



**Tecnológico  
de Monterrey**

ITESM- Campus Puebla

Modelación de sistemas multiagentes con gráficas  
computacionales (Gpo 1)

**Actividad Integradora**

**Alumno:**

Jorge Tato A01733616

01/12/21

## Descripción del Problema

¡Felicidades! Eres el orgulloso propietario de 5 robots nuevos y un almacén lleno de cajas. El dueño anterior del almacén lo dejó en completo desorden, por lo que depende de tus robots organizar las cajas en algo parecido al orden y convertirlo en un negocio exitoso.

Cada robot está equipado con ruedas omnidireccionales y, por lo tanto, puede conducir en las cuatro direcciones. Pueden recoger cajas en celdas de cuadrícula adyacentes con sus manipuladores, luego llevarlas a otra ubicación e incluso construir pilas de hasta cinco cajas. Todos los robots están equipados con la tecnología de sensores más nueva que les permite recibir datos de sensores de las cuatro celdas adyacentes. Por tanto, es fácil distinguir si un campo está libre, es una pared, contiene una pila de cajas (y cuantas cajas hay en la pila) o está ocupado por otro robot. Los robots también tienen sensores de presión equipados que les indican si llevan una caja en ese momento.

Lamentablemente, tu presupuesto resultó insuficiente para adquirir un software de gestión de agentes múltiples de última generación. Pero eso no debería ser un gran problema ... ¿verdad? Tu tarea es enseñar a sus robots cómo ordenar su almacén. La organización de los agentes depende de ti, siempre que todas las cajas terminen en pilas ordenadas de cinco.

### 1. Realiza la siguiente simulación:

- a. Inicia las posiciones iniciales de las  $k$  cajas. Todas las cajas están a nivel de piso, es decir, no hay pilas de cajas.
- b. Todos los agentes empiezan en posiciones aleatorias vacías.
- c. Se ejecuta el tiempo máximo establecido.

### 2. Deberás recopilar la siguiente información durante la ejecución:

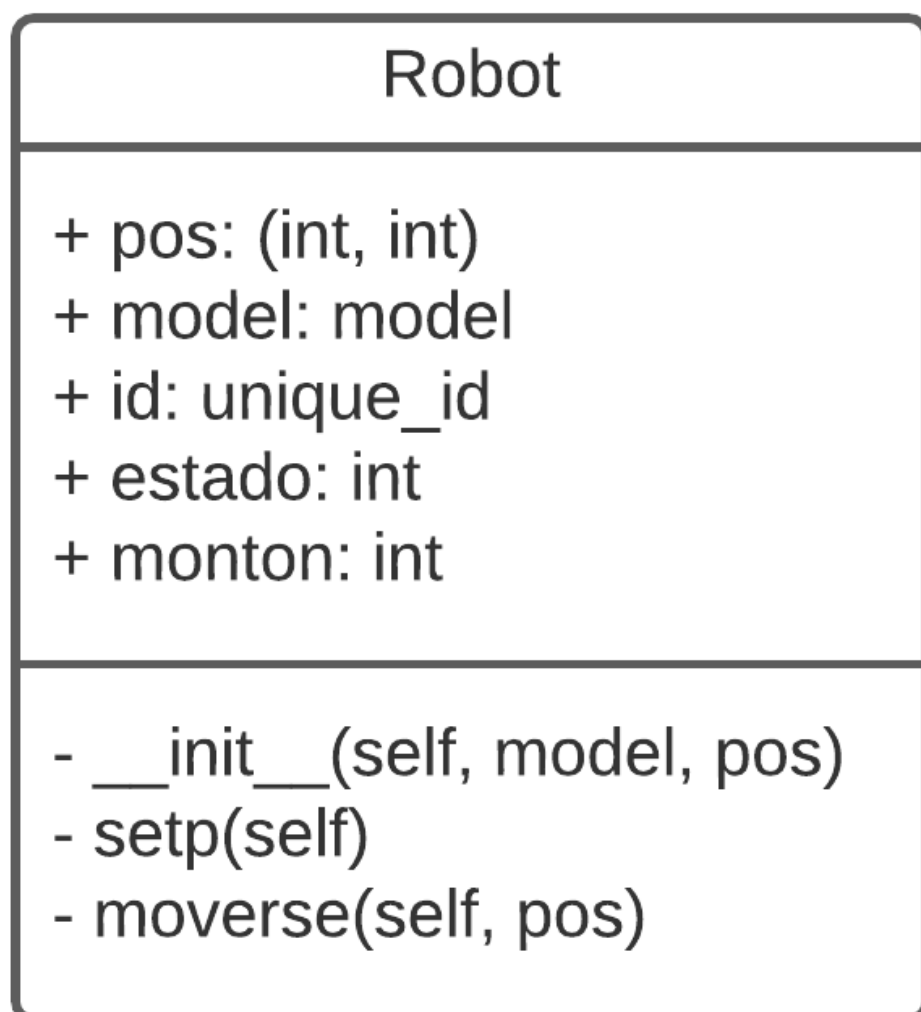
- a. Tiempo máximo hasta que todas las cajas están en pilas de máximo 5 cajas.
- b. Número de movimientos realizados por todos los robots.
- c. Analiza si existe una estrategia que podría disminuir el tiempo dedicado, así como la cantidad de movimientos realizados. ¿Cómo sería? Descríbela.

## Agentes Involucrados en el Problema

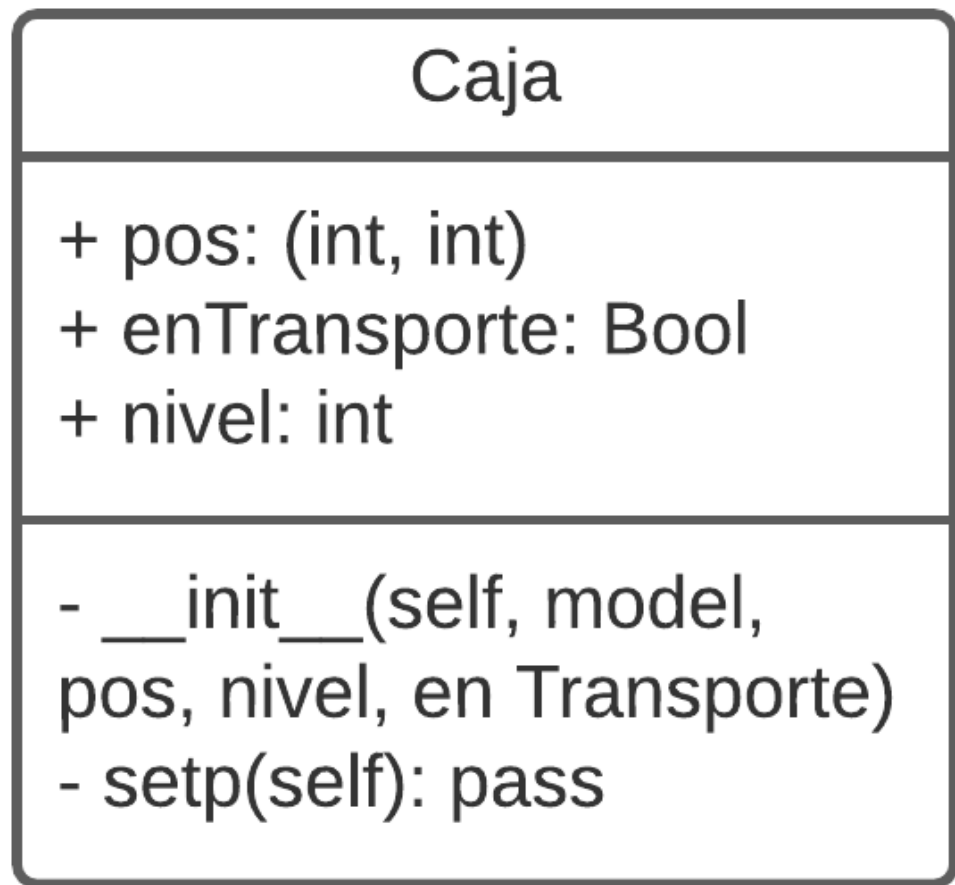
1. **Robot:** Estos agentes serán quienes estarán moviéndose en la simulación, a su vez serán los encargados de ir por las cajas para acomodarlas, por lo que tiene una lógica de programación importante.
2. **Caja:** Estos agentes son sencillos, simplemente serán los encargados de representar a las cajas, no poseen una lógica de programación compleja y no se mueven o interactúan de manera complicada.

## Diagramas de Clases

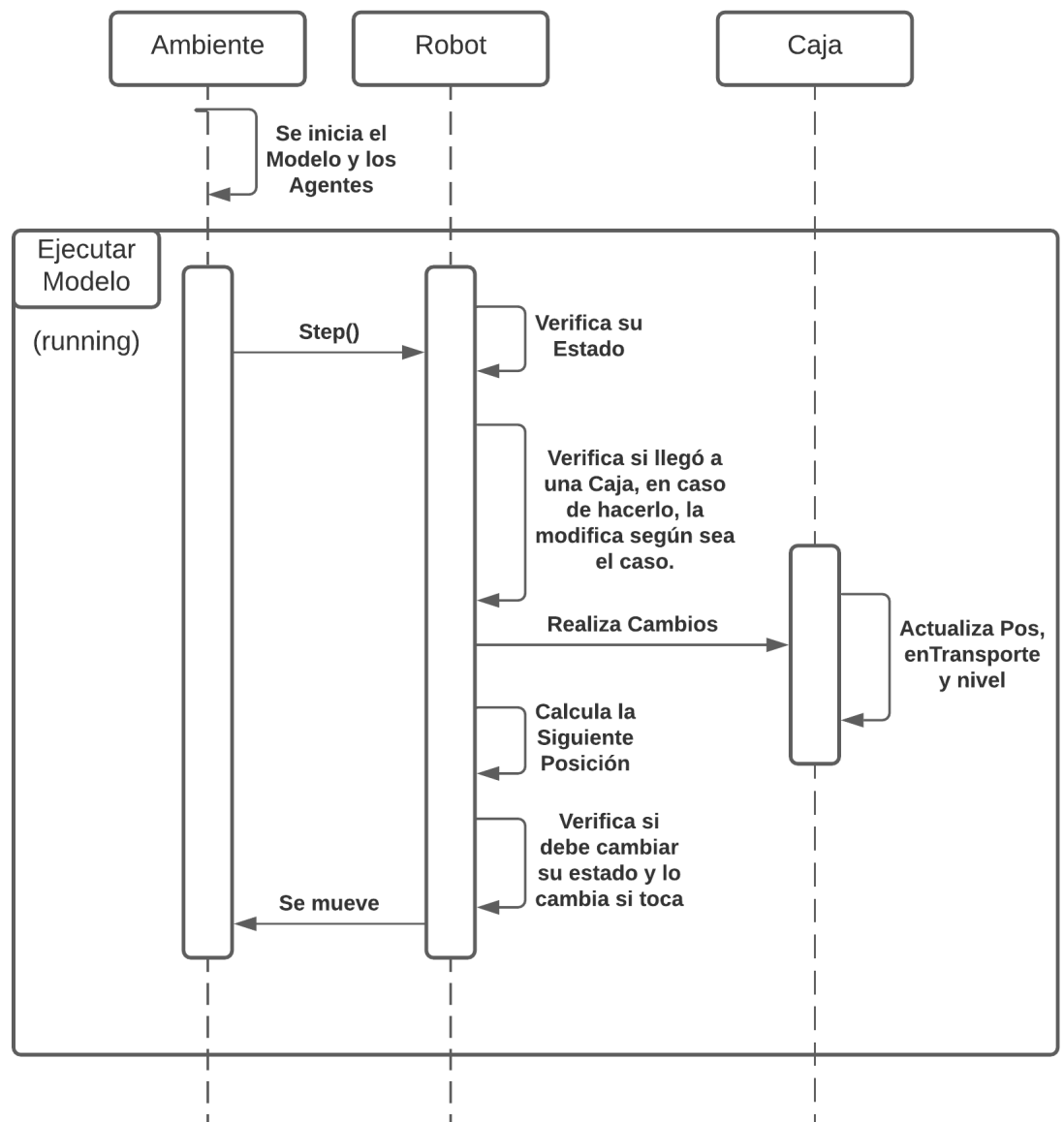
1. **Robot:**



## 2. Caja:



### 3. Diagrama de Interacción:



## **Estrategia Cooperativa para la Solución del Problema**

El algoritmo en general sigue la siguiente lógica:

Primero crea a los Robots y los acomoda de manera aleatoria, procurando no poner ninguno en  $X = 0$ , dado que ahí van a ir los cúmulos de 5 cajas.

Luego se crean las Cajas, también se acomodan de manera aleatoria, nuevamente procurando no poner ninguna caja en  $X = 0$ , dado que ahí van los grupos de cajas, también se procura que no se cree alguna caja donde ya haya un robot, finalmente se crea una lista con las posiciones de las cajas.

Ahora, se inician los pasos y lo primero que harán los robots será de partir de un estado "0" el cual indica que deben ir por una caja, se les asigna una caja de la lista con las posiciones de las cajas de manera aleatoria según el id del robot, por ejemplo, el robot con  $id = 1$ , le tocará la caja `listaCajas(0)`, por lo cual, no hay manera que 2 robots les toque la misma caja. Una vez el robot llega a la caja que debe recoger, se elimina dicha caja de la lista con las posiciones de las cajas. Llegado a este punto, los robots pasan a un estado "1" el cual indica que deben ir a depositar la caja a un grupo de cajas, siempre la primera caja que recogen la depositan en el grupo 1, una vez depositan la caja en el grupo 1, vuelven a pasar a un estado "0" para buscar otra caja y el ciclo se repite, cabe resaltar que nunca depositan las cajas en un mismo grupo, una vez depositan una caja en el grupo 1, la siguiente la depositan en el grupo 2 y así sucesivamente, como son 5 robots, los grupos de cajas siempre quedarán de 5.

Finalmente, cuando el robot detecta que ya no hay más cajas que recoger, osea, una vez que la lista con coordenadas de cajas está vacía, procede a tomar su lugar de descanso el cual es  $X = self.id$  y  $Y = 2$ , por lo que al final quedan ordenados en una fila con altura  $Y = 2$ .

**Analiza si existe una estrategia que podría disminuir el tiempo dedicado, así como la cantidad de movimientos realizados. ¿Cómo sería? Descríbela.**

Definitivamente existe una estrategia que podría disminuir el tiempo dedicado así como la cantidad de movimientos realizados, para ello se debe contemplar lo siguiente:

Dado que tanto los robots como las cajas parten de lugares aleatorios, el generar un orden en los espacios donde se agrupan las cajas (En el caso de la simulación las cajas se empiezan a apilar abajo a la izquierda), aumenta tanto el tiempo dedicado como la cantidad de movimientos realizados, por lo tanto, lo más eficiente sería que cada robot vaya directo a la caja que tengan más cerca desde su punto de partida, una vez hecho esto, deben identificar la siguiente caja más cercana, dicha caja será donde comenzarán a apilar las cajas, de esta manera se obtendría el proceso más eficiente tanto en tiempo dedicado como en la cantidad de movimientos, no obstante tiene varias desventajas, en primer lugar generaría montones de cajas en lugares aleatorios, lo cual podría ser poco deseado, por otro lado también podría generar montones de cajas de más, ya que si son pocas cajas podría haber varios montones de menos de 5 cajas.

Por otro lado, ignorando donde se van a poner los montones de cajas, otra manera de efficientar los tiempos y movimientos es que los robots una vez que depositen una caja en un montón, busquen la siguiente caja más cercana a ellos, y que esa caja sea la que vayan a recoger y volver a dejar en el montón, en la simulación actual a los robots se les asignan cajas según su orden de ID por lo que no es eficiente en términos de distancia, ya que o bien les puede tocar ir por una caja cercana a ellos como ir por una lejana a ellos.

También, desde el punto de vista de los recorridos, se podría implementar un algoritmo que prevea si dos robots se van a encontrar, para modificar previamente sus rutas, de tal manera que se envíe que se encuentren y uno se tenga que detener, no obstante esto agrega mucha lógica de programación para un evento que es poco probable que suceda, tal vez el aumento en el costo del algoritmo no lo valga.

Asimismo se puede revisar si un robot se dirige a un montón de cajas que se llenará antes de que él llegue a depositar su caja, para que no tenga que perder tiempo en moverse a otro montón sino que previamente se calcula y se redirecciona otro montón aunque el más próximo no esté lleno (Pero lo estará antes de que llegue).

**Link del Repositorio de GitHub:**

**<https://github.com/JorgeTato99/actividad-integradora-TC2008B.1>**