

**SOFTWARE ENGINEERING PROBLEM SPECIFICATION TABLE:**

<b>CLIENT</b>	Mercado Libre
<b>USER</b>	Sellers of Mercado Libre
<b>FUNCTIONAL REQUIREMENTS</b>	<ol style="list-style-type: none"><li>1. Register a new product</li><li>2. Make an Order</li><li>3. Remove from the inventory all elements purchased in an order</li><li>4. Add elements to the inventory</li></ol>
<b>PROBLEM CONTEXT</b>	<p>Mercado Libre has hired us to develop an application for online sales. For this purpose, they want the application to allow adding and removing products. They also want a functionality to be implemented that allows users to search for products based on various filters, such as name, description, price, available quantity, category, and the number of times purchased.</p>
<b>Non-FUNCTIONAL REQUIREMENTS</b>	<ol style="list-style-type: none"><li>1. The program must allow the user to add as many products as he wants.</li><li>2. The search engine must use the binary search algorithm.</li><li>3. The program to implement data persistence using JSON serialization model</li><li>4. The project must be developed using Test Driven Development</li></ol>

**Functional Requirements Analysis Table:**

Name or ID	1. Register a new product		
Summary	The system must be able to enter information about a new product		
Inputs	Input name	Data type	Selection or repetition condition
	name	String	the string cannot be empty
	description	String	the string cannot be empty
	price	double	price needs to be greater than 0
	amount	int	amount needs to be greater than 0
	category	Category	category must be within the given range
General activities necessary to achieve the results	The system will create the product object and depending on its category, it will be added to one database or another.		
Result or postcondition	The system will verify that the information is valid, and if it is, it will be added to the database and a message will be displayed informing that the product was added correctly, otherwise a message will be sent informing of the problem.		
Outputs	Output name	Data type	Selection or repetition condition
	Message	String	

Name or ID	2. Make an Order		
Summary	The system must allow the user to make an order. The order may have one or many products.		
Inputs	Input name	Data type	Selection or repetition condition
	buyerName	String	the string cannot be empty
	productsList	ArrayList<String>	While the input string is not equal to 0, the process of adding products will be repeated.  The requested quantity of the product must match the inventory.
	purchaseDate	LocalDate	
General activities necessary to achieve the results	The system verifies that the data is correct, as well as checking if the requested quantities are consistent with the inventory, and then generates the order, saving it in the database		
Result or postcondition	If there is no issue, the system will return a message confirming that the order has been successfully added and add the order to the database. Otherwise, it will return an error message.		
Outputs	Output name	Data type	Selection or repetition condition
	message	String	

Name or ID	3. Remove from the inventory all elements purchased in an order		
Summary	When an order is placed, the requested quantity of the product will be deducted from the inventory (only if the requested quantity is consistent).		
Inputs	Input name	Data type	Selection or repetition condition
	productName	String	The product name must match one in the inventory
	amount	int	The quantity must be greater than 0.
General activities necessary to achieve the results	The system will search for the product in the database based on the name and subtract the quantities from the inventory.		
Result or postcondition	The system will verify the requested quantity of the product. If it is coherent, it will subtract it from the inventory; otherwise, it will throw an exception.		
Outputs	Output name	Data type	Selection or repetition condition
	eMessage	AmountException	If the requested quantity is inconsistent with the inventory

Name or ID	4. Add elements to the inventory		
Summary	The system will allow the user to select a product and increase its quantity in stock.		
Inputs	Input name	Data type	Selection or repetition condition
	productName	String	The product name must match one in the inventory
	amount	int	The quantity must be greater than 0.
General activities necessary to achieve the results	The system will search for the product in the database based on the name and increase its stock level.		
Result or postcondition	The system will search for the product and add the quantities, if the quantities to be added are negative it will throw an exception		
Outputs	Output name	Data type	Selection or repetition condition
	eMessage	negativeException	only if the amount is lesser than 1.

Name or ID	5. The search engine of products must allow the user to search by range for the numbers or by interval for the strings		
Summary	The system must be able to filter the information by a range or an interval given by the user.		
Inputs	Input name	Data type	Selection or repetition condition
	minorLimit	Integer	If the user is filtering by numbers
	mayorLimit		
	initLetter	String	If the user is filtering by a letter
	initPrefix		If the user is filtering by a prefix
	lastLetter		If the user is filtering by a letter
	lastPrefix		If the user is filtering by a prefix
General activities necessary to achieve the results	The system must search all elements that are in the interval or range and return allthese elements		
Result or postcondition	An array with the elements is ready to be sorted and showed to the user		
Outputs	Output name	Data type	Selection or repetition condition
	array	Product[]	If the user has entered an interval or range to filter the information

Name or ID	6. The system must allow the user to search for any products by a given filter.		
Summary	The system must have a search engine for the products. The search engine must allow the user to search for any products by their name, price, category, and number of times bought.		
Inputs	Input name	Data type	Selection or repetition condition
	filter	integer	If the user is searching for any products using a given filter
	array	Product[]	If the system has found products whit the previous filter
General activities necessary to achieve the results	when the system knows what is the filter that will use, it will will organize the products to be shown.		
Result or postcondition	The system has extracted the required information and has sorted that to be shown.		
Outputs	Output name	Data type	Selection or repetition condition
	products	String	If the user is searching for any products and they exist in the system
	error	Exception	If the products required are not saved in the system yet

Name or ID	7. The system must allow the user to search for any orders by a given filter.		
Summary	The system must have a search engine for the made orders. The search engine must allow the user to search for any orders by their buyer name, final price, and date of purchase.		
Inputs	Input name	Data type	Selection or repetition condition
	filter	Integer	If the user is searching for any orders using a given filter
General activities necessary to achieve the results	when the system knows what is the filter that will use, it will search the required information and will organize that to be shown.		
Result or postcondition	The system has extracted the required information and has sorted that to be shown.		
Outputs	Output name	Data type	Selection or repetition condition
	orders	String	If the user is searching for any orders and they exist in the system
	error	Exception	If the orders required are not saved in the system yet

Name or ID	8. Persistence		
Summary	The system must be able to persist the information that an user has inserted in its although the program will be closed.		
General activities necessary to achieve the results	For that, when the program will be closed all changes or new information will be saved in a txt document to be charged when the user open the program again.		
Result or postcondition	All information will be saved to be charged when the user open the program again.		