**Lista de Requerimientos Integradora 2**

**Presentado por Juan Camilo Amorocho, Joshua Rivera y David Troya**

**Problem Definition**

The central problem lies in the lack of a game that allows one or more players to interact with a graph of at least 50 vertices and 50 edges, applying at least two specific graph algorithms. This lack hinders the practice and understanding of academic concepts related to graphs. The required solution is to develop a game that meets the academic requirements, is intuitive and pedagogical, and gives players the opportunity to apply graph algorithms effectively in a playful and educational environment.

| CLIENT | User |
| --- | --- |
| USER | User |
| FUNCTIONAL REQUIREMENTS | **Rf1-** Creation of a Graph  **Rf2-** Single and multiplayer gameplay  **Rf3**- Maps Selection  **Rf4-** Player Management  **Rf5**- Player Score  **Rf6**- Graph Algorithms |
| NON FUNCTIONAL REQUIREMENTS | **Rnf-** Simple and easy interface that allows players to perform their processes with ease from their cell phone or web browser.  **Rnf-** The system must have adequate performance  **Rnf-** The system must have good maintainability  **Rnf-** The system must be scalable to accommodate an increasing number of players without performance degradation. |

| Name | **Rf1-** Creation of a graph | | |
| --- | --- | --- | --- |
| Summary | The system must allow the creation of a graph with a minimum of 50 vertices and 50 edges, which will allow the game to be modeled using this data structure. This requirement is essential to meet the project criteria and ensure that the game has the required complexity in terms of vertices and edges. | | |
| Inputs | Name | Type | Condition |
| Minimum number of vertices and edges. | Int | The system must allow the user to define the data structure that will represent the network. This input is essential to determine how vertices and edges will be stored and manipulated in the game. |
| PostCondition | After the user specifies the minimum number of vertices and edges, and the data structure to represent the network, the system will create a valid network that meets the requirements of at least 50 vertices and 50 edges. | | |
| Outputs | Name | Type | Condition |
| ConfirmationMessage | String | Generated after a successful election to inform the user that the process has been successfully completed. |

| Name | **Rf2-** Single and multiplayer gameplay | | |
| --- | --- | --- | --- |
| Summary | The system must provide a user login functionality, allowing users to enter their nickname and password for authentication. The password input field should hide the characters entered for security reasons. Upon clicking the "Login" button, the system should verify the provided credentials and grant access if they are correct. If successful, the system should proceed to the user's session. | | |
| Inputs | Name | Type | Condition |
| Number of Players | Int | The system should allow a variable number of players to participate in the game, with a minimum requirement of 1 player for single-player mode and a minimum of 2 players for multiplayer mode. |
| Name of Players | Int | In multiplayer mode, the system should allow each player to enter their name. The number of names in the array should correspond to the number of players in the game. |
| Game Mode Selection | Enum | The system should provide an option for users to select either "SinglePlayer" or "Multiplayer" game mode. The game mode selected will determine the number of players and the gameplay experience. |
| PostCondition | After selecting the game mode (SinglePlayer or Multiplayer) and specifying the number of players, the system will provide an interface for player interaction. In multiplayer mode, the system will collect and display the names of all participating players. The gameplay will commence according to the selected mode, with players able to interact with the game using the provided user interface elements. | | |
| Outputs | Name | Type | Condition |
| ConfirmationMessage | String | Generated after a successful election to inform the user that the process has been successfully completed. |

| Name | **Rf3-** Maps Selection | | |
| --- | --- | --- | --- |
| Summary | Users can select from different maps, regions, or places, which involves the integration of map-related features such as map display, location search, and interaction with geospatial information. This requirement can be relevant for applications related to navigation, tourism, logistics, and various other areas where location selection is essential. | | |
| Inputs | Name | Type | Condition |
| Map Selection | Int | Users can select a map from a predefined list of available maps. The dropdown should be populated with map options, and the user's selection is a required input for the system to proceed. |
| PostCondition | After the user selects a map from the dropdown, the system will have the user's choice recorded as the selected map for the game session. The system is ready to proceed with the selected map for gameplay, and any relevant game settings or configurations are updated to reflect the chosen map. | | |
| Outputs | Name | Type | Condition |
| ConfirmationMessage | String | Generated after a successful election to inform the user that the process has been successfully completed. |

| Name | **Rf4-** Player Management | | |
| --- | --- | --- | --- |
| Summary | This requirement involves the management of player-related data and interactions within the application. It includes features for registering and logging in players, managing player information, updating scores, tracking player progress, and allowing players to manage their profiles, such as personal information and avatars. | | |
| Inputs | Name | Type | Condition |
| Player information | String | The system should accept and manage player information, including player names, scores, game progress, and other relevant details. |
| Player Profile Management |  | The system should allow players to manage their profiles, including updating personal information, changing passwords, or customizing avatars. |
| Player register | String | The system should allow users to register as players, requiring them to input their desired usernames, passwords, and other relevant details as needed. |
| Player Login | String | The system should provide a login functionality where registered players can enter their usernames and passwords to access their accounts. |
| PostCondition | After a player registers or logs in, the system stores their player information and allows them to access and manage their profiles. The system also tracks player progress and updates scores based on in-game performance. Player data remains accurate and up to date, ensuring that players have a seamless and personalized experience within the application. | | |
| Outputs | Name | Type | Condition |
| ConfirmationMessage | String | Generated after a successful election to inform the user that the process has been successfully completed. |

| Name | **Rf5-** Player Score | | |
| --- | --- | --- | --- |
| Summary | This requirement pertains to the management and tracking of player scores within the game. It includes recording and storing player scores, implementing a scoring algorithm, and potentially tracking and displaying high scores for a competitive element in the game. | | |
| Inputs | Name | Type | Condition |
| Player Score | Int | The system should accept and record player scores as they progress in the game. |
| Score Storage | Array | The system should store and manage player scores, associating them with player profiles and game progress. |
| PostCondition | After playing the game, player scores are accurately calculated, stored, and associated with player profiles. The system updates scores based on in-game actions, achievements, or performance. Optionally, high scores are tracked and available for competitive comparison, enhancing the gaming experience. Player score data remains reliable and up to date. | | |
| Outputs | Name | Type | Condition |
| ConfirmationMessage | String | Generated after a successful election to inform the user that the process has been successfully completed. |