

Algoritmos y programación II.

Tarea Integradora 2.

Snakes and Ladders.

Gianni Estiven Buena vides.

Sebastian Navia Ramirez.

Facultad de ingeniería.

Ingeniería de sistemas e Ingeniería Telemática

Santiago de Cali, abril 30 de 2021.

## Requerimientos funcionales.

**Req 1:** Generar un tablero  $n \times m$ , donde se vea contenido las escaleras, serpientes y jugadores de la partida. El programa le mostrará una cuadrícula formada por corchetes, con las casillas numeradas correctamente y con la ubicación de las escaleras y las serpientes.

**Req 1.1:** Generar el tablero, la cantidad de columnas y filas las ingresa el usuario.

**Req 1.2:** Siempre la primera casilla comienza abajo a la izquierda, luego la ficha de cada jugador se mueve hacia la derecha hasta el borde del tablero, después sube una casilla y se mueve en dirección contraria a como venía originalmente, la casilla final es la meta de llegada.

**Req 1.3:** Las casillas deben tener una enumeración e a la izquierda

**Req 2:** Se deben generar  $p$  jugadores que interactúan directamente con el tablero.

**Req 2.1:** La cantidad de jugadores deben ser ingresados por el usuario.

**Req 2.2:** Se guarda la cantidad de movimientos de cada jugador.

**Req 2.3:** A los jugadores se les asigna un símbolo para reconocerlo fácilmente en el tablero. (El usuario los ingresa).

**Req 3:** Se debe generar las serpientes en el juego unen una casilla con otra cualquiera en una fila inferior. Las serpientes se identifican con letras mayúsculas del alfabeto iniciando en A.

**Req 3.1:** La función de las serpientes es poner un obstáculo en el camino, si te topas con ellas te bajaran directamente a la casilla donde se encuentra su cola.

**Req 4:** Se debe generar las escaleras en el tablero, donde unen una casilla con otra cualquiera en una fila superior y se denotan con un número, tanto en la parte superior y en la parte inferior.

**Req 4.1:** Su función es ayudar al jugador mejorando su posición. Si un jugador cae en la base de la escalera, este podrá avanzar todas las casillas hasta quedar al final de la escalera.

**Req 5:** El programa debe iniciar con un sencillo menú con 3 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.

**Req 5.1:** La primera opción inicia el juego.

**Req 5.2:** La segunda opción te permite ver el tablero y las posiciones de los jugadores.

**Req 5.3:** La tercera opción te permite cerrar el programa.

**Req 6:** El programa recibe 5 números enteros separados por consola por consola ( $n$ ,  $m$ ,  $s$ ,  $e$  y  $p$ ), respectivamente se refieren a las filas, las columnas, serpientes, escaleras y jugadores.

**Req 7:** Se debe crear un dado que vaya del número 1 al 6, los números que se generen son al azar.

**Req 8:** Si escribe la palabra num y luego se da salto de línea, el programa mostrará la misma cuadrícula que muestra al inicio, con las casillas numeradas, las serpientes y las escaleras.

**Req 9:** Si escribe la palabra simul y luego se da salto de línea, el programa empezará en modo simulación, que 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.

**Req 10:** Si escribe la palabra menu y luego se da salto de línea, el juego se corta sin terminar y el programa regresa al menú principal, mostrándose sus opciones.

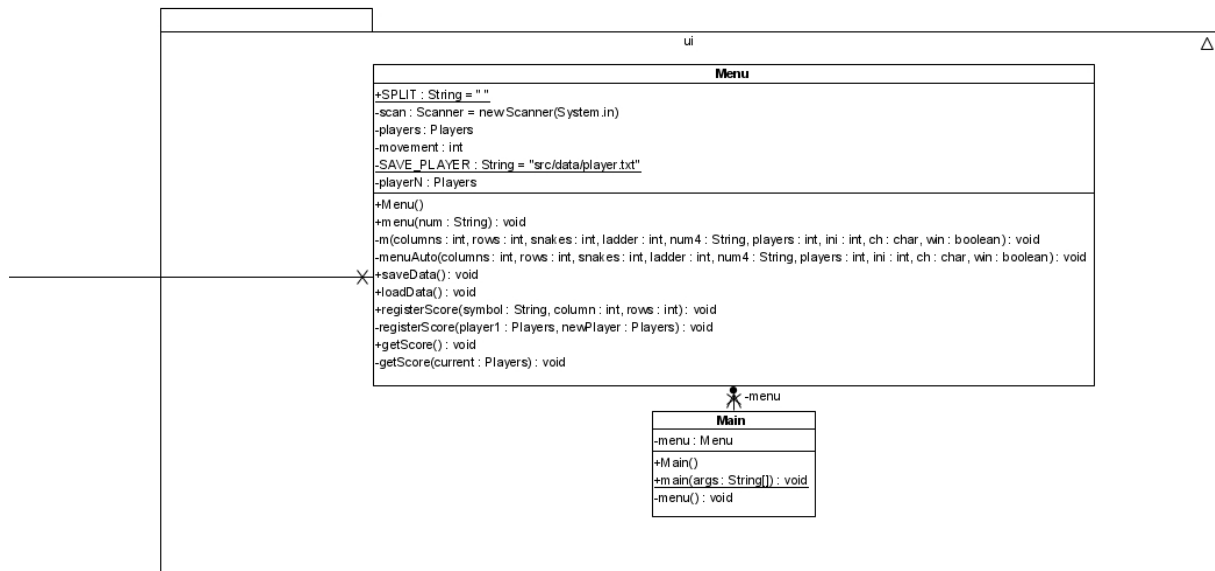
**Req 11:** Al final de cada partida se genera los puntajes y posiciones de cada jugador.

Req 12: Acabar el juego cuando un jugador llegue a la última casilla.

Req 12.1: Imprimir el símbolo del jugador que ganó.

Req 12.2: Pedir el nombre del jugador ganador. Desplegar el menú nuevamente.

Req 13. Se calcula el puntaje del jugador ganador multiplicando las casillas del tablero por los movimientos del jugador.



# model

