# **Identificación del problema y análisis de requerimientos**

## **Caso de Estudio:**

|  |  |
| --- | --- |
| **CLIENTE** | Empresa de desarrollo de videojuegos |
| **USUARIO** | Usuarios que utilicen el juego |
| **REQUERIMIENTOS FUNCIONALES** | RF1: Nueva partida y Crear usuario  RF1.2: Poner una tubería  RF1.3: Simular tubería  RF2: Ver puntaje |
| **CONTEXTO DEL PROBLEMA** | La compañía desarrolladora de juegos busca crear un algoritmo recursivo para fabricar un videojuego que gire en torno a un simulador de conductos o tuberías. El objetivo primordial de este entretenimiento radica en diseñar tres categorías distintas de tuberías para enlazar una fuente (F) y un sumidero (S). El problema del juego se concentra en este tipo de conductos:  **Conducción horizontal**: Habilita el curso del líquido en sentido lateral y se ancla sin intermediarios a la fuente o el sumidero.  **Conducto curvilíneo**: Cambia la orientación del flujo en un ángulo de 90 grados y solo puede encajar con otras conducciones horizontales o verticales. Está vetada su conexión inmediata a la fuente o el sumidero.  **Tubo vertical:** Facilita la progresión vertical del agua y puede conectarse directamente a la fuente o al sumidero.  El programa dispondrá de la alternativa de "Comenzar Partida" en la cual el usuario introducirá su seudónimo. Acto seguido, el sistema concebirá de modo aleatorio un tablero que representará la disposición de la fuente y el sumidero. La misión del participante conllevará establecer las uniones apropiadas entre la fuente y el sumidero.  En el transcurso del juego, el software requerirá que señale la posición deseada para insertar un conducto, precisando la fila y la columna. Una vez que el jugador haya concretado las conexiones con acierto, se le presentará la alternativa de emplear la función de "Simulación" para constatar si las uniones permiten un flujo fluido del agua.  Para concluir, el programa incorporará la función de consultar la puntuación, facultando al jugador para verificar su puntaje total y evaluar su desenvolvimiento en el juego. |
| REQUERIMIENTOS NO FUNCIONALES | RNF1: Usabilidad: La interfaz debe ser intuitiva y fácil de usar, permitiendo a los jugadores comprender las mecánicas del juego y colocar tuberías sin confusiones.  RNF2: Estabilidad: El juego debe ser robusto y confiable, minimizando la posibilidad de fallos, bloqueos o cierres inesperados que puedan afectar la experiencia de juego.  RNF3: El software debe ser capaz de generar cada puntaje en un corto plazo  RNF4: El software debe ser confiable a la hora de poner una tubería en la posición correcta digitada por el usuario. |

|  |  |  |  |
| --- | --- | --- | --- |
| Identificador y nombre | *R-F1: Nueva partida y crear usuario-* | | |
| Resumen | El programa debe permitirle al usuario la funcionalidad de iniciar una nueva partida, creando un perfil con un NickName valido para cada jugador, para que asi una vez se haya registrado el usuario, el software genera automática y aleatoriamente un tablero con la ubicación de el drenaje y la fuente. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición valores validos |
| Opcion | Int | Solo usar números enteros positivos |
|  | Nombre | String | Tendrá un máximo de 10 caracteres, entre números enteros positivos y letras |
| Resultado o Postcondición | Opcion seleccionada con éxito.  La opción no fue seleccionada, por favor intente de nuevo  Usuario registrado con éxito.  Este usuario no ha podido ser registrado. | | |
| Salidas | Nombre salida | Tipo de dato | Formato |
| Tablero de Juego | String | Tablero 8x8 ubicando la fuente y el drenaje de forma aleatoria. |

|  |  |  |  |
| --- | --- | --- | --- |
| Identificador o nombre | ***-RF1.2-Poner Tubería-*** | | |
| Resumen | El programa le ofrece al usuario la opción de poner tubería, lo que implica ingresar la posición (fila y columna), junto con el tipo de tubería. El programa imprime en pantalla el tablero con la edición realizada y nuevamente el menú, puesto que el usuario puede editar una posición del tablero las veces que desee. | | |
| Entradas | Nombre de Entrada | Tipo de Dato | Condición valores validos |
| Opcion | Int | Solo usar números enteros positivos |
|  | Primera posición | Int | (Fila) |
|  | Segunda posición | Int | (Columna) |
| Resultado o Postcondición | Opcion seleccionada correctamente | | |
| Salidas | Nombre salida | Tipo de dato | Formato |
| Imprime tablero y menú nuevamente | String | Imprime tablero [8]x[8] mostrando los cambios hechos con el menú para saber la opción que realiza nuevamente el jugador. |

|

|  |  |  |  |
| --- | --- | --- | --- |
| Identificador o Nombre | ***-RF1.3-Similar Tubería-*** | | |
| Resumen | El software le permite al usuario seleccionar la opción de simular para verificar si la tubería esta correctamente construida, en caso de que este bien se muestra, “La solución es correcta” y se termina la partida; pero en caso contrario el programa muestra “La solución no es correcta” y termina la partida. El programa para realizar esto ejecuta un algoritmo recursivo para garantizar que haya un camino continuo entre la fuente y el drenaje. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición valores validos |
| Opcion | Int | Solo usar números enteros positivos |
| Resultado o Postcondición | Opcion seleccionada correctamente  La solución es correcta  La solución no es correcta | | |
| Salidas | Nombre salida | Tipo de dato | Formato |
| Mensaje de validación | String | Tendrá un máximo de 10 caracteres, que verifica la función correcta del método |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | ***-RF2-Ver puntaje-*** | | |
| Resumen | En el programa cuando se termina una partida correctamente, el jugador entra en la tabla de puntajes, la puntuación se calcula según la formula dada que es: Puntos = tuberiasUsadas \* 100-(60-tiempoEnSegunfos) \*10. Estos puntajes se almacenan en un árbol binario de búsqueda y pueden ser vistos si se selección la opción “Ver puntaje” en el menú principal. | | |
| Entrada | Entrada nombre | Tipo de dato | Condición valores validos |
| Opcion | Int | Solo usar números enteros positivos |
| 3Resultado o Postcondición | Opcion seleccionada correctamente | | |
| Salidas | Nombre salida | Tipo de dato | Formato |
| Mensaje Puntos obtenidos | String | Entre números enteros positivos y letras, se muestra en pantalla el total de puntos que ha ganado el usuario en el juego |