

# Proyecto de Agentes Inteligentes

Presentado por:  
Mario Andrés Yusti Mejia

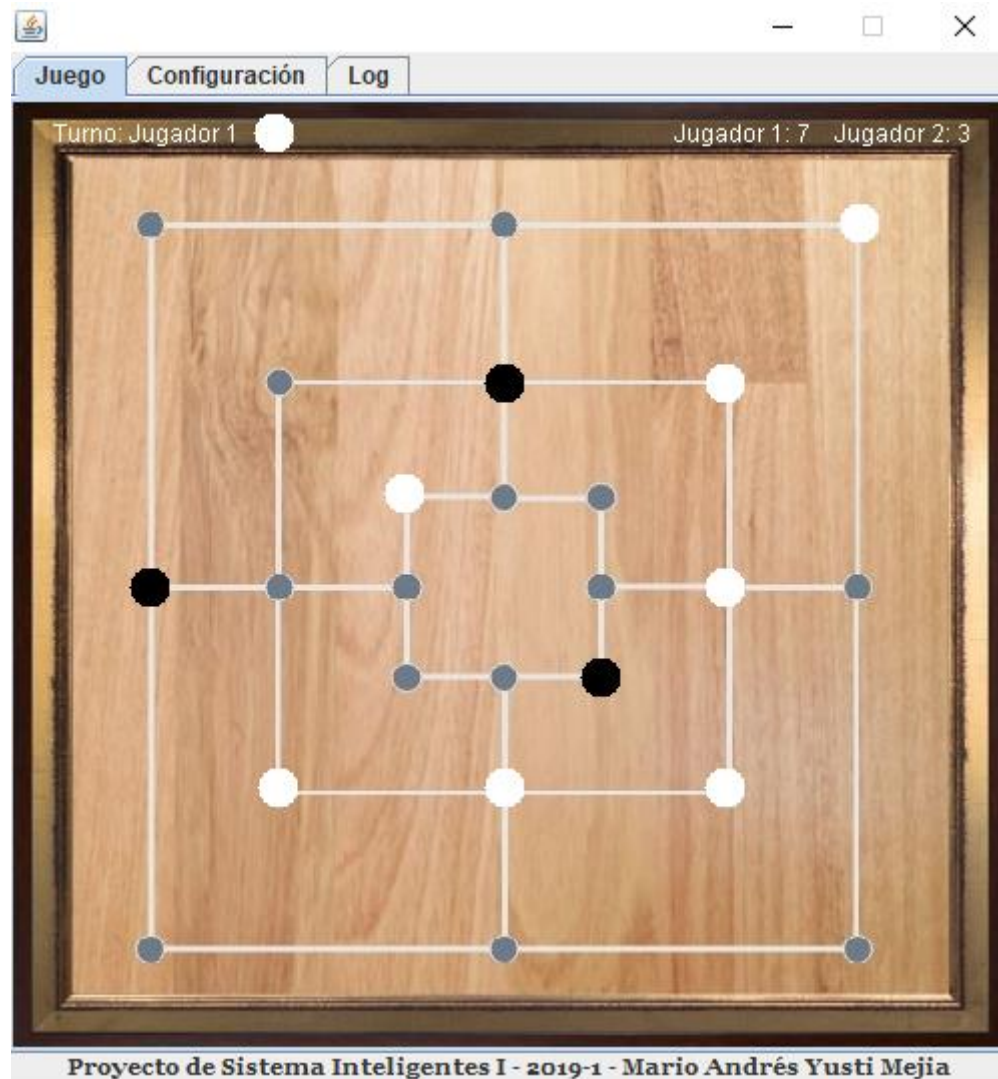
Profesor:  
Jairo Ivan Vélez Bedoya

Sistemas Inteligentes I  
Ingeniería en Sistemas y Computación  
Faculta de Ingeniería  
Universidad de Caldas  
Manizales  
Agosto 2019


## Interfaz Gráfica

La interfaz gráfica cuenta con 3 secciones:


1. Sección de Juego: En esta sección podemos encontrar el estado de las fichas de ambos jugadores, su cantidad de fichas y el jugador en turno.



2. Sección de Configuración: En esta sección se puede definir el nombre de los agentes jugadores e intentar conectarlos al tablero (probar que estén activos). Además, es posible cambiarles el color de las fichas. Podemos ver el índice de cada punto en el tablero y tenemos el botón de jugar para iniciar el juego y el de reiniciar para limpiar el tablero y el registro.

Jugador 1:  

Estado: **Desconectado**

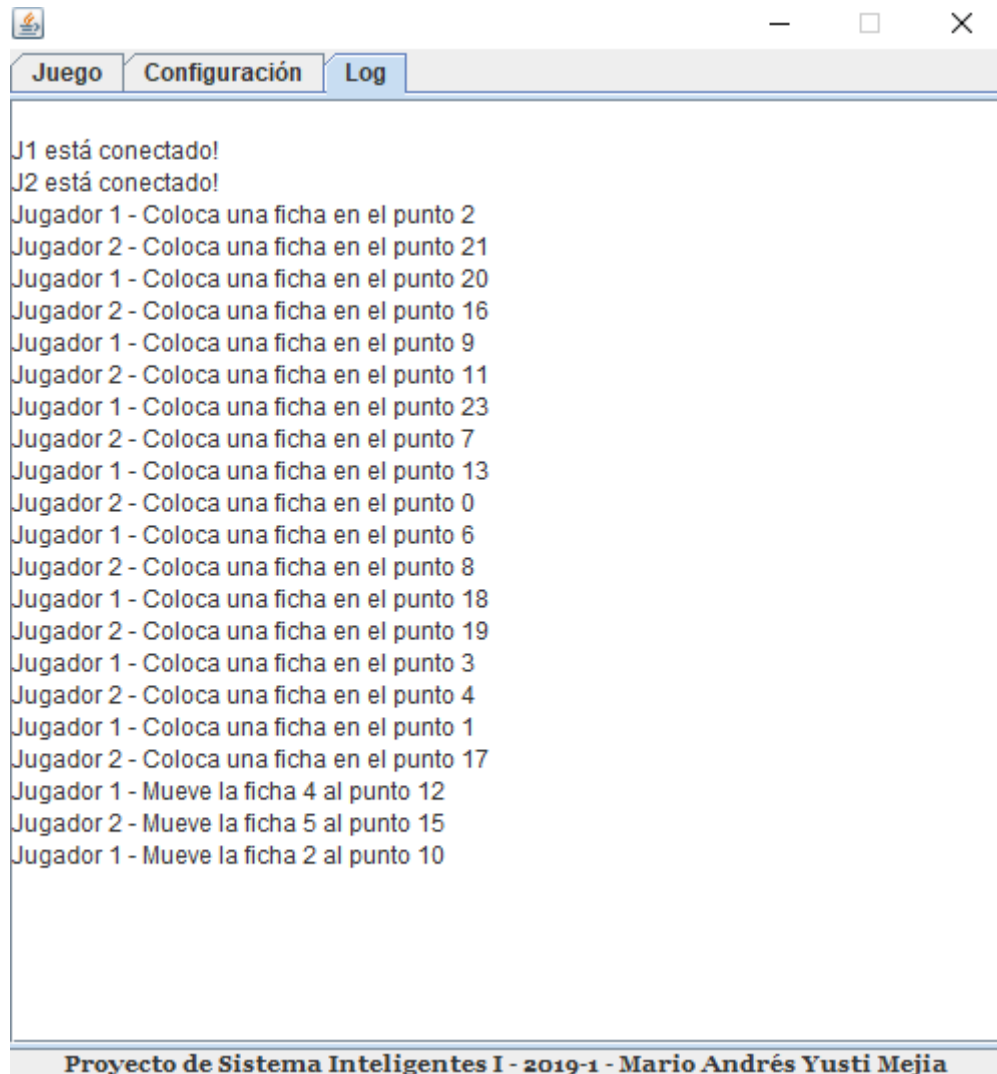
Jugador 2:  

Estado: **Desconectado**

☐ Mostrar el número de los Puntos

Proyecto de Sistema Inteligentes I - 2019-1 - Mario Andrés Yusti Mejia

3. Sección de historial (Log): Se muestra el historial de todos los eventos que ocurren en el juego.



## Arquitectura de Agentes

El proyecto consta de dos tipos de agentes: agente tablero y agente jugador.

**Agente tablero:** Se encarga llevar el control de las fichas de los jugadores y de la visualización por interfaz gráfica del juego de las acciones de los jugadores. Mantiene la lista actualizada de las fichas de los jugadores y su ubicación en el tablero.

**Agente jugador:** Se encarga de calcular la mejor jugada para él y de decidir la mejor acción a ejecutar. Tienen una copia actualizada de las fichas en el tablero.

El agente tablero envía una copia actualizada del tablero y la fase de juego en la que se encuentra al jugador que tiene el turno y este le responde con la acción que realizará. El tablero realiza la acción y vuelve a hacer el mismo proceso con el otro jugador.



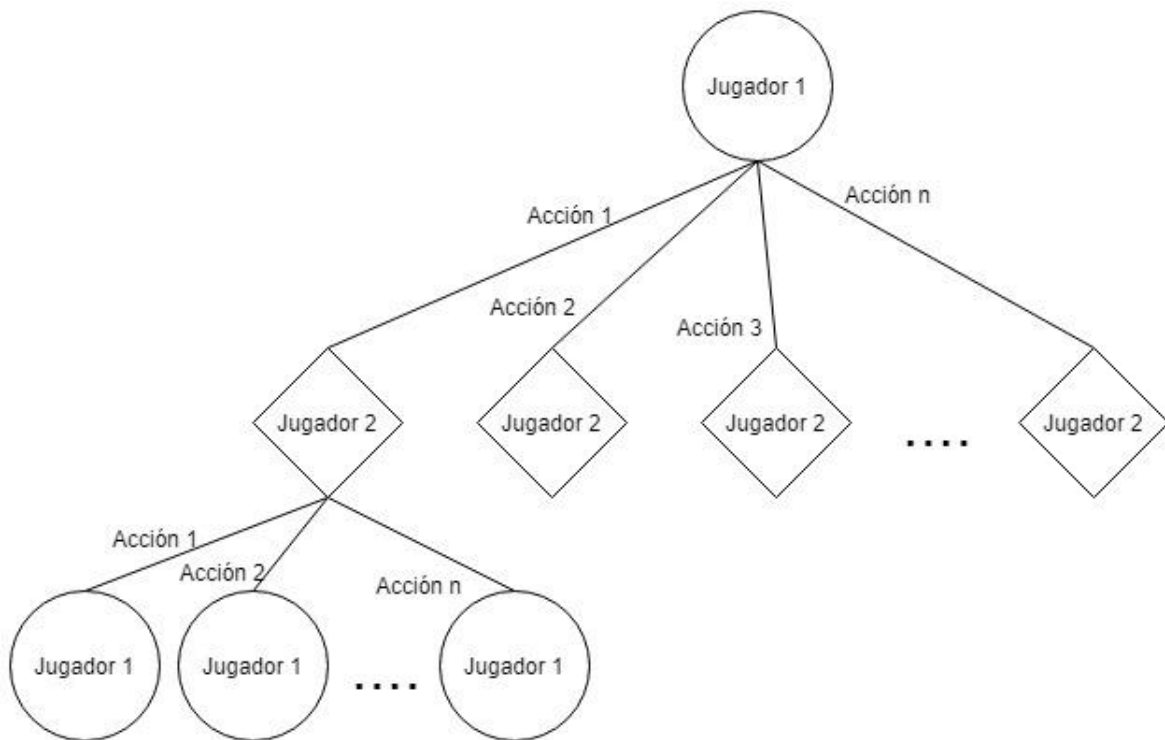
## Estrategia de juego

Para la implementación del juego se utilizó la teoría de juegos y la búsqueda informada.

Tomando el concepto de Teoría de Juego: *“Es una rama de la economía que estudia las decisiones en las que para que un individuo tenga éxito tiene que tener en cuenta las decisiones tomadas por el resto de los agentes que intervienen en la situación.”*

*No tenemos que preguntarnos qué vamos a hacer, tenemos que preguntarnos qué vamos a hacer teniendo en cuenta lo que pensamos que harán los demás, ellos actuarán pensando según crean que van a ser nuestras actuaciones.”*

Se utilizó la estructura de un árbol de recursión con la siguiente estructura:



Las acciones son las ramas de este árbol y cada nivel representa el turno de un jugador. Una acción representa la creación de un nuevo estado (nuevo nodo del árbol) que usará el jugador del siguiente turno. Una acción puede ser:

- Movimiento de una ficha
- Colocación de una ficha

Y en consecuencia de la acción, esta puede estar acompañada de una Captura de ficha.

El principal objetivo es encontrar una acción con la cual se obtenga el mayor beneficio para el jugador teniendo en cuenta que el otro jugador también buscará realizar la acción que mayor le beneficie.

En cada estado, el jugador al que le corresponde el turno debe por cada ficha, evaluar el beneficio de todos los posibles movimientos que puede realizar. Para esto es necesario realizar la acción (crear nuevo estado del tablero) y revisar el beneficio que puedo obtener desde ese estado.

**Clave 1:** En cada turno, el jugador correspondiente siempre buscará realizar la acción que más lo beneficie.

**Clave 2:** El juego termina cuando un jugador ya no se pueda mover, es decir, cuando ninguna de sus fichas se pueda mover o cuando ya no pueda completar líneas de 3 (solamente le queden 2 fichas).

#### **Heurística:**

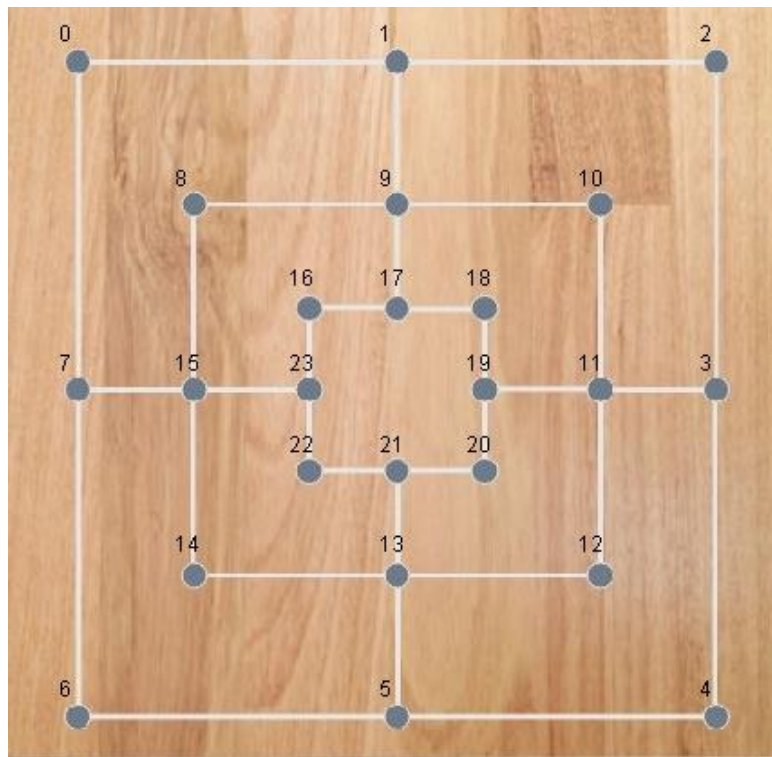
- Si el jugador en turno ya no se puede mover, pero el adversario sí, significa que perdió. Valor: -9.
- Si el jugador en turno se puede mover, pero el adversario no, significa que ganó. Valor: 9.
- Si al mover una ficha, obtengo una línea de tres, significa que puedo capturar una ficha del adversario. Valor 2.
- Si al mover una ficha en un punto donde evito una línea de tres fichas del adversario. Valor 1.
- Si ningún jugador puede moverse significa que el juego termina en empate. Valor 0.
- Si visito un estado ya visitado tiene un valor de 0.
- Cualquier otro movimiento tiene valor 0.

**Clave 3:** En cada turno el jugador buscará la acción que más lo beneficie, es decir, con mayor heurística. Pero esta acción es la que menos beneficie al jugador con turno anterior (nivel superior) y así sucesivamente hasta llegar al nodo raíz. (Es necesario que el valor heurístico que le retorne el nodo del siguiente turno sea multiplicado por -1)

## **Estrategia de implementación**

La **interfaz gráfica** es un agente con interfaz gráfica. Se utilizó un JFrame y un JPanel para dibujar el tablero.

Para la implementación del **tablero** se utilizaron estructuras de datos como listas, grafos y arreglos. El tablero está compuesto por 24 puntos y estos están enumerados del 0 al 23 como se muestra en la siguiente imagen.



Existen dos tipos de fichas y puntos en el Juego: Los de interfaz gráfica GUI usados por el agente tablero para poder visualizar y los lógicos utilizados por los agentes jugador para poder calcular.

Los puntos de GUI tienen su respectiva X y Y en el tablero.

Los puntos en los agentes jugador es un grafo que representa el tablero. Cada uno tiene su id y lista de vecinos.

Las fichas de GUI tienen el punto del tablero en el que se encuentran y su respectivo X y Y que finalmente el mismo que el del punto (tiene su propio X y Y para que se puedan mover de forma independiente en la interfaz gráfica del tablero).

Las fichas en los agentes jugador es simplemente un arreglo de enteros. El índice del valor es el id de la ficha y el valor es el punto en el que se encuentra.



Además de eso y para optimizar las búsquedas y los cálculos se tiene en el agente jugador tablas de líneas. Es decir, existe una tabla que almacena cada id de los puntos que participan en cada una de las líneas de tres puntos posibles en el tablero y la tabla inversa a esta, que almacena los id de las líneas de cada uno de los puntos en el tablero.

El agente tablero le envía al agente jugador en turno un objeto Serializable con el estado de las fichas de cada jugador y un mensaje que identifica la fase de juego. El agente jugador le responde con otro objeto Serializable con el índice del punto, índice de la ficha, índice de la ficha capturada si se necesita y el mensaje que identifica la acción a realizar.

### **Mensajes:**

- 'C' – Conexión entre agentes
- 'F' – Colocar ficha
- 'M' – Mover ficha
- 'P' – Mover ficha y Capturar ficha de adversario
- '-' – Jugador no puede moverse (Gana el adversario)
- 'E' – El juego termina en empate

### **Calculo de la mejor acción:**

El agente jugador recorre todas sus fichas y por cada ficha lista los puntos posibles a donde puede mover (Todos los puntos si tiene 3 fichas o los puntos vecinos al punto donde se encuentra la ficha). Luego por cada punto posible revisa si está libre (no hay alguna ficha ahí) y al estar libre, crea un nuevo estado de sus fichas y revisa si el jugador hizo una línea de tres, sino revisa si al menos le bloqueo una línea de tres al adversario. Luego de ahí vuelve a llamar a otra función que devuelve el beneficio del siguiente turno, el valor devuelto por esta función es el mayor beneficio que puede obtener el adversario después de este movimiento y debe ser multiplicado por -1: Si el beneficio del adversario es positivo, para mí es negativo y si el beneficio del adversario es negativo, para mí eso es positivo.

Debo guardar siempre el mayor beneficio obtenido y dependiendo de este, debo escoger la jugada (o acción) a realizar. Si hay más de una jugada con el mismo beneficio, son guardadas en una lista y luego selecciono al azar una de estas (Esto evita ciclo infinito con las mismas jugadas en el tablero).

## Decisiones inteligentes

Lista de decisiones ordenadas por su prioridad:

1. Si la colocación realiza una línea de tres la hago.
2. Si la colocación evita una línea de tres del adversario la realizo, sino se coloca en un punto aleatorio.
3. Si un movimiento hace una línea de tres lo hago (puede capturar una ficha).
4. En el momento de una captura, se elige la ficha del adversario que evita que el jugador realice una línea de tres, sino se selecciona una ficha al azar.
5. Si un movimiento evita una línea de tres del adversario lo hago, sino se realiza el movimiento al azar.

Es necesario tener en cuenta que no se revisen estados repetidos, es por eso que se tiene una función que revisa eso por cada turno de la siguiente forma: tomamos los 24 puntos del tablero (identificados del 0 al 23 como se mencionó anteriormente) y los tomo como un número binario de 24 bits. Cada bit identificado del 0 al 23 como los puntos. Es decir que tengo 16.777.216 de números distintos. Estos son los números que identifican a cada estado. Por ejemplo, el jugador 1 tiene sus fichas en los puntos: 0,3,4,5,10,12,20 entonces el número que lo identifica es  $2^0 + 2^3 + 2^4 + 2^5 + 2^{10} + 2^{12} + 2^{20} = 1.053.753$  (lo mismo para el jugador 2). Este número es único debido a que en un punto solo puede haber una ficha.

Nota: Cada acción me crea un nuevo estado para que el otro jugador juegue. Cada nuevo estado tiene unas acciones posibles, teniendo en cuenta las fichas y los puntos disponibles. Teniendo en cuenta que hay dos jugadores, 9 fichas por jugador, 24 puntos en el tablero podemos concluir que existen demasiados estados como para revisar todas las posibilidades de juego (Desbordamiento de pila seguro).

Se tiene un parámetro de profundidad para sirve limitar el número de niveles que se revisan (jugadas posteriores).

# Diagrama de Clases

