

Project developers:

Requirements Analysis Table.

User	MIO users
Functional requirements	FR1 – Search best route. FR2 – Search routes. FR3 – develop graphical user interface.
Context of the problem	<p>In the city of Cali, the Integrated Mass Transportation System of the West (Transporte Masivo de Occidente, MIO) is widely used by the population, covering 93% of the supply and mobilizing approximately 490 thousand passengers per year.</p> <p>Given the high level of use and the importance of MIO as a means of transport for Cali residents, there is a need to find an efficient solution to get from one station to another without having to make unnecessary journeys. In this sense, it seeks to determine the shortest path between two MIO stations, considering the structure of the routes and the pre-established areas.</p>
Non-functional requirements	
Product requirements	

Functional Requirements Analysis Tables.

Name or identifier	FR1 – Search best route.		
Summary	Given an origin station and a destination station, the system should utilize Dijkstra's algorithm on graphs to determine and display the most optimal routes, including their names and the stations they traverse.		
Inputs	Input name	Datatype	Selection or repetition condition
	OriginStation	String	The entered string must be different from "" or null.

	DestinationStation	String	The entered string must be different from "" or null.
General activities necessary to obtain the results	<ol style="list-style-type: none"> 1. Read and validate the user's input. 2. Retrieve the network graph of MIO stations and their connections. 3. Apply Dijkstra's algorithm on the graph to find the shortest path considering the distances or travel times. 		
Result or post-condition	A list with the information about the best routes and the stations it traverses.		
Outputs	Output name	Datatype	Selection or repetition condition
	bestRoute.	String	The entered Station must be included in the graph. The station must have at least one way to connect them. The resulting route must be the most efficient.

Name or identifier	FR2 – Search routes.		
Summary	Given an origin station and a destination station, the system should display a list of all the routes that connect them, allowing the user to choose any route, even if it is not the most optimal one.		
Inputs	Input name	Datatype	Selection or repetition condition
	OriginStation	String	The entered string must be different from "" or null.
	DestinationStation	String	The entered string must be different from "" or null.

General activities necessary to obtain the results	<ol style="list-style-type: none"> 1. Read and validate the user's input. 2. Retrieve the network graph of MIO stations and their connections. 3. Use DFS algorithm to find all the possible routes that connects the stations. 		
Result or post-condition	A list with all the possible routes between the origin and the destination stations.		
Outputs	Output name	Datatype	Selection or repetition condition
	routesList	String	<p>The stations must have at least one way to connect them.</p> <p>The station and the routes that connect them must be in the graph.</p>

Name or identifier	FR3 – develop graphical user interface.		
Summary	<p>The system must implement a graphic interface that provides options for the user to perform specific actions. The system should capture and process the user's actions and parameters entered, perform the corresponding operations, and display the results in the interface. This way, the user can intuitively interact with the system and obtain the desired responses or results.</p>		
Inputs	Input name	Datatype	Selection or repetition condition
	None		
General activities necessary to obtain the results	<ol style="list-style-type: none"> 1. Capture the user action. 2. Validate and process the user's selected action when needy. 3. Perform the operation according to the selection of the user. 4. Show the result of the operation performed in the system. 		
Result or post-condition	The system responds to the user's action showing the result of the operation or an error message when needy.		

Outputs	Output name	Datatype	Selection or repetition condition
	Result		If the user's actions are valid.