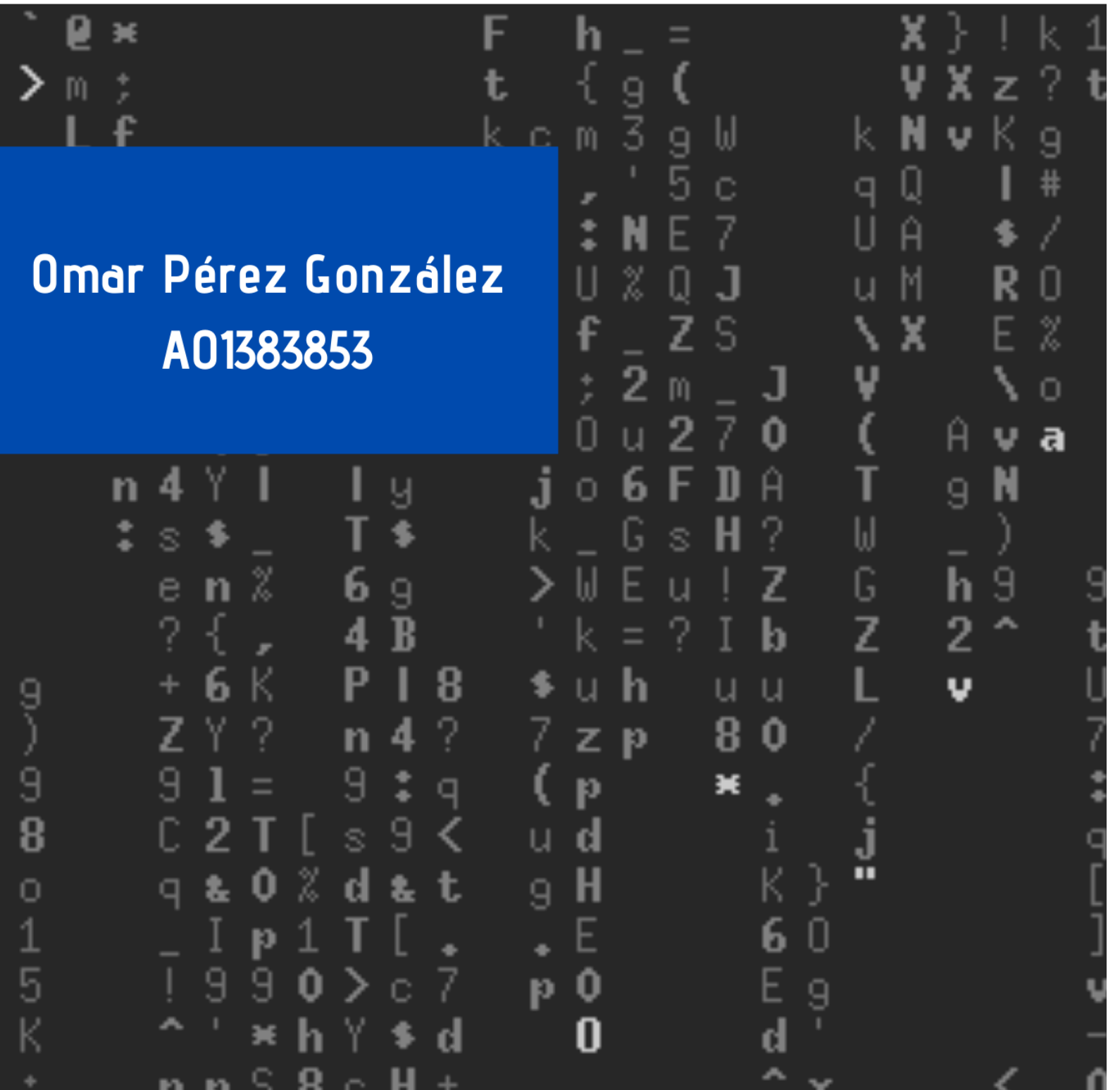
ACTIVIDAD INTEGRADORA

PARTE 1

Tecnológico de Monterrey

Omar Pérez González
A01383853



Introducción

La inteligencia artificial se refiere a los sistemas o máquinas que imitan la inteligencia humana para realizar tareas y que tienen capacidad de mejorar iterativamente a partir de la información que recopilan, con esta inteligencia, los sistemas pueden comprender más rápido los problemas, analizar información proveniente de enormes conjuntos de datos, así como encontrar patrones en problemas y así proveer la mejor solución a este mismo.
Con esta actividad integradora se planea simular la organización de una habitación de dimensiones MxN en la cual se encuentran cajas dispersas por esta y 5 robots deben organizar las cajas en pilas de no más de 5 cajas para así obtener como resultado una habitación ordenada.

Las restricciones de esta simulación son que cada robot está equipado con ruedas omnidireccionales, por lo que puede conducir en cuatro direcciones (adelante, atrás, izquierda y derecha), estos pueden recoger cajas en celdas de cuadrículas adyacentes con sus manipuladores para después llevarlas a alguna de las pilas, estas pilas se pueden crear en caso de que no exista alguna o se encuentre llena (hay 5 cajas en esta misma). Para estos robots es fácil distinguir si tienen una pared, una pila, y otro robot enfrente, estos llevan también sensores que indican si llevan una caja en ese momento.

Con esto establecido es posible crear un sistema de gestión de agentes múltiples, por lo que la simulación debe contar con:

- Inicializar las posiciones de las K cajas, todas las cajas están a nivel de piso, no hay pilas de cajas en el inicio.
- Todos los agentes empiezan en posiciones aleatorias vacías.
- Se ejecuta al tiempo máximo establecido.

Durante la simulación se debe recopilar:

- Tiempo necesario hasta que todas las cajas estén en pilas de máximo 5 cajas.
- Número de movimientos realizados por los robots.

El código a la solución de esta actividad integradora se subirá a un repositorio en GitHub, mientras que este documento también se subirá a la misma plataforma, pero contendrá los distintos diagramas hechos para la solución del problema, este documento también contendrá posibles estrategias de mejora para la solución del problema, así como las conclusiones de esta actividad integradora.

Diseño del modelo

Para el diseño del modelo se tuvo en cuenta todo lo mencionado anteriormente, siendo las restricciones una parte importante a la hora de crear un modelo, donde estas sirven para establecer lo que se puede y no se puede hacer en la simulación, con esto establecido podemos hablar de los agentes, así como las características que estos deben tener, estos son:

- **Robot:**
  Este agente será el encargado de recorrer la habitación e ir escaneando en busca de cajas y pilas en las cuales deben colocar las cajas. Las características y restricciones que posee este agente son:
  - El robot se puede mover en cuatro direcciones (adelante, atrás, izquierda, derecha).
  - Se puede mover por toda la habitación mientras no exista una pila llena u otro robot en cualquiera de sus direcciones.
  - Al agarrar una caja el agente cambiará a un color más claro (representando el sensor de presión que tienen).
  - Al tener una caja el robot no podrá pasar por encima de otras cajas.
  - Al no tener una caja el robot no podrá pasar por encima de pilas.
  - El movimiento del robot se elige aleatoriamente de una lista que fue construida en base a los requerimientos necesarios.

  Este agente deberá cooperar con los otros para poder tener un correcto funcionamiento siendo que no se necesita un ambiente competitivo entre estos.
  Los estados con los que cuentan el Robot son:
  - CAJA_ENCIMA: Cuando el Robot tiene una caja encima.
  - SIN_CAJA: Cuando el Robot no tiene una caja encima.

- **Piso:**
  Este agente se encarga de cambiar de estado en base a los movimientos del agente Robot, los estados y sus características son:
  - NADA: En este estado es cuando no se encuentra una caja, pila o pila llena en el piso, por aquí puede moverse el Robot sin problema.
  - CAJA: Existe una caja en el piso, el Robot puede tomar la caja si no lleva cargando ya una caja.
  - PILA: Una pila de cajas menor a cinco, donde el Robot podrá colocar cajas y no podrá pasar por ahí si no tiene cajas cargando.
  - PILA_LLENA: En este estado la pila tiene cinco cajas apiladas.

En el modelo creado es posible cambiar las dimensiones de la habitación, así como el tiempo de ejecución máximo al que quisiéramos someter la simulación, al final se

debe poder visualizar la cantidad de pilas y cajas (si quedaron) que se pudieron crear y ordenar respectivamente.
En los siguientes diagramas se puede observar tanto el de protocolos como el de clases, en el primero de estos es posible ver el funcionamiento de la lógica del agente, mientras que en el otro es posible ver las variables con las que cuentan, así como las funciones de las mismas. Todos estos diagramas representan gráficamente el funcionamiento del modelo, los diagramas son:

Diagrama de clases

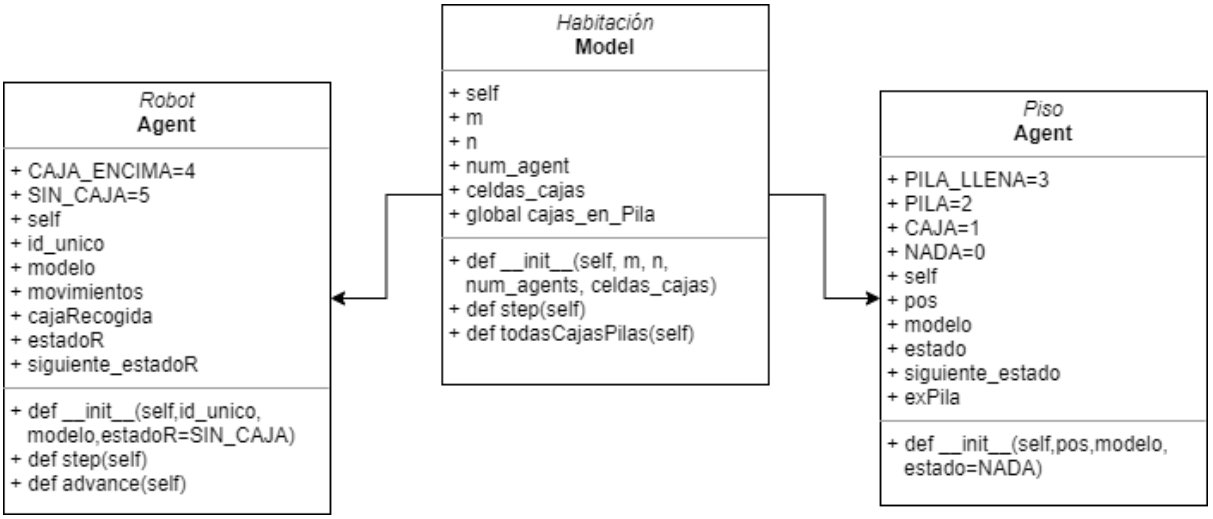


Diagrama de protocolos del Piso

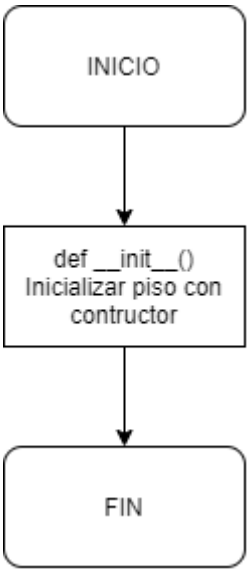


Diagrama de protocolos del Modelo

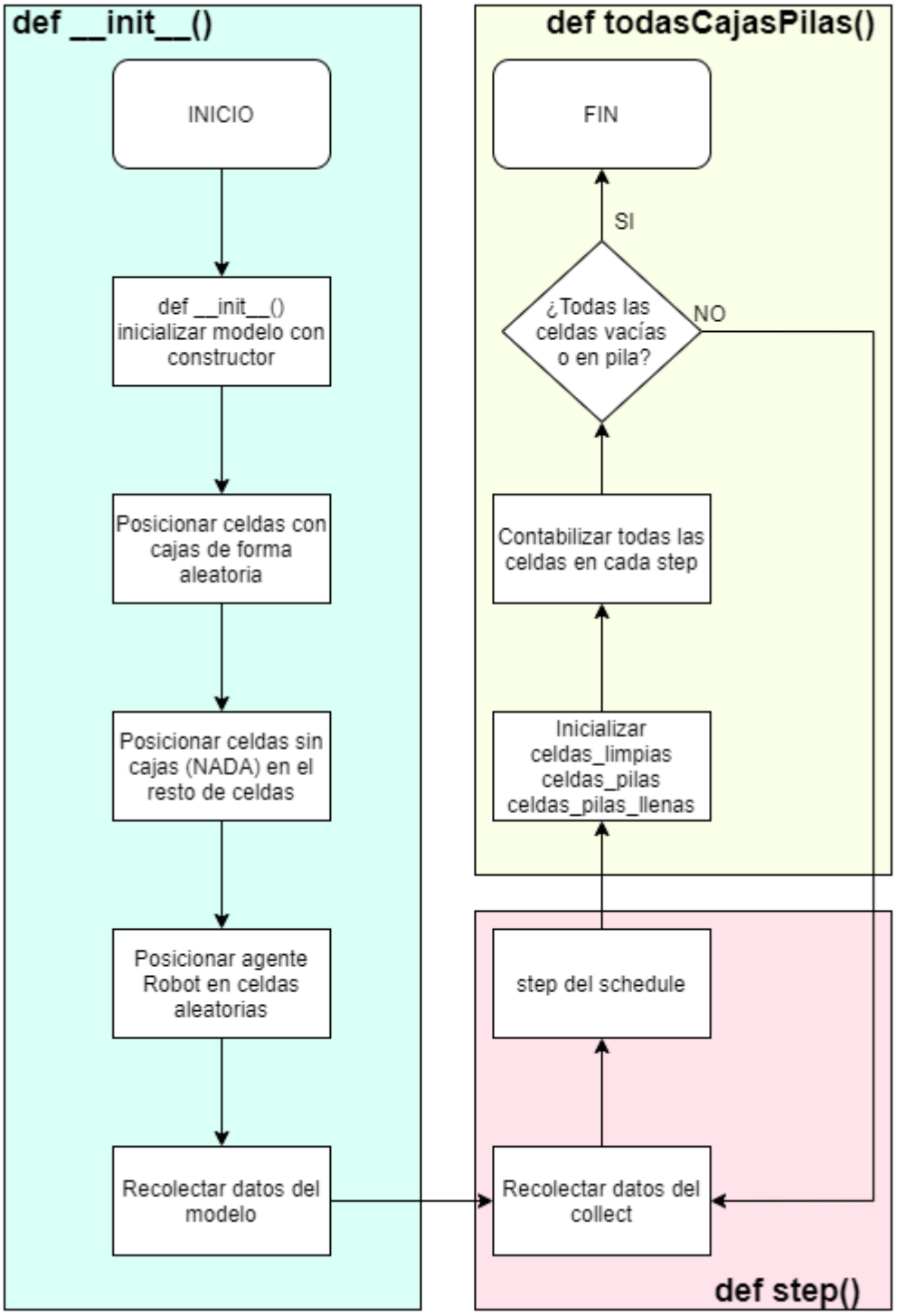
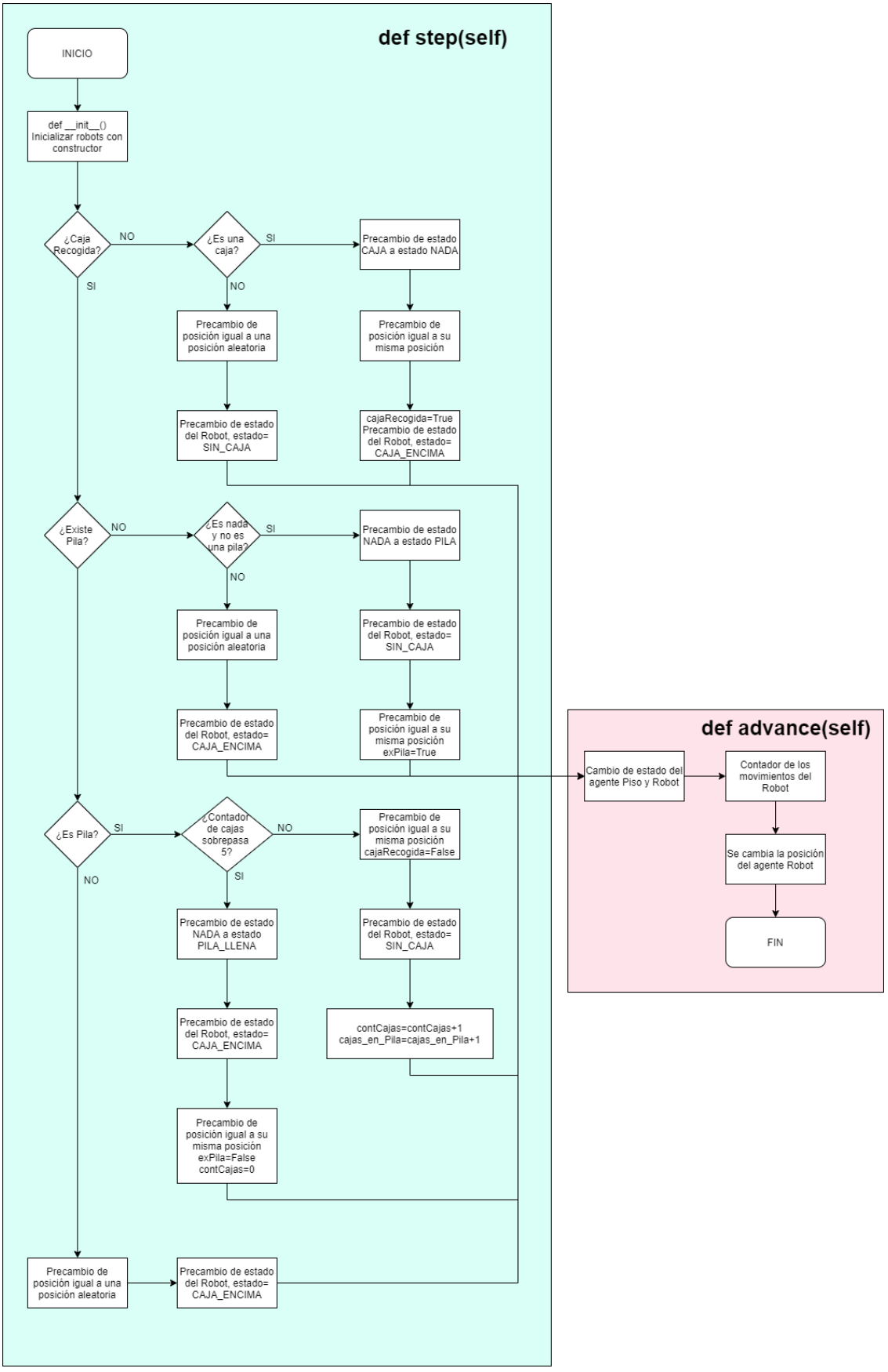


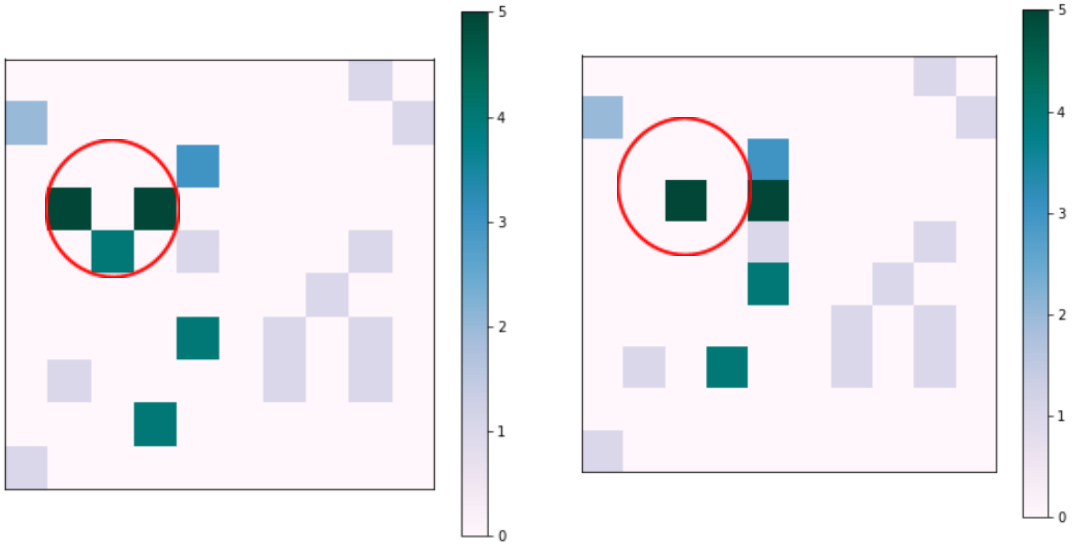
Diagrama de protocolos Robot



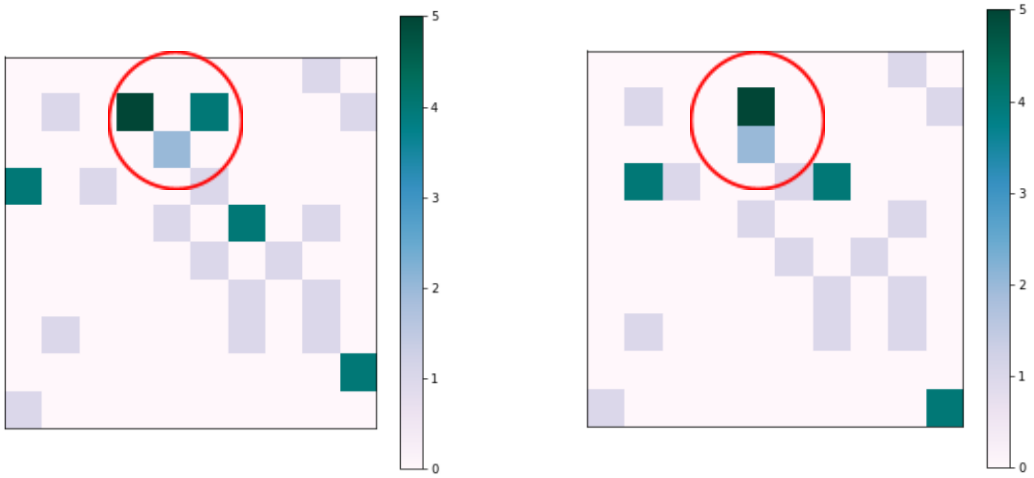
Otras estrategias de solución

Una de las principales problemáticas a la hora de la simulación fue que los Robots se empalman en momentos específicos de esta, a la hora de hacer pruebas es posible observar estos comportamientos:

- Cuando dos Robots se encuentran en diagonal y se mueven hacia el mismo cuadrado gracias a la elección aleatoria de casilla, gráficamente este comportamiento se vería como:



- Cuando los Robots se encuentran con una celda más de separación y se mueven a la misma celda, gráficamente sería:



El problema de los casos de prueba puede servir para desarrollar la estrategia para que los Robots sean más inteligentes y no elijan su movimiento en base a la aleatoriedad, con esto puede ser posible la creación de un algoritmo donde se tienen estas opciones:

- Guardar la posición de la pila actual para que los Robots la busquen y sus movimientos no sean aleatorios, donde estos irán con propósito al lugar establecido en la variable de posición pila, siempre teniendo en cuenta que no deben pasar por encima de otras cajas, pilas llenas y Robots.

- Tener en cuenta la cantidad de cajas que existen y así crear distintas pilas en partes de la habitación, donde el Robot deberá elegir la pila más cercana a él y dejar la caja en esa posición, si la pila cercana ya está llena deberá ir a buscar la pila lejana.

Esta nueva estrategia es una recomendación para futuras implementaciones, donde por tiempo y conocimientos no fue posible implementarlo en la solución, pero aún así se tomaron en cuenta posibles mejoras para esta actividad integradora.

Conclusiones

En la primera y última versión del programa hubo varias diferencias a la hora de evaluar distintos casos de prueba, donde en la primer versión, en la cual los Robots podían pasar por encima de las cajas, pilas, pilas llenas y Robots el tiempo de ejecución fue un 50% menos que en la versión del programa donde los Robots tenían las restricciones del movimiento y no podían pasar por encima de las cajas y demás objetos, esta comparación es irrelevante pero necesaria para analizar el cómo un cambio en el algoritmo puede cambiar toda la simulación y sus tiempos de ejecución.

Hablando de los resultados de una de las simulaciones donde los parámetros iniciales fueron:

- M=10
- N=10
- NUM_AGENTES=5
- CELDAS_CAJAS=23
- TIEMPO_MAXIMO_EJECUCION=2

Con estos parámetros se obtuvo un tiempo máximo de ejecución de 00:00:00.464473 siendo este mucho menor que el establecido de 2, por lo que el algoritmo de movimiento aleatorio de los robots puede ser útil, pero no eficiente a la hora de solucionar el problema, por lo que un mejor algoritmo en el movimiento de los Robots puede ayudar a decrementar el tiempo de ejecución hasta en un 30% (aproximado), la cantidad de movimientos que se obtuvieron con los parámetros anteriores fueron de 4145 siendo una gran cantidad de movimientos, pero a la vez una gran cantidad de tiempo y agentes que intervienen entre sí, por lo que el tiempo y cantidad de movimientos si “hacen sentido”.

La actividad integradora parte uno me ha ayudado a ser un mejor programador, donde al inicio del curso no entendía nada de Mesa Python pero con el tiempo fui entendiéndolo hasta el punto de poder concluir con esta actividad satisfactoriamente y orgulloso del progreso que logré al terminarla (siendo que la primera actividad ni siquiera funcionó), por lo que esto significa un gran paso tanto conceptual como emocional para mí, donde el culminar esto me levanta el ánimo y me hace creer que entendí los temas del curso y entendí el funcionamiento de esta biblioteca de Python, esta actividad solo es una “demostración” de lo que se puede venir para la situación problema que crearemos, donde esto será más complicado y necesitaremos de algoritmos mucho más complejos que nos ayuden con el modelo de la intersección vehicular.

Bibliografía

- Covantes, E. (s.f.) 1. Agentes inteligentes [Diapositivas de Powerpoint]. Departamento de Ciencias Computacionales, Instituto Tecnológico y de Estudios Superiores de Monterrey.

- Oracle (s.f.) ¿Qué es la inteligencia artificial (IA)?. Obtenido de: https://www.oracle.com/mx/artificial-intelligence/what-is-ai/

- Project Mesa (2016) Mesa: Agent-based modeling in Python 3+. Obtenido de: https://mesa.readthedocs.io/en/master/index.html#

- Project Mesa (2016) mesa package. Obtenido de: https://mesa.readthedocs.io/en/master/mesa.html