



**TECNOLÓGICO  
DE MONTERREY®**

# **Modelación de sistemas multiagentes con gráficas computacionales.**

Actividad Integradora

Hazel Astrid Ángeles Quevedo.

A01275792

23/11/2021

## Contexto:

Se pide un análisis sobre la ejecución de un programa con robots dentro de un almacén lleno de cajas inicializar las posiciones iniciales de las K cajas. Todos los agentes empiezan en posición aleatorias vacías. Se ejecuta en el tiempo máximo establecido. Durante la ejecución, se recopilará el tiempo necesario hasta que todas las cajas están en pilas de máximo 5 cajas y el número de movimientos realizados por todos los robots.

## Análisis:

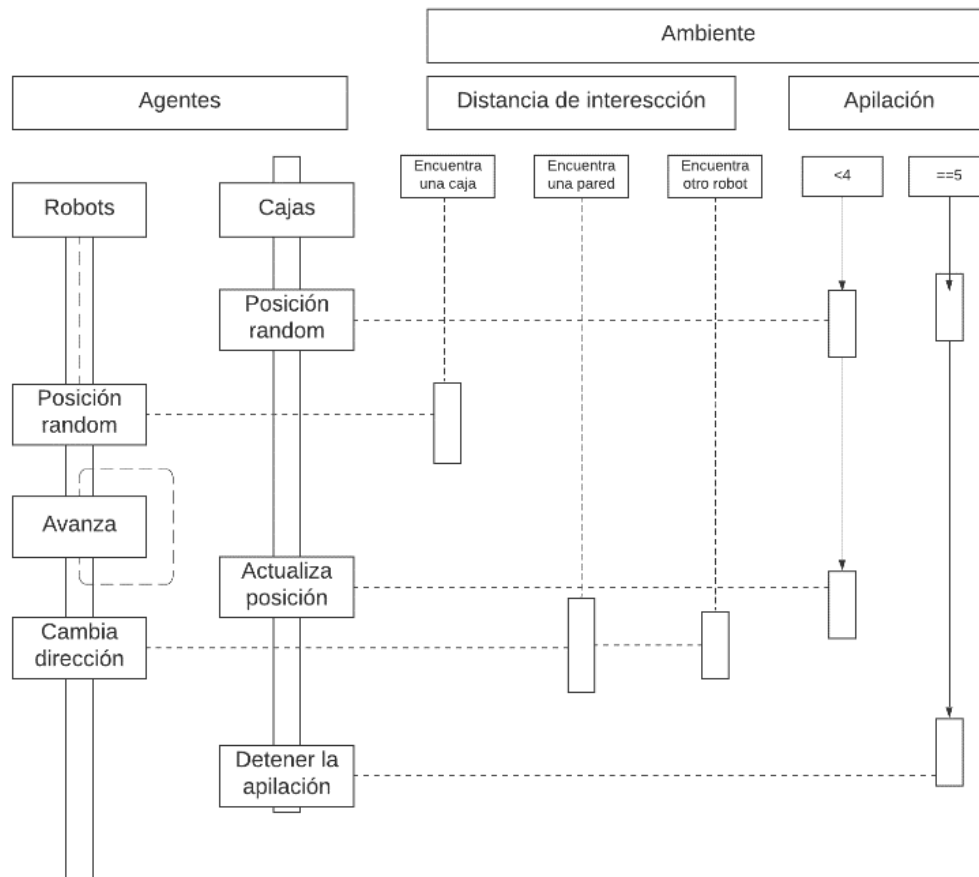
Según la descripción del problema se debe realizar un programa de multiagentes que se muevan de manera aleatoria para que los robots pueden llevar las cajas y apilarlas de cinco en cinco. Se inician en posiciones rango por lo que en cada interacción el tiempo que le puede llevar será diferente.

Para realizar este programa se requieren de múltiples funciones primeramente una función para crear el tablero, así como los robots y cajas de manera rango. Necesitamos una función que permita mover a los robots en cualquier dirección y en caso de encontrarse con algún obstáculo que cambie su dirección para no seguir chocando con este, además de que no se salga del tablero. Así mismo necesitamos una función que nos dará información sobre las cajas, actualizar su posición en la que se encuentra y obtener en caso de que ya se encuentra otra caja apilada sobre esta.

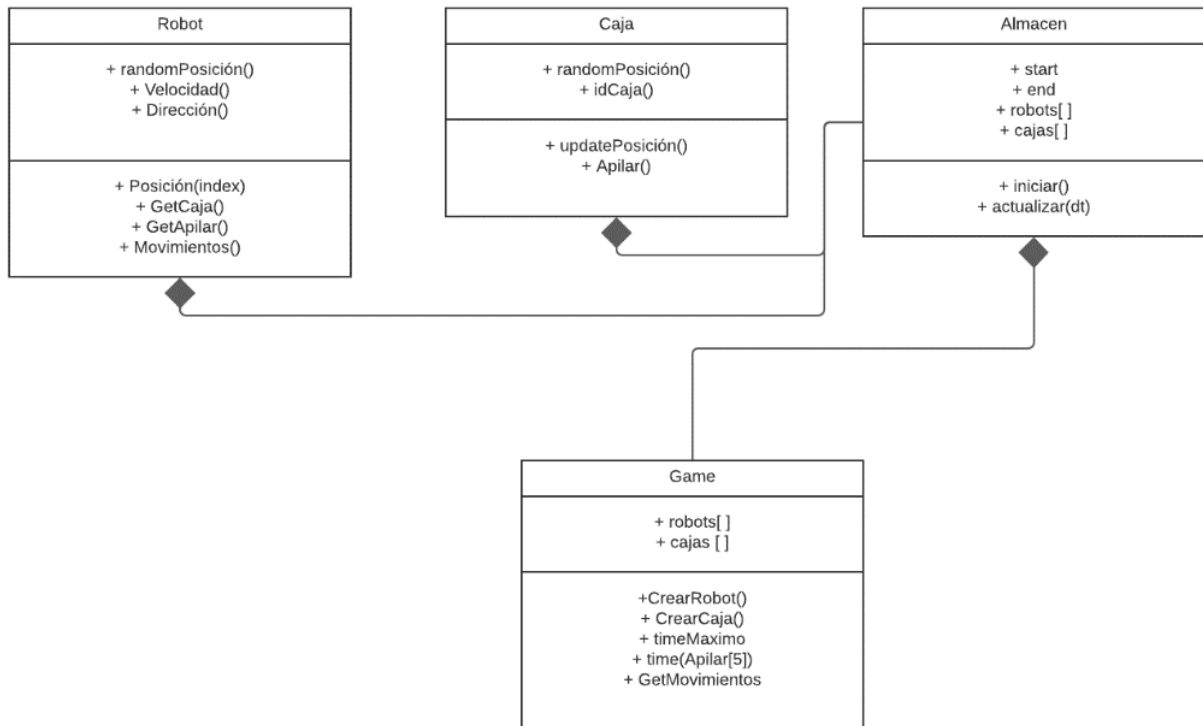
Para el ambiente necesitamos manejar la parte de que reciba las interacciones con todos los agentes involucrados y se comuniquen con ellos, aunque estos no se comuniquen directamente entre sí. Así se necesita una parte en donde obtendremos la parte de la ejecución, que será cuantos movimientos se tuvieron que realizar y el tiempo de ejecución que tardaron en completar su objetivo los robots.

Mediante el diagrama de protocolos podemos ver la interacción que se tendrán los agentes con el ambiente de manera que puedan interactuar y formar un ambiente dinámico de parte de los robots, estos podrán avanzar y completar su misión que es llevar cajas y apilarlas.

Los diagramas de protocolo se presentan con una definición deliberadamente amplia, que identifica vagamente el perímetro dentro del cual los estudiantes pueden moverse en el proceso de diseño. Primero, se introduce el concepto de protocolo de investigación. Aquí, se utiliza una definición específicamente amplia: los diagramas de protocolo se describen simplemente como llevara la interacción entre los agentes. Al mismo tiempo, describir el diagrama como un artefacto visual compacto plantea un desafío de diseño estimulante para los estudiantes, ya que pueden apreciar la necesidad de condensar un proceso muy complejo en una cantidad limitada de espacio.



Con el diagrama de clases podemos tomar en cuenta de manera mas clara la interacción que podrían tener entre las clases los diferentes elementos dentro del programa, primero tenemos a Game el cual será el encargado de conformar y obtener los resultados finales para su análisis, como el tiempo máximo, el tiempo que tardan los robots en completar la misión y los movimientos que se tuvieron que realizar. Luego tenemos el almacén que recibir y será el encargado de realizar la interacción entre las cajas y los robots con un inicio, fin y los elementos, además de actualizar el tiempo que está tomando. Las cajas tendrán propiedades de inicio como su posición rango y que numero de caja es para en el momento en que el robot interactúe con esta podamos saber de que caja se trata, puede actualizar su posición y saber si aun se puede apilar o no. El robot tiene una posición rango de inicio, una velocidad a la que se moverá, y la dirección hacia donde ira según lo en encuentre a su alrededor, puede obtener su posición, hacer un conteo de sus movimientos y darse cuenta si hay una caja para tomarla y apilarla en caso de que se pueda.



## Conclusión:

Los multiagentes pueden solucionar problemas complejos de manera eficaz como es el caso de este problema solucionado, sin embargo puede llegar a tomar un cierto tiempo dependiendo de cuantos agentes y cajas tengamos para realizar la ejecución, esto debido a que las posiciones son random y los robots de igual forma se mueven de manera random hasta que encuentran una caja o un obstáculo para cambiar su dirección, por lo que la manera en que se podría optimizar mejor el tiempo es hacer que los robots reciban las posiciones de las cajas y que estos tengan como meta encontrar estas posiciones para no dar vueltas en otros lados y únicamente dirigirse hacia su objetivo, habría que mejorarse algunas cosas como que cada robot reciba posiciones diferentes de las cajas y que sean las que estén más cercas a los robots por que de otra manera puede haber varios robots en busca de una misma caja o van hacia cajas que tal vez ya fueron tomadas por otros robots. Para esta implementación se necesitaría hacer uso de una inteligencia más avanzada y que con cada ejecución pueda aprender de manera más optima para dar las posiciones de las cajas mucho más cercanas con un orden optimo.