



Documentación Actividad Integradora 1

Jesús Sebastián Jaime Oviedo A01412442

Sergio Hiroshi Carrera Monnier A01197964

Juan Daniel Rodríguez Oropeza A01411625

Jackeline Conant Rubalcava A01280544

William Frank Monroy Mamani A00829796

Premysl Pilar A01760915

Monterrey, Nuevo León, México

Diciembre 02, 2022

Ing. César Raúl García Jacas

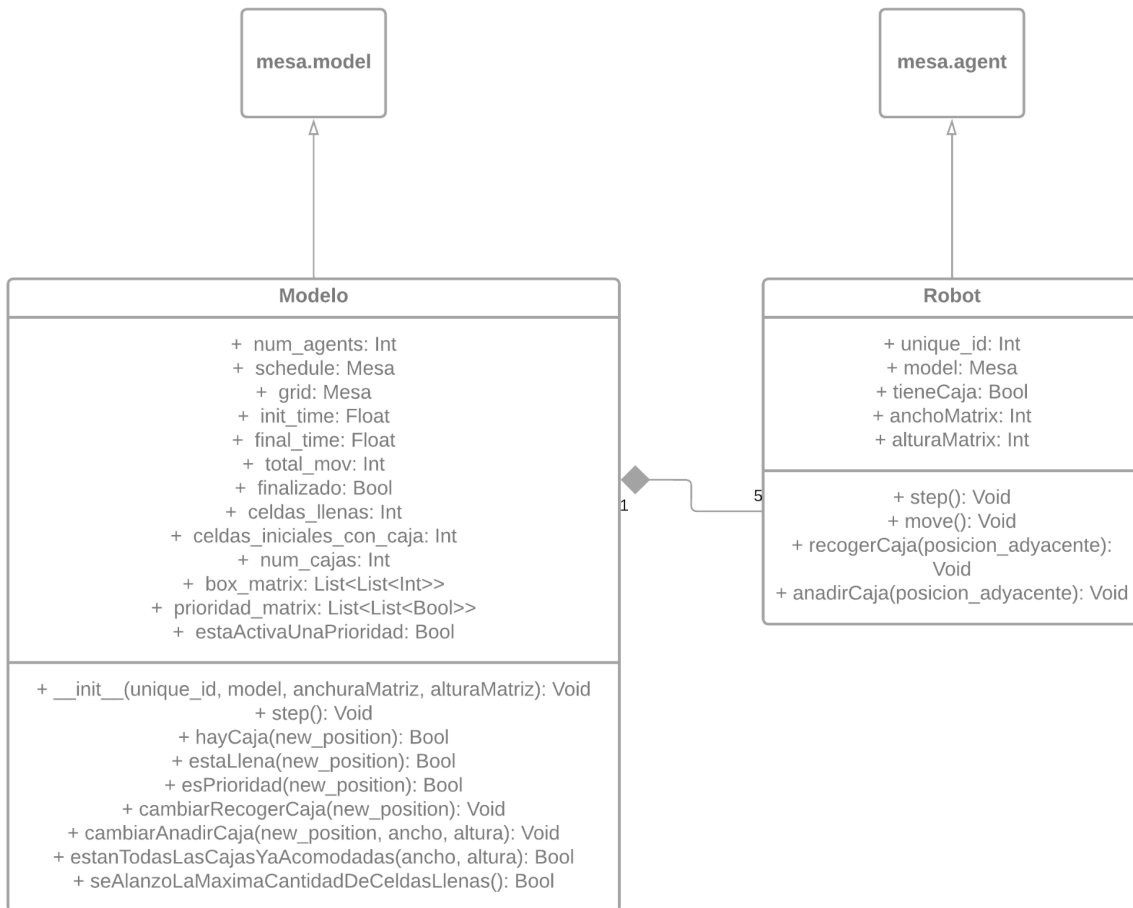
Ing. Sebastián Ulises Adán Saldívar

Instituto Tecnológico y de Estudios Superiores de Monterrey

**Modelación de Sistemas Multiagentes con Gráficas Computacionales
(Gpo 301)**

Partes 1 - Mesa Python

Diagrama de Clases

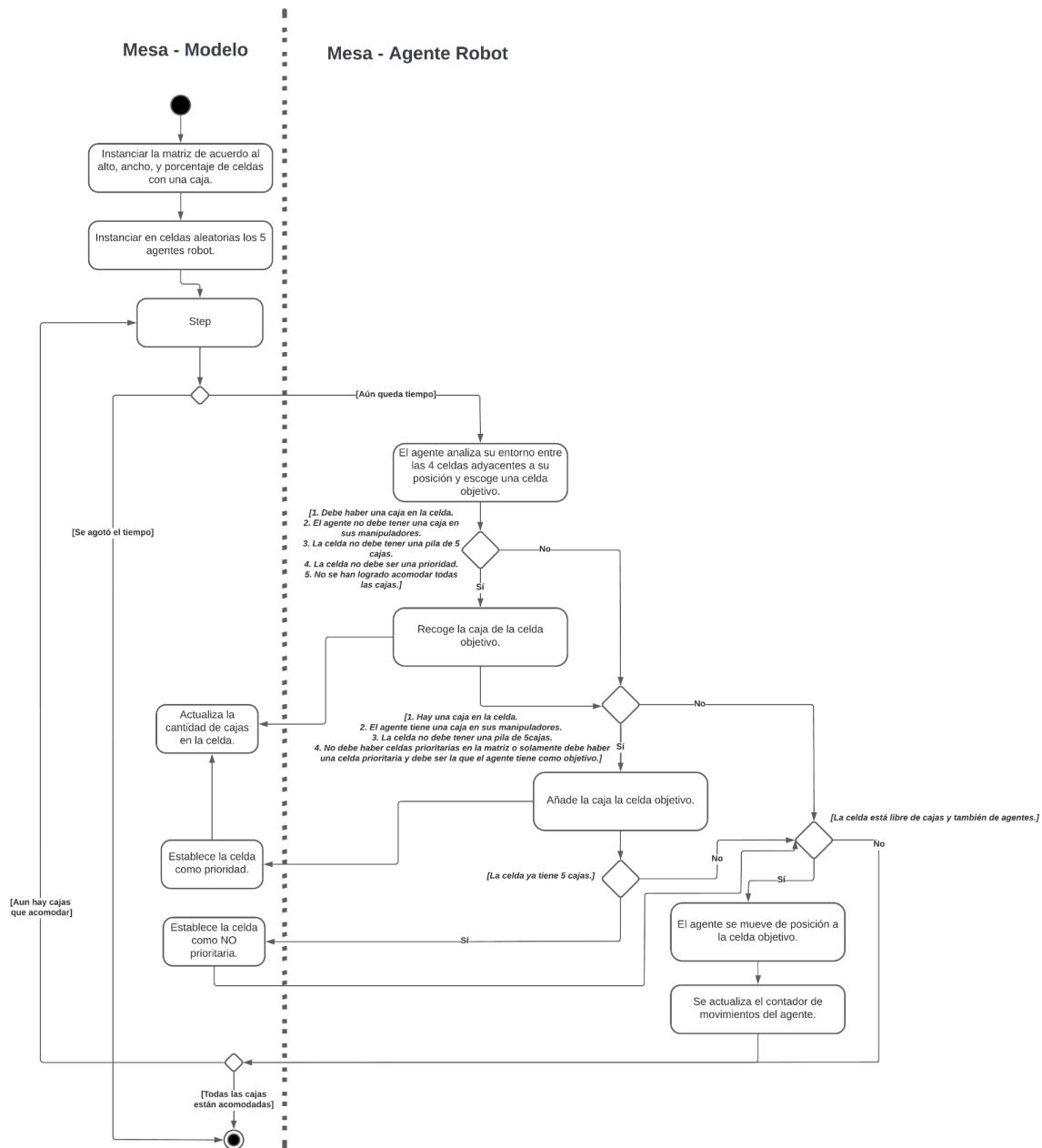


Los agentes se comunican entre ellos y evalúan condicionales como hayCaja(), estaLlena(), esPrioridad() y tieneCaja(), analizando su entorno y tomando decisiones respecto a éstas.

El modelo principalmente aplica cambios de valor a las celdas para indicar la cantidad de cajas y establecer cuál es prioridad, en base con las posiciones que recibe de las llamadas de los agentes a los métodos de su clase (la del modelo). A su vez, también hace comprobaciones y evaluaciones a nivel global de la matriz.

En resumen, 5 robots tienen un espacio para cargar una caja y apilar las cajas desperdigadas en un almacén, para armar torres de 5 cajas.

Protocolo de Agentes (Diagrama de Actividades)



Estrategia Cooperativa

Los agentes al trabajar en conjunto para apilar las cajas, cada quien atento a sus 4 celdas aledañas, por los 5 agentes, permite que en cada movimiento 20 espacios sean observados si tienen caja o no.

Luego de que un agente recogiese una caja y después la añada en una celda, a esta última se le asigna como prioridad para que todo los agentes trabajen en hacer

es pila, una vez están las 5 cajas en la celda, ésta ya no es considerada como celda prioritaria, y como está llena, ningún otro agente puede recoger cajas de esa celda. Solamente puede haber una celda prioritaria a la vez en la matriz durante la simulación.

Posteriormente, un agente añade una caja en otra celda, y se vuelve a repetir el proceso hasta que todas las cajas ya estén acomodadas. El agente solamente agregará una caja a las celdas que ya cuenten con al menos una caja y menos de 5. Es mejor trabajar con las celdas ya “empezadas”, es decir, las que tienen al menos una caja, que empezar desde cero en una celda vacía.

Esto hace que en grupo solucionen el problema de la recolección y apilado de las cajas del almacén de forma ordenada.

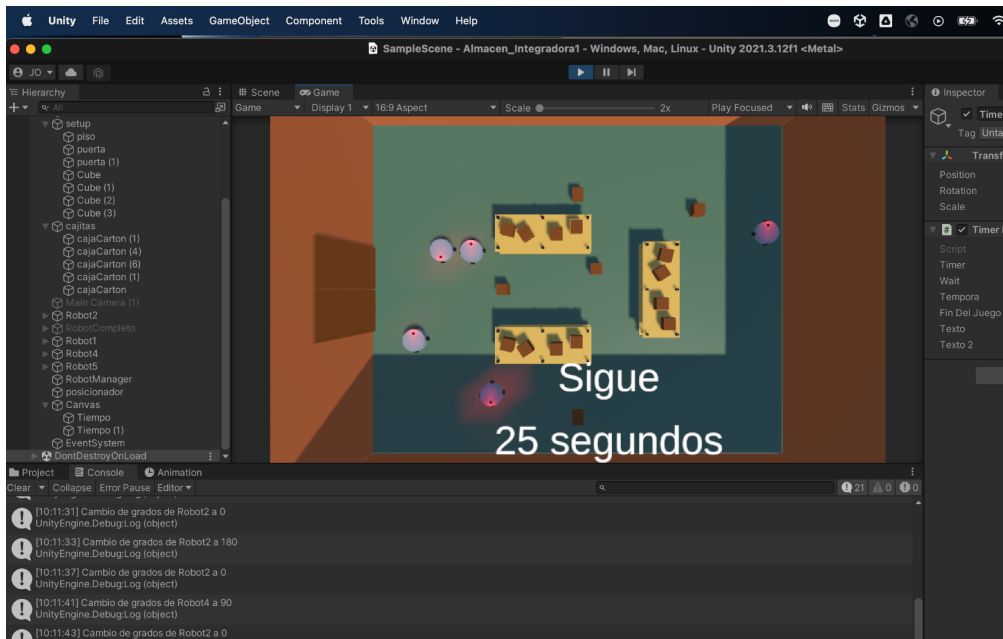
Analiza si existe una estrategia que podría disminuir el tiempo dedicado, así como la cantidad de movimientos realizados. ¿Cómo sería? Descríbela.

Un ajuste que se podría realizar para agilizar el proceso, es que pueda haber más de una celda prioridad a la vez para poder llenar más celdas con pilas de 5 cajas de forma simultánea, lo que conlleva que los agentes ya no anden dando vueltas sin poder añadir cajas más que a un celda en específico.

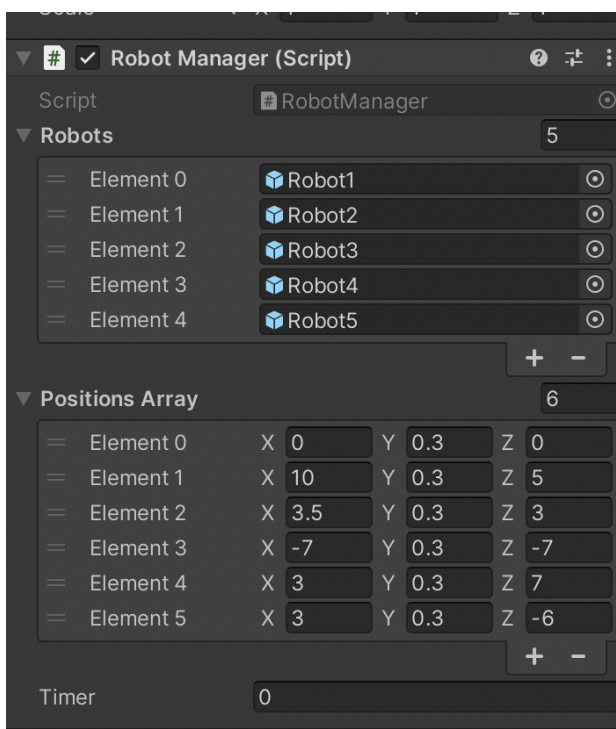
También se podría añadir que se establezca que los agentes solamente añadan cajas a las celdas que estén en el contorno u orilla de la matriz, de esta forma sería más fácil y ordenado visualizar los datos, además, de que los agentes tendrían un destino predeterminado al cual ir cuando tienen una caja, en lugar de vagar por la matriz a ver si se encuentran una celda prioritaria para agregar una caja.

Parte 2 - Unity

Por medio de un contador como el anterior, se ejemplifica gráficamente cómo serían nuestros robots en el almacén.



Estos espacios están predeterminados, pero su elección es aleatoria.



Para el tiempo máximo, cuando termina de limpiarse todas las cajas, se quita el sigue del contador y reporta el **tiempo final para recoger** las 12 cajas planteadas aquí.