

REQUERIMIENTOS FUNCIONALES “SNAKES AND LADDERS”

El sistema debe estar en la capacidad de:

- **Req1. *Mostrar*** un menú con tres opciones. La primera opción es para jugar, la segunda opción es para ver el tablero de posiciones y la tercera opción es salir del programa.
 - ◆ **Req1.1 Iniciar** un nuevo juego cuando el usuario digita la primera opción.
 - ◆ **Req1.2 Mostrar** el ranking de ganadores como un listado de los nombres o nicknames de los jugadores, sus símbolos y sus respectivos puntajes, además de la información de la partida, resultado de recorrer el árbol binario de búsqueda inorden.
 - ◆ **Req1.3 Salir** del juego. Esto significa terminar la ejecución del programa.
- **Req2. *Crear*** el tablero de juego según los parámetros ingresados por el usuario (n m s e p).
 - ◆ **Req2.1 Crear** n filas.
 - ◆ **Req2.2 Crear** m columnas.
 - ◆ **Req2.3 Crear** s serpientes uniendo una casilla con otra casilla cualquiera en una fila inferior. Ninguna serpiente inicia en la casilla $n \times m$ y ninguna casilla de inicio o fin de serpiente debe coincidir con otro inicio o fin de serpiente.
 - ◆ **Req2.4 Crear** e escaleras uniendo una casilla con otra casilla cualquiera en una fila superior. Ninguna escalera inicia en la casilla 1 y ninguna casilla de inicio o fin de escalera debe coincidir con otro inicio o fin de escalera.
- **Req3. *Mostrar*** una cuadrícula de n filas por m columnas, con s serpientes y e escaleras.
 - ◆ **Req3.1 Identificar** cada casilla de la cuadrícula con un número.
 - **Req1.1.1 Enumerar** cada casilla de la siguiente forma: inicia en la casilla inferior izquierda con el número 1, sigue en la casilla inmediatamente a la derecha y así hasta terminar la fila. Luego sube en esa misma columna y se regresa hacia la izquierda, intercalando así la dirección en cada fila.
 - ◆ **Req3.2 Mostrar** s serpientes identificadas por letras mayúsculas (una letra para cada par de serpientes).
 - ◆ **Req3.3 Mostrar** e escaleras identificadas por números (una número para cada par de escaleras).

- **Req 4. Mover** un jugador de una casilla a otra dependiendo el puntaje obtenido en el tiro de dado. En caso de caer en una casilla que contenga el inicio de una serpiente o escalera deberá subir o bajar respectivamente al final de la misma.
 - ◆ **Req4.1 Mostrar** un mensaje que indique la espera de un salto de línea para continuar con el siguiente turno.
 - ◆ **Req4.1 Mostrar** el tablero actual después del movimiento con la posición de los jugadores en las casillas.
- **Req5. Simular** el juego cuando el usuario ingrese el comando *simul*. Esto consiste en ir mostrando lo que cada jugador a su turno juega, con el tablero correspondiente de cada nueva posición, esperando 2 segundos entre cada jugada, pero sin esperar ningún salto de línea.
- **Req6. Mostrar** el menú principal cuando el usuario ingrese el comando *menu*. El juego se corta sin terminar y el programa regresa al menú principal, mostrándose sus opciones.
- **Req7. Mostrar** el tablero principal con la enumeración de las casillas, las serpientes y escaleras cuando el usuario ingrese el comando *num*. Luego esperará entonces un salto de línea para mostrar el tablero actual (con las serpientes, escaleras y posición actual de los jugadores) y continuar con el juego.
- **Req 8. Crear** un ranking de ganadores usando un árbol binario de búsqueda ordenado inversamente por puntaje.
 - ◆ **Req8.1 Crear** un nodo el cual contenga el nombre del jugador y los parámetros del juego: n, m, s, e y p. Además de los símbolos usados por todos los jugadores, y el símbolo ganador.
 - ◆ **Req 8.2 Agregar** un jugador ganador al árbol binario.
- **Req9. Mostrar** un mensaje cuando un jugador haya ganado la partida, mencionando el número de movimientos que tuvo.
 - ◆ **Req9.1 Registrar** el jugador ganador solicitando un nickname y almacenar la información necesaria de este en el ranking de jugadores (Req 8.2).

