|  |  |
| --- | --- |
| Cliente | Andrés Aristizábal |
| Usuario | Users |
| Requerimientos funcionales | *Card Distribution: The system must be able to randomly and fairly distribute a specific number of cards to each player at the beginning of the game.*  *Play Cards: Players must be able to select and play a card from their hand during their turn, as long as it complies with the rules of matching color, number, or symbol with the top card on the discard pile.*  *Draw Cards: If a player cannot play any card during their turn, the system must allow them to automatically draw a card from the deck.*  *Rule Verification: The system must continuously verify that players' moves comply with the rules of the Uno game, such as playing a valid card or drawing a card when necessary.*  *Game End: The system must detect when a player runs out of cards in hand and declare the first player to be out of cards as the winner of the game.*  *User Interface: The system must provide an intuitive user interface that displays the game state, the cards in each player's hand, the cards on the discard pile, and the available options on each turn.*  *Special Card Management: The system must be able to handle the special cards of the Uno game, such as Change Color, Draw 2, Reverse, and Skip, and apply their effects correctly during the game.*  *Player Order Update: After playing a special card that affects the order of play, the system must automatically rearrange the order of players according to the rules of the game.*  *These are just some examples of functional requirements for the Uno game. Depending on the complexity and specific features of the game, there may be more detailed functional requirements.* |
| Contexto del problema | The problem consists of implementing a Uno card game in Java, which requires the use of various data structures such as stacks, queues, hash tables, and priority queues to manage different aspects of the game. The standard rules of the Uno game must be followed, where players try to get rid of cards in their hands by playing cards that match in color, number, or symbol with the top card on the discard pile. |
| Requerimientos no funcionales | **Efficiency**: The game must run efficiently, with fast response times for a smooth gaming experience.  **Security**: The code must follow secure coding standards to avoid vulnerabilities and security errors.  **Usability**: The user interface must be intuitive and easy to use, allowing players to interact with the game easily.  **Robustness**: The game must be robust and handle unexpected situations or errors during execution correctly.  **Scalability**: The game must be able to handle a variable number of players and dynamically adapt to different game configurations. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | distributeCardsToPlayers | | | |
| Resumen | Distributes a specific number of cards to each player from a deck. | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| Number of players | Int | |  |
| Deck | String | |  |
|  |  | |  |
| Resultado o Postcondición | Each player has a specific set of cards assigned. | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
|  | |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | playCard | | | |
| Resumen | :Allows a player to play a card on their turn. | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| Card to play,. | Int | |  |
| top card on the discard pile | Int | |  |
|  |  | |  |
| Resultado o Postcondición | The game state is updated according to Uno rules | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
|  | |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | drawCard | | | |
| Resumen | Allows a player to draw a card from the deck if they cannot play any card. | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| Deck of cards. | Int | |  |
|  | Int | |  |
|  |  | |  |
| Resultado o Postcondición | The player receives a new card in their hand. | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
| The drawn card. | |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | checkEndGame | | | |
| Resumen | Checks if a player has run out of cards in hand to determine the end of the game. | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| Players' hands. |  | |  |
|  |  | |  |
|  |  | |  |
| Resultado o Postcondición | Determines if there is a winner or if the game continues. | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
| Winning player (if any) or indication that the game continues. | |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | updatePlayersOrder | | | |
| Resumen | Updates the order of players after a special card is played. | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| Priority queue with the order of players. |  | |  |
|  |  | |  |
|  |  | |  |
| Resultado o Postcondición | The order of players is rearranged according to Uno rules. | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
|  | |  |  |