# Juan Diego Gil

Esue Torres Morales

Cesar Augusto Jimenez Reina

# Identificación del problema y análisis de requerimientos

## Caso de Estudio:

|  |  |
| --- | --- |
| Cliente | Empresa de juegos |
| Usuario | Los jugadores de todas las edades que disfrutan de juegos de cartas digitales. |
| Requerimientos funcionales | RF\_1 Distribución de cartas.  RF\_2 Jugar Carta.  RF\_3 Robar Carta.  RF\_4 Cartas Especiales.  RF\_5 Crear Interfaz de usuario.  RF\_6 Comenzar juego.  RF\_7 Terminar Juego.  RF\_8 Revisar Cartas.  RF\_9 Manejar turno de los jugadores. |
| Contexto del problema | El juego “Uno” es un juego de cartas popular que se juega con una baraja especial. El objetivo del juego es ser el primer jugador en quedarse sin cartas. Para digitalizar este juego, se requiere un sistema que pueda manejar la lógica del juego y la interacción con el usuario de manera eficiente y atractiva. |
| Requerimientos no funcionales | 1. **Rendimiento**: El juego debe responder a las acciones del usuario sin retrasos perceptibles. 2. **Usabilidad**: La interfaz de usuario debe ser intuitiva y fácil de usar para personas de todas las edades. 3. **Escalabilidad**: El sistema debe ser capaz de soportar un número creciente de usuarios simultáneos. 4. **Seguridad**: El juego debe proteger la información del usuario y garantizar que las partidas no puedan ser manipuladas por terceros. 5. **Compatibilidad**: El juego debe ser compatible con diferentes sistemas operativos donde se pueda ejecutar Java. 6. **Mantenibilidad**: El código debe estar bien documentado y estructurado para facilitar futuras actualizaciones o correcciones. 7. **Internacionalización**: El juego debe soportar la localización para diferentes idiomas y regiones. 8. **Accesibilidad**: El juego debe ser accesible para usuarios con discapacidades, incluyendo opciones de alto contraste y soporte para lectores de pantalla. 9. **Estética**: El diseño visual del juego debe ser atractivo y mantenerse coherente con la temática de “Uno”. 10. **Portabilidad**: El juego debe ser fácilmente trasladable entre distintas plataformas que soporten Java sin necesidad de cambios significativos en el código. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | RF\_1 Distribución de cartas. | | | |
| Resumen | Al inicio del juego, el sistema debe repartir 7 cartas a cada jugador de una baraja de 108 cartas. | | | |
| Entradas | Nombre entrada | Tipo de dato | | Condición valores válidos |
| No aplica | - | |  |
|
|
|
| Resultado o Postcondición | Cada jugador tiene 7 cartas en su mano. | | | |
| Salidas | Nombre salida | | Tipo de dato | Formato |
| playerCard | | List | Cada objeto Carta debe ser único y parte de una baraja de 108 cartas. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | RF\_2 Jugar Carta. | | | |
| Resumen | En su turno, el jugador debe poner una carta en el montón de descarte que coincida en color, número o símbolo con la carta superior. | | | |
| Entradas | Nombre entrada | Tipo de dato | | Condición valores válidos |
| No aplica |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
| Resultado o Postcondición | La carta seleccionada se coloca en el montón de descarte. | | | |
| Salidas | Nombre salida | | Tipo de dato | Formato |
| SelectedCard | | Card | Debe coincidir en color, número o símbolo con la carta superior del montón de descarte. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | RF\_3 Robar Carta. | | | |
| Resumen | Si el jugador no puede jugar, debe robar una carta del mazo. | | | |
| Entradas | Nombre entrada | Tipo de dato | | Condición valores válidos |
| No aplica |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
| Resultado o Postcondición | El jugador añade una carta a su mano. | | | |
| Salidas | Nombre salida | | Tipo de dato | Formato |
| robbedCard | | Card | Debe ser una carta válida dentro de la baraja de 108 cartas. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | RF\_4 Cartas Especiales. | | | |
| Resumen | El juego incluye cartas especiales como cambio de color, roba 2, revertir y salto, cada una con una acción específica. | | | |
| Entradas | Nombre entrada | Tipo de dato | | Condición valores válidos |
| specialCard | Card | | Debe ser una de las cartas especiales válidas (cambio de color, roba 2, revertir, salto). |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
| Resultado o Postcondición | Se ejecuta la acción especial correspondiente a la carta. | | | |
| Salidas | Nombre salida | | Tipo de dato | Formato |
| specialAction | | String | Debe describir la acción especial realizada por la carta jugada. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | RF\_5 Crear Interfaz de usuario. | | | |
| Resumen | El sistema debe proporcionar una interfaz gráfica para que los jugadores interactúen con el juego. | | | |
| Entradas | Nombre entrada | Tipo de dato | | Condición valores válidos |
| No aplica |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
| Resultado o Postcondición | La interfaz gráfica se muestra al usuario. | | | |
| Salidas | Nombre salida | | Tipo de dato | Formato |
| graphicInterface | | GUI | Debe ser visualmente comprensible y permitir la interacción con el juego. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | RF6\_Comenzar Juego | | | |
| Resumen | El sistema debe ofrecer una opción para iniciar una nueva partida de Uno. | | | |
| Entradas | Nombre entrada | Tipo de dato | | Condición valores válidos |
| No aplica |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
| Resultado o Postcondición | Se establece el estado inicial del juego y comienza la partida. | | | |
| Salidas | Nombre salida | | Tipo de dato | Formato |
| initialState | | State | Debe representar el estado inicial de una nueva partida de Uno. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | RF7\_Terminar Juego | | | |
| Resumen | El sistema debe identificar cuando un jugador se queda sin cartas y declararlo ganador, terminando la partida. | | | |
| Entradas | Nombre entrada | Tipo de dato | | Condición valores válidos |
| No aplica |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
| Resultado o Postcondición | La partida termina y se declara un ganador. | | | |
| Salidas | Nombre salida | | Tipo de dato | Formato |
| winner | | String | Debe ser el nombre del jugador que se quedó sin cartas primero. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | RF8\_ **Revisar Cartas** | | | |
| Resumen | Los jugadores deben poder ver sus cartas en cualquier momento durante la partida.. | | | |
| Entradas | Nombre entrada | Tipo de dato | | Condición valores válidos |
| No aplica |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
| Resultado o Postcondición | El jugador conoce las cartas que tiene en su mano. | | | |
| Salidas | Nombre salida | | Tipo de dato | Formato |
| handCard | | ListCard | Debe mostrar todas las cartas que el jugador tiene en su mano. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | RF\_9 Manejar Turno de Jugadores | | | |
| Resumen | El sistema debe controlar el orden de los turnos y pasar el turno al siguiente jugador adecuadamente. | | | |
| Entradas | Nombre entrada | Tipo de dato | | Condición valores válidos |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
| Resultado o Postcondición | El turno pasa al siguiente jugador en la secuencia. | | | |
| Salidas | Nombre salida | | Tipo de dato | Formato |
| actualPlayer | | String | Debe indicar el nombre del jugador cuyo turno es jugar. |