto a modificar: Producto 5 stockToAdd = 0	El stock del producto queda en 15.

R4: Search product by unique attribute

Objetivo de la Prueba: Comprobar el correcto funcionamiento de los métodos de búsqueda por un único atributo sin ser intervalo de los productos, para ello se testea la búsqueda por cada atributo cuando no se encuentra coincidencias y cuando se encuentra una o varias.

Clase	Método	Escenario	Valores de Entrada	Resultado esperado
Controller	searchProductByName	setupStage4	name = “Guante”	Se debe indicar que el producto fue encontrado y retornar el producto “Guante”.
Controller	searchProductByPrice	setupStage4	price = “10”	Se debe indicar que el producto fue encontrado y retornar el producto “Barnie”.
Controller	searchProductByCategory	setupStage4	category = ENTERTAINMENT.	Se debe indicar que los productos fueron encontrados y retornar “Barnie” y “Movie”.
Controller	searchProductByAmountAvailable	setupStage4	amount = 0	Se debe indicar que el producto fue encontrado y retornar el producto “Barnie”.
Controller	searchProduct	setupStage	name = “Ropita”	Se debe indicar que el producto no fue

	ctByName	4		encontrado.
Controller	searchProductByPrice	setupStage 4	price = 10	Se debe indicar que los productos fueron encontrados y retornar “Camisa” y “Movie”.

R5: Search product by a range

Objetivo de la Prueba: Comprobar el correcto funcionamiento de los métodos de búsqueda por rango de los productos, para ello se testea la búsqueda por cada atributo cuando no se encuentra coincidencias y cuando se encuentra una o varias.

Clase	Método	Escenario	Valores de Entrada	Resultado esperado
Controller	searchRangeProductByName	setupStage 4	lowerLimit = “A” upperLimit = “D”	Se espera que retorne: <ul style="list-style-type: none"> - producto 2: name=“Barnie” - producto 3: name=“Camisa”
Controller	searchRangeProductByPrice	setupStage 4	lowerLimit = 9 upperLimit = 11	Se espera que retorne: <ul style="list-style-type: none"> - producto 1: name=“Guante”, precio = 10 - producto 3: name=“Camisa”, precio = 10
Controller	searchRangeProductByAmountAvailable	setupStage 4	lowerLimit = 10 upperLimit = 14	Se espera que retorne: <ul style="list-style-type: none"> - producto 1: name=“Guante”, amount= 10 - producto 3: name=“Camisa”, amount= 11
Controller	searchRangeProductByName	setupStage 4	lowerLimit = “X” upperLimit = “Z”	Se indica no hay productos en ese intervalo
Controller	searchRangeProductByPrice	setupStage 4	upperLimit = 50 lowerLimit = 50	Lanza Exception y el mensaje dice que el límite inferior de búsqueda no puede ser igual al superior.
Controller	searchRangeProductByPrice	setupStage 4	upperLimit = 20 lowerLimit = 50	Lanza Exception y el mensaje dice que el límite inferior de búsqueda mayor al superior.
Controller	searchRangeProductByAmountAvailable	setupStage 4	lowerLimit = -10 upperLimit = -2	Lanza Exception y el mensaje dice que el límite inferior de búsqueda y el superior deben ser mayores a 0.

R6: Search order by unique attribute

Objetivo de la Prueba: Comprobar el correcto funcionamiento de los métodos de búsqueda de los pedidos por un único atributo sin ser intervalo, para ello se testea la búsqueda por cada atributo, cuando no se encuentra coincidencias y cuando se encuentra una o varias.

Clase	Método	Escenario	Valores de Entrada	Resultado esperado
Controller	searchOrderBynameClient	setupStage5	NameClient : “Diaz”	Se obtiene la orden 1 y 2.
Controller	searchOrderByPriceTotal	setupStage5	priceTotal : “243.50”	Se obtiene la orden 3.
Controller	searchOrderByDate	setupStage5	date: “4/10/2022”	Se obtiene la orden 3.
Controller	searchOrderBynameClient	setUpstage5	NameClient: “Lucila T.”	No se obtiene ninguna orden.
Controller	searchOrderByDate	setUpstage5	date: “12/12/2024”	No se obtiene ninguna orden.
Controller	searchOrderByPriceTotal	setupStage5	priceTotal: 10223	No se obtiene ninguna orden.

R7: Search order by range

Objetivo de la Prueba: Verificar que los métodos searchOrderByRange de la clase Controller funcionen correctamente, buscando el Order solicitado a través de un rango numérico de precio pasado por parámetro.

Clase	Método	Escenario	Valores de Entrada	Resultado esperado
Controller	searchOrderByPriceRange	setupStage5	upperLimit = 50 lowerLimit = 20	Se muestran las siguientes order: Atributos de la Order: NameClient = "Diaz", priceTotal = 23.50, date = 1/10/2022 Atributos de la Order: NameClient = "Diaz", priceTotal = 23.50, date = 1/12/2022
Controller	searchOrderByPriceRange	setupStage5	upperLimit = 10 lowerLimit = 20	Lanza Exception y el mensaje dice que el límite inferior de búsqueda no puede ser mayor al superior.
Controller	searchOrderByPriceRange	setupStage5	upperLimit = 50 lowerLimit = 50	Lanza Exception y el mensaje dice que el límite inferior de búsqueda no puede ser igual al superior.
Controller	searchOrderByPriceRange	setupStage5	upperLimit = "abc" lowerLimit = "abc"	Lanza Exception y el mensaje dice que el rango está en un formato incorrecto, los datos deben ser numéricos.
Controller	searchOrderByBuyerNameRange	setupStage5	lowerLimit = "A" upperLimit = "D"	Se espera que retorne: - order 1: clientName ="Diaz" - order 2: clientName ="Diaz"
Controller	searchOrderByBuyerNameRange	setupStage5	lowerLimit = "D" upperLimit = "O"	Se espera que retorne: - order 1: clientName ="Diaz" - order 2: clientName ="Diaz" - order 3: clientName ="Ñerby Hernandez"
Controller	searchOrderByBuyerNameRange	setupStage5	lowerLimit = "M" upperLimit = "O"	Se espera que retorne: - order 3: clientName ="Ñerby Hernandez"
Controller	searchOrderByBuyerNameRange	setupStage5	lowerLimit = "X" upperLimit = "Z"	Se indica no hay orders en ese intervalo