



Tecnológico de Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Santa Fe

Modelación de sistemas multiagentes con gráficas Computacionales (Gpo 301)

Nombre del profesores:

Octavio Navarro Hinojosa

Actividad:

MA. Actividad: Roomba

Alumno:

Nicolas Quintana Kluchnik |

A01785655

Fecha:

22 de octubre del 2025

Problema que se está Resolviendo:

El problema consiste en desarrollar un sistema de agentes autónomos capaces de limpiar de forma eficaz una habitación con distribución aleatoria de suciedad, obstáculos, y estaciones de carga.

Propuesta:

Se propone implementar un sistema basado en agentes reactivos con capacidades de planificación utilizando el framework Mesa para simulación multiagente. El sistema utiliza un enfoque de priorización de acciones que permite al Roomba tomar decisiones inteligentes sobre su próximo movimiento.

Diseño de los Agentes:

Objetivo: El objetivo principal de los agentes roombas es de limpiar la mayor suciedad posible mientras mantienen la energía suficiente para continuar operando.

Capacidad efectora:

Los agentes roomba tienen las siguientes capacidades de acción:

Acción	Descripción	Efecto
Movimiento	Desplazarse a una celda adyacente de forma vertical, horizontal o diagonal	Consumir una energía por desplazamiento
Limpieza	Remover basura de la celda actual si contiene	Incrementa el contador de basura recolectada
Carga	Cargar batería en la estación asignada	Regenera 5 de energía por paso de la simulación
Detención	Detenerse/Morir cuando la barra de energía llega a 0	El agente se muere y desaparece de la simulación

Percepción:

Elemento Percibido	Como lo percibe	Información obtenida
DirtAgent	Ánálisis de celdas vecindarias	Presencia de basura en las celdas adyacente/s
Obstáculos	Ánálisis de celdas vecindarias	Ubicación de obstáculos y/o bordes en cual no se puede mover cuya celda ocupada
Self.energy (propia energía)	Sensor interno	Nivel actual de batería
Estación de carga	Coordenadas de su celda inicial	Su estación de carga
Posición actual	Coordenadas de celda	Su ubicación en el entorno
Celdas vacías	Ánálisis de celdas	Las celdas disponibles para

	vecindarias	moverse
--	-------------	---------

Proactividad:

El Roomba es un agente proactivo que toma iniciativas inteligentes:

- Búsqueda activa de suciedad: Prioriza moverse hacia celdas con basura detectada
- Planificación de recarga: Anticipa la necesidad de energía y se dirige a la estación antes de agotarse (tienen un colchón en el código para que lleguen a su estación sin problema)
- Exploración: Cuando no hay suciedad cercana, explora aleatoriamente para descubrir nuevas áreas que no ha descubierto
- Decisiones autónomas: Elige su próximo movimiento sin intervención externa basado en el estado actual

Métricas de Desempeño:

Métrica	Fórmula	Desempeño
Basura Recolectada/Limpiada	Suma de trash_count del agente	Cantidad de suciedad limpiada de forma cumulativa
Energía	Valor actual de energy	Nivel actual de batería (0-100)
Movimientos totales	Suma de Movement_count	Cantidad de celdas recorridas durante la simulación
Total de celdas limpias	(Celdas limpias / Celdas totales del ambiente) * 100	Eficiencia de limpieza del ambiente

Arquitectura de los agentes:

1. Percepciones del ambiente
 - a. Toma de decisiones
 - i. Evaluar estado de energía
 - ii. Detectar suciedad en celdas vecindarias
 - iii. Calcular ruta a estación de carga si la energía llega a un cierto número
 - b. Selección de acción
 - i. Si la energía llega a la condición de ≤ 30 , va a su estación de carga
 - ii. Si detecta basura en una celda vecindaria, va a limpiarla
 - iii. Si no detecta nada, sigue explorando

Características del ambiente:

Característica	Descripción
----------------	-------------

Tipo de grid	OrthogonalMooreGrid (8 direcciones)
Tamaño	28x28 pero es personalizable el tamaño
Secuencial	Las decisiones afectan los estados futuros
Observabilidad	Parcialmente accesible ya que solo pueden observar las celdas cercanas
Determinístico	El estado siguiente completamente depende del estado actual
Dinamicidad	Entorno cambia con el tiempo

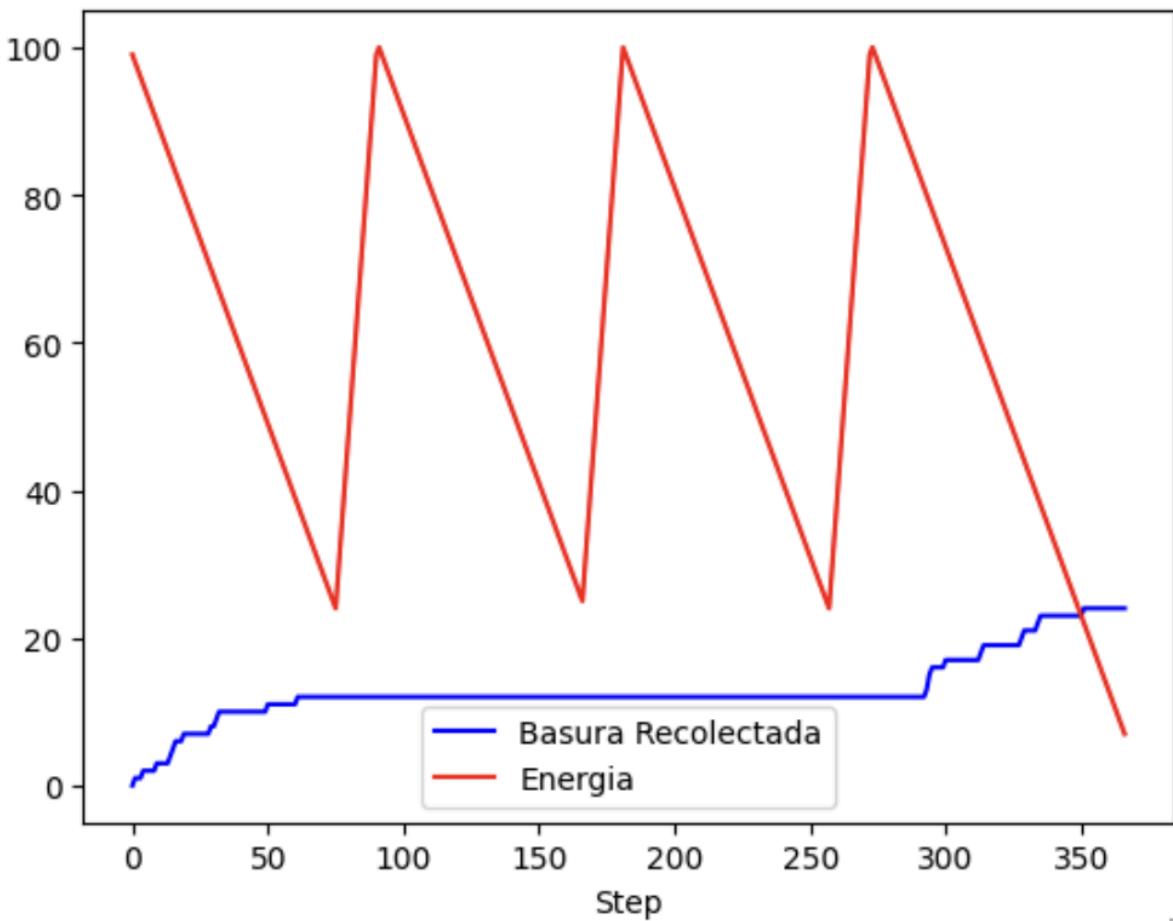
Elementos de los Agentes:

Elemento	Tipo	Cantidad	Rol
Roomba	RandomAgent	Las que el usuario dese	Agente limpiador principal
Suciedad	DirtAgent	0.15 (15% del ambiente)	Objetivo a limpiar
Obstáculos	ObstacleAgent	0.1 (10% del ambiente)	Obstáculos fijos
Estación de carga	ChargingStationAgent	La misma cantidad que los roombas	Recarga la batería de los roombas
Bordes	BorderAgent	Perímetro	Límites del ambiente

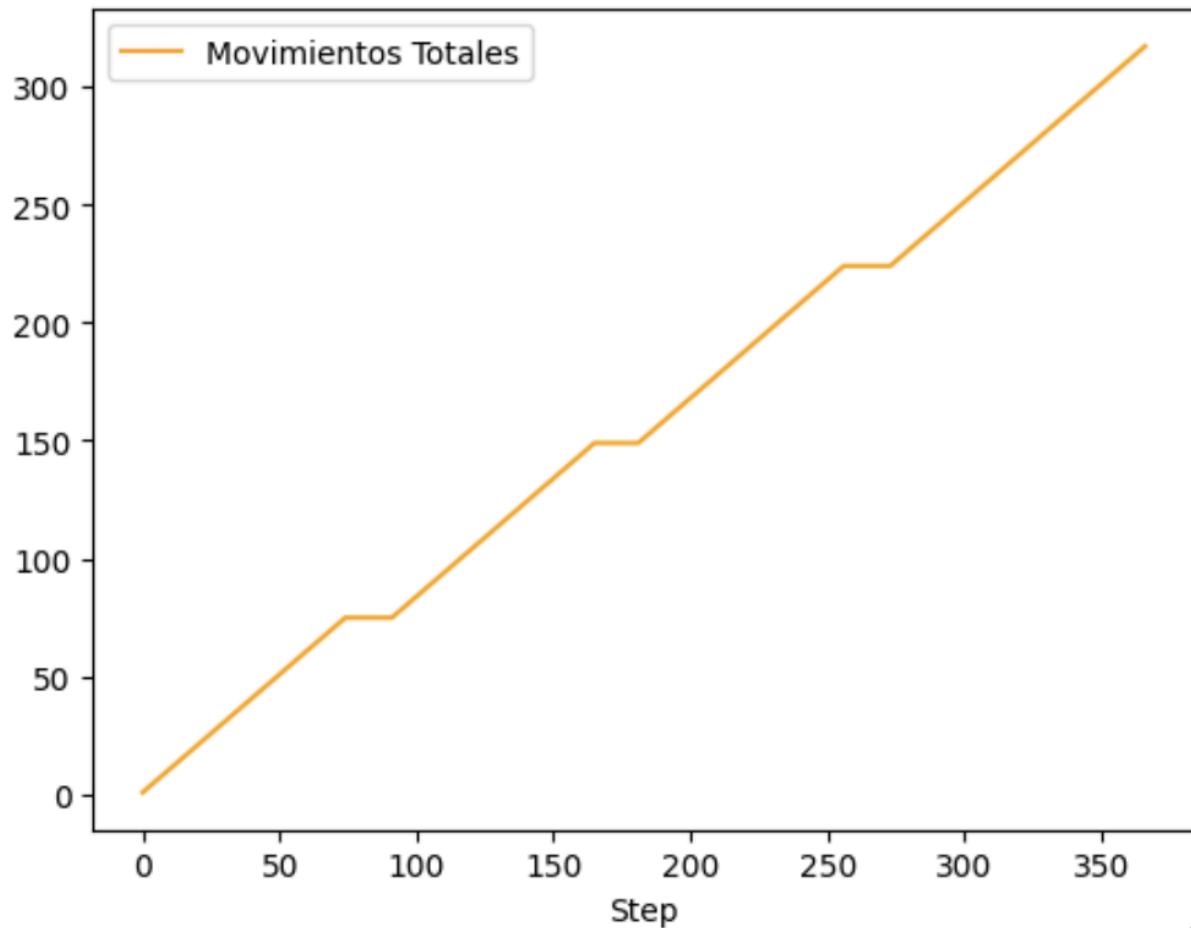
Estadísticas Recolectadas:

Simulación 1 (1 roomba):

En esta simulación se usó un solo roomba, en cual estaba preestablecido que comience en la celda 1,1 con su carga de estación en la misma celda. Lo que podemos ver en la gráfica abajo, la cual nos enseña 2 cosas, la energía del roomba y la cantidad de basura colectada. Esta información nos dice 3 cosas muy claras, uno, al principio de la simulación, el roomba limpian muy rápido la basura, esto puede ser porque hay mucha basura cerca de su estación de carga, dos, cuando el roomba carga, la batería si llega al 100, ya que no tiene un promedio que tiene que calcular, y finalmente, tres, llega un punto en cual muy poca basura es recolectada, esto nos dice que no está encontrando basura con el sistema que tiene implementado. Esto se puede mejorar a base de un cambio en el código en cual guarde su última celda antes de empezar a regresar a la estación de carga, para cuando ya tenga su batería en 100, regrese a esa celda y empiece a explorar desde ahí.

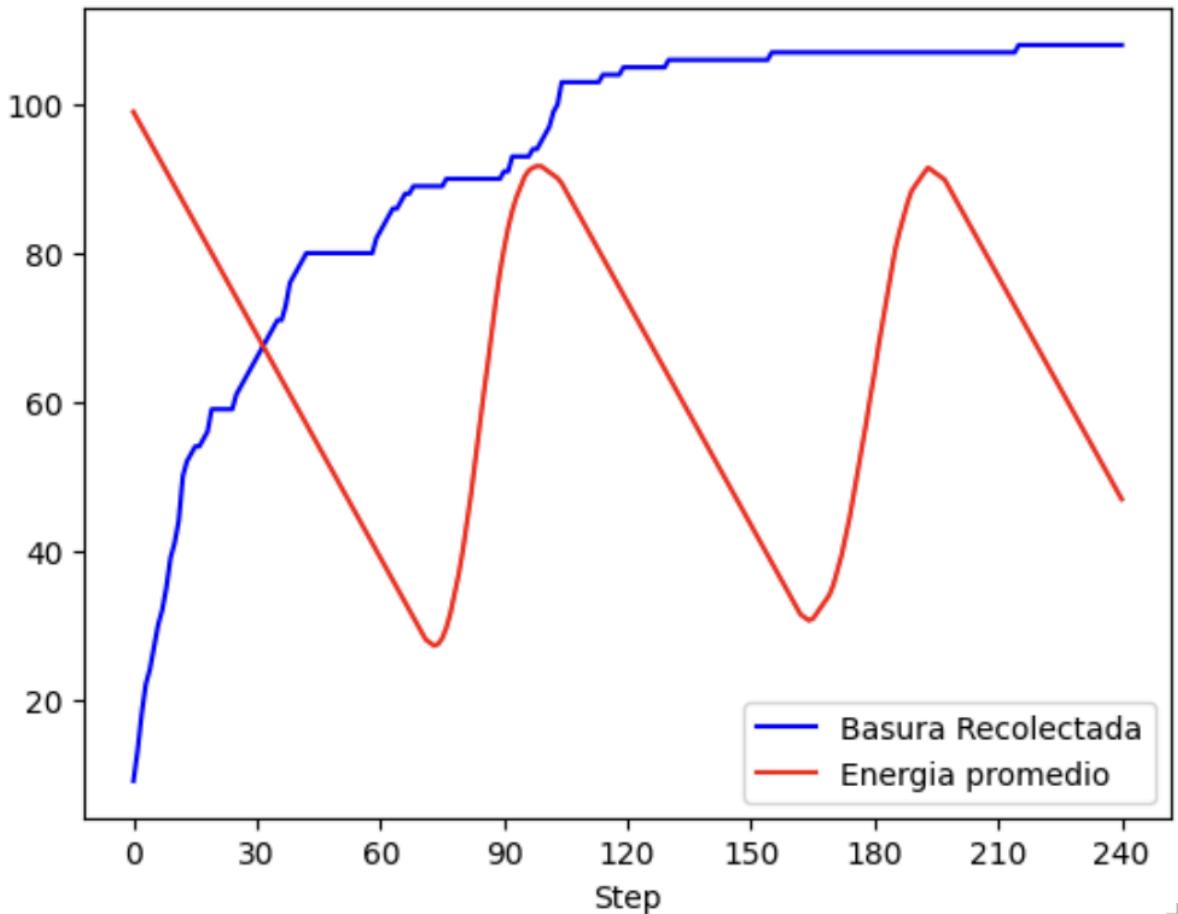


En la siguiente gráfica, podemos observar el movimiento de el roomba en cada step. Una cosa interesante es que por el rango de 80 a 90 steps, podemos ver que la cantidad de movimientos por step es completamente parada, pero hay una razón muy simple para esto. Esto pasa ya que el roomba está recargando sus pilas en sus estaciones, y como no hay movimiento, no se registra en la gráfica.

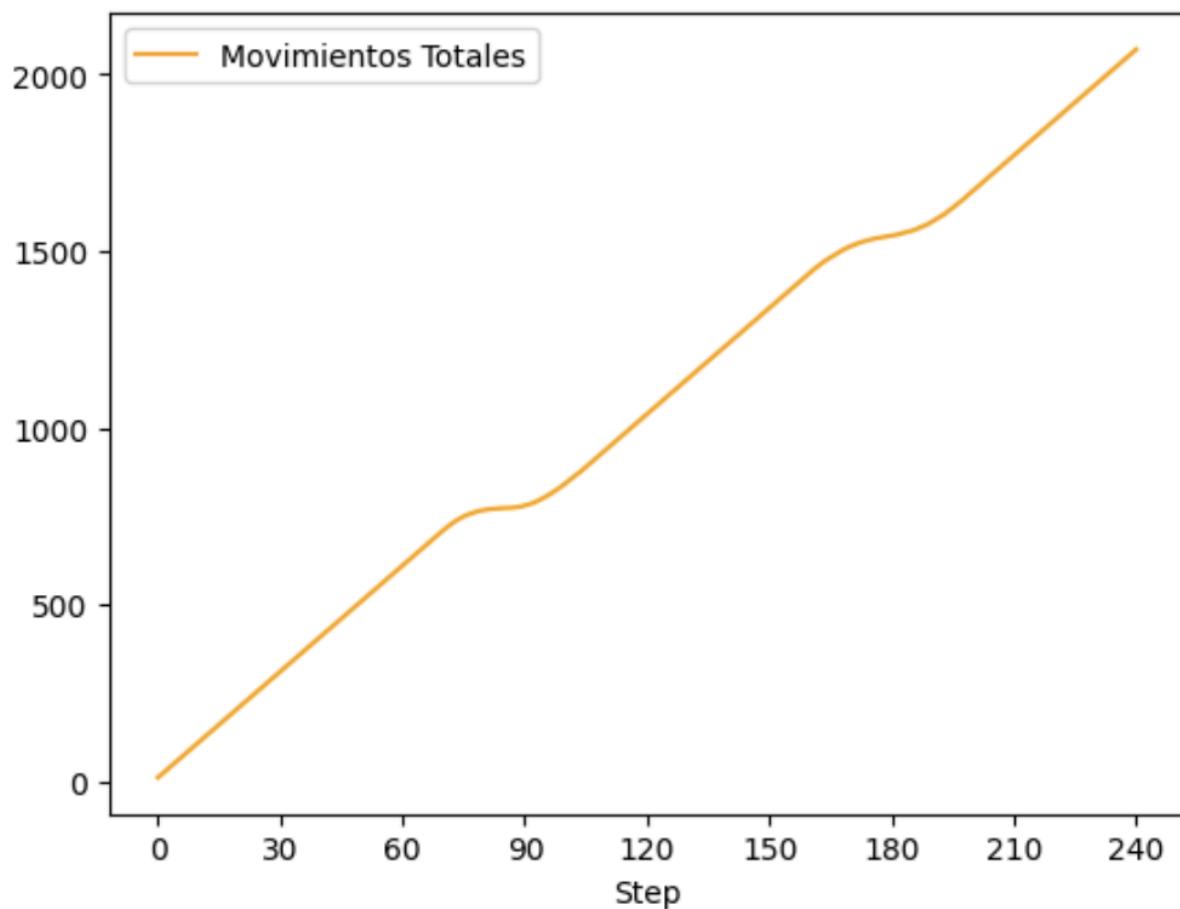


Simulación 2 (9 roombas):

En esta simulación, se utilizaron 9 roombas, las cuales estaban establecidas de forma aleatoria, con su estación de carga y basura también, en posiciones aleatorias. Lo que podemos ver en la gráfica de abajo, la cual nos enseña 2 cosas, la energía de promedio de los 9 roombas, y la cantidad de basura colectada. Esta información nos dice 3 cosas muy claras, uno, al principio de la simulación, los roombas limpian muy rápido la basura, esto puede ser porque hay mucha basura cerca de sus estaciones de carga, dos, el promedio de carga después que todos los roombas cargan, no llega al 100, esto es porque hay unos roombas que llegan a sus estaciones más rápido ya que se encontraban más cerca de sus estaciones, y finalmente, tres, llega un punto en cual muy poca basura es recolectada, esto nos dice que ya queda muy poca o ya no hay basura en la simulación.



En la siguiente grafica, podemos observar los movimientos de los roombas en cada step, aunque se ve que son mucho movimiento por step ya que solo se pueden mover una vez por step, los 9 roombas se mueven de forma simultánea y esto hace que se registren 9 movimientos por step. Otra cosa interesante es que por el rango de 80 a 90 steps, podemos ver que la cantidad de movimientos por step baja de forma sorprendente, pero hay una razón muy simple para esto. Esto pasa ya que los roombas están recargando sus pilas en sus estaciones, y como no hay movimiento, no se registra en la grafica.



Conclusión:

Este sistema implementa un Roomba autónomo inteligente que navega y limpia un ambiente con obstáculos usando un enfoque de toma de decisiones jerárquico. El agente es capaz de percibir su entorno local, priorizar objetivos (limpiar o. recargar), y actuar de manera autónoma maximizando la eficiencia de limpieza bajo restricciones de energía limitada.