



**Tecnológico  
de Monterrey**

**TC2008B.523**

**Modelación de sistemas multiagentes con  
gráficas computacionales**

**Actividad Integradora**

**A01658969**

**Alejandro Hernández Ramos**

## **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 cuántas 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.

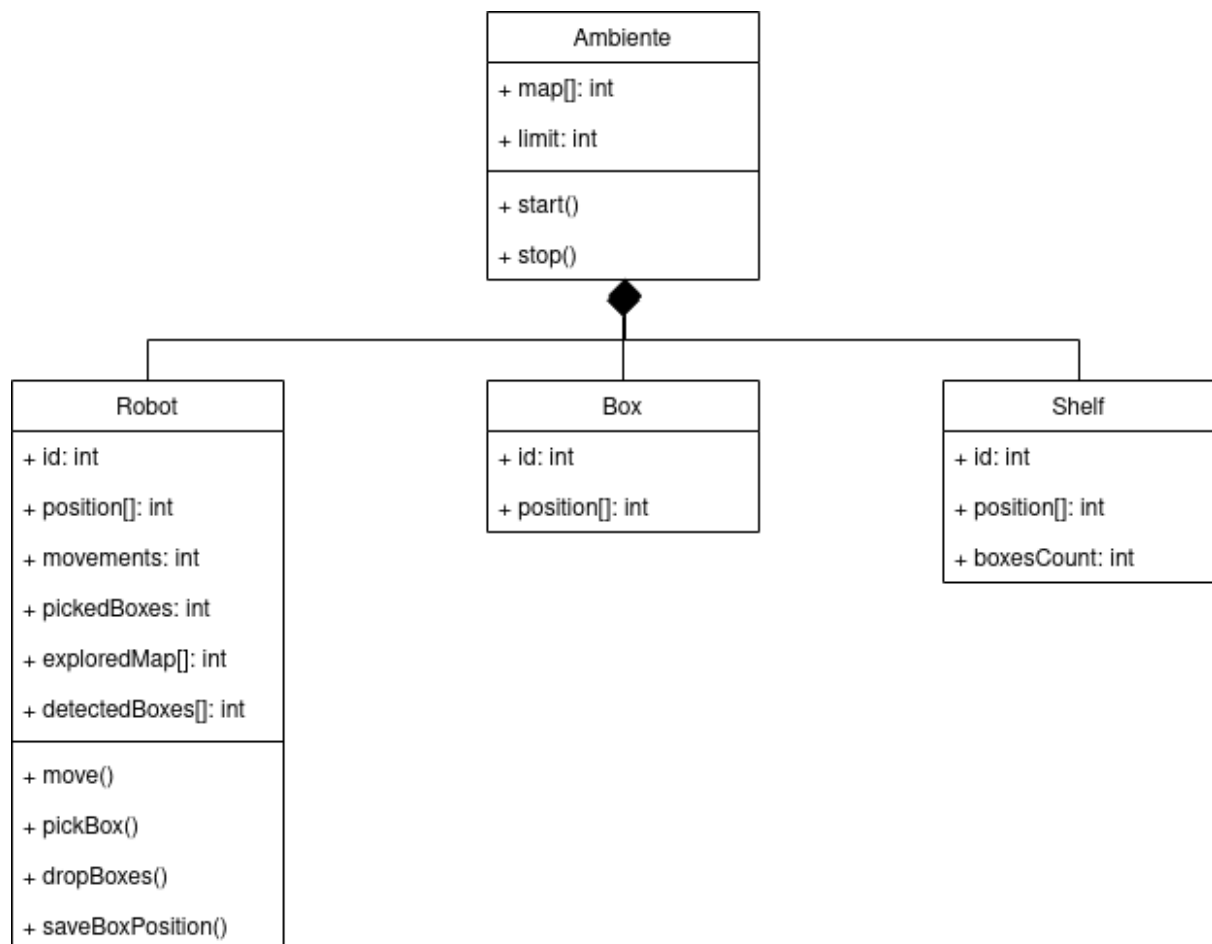
Lamentablemente, tu presupuesto resultó insuficiente para adquirir un software de gestión de agentes múltiples de última generación. Pero eso no debería ser un gran problema ... ¿verdad? Tu tarea es enseñar a sus robots cómo ordenar su almacén. La organización de los agentes depende de ti, siempre que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

## **Sistemas multiagentes**

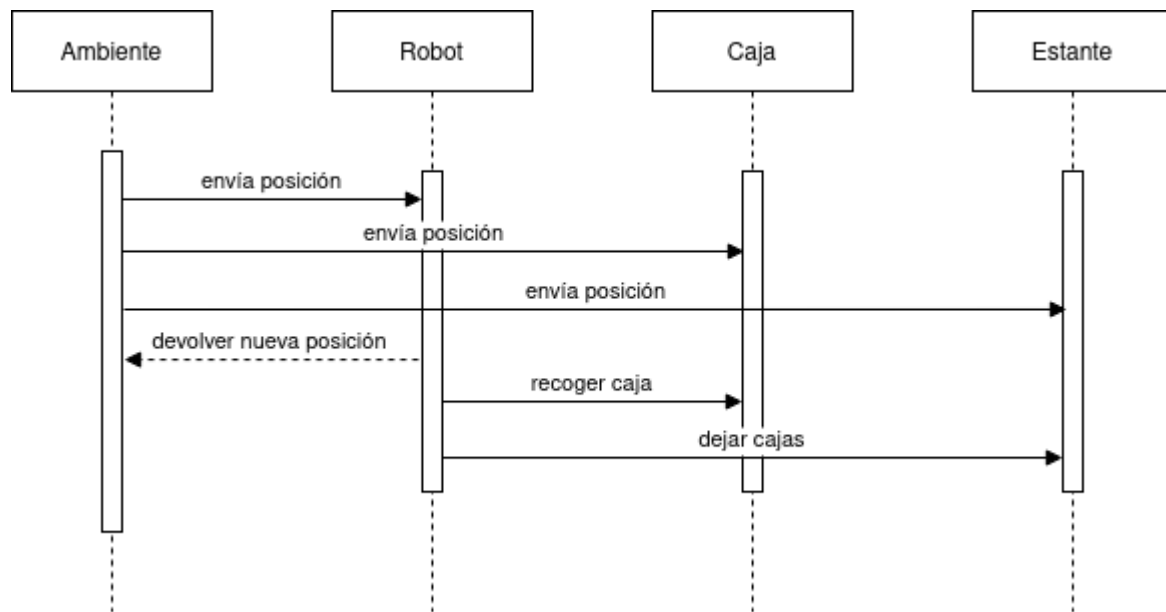
Para solucionar este problema mi propuesta es crear un mapa del almacén con una matriz 2x2 donde se señalen los puntos donde se encuentran los estantes, las cajas y los robots serán inicializados en posiciones aleatorias. Cada robot almacena la cantidad de movimientos realizados. Teniendo en cuenta que el robot puede detectar lo que se encuentra en las celdas adyacentes, en caso de detectar una caja deberá desplazarse a esa posición para recogerla. Para reducir el tiempo, cada robot contará

con un registro de las celdas recorridas para evitar pasar nuevamente por un camino ya explorado y así tener oportunidad de encontrar más cajas. Ya que pueden detectar las 4 celdas adyacentes, también para reducir el tiempo, sería una buena opción que hicieran un barrido ya sea por filas o columnas; por ejemplo, si iniciamos un tablero de 3xN, tendríamos todos los elementos del tablero detectados recorriendo sólo la columna del centro. Una vez que se tengan las 5 cajas apiladas podemos utilizar Dijkstra o BFS para encontrar el camino más corto al estante más cercano. En caso de localizar una caja en camino al estante, este podrá almacenar la posición para posteriormente recogerla.

## Diagrama de clase



## Protocolos de agentes



Como podemos observar en el diagrama, los robots no se comunican entre sí, si no que la información de la posición de cada uno es enviada al ambiente y esta a los robots. Los robots pueden recoger y dejar cajas en los estantes, sin embargo, no se espera que estos tengan una interacción de regreso.

## Gráficas computacionales

Assets utilizados:

<https://assetstore.unity.com/packages/3d/characters/robots/robot-jr-1-animated-mod1-mod2-182628>

<https://assetstore.unity.com/packages/3d/environments/sci-fi/sci-fi-construction-kit-modular-159280>

<https://assetstore.unity.com/packages/3d/props/industrial/cardboard-boxes-pack-hd-202666>

