

Simulación Multiagente Gestión de Proyectos.

Roger Fuentes Rodríguez
Kevin Manzano Rodríguez
Jackson Vera Pineda

September 30, 2024

1 Introducción

En este informe presentamos el desarrollo de una simulación multiagente basada en una arquitectura BDI (Beliefs, Desires, Intentions). El enfoque de esta simulación está en la interacción entre dos tipos de agentes: el **Project Manager (PM)** y los **Trabajadores**, enfocándonos en la capacidad de gestión y dirección del primero. Además, se modeló un entorno dinámico donde los agentes interactúan con proyectos, definidos mediante una ontología que relaciona las tareas, recursos, riesgos y oportunidades del mismo.

El objetivo principal fue simular la gestión de proyectos en un entorno de trabajo utilizando técnicas de inteligencia artificial, incluyendo **algoritmos genéticos**, **lógica difusa**, **aprendizaje reforzado**, y **LLMs** lo que permitió optimizar la asignación de tareas y la resolución de problemas en tiempo real.

2 Arquitectura BDI

El modelo BDI permite a los agentes razonar sobre sus creencias, deseos e intenciones, adaptándose a situaciones dinámicas y cambiantes en el contexto del proyecto. Los agentes toman decisiones basadas en su percepción del entorno y en los objetivos que se les asignan.

2.1 Beliefs (Creencias)

El agente Project Manager mantienen una representación interna de su entorno, que incluye información sobre el estado del proyecto, los recursos disponibles, los riesgos identificados, el progreso de las tareas y de los agentes con los que interactúa, en este caso los trabajadores. En sus creencias posee un conjunto de reglas que modifican sus deseos, y un subconjunto de dichas reglas serán las reglas activas, las que se aplicarán cada vez que el agente actúa. Dichas reglas no son un conjunto estático, ya que existen algunas que pueden crear nuevas reglas y activarlas o eliminar reglas ya existentes. Para la creación de reglas el agente toma ciertas condiciones en que se encuentra el medio, y se apoya en un LLM, pasándole dichas condiciones y el conjunto de deseos, esperando como respuesta un subconjunto de condiciones que implicarían la activación de un subconjunto de deseos, elegimos esta forma para mantener cierto sentido sobre las reglas. De esta forma creamos un conjunto variado de reglas para diferentes situaciones, permitiendo al agente adaptar su comportamiento según el estado del medio en que se encuentra y cambiando las prioridades del proyecto consigo. Las reglas tienen un peso, el cual será modificado a medida que se aplique y se observe su efectividad. El Project Manager no va a crear reglas continuamente, de ser así el conjunto de las mismas sería exponencial además de repetir reglas o tener poca variedad entre ellas, para decidir cuándo crearlas nos hemos apoyado en la lógica difusa. Nuestro agente se apoya en una función que caracterizará el medio en el que se encuentra, considerando el progreso del proyecto. Usamos funciones de pertenencia como Nivel de motivación, recompensas obtenidas, tiempo transcurrido, etc ; pasando dichos parámetros obtenemos sus valores fusificados, aplicamos las reglas definidas y obtenemos un valor categórico de como va el proyecto(mal, normal, bien). Esto nos permite evaluar la efectividad de las reglas que se han estado aplicando hasta ese momento y nos indica la necesidad de generar nuevas.

Los agentes Trabajadores mantienen una representación interna de las tareas que tienen asignadas, el progreso en las mismas, la motivación del equipo, algunas relaciones con otros agentes y ciertos parámetros que caracterizan al agente, como cuan trabajador es o su capacidad de resolución de problemas. Al igual que el Project Manager también posee un conjunto de reglas en sus creencias que modificarán sus deseos.

2.2 Desires (Deseos)

Los deseos reflejan los objetivos tanto a corto como a largo plazo que el agente intenta alcanzar, como completar un número máximo de tareas, evi-

tar problemas o minimizar los recursos utilizados. Se incluyen deseos básicos de los seres humanos como descansar, mantener cierto nivel de motivación o cooperar con otros agentes. Los deseos de los trabajadores son más personales, sin embargo los del Project Manager contemplan además los deseos con el proyecto y con el equipo en general.

2.3 Intentions (Intenciones)

Las intenciones son los planes que el agente decide llevar a cabo en función de sus deseos, sus creencias y sus percepciones; como asignar recursos a tareas específicas o escalar problemas al Project Manager. Están más cerca de lo que el agente puede hacer para cumplir esos deseos, aquí se maneja cómo el agente actuará ante deseos contradictorios como descansar y trabajar, digamos que aquí se define el orden de prioridad de los deseos y la acción que devolverá el agente.

3 Algoritmo Genético para la Planificación de Tareas

El **Project Manager** utiliza un algoritmo genético para encontrar la mejor permutación posible de las tareas del proyecto debido a que no hay garantía en que exista una permutación en la que se cumplan todas las restricciones de las tareas. A continuación, describimos el proceso:

3.1 Población y Generación de Individuos

Se parte de una población inicial compuesta por permutaciones siguiendo una heurística **greedy** para garantizar como mínimo las dependencias entre las tareas. Cada individuo de la población representa una permutación de las tareas del proyecto. Obtenemos un orden topológico de las tareas, con una modificación en el momento de la elección de los nodos con indegree 0, este se elige de manera aleatoria. Luego de establecido el orden, los nodos quedan divididos en niveles, sobre estos niveles se realizan cambios de manera aleatoria.

3.2 Selección y Cruzamiento

Para seleccionar los individuos que pasarán a la siguiente generación, implementamos un método de **selección por torneo**, teniendo en cuenta un factor de elitismo que puede variar. Este método selecciona a los individuos

más aptos para cruzarse y producir nuevos descendientes. Posteriormente, los individuos se cruzan y las mutaciones se incorporan como nuevos individuos, lo que permite introducir variabilidad en la población. Para el cruce de dos permutaciones dadas, p_1, p_2 , se elige un punto de corte aleatorio de la primera permutación luego se van agregando nodos en el orden de la segunda permutación que no incumplan con las dependencias, y cuando termina este proceso puede darse el caso de que falten elementos por colocar, por lo tanto se rellenan con elementos en el orden de la primera permutación pues ya en esta se ha garantizado que se cumplen las dependencias.

3.3 Mutación

La mutación en este contexto implica la creación de nuevas permutaciones de tareas, introduciendo cambios aleatorios menores en los individuos seleccionados para diversificar la búsqueda del óptimo global. Se prioriza la diversidad de la dificultad entre las tareas sucesivas, lo que permite un mejor balance en la distribución de las tareas. Dada una permutación p , se elige un nodo aleatorio, luego se "empuja" este nodo en los niveles del orden topológico, de manera que se garantice que no se rompan las dependencias (los nodos que dependen de el también se empujarán).

3.4 Función de Optimización

Definimos dos funciones de optimización para el algoritmo. Una función pondera las restricciones cumplidas e incumplidas de las tareas en la permutación, como su posición respecto a su fecha mínima de comienzo y su fecha de terminación máxima o sus dependencias hacia otras tareas. Esta función devuelve un valor numérico y resultó más efectiva. La otra función utilizada para evaluar la calidad de cada permutación de tareas incluye varios factores que pueden ajustarse mediante reglas de **lógica difusa**, esta función nos brinda una mayor flexibilidad a la hora de darle prioridades a ciertos parámetros sobre otros, ya que al crear reglas nuevas no tendríamos que modificar el código cada vez que cambie el objetivo del proyecto o crear varias funciones muy parecidas, el equipo propone crear un conjunto de conjuntos de reglas donde cada conjunto tribute a priorizar alguno de los parámetros a evaluar en las tareas; por ejemplo otorgar mayor peso al número de tareas posibles a completar frente a la calidad de las mismas.

Se incorporaron penalizaciones basadas en el incumplimiento de dependencias entre tareas, el exceso de tareas de dificultad similar y las tareas que exceden su *deadline*. Además, se aplican bonificaciones por mantener una

variedad de dificultades consecutivas y completar las tareas de alta prioridad a tiempo.

4 Interacción entre Agentes

Una vez que el **Project Manager** obtiene la mejor permutación de tareas, comienza la interacción con los **Trabajadores**. El PM asigna tareas a los trabajadores, supervisa su desarrollo y ajusta las estrategias en función del progreso.

4.1 Asignación de Tareas Basada en Habilidades

La asignación de tareas se basa en las habilidades de los trabajadores y la dificultad de las tareas. El PM utiliza una heurística que busca minimizar la diferencia entre la dificultad de la tarea y la capacidad de resolución de problemas del trabajador, garantizando una asignación eficiente.

4.2 Generación de Hitos

El PM genera **hitos** basados en proyecciones del avance del proyecto. Estos hitos no son estáticos, sino que se generan dinámicamente según la situación actual del proyecto. Para esto, el agente evalúa el estado del proyecto utilizando **lógica difusa**, lo que le permite proyectar los hitos de manera flexible, ajustándose a cambios en el entorno. Dependiendo del estado del equipo (motivación, productividad, disponibilidad de recursos), los hitos pueden ser conservadores, normales o entusiastas. El cumplimiento de los mismos también influye sobre la percepción del agente del estado del proyecto.

5 Modelado del Proyecto con Ontologías

Para representar los proyectos y su estructura, utilizamos una **ontología** que incluye los siguientes componentes:

- **Recursos:** Elementos que el proyecto requiere para completarse, relacionados con las tareas.
- **Riesgos:** Posibles problemas que podrían surgir durante el proyecto, como retrasos en la ejecución o falta de personal.
- **Oportunidades:** Situaciones favorables que pueden mejorar el desempeño del proyecto, como alta motivación o ahorro de costos.

- **Tareas:** Actividades específicas que deben completarse, cada una con su duración, prioridad, dificultad, dependencias y recursos necesarios.

Este enfoque permite una representación estructurada y detallada del proyecto, facilitando la toma de decisiones por parte del PM, incluyendo la creación de hitos personalizados.

6 Procesamiento de Tareas desde Texto

Para modelar la entrada de tareas en formato de texto natural, utilizamos una **API de Geminis** con un *prompt* diseñado para convertir un texto de entrada en una lista estructurada de tareas, cada una con sus respectivos atributos y dependencias. Esto nos permitió automatizar la creación de tareas dentro de la simulación y garantizar que las tareas ingresadas manualmente siguieran una estructura consistente.

7 Resumen de la Simulación

Dado el contenido registrado de la simulación, se le envía esta información al *LLM* para que genere un informe que contenga el estado del proyecto, el progreso de las tareas y las recomendaciones para la siguiente iteración.

8 Aprendizaje Reforzado

El **Project Manager** no solo asigna tareas y supervisa a los trabajadores, sino que también incorpora un sistema de **aprendizaje reforzado**. A través de la experiencia, el PM aprende a optimizar la asignación de tareas basándose en las cualidades de los trabajadores y sus respuestas ante situaciones complejas. El aprendizaje reforzado permite que el PM mejore con el tiempo y optimice las decisiones de asignación de tareas de manera adaptativa, maximizando la eficiencia y el uso de los recursos del proyecto. Entre sus creencias contiene cuánta confianza posee en cada trabajador, y a medida que los trabajadores se desempeñan la va modificando, al igual que las relaciones de cooperación entre los mismos. Además de tener una memoria para recordar las decisiones tomadas respecto a los problemas surgidos y aprender de la efectividad de las mismas.

9 Estudio Estadístico

Resulta interesante descubrir como las dinámicas entre el Project Manager y los Trabajadores, en nuestra simulación, nos proveen respuestas a diferentes interrogantes. En nuestro estudio nos centramos en el desempeño de equipos, y su habilidad para resolver tareas en un tiempo determinado; con este fin, se definió un proyecto (conjunto de tareas) lo suficientemente complejo que permita evaluar, de manera justa, grupos con diferentes características.

9.1 Tiempo de Finalización

Una de las variables de interés para nuestro análisis es el tiempo de finalización de las tareas. Todos los equipos se enfrentaran al mismo proyecto, con una ventana de tiempo holgada, de ahí que la completitud de la tarea no debe ser relevante; sin embargo, la capacidad de los equipos para, dentro de esta ventana, reducir el tiempo de ejecución de todas las tareas es un indicador relevante de su desempeño.

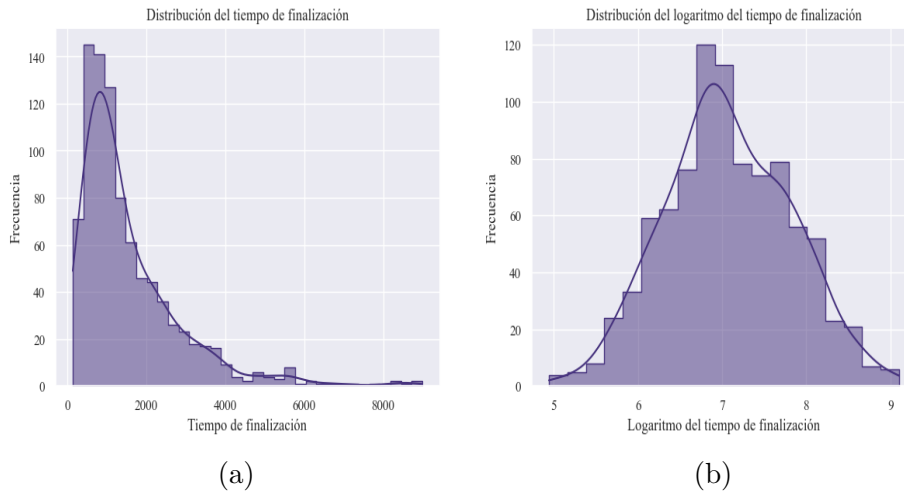


Figure 1: Histograma de la variable tiempo de finalización.

10 Conclusiones

La simulación multiagente con la arquitectura BDI nos permitió modelar un entorno de trabajo colaborativo donde los agentes toman decisiones complejas de manera autónoma. El uso de un algoritmo genético para la optimización de tareas, junto con la lógica difusa para la evaluación de las situaciones,

resultó en una mayor flexibilidad y adaptabilidad del sistema. La incorporación de aprendizaje reforzado mejoró las capacidades del **Project Manager**, haciéndolo más eficiente en la gestión de proyectos y la optimización del uso de los recursos a largos plazos.