



# Tecnológico de Monterrey

**Instituto Tecnológico y de Estudios Superiores de Monterrey**

***Campus Estado de México***

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

Evidencia 1. Actividad Integradora

TC2008B

Grupo: 301

Equipo: 4

**Integrantes del equipo:**

*Aleny Sofia Arévalo Magdaleno | A01751272*

*Luis Humberto Romero Pérez | A01752789*

*Valeria Martínez Silva | A01752167*

*Pablo González de la Parra | A01745096*

**Fecha de entrega:**

**21 de Noviembre de 2022**

## Descripción del medio ambiente

De acuerdo con la descripción de la actividad integradora, el ambiente de este proyecto simula la organización de un almacén de cajas con ayuda de un grupo de robots. El almacén consiste en una superficie cuadrada rodeada por cuatro paredes, y  $k$  número de tarimas definido por la cantidad de cajas a ordenar. A continuación se mencionan ciertas características relevantes del ambiente:

- 1 cuadrícula de 15x15
- 4 bordes representados por paredes
- $j$  número de tarimas ( $k = j/5$ )

A continuación se listan los tipos de agente que involucra el sistema:

- Agente “Robot”
- Agente “Caja”

Contamos con 5 agentes de tipo “Robot”, los cuales se encargan de simular la organización de un almacén, acomodando los  $k$  agentes “Caja” distribuidas en el suelo sobre las tarimas disponibles, de tal manera que queden ordenadas en pilas de máximo 5 cajas. Como se menciona anteriormente, el número de tarimas se calcula de acuerdo al número de cajas ( $k$ ) que se requieran acomodar. Esto se logra de la siguiente manera:

- Si  $k$  es menor o igual a 20 cajas
  - Se generan las tarimas en las esquinas de la cuadrícula
- Si  $k$  es mayor a 20 cajas
  - Además de las tarimas en las esquinas, se generan el número suficiente de tarimas para apilar las cajas disponibles. Esto se logra cambiando los valores  $x$  y  $y$  respectivamente con el fin de acomodarlas en los extremos de la cuadrícula.

Aunado a esto, las variables iniciales de nuestro ambiente permiten ser modificadas directamente desde Unity, al igual que en la visualización dentro del navegador web deseado. A continuación se describen de manera más detallada las características del ambiente.

- **Mayormente inaccesible**

El ambiente es mayormente inaccesible debido a que el agente no tiene acceso al estado completo o total del ambiente. Es decir, se requiere de un estado interno con el fin de poder tomar una decisión de acción. Dentro de nuestro ambiente se considera que este es mayormente inaccesible debido a la limitación que poseen los agentes que lo conforman.

En el caso del agente “Robot”, su conocimiento del ambiente es un poco más extenso, pues conoce su ubicación y la de sus cuatro vecinos, la ubicación de las tarimas, y un diccionario con la ubicación de las cajas detectadas por otros robots. Por otro lado, el agente “Caja” solo puede acceder a la celda en la que se encuentra, y si esta está siendo transportada. Es decir, para este agente el resto del ambiente se encuentra mayormente “oculto”.

- **Mayormente determinista**

El ambiente es mayormente determinista debido a que el estado actual y las acciones realizadas por el agente llegan a determinar el estado siguiente del ambiente. Es decir, se

puede estar al tanto del mayor número de aspectos accesibles del ambiente. Esto es notable cuando se observa el comportamiento y la interacción que poseen ambos agentes.

En el caso del agente “Robot”, la posición actual, la lista de cajas por recoger, la distancia de las tarimas (calculada iterativamente), al igual que la presencia de otros agentes (tanto “Robot” como “Caja”) a su alrededor determina de manera mayormente objetiva su siguiente movimiento. En el caso del agente “Caja”, su siguiente posición depende directamente de los cambios de movimiento que tenga el agente “Robot”. Sin embargo, no podemos decir que es un ambiente completamente determinista pues el agente “Robot” no siempre conoce la ubicación de todas las cajas que se encuentran a su alrededor, al igual que si algún robot trae consigo una caja, la cual afecte su trayecto.

- **No episódico**

El ambiente es no episódico debido a que las acciones del agente no se dividen en episodios o ciclos previamente establecidos. Es decir, el agente tiene que pensar de manera adelantada para actuar. En este caso, el comportamiento de los agentes al igual que su interacción con el ambiente no se encuentra ligado de manera episódica. Esto debido a que las acciones que se realizan dependen completamente de las acciones producidas anteriormente. Dentro de nuestro ambiente, las acciones de los agentes “Robot” dependen mayormente de la celda a la que hayan decidido avanzar y el contenido de dicha celda, mientras que en los agentes “Caja”, el cambio en su posición depende directa y objetivamente de la posición del agente “Robot” que los haya recogido, como se mencionó anteriormente.

- **Dinámico**

El ambiente es dinámico debido a que existe la posibilidad de que el ambiente presente modificaciones mientras que el agente delibera. Es decir, el ambiente cambia con el paso del tiempo. Debido a la naturaleza de ciertas simulaciones, a veces se considera a un ambiente como estático debido a la posibilidad de manipular el tiempo dentro de esta. En nuestro caso, nuestro ambiente es dinámico debido a que este cambia respecto al tiempo.

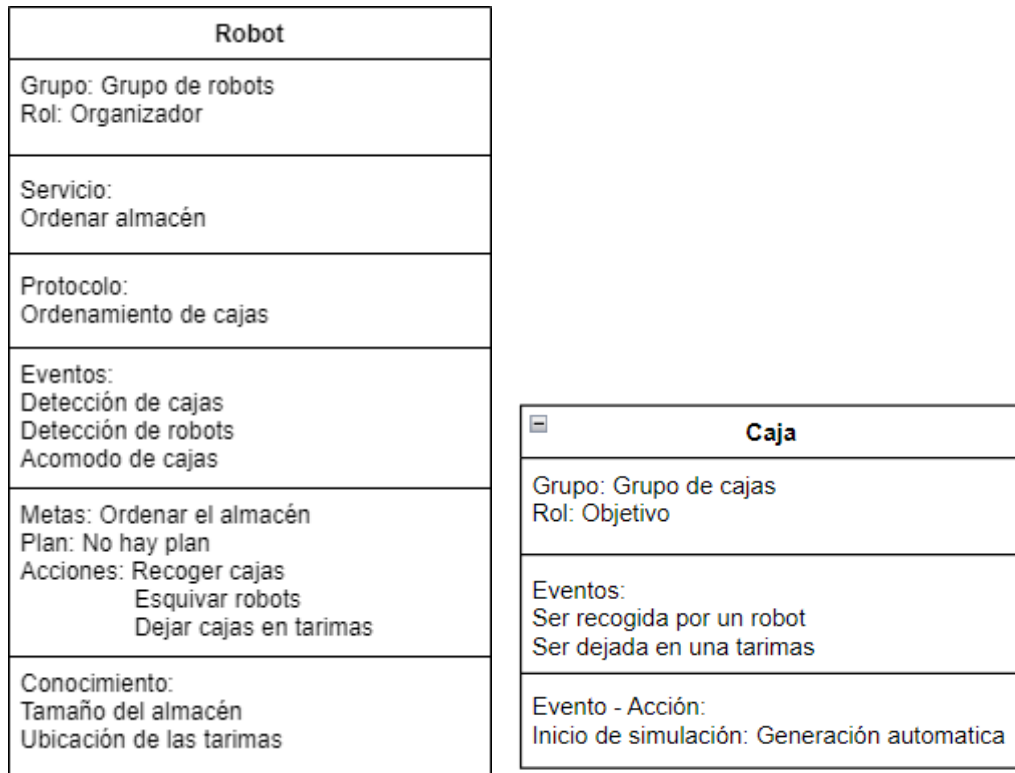
Esto es notable debido a que, mientras los agentes deciden su siguiente movimiento (o deliberan), el ambiente se encuentra cambiando debido a la deliberación de los demás agentes que lo constituyen. Por ejemplo, cuando un agente “Robot” se encuentra decidiendo su siguiente movimiento, otro agente “Robot” pudo haber detectado una caja y haberla recogido o añadido a la lista de cajas por recoger. Esto es incluso más notable cuando un agente aparta una celda en la que se quiere mover, mientras que otros agentes se encuentran decidiendo su siguiente movimiento.

- **Discreto**

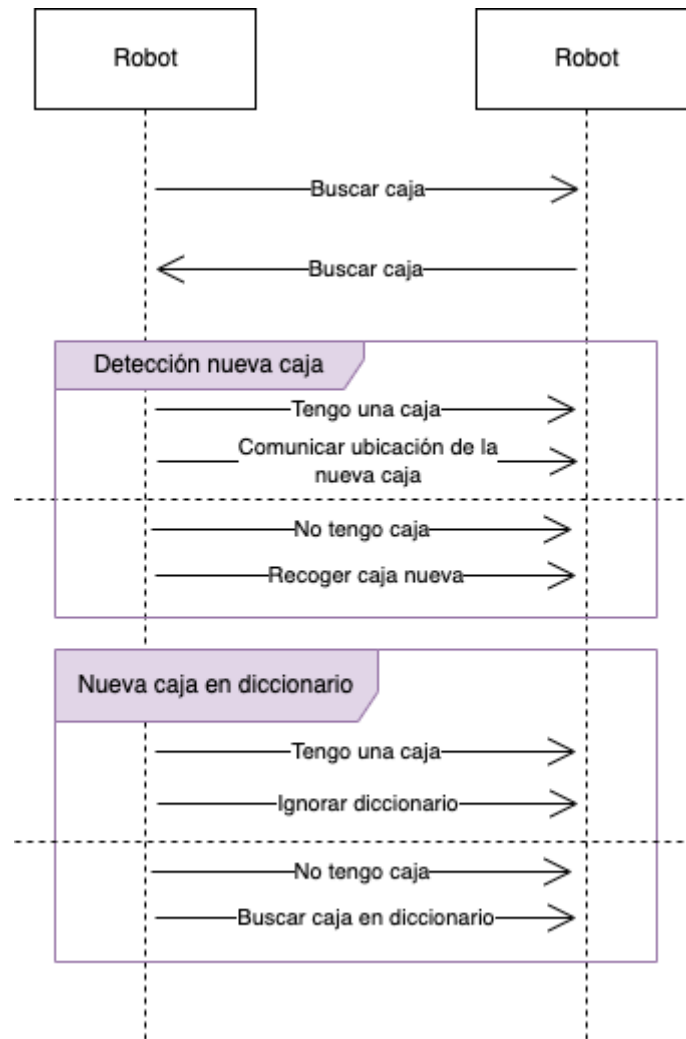
El ambiente es discreto debido a que existe un número limitado de acciones y percepciones. Es decir, es posible listar o enumerar estos hechos en el ambiente. Esto es más notable en simulaciones, las cuales prohíben la utilización de rangos (valores continuos) para determinar una o muchas variables. Dentro de nuestro ambiente, este se considera discreto debido a que se pueden conocer todas las acciones o percepciones que un agente pueda llegar a ejecutar. En el caso de los agentes “Robot”, se conocen los posibles movimientos que cualquier agente de este tipo puede llegar a realizar, como es el avanzar, recoger una caja, añadir caja al

diccionario de cajas sin recoger, y depositar la caja en una tarima. Por otra parte, los agentes “Caja”, al poseer solamente una acción que cambie su entorno, que es ser cargadas por el agente “Robot”, es más notable observar que estas acciones se pueden listar de manera concisa.

### Diagramas de clases de agentes



### Diagrama de protocolo de interacción



### Estrategia cooperativa de nuestro sistema y descripción de agentes

De acuerdo con los requisitos de la actividad integradora, el sistema multiagentes construido se encarga de acomodar o limpiar un almacén, al utilizar robots que buscan, recogen y apilan cajas en las distintas tarimas disponibles de manera respectiva. Conformado por un agente “Robot” y un agente “Caja”, estos se encargan de representar el funcionamiento correcto del sistema dentro de nuestro ambiente. A continuación se describen los dos agentes, al igual que sus comportamientos de manera detallada.

#### *Agente “Caja”*

En el caso de los agentes “Caja”, su funcionalidad se encuentra directamente relacionada con el agente “Robot”. Esto debido a la estrecha relación que existe sobre sus comportamientos. Al comenzar la simulación, estas aparecen en lugares aleatorios dentro de la cuadrícula. Y se mantienen en la misma posición (o inmóviles) hasta que un agente “Robot” se encargue de

detectarlas, y finalmente de recogerlas. Al ser recogidas, las cajas se mueven de acuerdo a la posición del robot encargado de cargarlas. Esto continúa hasta que se llega a una tarima (con una cantidad menor a 5 cajas), en donde la caja se traslada a esa tarima, y termina su recorrido. Es entonces cuando el movimiento de la caja se bloquea, y vuelve a regresar a su estado inicial estático.

Como se puede observar, el comportamiento del agente “Caja” es simple y mayormente condicionado por el agente “Robot”. Sin embargo, aunque sus acciones sean sencillas de modelar, este forma gran parte de las decisiones y comportamiento de los demás agentes dentro de nuestro ambiente, al igual que se encargan de comprobar el éxito y optimización de nuestro sistema al modificar sus atributos, como el número de cajas por simulación.

### ***Agente “Robot”***

Como se mencionó anteriormente, se cuenta con otro agente denominado agente “Robot”, el cual se encarga de limpiar y organizar el almacén modelado en la situación problema. Este agente puede ser considerado una mezcla entre reactivo y no reactivo, debido a sus etapas de búsqueda aleatoria, al igual que por la presencia de una estrategia cooperativa, las cuales se explicarán a detalle más adelante. Al comenzar la simulación, estos agentes “Robot” aparecen en posiciones aleatorias. Cuando no hayan detectado una caja que recoger, estos se mueven de manera aleatoria a celdas vacías dentro de su “vecindario”. En el momento que se detecta una caja, el agente “Robot” la recoge, modificando su estado a “cargando” (o has\_box). Esto representa que un robot trae consigo una caja, y su objetivo es dejarla en una tarima. Para decidir a qué tarima debe dejar la caja que está cargando, se toma en cuenta los siguientes pasos:

1. El robot tiene una caja.
2. Registrar todos los vecinos vacíos disponibles (vecindario).
3. Por cada vecino, sacar la distancia de todas las tarimas no llenas (menor a 5 cajas).
4. Sacar el menor valor de la distancia con la siguiente fórmula.

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Donde  $(x_2, y_2)$  representa las coordenadas de la tarima, y  $(x_1, y_1)$  representa la celda del vecino actual.

5. Sacar el menor valor de la distancia de cada vecino respectivamente.
6. Se avanza al vecino (o coordenadas) con el menor valor de distancia.

Con esto aseguramos que, no solo un agente “Robot” elige siempre la tarima más cercana a su posición, sino que también toma el camino más directo (o corto) posible hacia esa tarima. Esto optimiza de manera drástica la eficiencia de los robots en organizar el almacén.

### ***Estrategia cooperativa***

Aunado a esto, existe otra funcionalidad, la cual incorpora una estrategia cooperativa entre estos agentes del mismo tipo. Cuando un robot trae consigo una caja, y dentro de su recorrido (explicado anteriormente) se encuentra con otra caja (debido a que estos robots solo pueden cargar una caja a la vez) este pide ayuda. El procedimiento de asignar y recoger una caja asignada consiste en lo siguiente:

1. El robot tiene una caja y se encuentra con otra.
2. Guarda la posición o coordenadas de la caja que está en el suelo (no recogida) de manera global, y sigue con su recorrido.
3. Al primer robot que no tiene una caja (en ese momento) se le asigna esta caja no recogida.
4. Se sigue el mismo procedimiento de calcular la distancia a la tarima más cercana, pero ahora  $(x_1, y_1)$  representa la posición de la caja que se le fue asignada al robot vacío.
5. Cuando llega a la caja asignada, se elimina esa caja de manera global, y se sigue el procedimiento normal de llevarla a la tarima más cercana
6. Si el robot en su recorrido de alcanzar la caja que se le fue asignada se encuentra con otra caja (completamente distinta), este la recoge, y vuelve a declarar de manera global la caja que se le fue asignada en un principio (con el fin de que otro robot la pueda recoger)

Como se puede observar, este procedimiento permite la comunicación y cooperación entre los agentes “Robot”, al compartir parte de su conocimiento, con el fin de agilizar el procedimiento de recolección de agentes “Caja”. Los agentes “Robot” cuentan con el siguiente procedimiento de la arquitectura “Subsumption Architecture”.

Agente "Robot"	
9	No tiene caja y no detecta caja (Nada) -> Avanzar
7	No tiene caja, no tiene objetivo y detecta caja -> Recoger caja
8	No tiene caja y tiene objetivo -> Avanzar a objetivo
6	No tiene caja, tiene objetivo y detecta otra caja -> Pedir ayuda y recoger caja
5	No tiene caja y llega a su objetivo -> Recoger caja
4	Tiene caja y detecta otra caja -> Pedir ayuda
3	Tiene caja y no la puede soltar -> Avanzar a tarima más cercana
2	Tiene caja y la puede soltar -> Soltar caja en tarima
1	Detectar obstáculo -> Elegir nueva celda

### Implementación de estrategias cooperativas y mejoras

Como se puede observar previamente, el sistema incorpora la utilización de un sistema de colaboración entre los agentes "Robot", implementando una estrategia cooperativa para optimizar y reducir el tiempo en la recolección de cajas en el almacén. Sin embargo, de acuerdo con la descripción de la situación problema, al igual que la construcción del ambiente, existen otras estrategias cooperativas y formas de comunicación que podrían permitir incrementar la eficiencia de los agentes "Robot" en su objetivo de limpiar el almacén. A continuación se describen de manera detallada.

La primera estrategia cooperativa que se puede llegar a implementar consiste en que los agentes "Robot" dividan la cuadrícula en cuadrantes (asignados a cada agente) de acuerdo con la posición en la que aparecieron. Al hacer esto, cada agente se puede hacer cargo de comenzar a limpiar o recoger las cajas de su propio cuadrante. Esto es beneficioso debido a dos razones principales. La primera es que la utilización del acercamiento "Divide y vencerás" reduce significativamente el tiempo de búsqueda de cajas sin recoger debido a la reducción del espacio que un agente puede llegar a recorrer.

La segunda razón consiste en que, al darse cuenta de que no existan cajas sin recoger en uno de los cuadrantes, el agente "Robot" puede ser asignado un cuadrante, el cual todavía tenga un número de cajas sin recoger, disminuyendo el tiempo de recolección de manera drástica. En este caso, esta estrategia dependerá de ciertos factores como, el número de cuadrantes por división, la presencia de tarimas en cada cuadrante, la detección del número de cajas en cada cuadrante.



Este último factor es de suma importancia debido a que de este depende que los agentes “Robot” puedan terminar con su cuadrícula y ayudar a otro agente a terminar de recoger las cajas en las suyas. Claramente, la comunicación entre los agentes tiene que ser constante, no solamente para ponerse de acuerdo en cómo delimitar el perímetro de manera eficiente, sino también para pedir ayuda a otro agente cuando este haya acabado con su propio cuadrante.

La segunda estrategia cooperativa que se pudiera implementar, consiste en la asignación de responsabilidades. Esta consiste en asignar a unos agentes “Robot” como buscadores, mientras que otros ejercen la tarea de recolectores. Como su nombre lo indica, los agentes denominados buscadores no pueden recoger cajas; sin embargo, estos se encargan de detectar las cajas que existen junto con sus posiciones, con el fin de que los recolectores las lleven a las tarimas correspondientes.

Esta técnica permite optimizar el movimiento de los buscadores con el fin de no estar teniendo que regresar a la tarima y perder el progreso en la zona en la que se encontraban. De igual manera, esta comunicación se puede mejorar al solamente llamar a los buscadores cuando se hayan encontrado un cierto número de cajas, evitando un cuello de botella en los desplazamientos de los buscadores.

Aunque existen diversas estrategias cooperativas, si nos enfocamos en la que utilizamos para la resolución de la actividad integradora, esta se podría mejorar de la siguiente manera. Para empezar, cuando un agente “Robot” con una caja detecte una caja, actualmente se llama al primer agente que se encuentre sin una caja. Sin embargo, sería más eficiente realizar una solicitud con el fin de que el robot sin caja que se encuentre más cercano sea el que vaya a recoger esa caja.

También, existen ciertos casos en los que dos o más agentes se quedan atorados debido a que se encuentren tratando de acceder a la misma ruta optimizada para llegar a su destino. Y aunque esto no suceda muchas veces, se podría mejorar implementando la condicional que, cuando existan dos agentes en el mismo grupo de celdas, uno de ellos avanzara a una dirección opuesta con el fin de permitir al paso de un agente a la vez. Esta comunicación sería fundamental para la resolución de estos casos extraordinarios.

Finalmente, una última adición a esta estrategia cooperativa consiste en que, cuando exista un agente “Robot” con una caja, este trazara el camino más rápido hacia una tarima. Y que todos los agentes que se encuentren actualmente en alguna de esas celdas, que se movieran o tuvieran prohibido visitarlas hasta que el agente hubiera dejado la caja en la tarima respectivamente. Esta podría ser una de las estrategias no solamente más notables sino también más realistas que se pudieran implementar a nuestra solución.

