

# Proyecto final de IA y Simulación

Juan Carlos Espinosa Delgado C-411

Raudel Gómez Molina C-411

Alex Sierra Alcalá C-411

28 de abril de 2024

# Contents

<b>1</b>	<b>Introducción</b>	<b>3</b>
<b>2</b>	<b>Modelación basada en Agentes</b>	<b>4</b>
2.1	Modelación del ambiente . . . . .	4
2.2	Tipos de Agentes . . . . .	4
2.2.1	Agente jugador . . . . .	4
2.2.2	Agente entrenador . . . . .	5
<b>3</b>	<b>Simulación de partidos</b>	<b>6</b>
3.1	Campo de juego y posiciones . . . . .	6
3.2	Atributos de los jugadores . . . . .	7
3.3	Acciones de los jugadores . . . . .	8
3.4	Acciones de los entrenadores . . . . .	9
<b>4</b>	<b>Integración con LLM</b>	<b>9</b>
<b>5</b>	<b>Detalles de implementación</b>	<b>10</b>
5.1	Mecanismo de simulación . . . . .	10
5.2	implementación de reglas y lógica difusa . . . . .	10
5.3	Encapsulamiento de las estrategias de los agentes . . . . .	10
<b>6</b>	<b>Análisis experimental</b>	<b>10</b>
<b>7</b>	<b>Conclusiones</b>	<b>10</b>
<b>8</b>	<b>Recomendaciones</b>	<b>10</b>

# 1 Introducción

El fútbol es un deporte de equipo muy popular en todo el mundo, y su simulación mediante técnicas de inteligencia artificial (IA) se ha convertido en un área de investigación activa. La simulación de partidos de fútbol permite a los investigadores y profesionales del deporte estudiar diferentes estrategias, tácticas y escenarios de juego sin la necesidad de partidos físicos reales.

Este informe presenta los resultados de un proyecto de simulación de un partido de fútbol utilizando técnicas de IA. El objetivo del proyecto era desarrollar un modelo de simulación que pudiera generar partidos de fútbol realistas y que permitiera a los usuarios experimentar con diferentes parámetros y estrategias para hacer predicciones de estos partidos.

## 2 Modelación basada en Agentes

Para realizar simulaciones de los juegos hacemos uso de agentes inteligentes, nuestro ambiente consistirá en el estado actual del juego, que está compuesto por sus jugadores, la estrategia que siguen, sus atributos de jugador actuales, los entrenadores, el terreno y las estadísticas actuales del partido. Nuestros agentes serán los jugadores y entrenadores y su función objetivo será ganar el partido.

### 2.1 Modelación del ambiente

Nuestro ambiente consta de las siguientes características:

1. **Accesibilidad:** Nuestro ambiente es totalmente accesible, cada jugador posee conocimiento acerca del resto de los jugadores y de sus posiciones en el terreno, de dónde se encuentra el balón y de las estadísticas actuales del partido. Importante aclarar que para los jugadores se tiene en cuenta un atributo de visión y cansancio, que limita las acciones que este puede tomar.
2. **Determinista o no determinista:** Nuestro ambiente es un ambiente no determinista, ya que, al no conocer el resultado de cada acción con certeza, existe incertidumbre acerca del estado del ambiente luego de realizar una acción.
3. **Episódico o secuencial:** Este ambiente es secuencial, ya que las decisiones del agente pueden influir de forma positiva o negativa en el futuro, por lo que tiene que razonar las consecuencias de sus acciones.
4. **Estático o dinámico:** Es un ambiente estático, ya que permanece inalterable mientras no se realice una acción sobre él, de hecho, todos los jugadores deben realizar su acción (la cual puede ser no hacer nada) para poder avanzar en el partido.
5. **Discreto o continuo:** Optamos por que el tiempo fuera discreto, aunque eso no aleje un poco de la realidad. En cada instancia de tiempo, cada jugador puede realizar un número constante de acciones (solo una). Cada jugador no puede decidir de forma independiente el instante exacto de tiempo en que realizará su acción, sino que se sigue un orden preestablecido en cada instancia de tiempo. Todas estas consideraciones simplifican de forma considerable el problema aunque lo alejan un poco de la realidad.

### 2.2 Tipos de Agentes

#### 2.2.1 Agente jugador

Nuestros jugadores son agentes puramente reactivos ya que basan su decisión enteramente en el presente, sin referencia a lo que haya pasado anteriormente. Estos simplemente responden directamente al ambiente. Implementamos varias estrategias que pueden ser usadas por el agente estas perfectamente pueden ser combinadas en un mismo juego por varios agentes y actúan de forma independiente.

Una de estas estrategias se basa en tipos de comportamiento entre los que se encuentran: ofensivo, defensivo, evitar cansancio y el respeto por la posición inicial impuesta por el entrenador. En dependencia de si los jugadores son defensas, mediocampistas o delanteros, tendrán una estrategia basada en una heurística, la cual funciona dándole pesos a cada uno de los comportamientos del agente, y un pequeño peso a un comportamiento aleatorio. De esta forma cada jugador puede variar su configuración de comportamiento en dependencia de su posición en la alineación.

Otra estrategia implementada se basa en la búsqueda del posible árbol de juego (esta implementación tuvo sus complicaciones ya que las transiciones de posibles acciones son no deterministas). Por esta razón optamos por evaluar todas las posibles acciones del agente y tomando cada una de estas simular el comportamiento del resto de los agentes y en la próxima instancia de tiempo volver a evaluar todas las posibles acciones del agente y así hasta cierto nivel que se especifica. Una vez finalizadas estas simulaciones se evalúa que tan bueno es la posición obtenida en el campo usando una función de evaluación y se toma la mejor decisión en base a este número. En esta función se le da una valoración

a qué tan bien está posicionado el equipo del jugador ofensivamente en caso de tener el balón o defensivamente en caso de que la posesión sea del equipo contrario. Para hacer esta valoración tenemos en cuenta la posición del balón y la distancia a la que este se encuentra de la portería rival, las oportunidades de pase, el marcador actual del partido y se hace una valoración también de la ventaja que tiene un equipo sobre otro en el terreno haciendo uso de las estadísticas del partido con diferentes pesos en donde por ejemplo se usan los goles, faltas, pases, pases, tiros y tarjetas, donde un gol tiene bastante mas peso que un pase o un disparo. Teniendo en cuenta los valores numéricos todos estos factores el jugador evalúa la posición ofensiva/defensiva de su equipo basado en reglas de lógica difusa donde por ejemplo, si el jugador considera que un atacante está "lejos" (según lo que considere lejos ese jugador) de la portería rival, este no tiene una posición ofensiva buena. Todas las reglas mencionadas anteriormente se aplican al algoritmo de evaluación usando lógica difusa.

Como en la vida real, un jugador no sabe cómo percibe el resto de los jugadores los datos de un partido, este no tiene conocimiento de sus funciones de evaluación en nuestra simulación, por lo que a la hora de ejecutar la búsqueda su predicción del partido no es del todo exacta, haciendo su pensamiento más realista. Para intuir las acciones del resto de los jugadores, hace uso de las heurísticas explicadas anteriormente, dependiendo si el jugador a predecir es defensor, centrocampista o delantero, teniendo en cuenta los comportamientos más probables de estos tipos de jugadores.

### 2.2.2 Agente entrenador

Nuestros agentes entrenadores, son agentes puramente reactivos ya que, al igual que los jugadores, basan su decisión enteramente en el presente, sin referencia a lo que haya pasado anteriormente.

Su primera tarea será la de elegir el 11 inicial. Para ello tienen distintas posibilidades de formaciones (4-3-3, 5-3-2, etc). Para cada formación eligen a los 11 jugadores haciendo uso de una heurística que consiste en que el mejor 11 posible a alinear es aquel que mejores jugadores tenga en general (basado en los atributos de los jugadores de nuestro dataset). Optamos por esta heurística de evaluación un tanto greedy ya que evaluar todas las combinaciones sobre el campo de los jugadores es un problema combinatorio muy costoso computacionalmente, esto tiene la desventaja que solo se evalúa el jugador que mejor para una posición y no existe una evaluación conjunta. El manager explora todas las combinaciones de 11 jugadores posibles de la plantilla en donde ningún jugador esté fuera de posición (por ejemplo no ve ninguna combinación en la que Messi sea portero) todo esto después de aplicar la heurística descrita anteriormente. Luego de elegir para cada formación su mejor alineación, el entrenador simula varias veces todas sus formaciones posibles y las del entrenador rival y decide alinear la que más porcentaje de victoria le dé. Al ser no determinista nuestra simulación y no saber el resultado de cada acción en el juego con certeza, no pudimos hacer usos de algoritmos como el **MTCS** para el cual está probada su eficacia en este tipo de contextos pero en el caso de los deterministas, por lo que se optó por la idea descrita anteriormente.

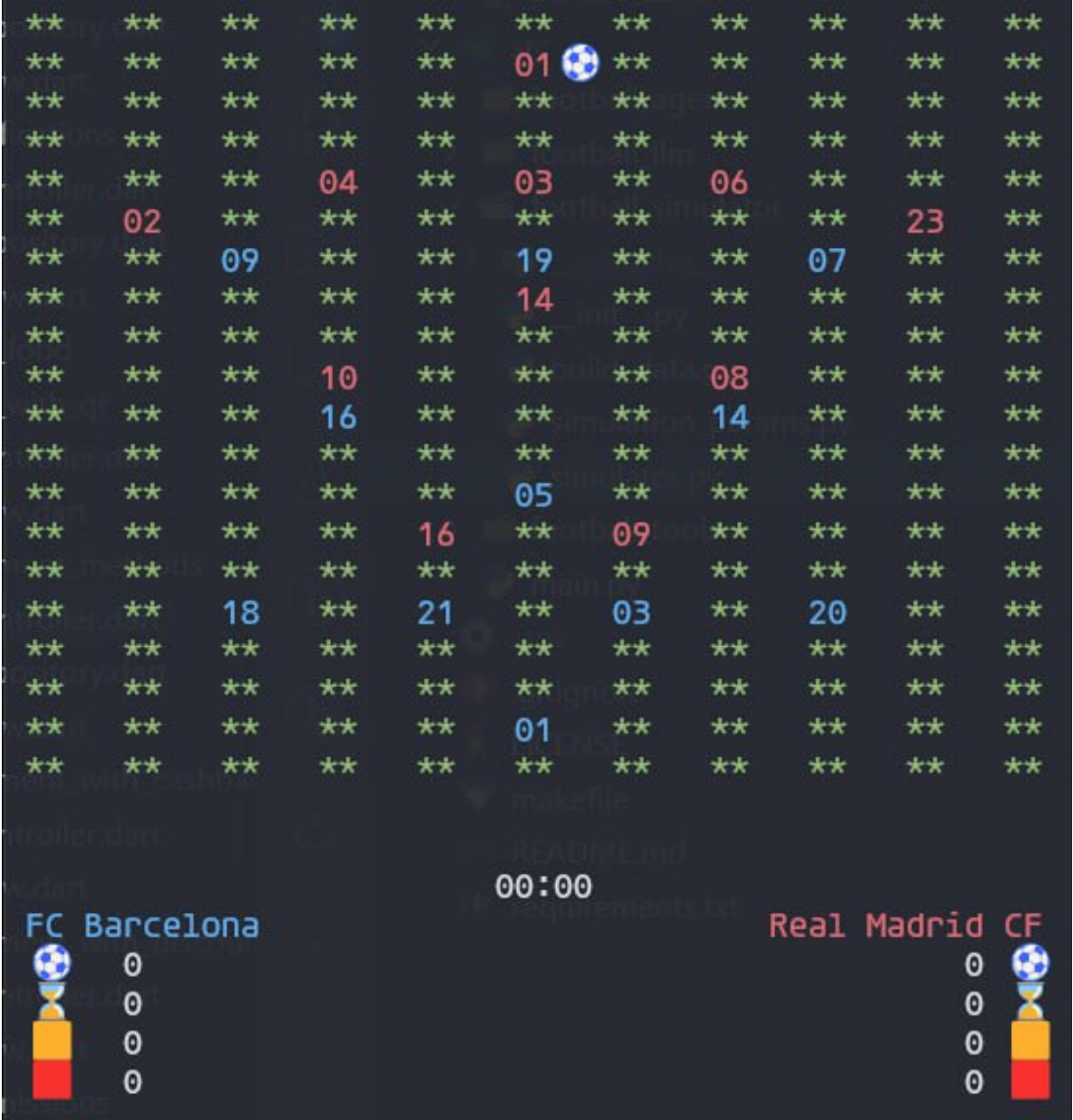
Una vez empezado el partido con las alineaciones decididas por cada entrenador, estos también pueden ejecutar acciones. Entre estas se encuentran hacer cambios de jugadores y cambios de formaciones. En los cambios de formaciones los entrenadores si podrán situar jugadores en posiciones que no aparezcan entre las posibles a jugar por estos, teniendo una penalización en sus atributos de habilidades (ya que en la vida real los jugadores que juegan fuera de sus posiciones habituales no son igual de buenos que en estas). Para tomar las decisiones sobre qué acción ejecutar, los entrenadores siguen el mismo algoritmo descrito para elegir la alineación inicial pero en este caso asumen que el técnico rival jugará random ya que no es posible probar todas las posibles acciones y llegar hasta el final del partido.

Otra estrategia implementada para los entrenadores es siguiendo la misma idea que la búsqueda de los jugadores. En este caso es mucho más fácil ya que solo se tiene en cuenta la interacción de los dos entrenadores, por lo que en este caso se puede hacer uso de una especie de **minimax** (con el no determinismo de las transiciones presente). Para la ejecución del mismo se evalúan todas las posibles acciones de ambos entrenadores y se simula hasta la próxima instancia de tiempo en que le corresponda jugar, así hasta llegar a bajar una cantidad de niveles y luego utilizar la misma evaluación del campo que utilizaban los jugadores más simplificada.

### 3 Simulación de partidos

#### 3.1 Campo de juego y posiciones

El terreno de juego se representa gráficamente con un array de 20 filas por 11 columnas donde se intenta simular un campo de fútbol real, con límites por los lados y por el fondo, y con dos porterías. Una vez que introducimos a los jugadores en el campo para simular un partido, ocupan las posiciones predeterminadas para cada posición, que varían dependiendo de la táctica utilizada. Aquí podemos ver un ejemplo de las formaciones iniciales de un partido y de las posiciones que existen:



DEMARCACIÓN	REPRESENTACIÓN	POSICIÓN
PORTERO	GK	Portero
DEFENSA	LB	Lateral izquierdo
	RB	Lateral derecho
	CB	Defensa central
CENTROCAMPISTA	CM	Medio centro
	CDM	Medio centro defensivo
	RM	Medio derecho
	LM	Medio izquierdo
DELANTERO	RW	Extremo derecho
	LW	Extremo izquierdo
	ST	Delantero centro

### 3.2 Atributos de los jugadores

Los datos de los equipos y jugadores son extraídos de un dataset del FIFA 22 de EA Sports. Un jugador tiene un nombre, un conjunto de posiciones, equipo al que pertenece, un dorsal y, además, doce atributos preestablecidos. A continuación una relación de en qué acciones se usa cada uno de los atributos:

SUCESOS POSIBLES EN LA SIMULACIÓN	ATRIBUTOS QUE UTILIZA EL JUGADOR CON EL BALÓN (ATACANTE)	ATRIBUTOS QUE UTILIZA EL JUGADOR SIN EL BALÓN (DEFENSOR)
Pase	Visión	-
Anticipación en el pase	Reacción y control de balón (del receptor del pase)	<u>Intercepción y defensa</u>
Regate (movimiento hacia una posición contigua con defensores cerca)	Velocidad, Regate	Velocidad, Defensa
Falta en el regate	-	Agresividad
Disparo a puerta	Disparo	Reflejos, estirada (del portero)

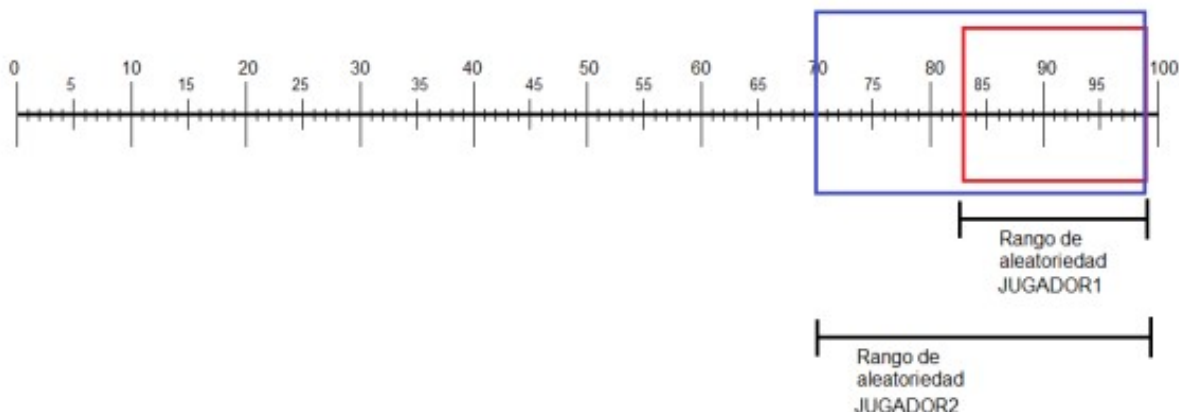
Además de los atributos vistos en la tabla anterior, hacemos uso de la resistencia de los jugadores. cada acción que realiza un jugador tiene un costo de estamina. Existe también la acción de moverse en el terreno, tanto con balón

como sin él, donde lo único que sucede es la disminución de estamina sin tener que utilizar ninguno de los atributos de la tabla.

Los atributos preestablecidos son valores numéricos previamente asignados en el dataset, con valores posibles del 1 al 99, pero que generalmente se encuentran entre 60 y 85, donde se evalúa de forma precisa cómo de bueno es un jugador en un aspecto físico o técnico a la hora de jugar un partido. Para poner un ejemplo, un atributo es la Velocidad. Si un jugador llamado JUGADOR1 tiene un valor de 83 de velocidad, y otro jugador llamado JUGADOR2 tiene un valor de 70 de velocidad, significa que el JUGADOR1 es más rápido que el JUGADOR2. Esto quiere decir que, a la hora de jugar el partido, es más probable que JUGADOR1 gane a JUGADOR2 en velocidad. Es importante no confundir esta última frase. Es más probable que JUGADOR1 gane a JUGADOR2 en velocidad, pero no significa que JUGADOR1 gane a JUGADOR2 siempre.

Para explicar cuál es la probabilidad de que gane uno u otro jugador, explicaremos a continuación cómo funciona el algoritmo de probabilidades:

Supongamos que ocurre un pase, donde el receptor (del equipo atacante) tiene 81 en reacciones y 85 en control del balón y el defensa que quiere interceptar tiene 68 en intercepción y 72 en defensa. Hacemos una normalización que no es más que la media de los atributos implicados del jugador y sacaremos un valor aleatorio entre ese factor y 99. Ese será nuestro factor de aleatoriedad.



Con estos datos, podemos afirmar que, si JUGADOR1 y JUGADOR2 disputasen una recepción de un pase en medio de un partido, JUGADOR1 tendría más posibilidades de ganarla, pero esto no implica que tenga que ganar siempre. Obviamente, cuanto más diferencia hay entre los valores de los jugadores, más fácil es ganar el algoritmo de aleatoriedad para el de valor superior. En caso de empate en los factores de aleatoriedad ganará el duelo el jugador a la defensa.

### 3.3 Acciones de los jugadores

Como ya mencionamos anteriormente las acciones de los jugadores ocurren en una instancia de tiempo en la que cada agente decide una única acción para realizar. El orden en que los agentes deciden sus acciones es secuencial y se basa en la posición que estos tengan en el campo, comenzando por el jugador que tiene la pelota. Las acciones que pueden realizar los jugadores son las siguientes:

1. **Nothing:** El jugador entre sus decisiones posibles tiene la de no hacer nada, esta es la única que no resta estamina.
2. **Pass:** El jugador con balón tiene la posibilidad de pasar la pelota a los compañeros que se encuentre en su rango de vision, el pase siempre llega con éxito.



3. **Move:** Todo jugador tiene la posibilidad de moverse sobre el terreno, la única limitante es que no se puede mover a un lugar donde ya haya un jugador.
4. **Dribble:** Junto con la acción de moverse con el balón, el jugador puede decidir si intentar regatear para que los defensores de las casillas contiguas a la casilla objetivo no le roben el balón.
5. **StealBall:** Los jugadores a la defensa tienen la posibilidad de robar el balón si alguien del equipo rival recibe un pase o se mueve con el balón hacia una casilla contigua a la suya.
6. **Shoot:** Los jugadores con balón siempre tienen la oportunidad de disparar, con una probabilidad de que el balón vaya a portería en dependencia de su habilidad de tiro y distancia a esta, y en caso de que vaya a puerta hay otra probabilidad de que sea gol.

### 3.4 Acciones de los entrenadores

Los entrenadores no pueden actuar en todas las instancias de tiempo al igual que los jugadores esto tratando de imitar la realidad en la que en todo momento los entrenadores no realizan sus acciones. Las acciones que ejecuten los entrenadores solo serán aplicadas una vez que se reorganice el juego en el campo, esto solo ocurre cuando se produce un disparo a portería. Las posibles acciones de los entrenadores son las siguientes:

1. **ManagerNothing:** El entrenador entre sus decisiones posibles tiene la de no hacer nada.
2. **ChangePlayer:** El entrenador tiene la posibilidad de cambiar jugadores del terreno por otros que pertenezcan a su lista de jugadores posibles a alinear siempre que le queden cambios.
3. **ChangeLineUp:** Todo entrenador tiene la posibilidad de cambiar la formación en medio de un partido, penalizando a los jugadores que queden jugando fuera de sus posiciones habituales.

## 4 Integración con LLM

El componente de **NLP** que usamos en nuestro proyecto fue darle la facilidad al usuario que desea realizar la simulación del juego de fútbol la posibilidad de configurar dicha simulación en lenguaje natural. Para completar esta tarea usamos la api de **Gemini** en su versión pro, la idea básica fue realizar trabajo de **prompt engineer** a la consulta realizada por el usuario y así obtener la consulta original en un formato que se acople a nuestro sistema.

Los posibles parámetros de configuración de la simulación son los siguientes:

- **Equipos implicados en la simulación:** para obtener estos parámetros optamos por filtrar primero los equipos por liga ya que era imposible proporcionarle al modelo todos los equipos de nuestro dataset, así que primero el usuario deberá especificar la liga y luego los equipos implicados en el juego. Por tanto primero se le pide al modelo que identifique de todas las posibles ligas la que especificó el usuario y luego se le proporcionan todos los equipos de dicha liga y se le pide seleccionar el equipo local y el equipo visitante en el siguiente formato: FC Barcelona vs CF Real Madrid.
- **Estrategia de los entrenadores para elegir la alineación:** las posibles estrategias para el entrenador en este caso son **random** y **simulate** las cuales están basadas en las estrategias descritas anteriormente. Por lo que entonces se le pide al modelo que identifique estos parámetros.
- **Estrategia de los entrenadores para elegir las acciones durante el juego:** las posibles estrategias para el entrenador en este caso son **random**, **simulate** y **minimax** las cuales están basadas en las estrategias descritas anteriormente. Por lo que entonces se le pide al modelo que identifique estos parámetros.

- **Estrategia de los jugadores para elegir las acciones durante el juego:** las posibles estrategias para los jugadores en este caso son **random**, **heuristic** y **minimax** las cuales están basadas en las estrategias descritas anteriormente. Por lo que entonces se le pide al modelo que identifique estos parámetros.

```
Describe tu simulación, especifica:
* liga
* equipo local
* equipo visitante
* estrategia del manager local para elegir la alineación
* estrategia del manager visitante para elegir la alineación
* estrategias de los jugadores para tomar decisiones

Deseo simular el clásico del fútbol español, que los entrenadores elijan las estrategias al azar y los jugadores tengan estrategias heurísticas
```

## 5 Detalles de implementación

### 5.1 Mecanismo de simulación

Como estamos haciendo uso de varios algoritmos de búsqueda para la ejecución de las acciones de los agentes durante el juego fue necesario la creación de un mecanismo de memoria sobre las acciones que se ejecutan. Para ello se encapsuló la lógica de cada acción en la clase **Action** a la cual se le implementaron las funciones **execute** y **reset**. Siguiendo esta idea se definió un mecanismo de **Dispatch** mediante el cual los agentes ejecutan sus acciones y estas se guardan en un pila (**stack**) para después poder ser reseteadas en el momento que así se requiera. En el caso de las acciones de los managers que no tiene un efecto inmediato se hace uso de una nueva pila (**lazy\_stack**) en la cual se introducen estas acciones y en el momento indicado se comprimen y se aplican al juego de manera análoga a lo anterior este mecanismo de acciones lazy también cuenta con un mecanismo de **reset**.

Para simular el juego en concreto se hace uso de una clase **Simulator** la cual se encarga de ejecutar una instancia de tiempo y resetear una instancia de tiempo.

Para proveer un mecanismo de simulación de los agentes se emplea la clase **SimulatorAgent** la cual posee una función para simular el resto del comportamiento del medio que así lo requiera el agente, una función para simular el resto de la instancia de tiempo correspondiente así como las respectivas funciones de reset de cada una de las funciones antes mencionadas. Además se le provee al agente una función para interactuar directamente con el **Dispatch** del juego.

### 5.2 implementación de reglas y lógica difusa

### 5.3 Encapsulamiento de las estrategias de los agentes

Para cada tipo de acción de los agentes (jugar para los jugadores, realizar acciones durante el partido y establecer alineación para los técnicos) se crea una interfaz que permite encapsular dicho comportamiento y que sea posible permitir más de una configuración sin afectar el resto del funcionamiento del agente.

## 6 Análisis experimental

## 7 Conclusiones

## 8 Recomendaciones

Para futuros trabajos se recomienda un estudio más profundo del problema que permita el desarrollo de mejores heurísticas de evaluación que las planteadas en este trabajo. Por ejemplo no nos fue posible desarrollar una buena

heurística para las acciones del manager y algunas de las introducidas para escoger la alineación o escoger el mejor cambio las consideramos que no evalúan la alineación en conjunto. Además de consideramos importante mejorar las heurísticas empleadas en la función de evaluación lo que permitirá un mejor rendimiento de los algoritmos de búsqueda. Otra heurística no abordada en este trabajo pudiera consistir en la planificación de acciones conjuntas y colaborativas para la realización de acciones en equipo.

El estudio de mejores estrategias de búsqueda para este tipo de escenarios que pueden llegar a ser colaborativos y no deterministas puede ser otro punto importante para mejoras futuras.

Otro aspecto que queremos resaltar y no exploramos del todo es el reajuste de los pesos con que cuentan las estrategias heurísticas de los jugadores que en futuras implementaciones se pudiera tratar de desarrollar un algoritmo que sea capaz de reajustar estos parámetros en base a la experiencia obtenida por el agente.