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

****

## **Caso de Estudio : Pipe Mania**

| Cliente | **Profesores de APO-2** |
| --- | --- |
| Usuario | Monitor del curso y jugadores |
| Requerimientos funcionales | RF0: Iniciar juego  RF1: Mostrar juego  RF2: Poner una tubería  RF3: Evaluar (Simular) la red de tuberías  RF4: Calcular el puntaje  RF5: Ver el puntaje |
| Contexto del problema | Pipe Mania que consiste en un simulador de tuberías, es un tablero 8x8 que se muestra por consola, en el que se colocan: tuberías verticales (| |), tuberías horizontales (=) y tubería circular (o). La idea del juego es hacer una tubería que llegue a la casilla Desagüe (D) desde la casilla Fuente (F).  Se debe poder guardar puntajes de los jugadores, detectar si la red tubería es válida, intercambiar tuberías, ver puntajes, mostrar la tubería actualizada y almacenar el nombre de los jugadores y su puntaje más alto. |
| Requerimientos no funcionales | RF0:El código debe ser escrito en Java, se debe usar listas enlazadas, árboles y métodos recursivos.  RF1: Gestión de pruebas: Se debe desarrollar un plan de pruebas que incluya la validación de las funcionalidades del prototipo y su escalabilidad. El plan de prueba debe definir los casos de prueba, los criterios de aceptación y las herramientas para ser utilizado para la ejecución y seguimiento de la prueba.  RF2: Rendimiento: el sistema debe poder manejar un gran volumen de transacciones y operaciones simultáneas sin afectar el rendimiento del sistema o la velocidad de respuesta.  RF3: Usabilidad: el menú que permite al usuario visualizar su puntaje dentro del juego debe ser intuitivo y fácil de usar. Además, la representación gráfica del tablero debe ser clara y atractiva para el usuario. |

Tabla de requerimientos:

| Identificador y nombre | *[RF0-Iniciar juego]* | | | |
| --- | --- | --- | --- | --- |
| Resumen | *El sistema le debe permitir al usuario crear una nueva partida, donde le permitirá ingresar el nickname del usuario. Posteriormente mostrará al usuario un tablero 8x8 ubicando la fuente(F) y el drenaje(D) al azar dentro del mismo tablero para así comenzar la conexión de la fuente y el drenaje en el juego. Además el sistema debe guardar el tiempo de inicio y final del juego en segundos.* | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| nickName | String | | *-El espacio no debe quedar vacío*  *- Solo entradas de tipo string o cadena de caracteres*  *-Que el nickname no se repita dentro de la lista de jugadores* |
| Resultado o Postcondición | El nickname del usuario se guardará en la lista de jugadores registrados dentro del sistema. | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
| - | | - | *-* |

| Identificador y nombre | *[RF1-Mostrar Juego]* | | | |
| --- | --- | --- | --- | --- |
| Resumen | *El tablero consta de dimensiones 8x8, en cada casilla habrá:*  *Una* ***‘X’*** *representando la ausencia de una tubería, Un* ***‘=’*** *representando una tubería horizontal, Un* ***‘||’*** *representando una tubería vertical, un* ***‘o’*** *representando una tubería circular, una* ***‘F’*** *de fuente de agua y una* ***‘D’*** *de desagüe* | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| - | - | | *-* |
| Resultado o Postcondición | Crea un dato String que representa la matriz de tuberías | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
| board | | String | *matriz 8\*8 que muestre la fuente y el drenaje ubicados al azar.* |

| Identificador y nombre | *[RF2-Poner una tubería]* | | | |
| --- | --- | --- | --- | --- |
| Resumen | *El sistema le permitirá al usuario poner una tubería en el tablero para conectar la fuente con el drenaje. El jugador ingresará la posición de la fila y la columna donde quiere poner la tubería dentro del tablero, sea (=) para horizontal, (o) para circular en caso de flujos de agua en ángulo de 90° y (||)para las tuberías verticales. Posteriormente la tubería se registrará dentro del tablero.* | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| row | int | | *-El espacio no debe quedar vacío*  *- Solo entradas de tipo entero, no valores de otro tipo.* |
|  | column | int | | *-El espacio no debe quedar vacío*  *- Solo entradas de tipo entero, no valores de otro tipo* |
|  | pipe | int | | *-El espacio no debe quedar vacío*  *- Solo entradas de tipo int, no valores de otro tipo.*  *-Solo se permitirá escoger entre tres opciones de tuberías para el usuario:*   1. ***‘=’*** 2. ***‘||’*** 3. ***‘o’*** |
| Resultado o Postcondición | El sistema actualizará la tubería registrada por el usuario dentro del tablero, paso a paso. | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
| - | | - | *-* |

| Identificador y nombre | *[RF3-Evaluar/Simular red de tuberías]* | | | |
| --- | --- | --- | --- | --- |
| Resumen | *Mientras el usuario está viendo la matriz de su juego, se le da la oportunidad de simular el recorrido de las tuberías de la fuente hasta el drenaje (la única forma de ganar).*  *Las tuberías horizontales solo pueden recibir y dar agua a los costados y las verticales por arriba y abajo, el tubo circular no se puede conectar a la fuente y al drenaje, el tubo circular sólo puede seguir el ciclo del agua en ángulos de 90°.*  *El flujo del agua puede ir de arriba a abajo, dar giros de 90° e ir de lado a lado.* | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| No existe |  | |  |
| Resultado o Postcondición | Simula el flujo de la tubería y crea una respuesta según si es válida o no.  Paso seguido almacena el resultado del jugador. | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
| valid | | String | *“Tubería válida, !Bien hecho¡”* |
| invalid | | String | *“Tubería invalida”* |

| Identificador y nombre | *[RF4-Calcular el puntaje]* | | | |
| --- | --- | --- | --- | --- |
| Resumen | *Cuando el usuario termina la partida, el sistema debe calcular el puntaje y registrarlo en la tabla de puntajes. El cálculo se realizará con la siguiente fórmula:*  Donde tuberiasUsadas es el total de piezas que usó el jugador para su solución y tiempoEnSegundos es el tiempo transcurrido entre el inicio de la partida y el final de la partida.  El tiempo en segundos se calculará restando la hora de inicio y fin de la partida. | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| - | - | | *-* |
| Resultado o Postcondición | El puntaje del usuario se guardará en la tabla de puntajes registrada dentro de una lista enlazada. | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
| - | | - | *-* |

| Identificador y nombre | *[RF5-Ver puntaje de otros jugadores]* | | | |
| --- | --- | --- | --- | --- |
| Resumen | *Se muestra una lista de los 10 puntajes más altos junto a la fecha y el nickname del jugador ingresó previamente y da la opción de mostrar la tubería final de la partida del jugador seleccionado* | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| selectedPlayer | int | | *menor a 11 y mayor a 0.* |
| Resultado o Postcondición | Muestra la partida del jugador seleccionado. | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
| ranking | | String | *Posición del jugador + nombre del jugador + puntaje + fecha.*  *Todo esto de cada jugador* |
| Board | | String | *Simula la tubería final del jugador seleccionado* |
| invalid | | String | *“puesto vacío”* |