



# Tecnológico de Monterrey

Campus Santa Fe

## **MA. Actividad: Roomba**

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 301)

Miranda Urban Solano A01752391

Miércoles 19 de noviembre de 2025

## Introducción

### Problemática

El presente trabajo tiene como objetivo desarrollar y evaluar un sistema de agentes autónomos capaces de realizar tareas de limpieza dentro de un entorno. Específicamente, se busca que los agentes tomen decisiones que les permita maximizar el nivel de limpieza alcanzable dentro de un tiempo de ejecución limitado, mientras se analiza el impacto que tiene el número de agentes en el desempeño como el tiempo requerido y la cantidad de movimientos realizados.

### Desarrollo

#### Análisis del ambiente y agentes

##### *Descripción ambiente*

El ambiente corresponde a una cuadrícula de tamaño  $M \times N$ , la cual se inicializa con:

- Un porcentaje aleatorio de celdas sucias.
- Un conjunto aleatorio de obstáculos que impiden el paso.
- Estaciones de carga asignadas aleatoriamente, en donde aparecen por primera vez los agentes.

##### *Descripción de los agentes*

Los agentes tienen como objetivo principal limpiar el cuarto lo más que pueda en el tiempo máximo de ejecución, sin quedarse sin batería. Para ello se consideran las siguientes características:

- Inicia con 100% de batería.
- Cada acción consume 1% de energía.
- Permanecer sobre una estación de carga durante un episodio recupera 5% de batería.
- Debe ser capaz de planificar su regreso a la estación de carga antes de agotarse.

#### *Características generales*

Con toda la información anterior, es posible determinar que el ambiente presenta las siguientes características:

- Inaccesible: El agente únicamente puede acceder a la información de cierto espacio.
- No determinista: Una acción no tiene un efecto garantizado en todas sus ejecuciones.
- No episódico: El desempeño del agente no depende del episodio anterior.
- Estático: Permanece sin cambios excepto por la realización de acciones de los agentes.
- Discreto: El número de acciones y percepciones es fijo y finito.

Por otro lado, las características de los agentes son las siguientes:

- Reactivos: Tienen la capacidad de percibir su entorno y responden a cambios que hay dentro de este, por ejemplo, saben qué hacer si se encuentran con celdas bloqueadas, sucias, o con estaciones de carga.
- Habilidades sociales: Al encontrarse con otro agente, comparten entre ellos la posición de su estación de carga.

### ***PEAS de los agentes***

- Performance: Los agentes son capaces de percibir la simulación, tomar acciones para sobrevivir (cargarse), y tomar decisiones para cumplir con su objetivo de limpieza (evitar obstáculos y limpiar celdas sucias).
- Ambiente: Su ambiente es una cuadrícula con zonas sucias, limpias y con estaciones de carga generadas de manera aleatoria.
- Actuadores: Para poder interactuar con su entorno, los agentes pueden moverse a cuadrículas vecinas sin obstáculos y preferentemente con suciedad para poder limpiar y cumplir con su objetivo, además conoce su zona de carga (o zonas de carga en general si ya tiene noción de otra) para poder cargarse en caso de tener poca batería.
- Sensores: Sus sensores son la “vista” que tiene de la cuadrícula que la rodea (celdas aledañas) y las características de estas.

### ***Arquitectura de subsunción de los agentes***

Los agentes deben de obtener información de su ambiente y evaluar para la toma de decisiones. Para ello se usa una jerarquía por capas para elegir acciones. A continuación, se muestran las posibles acciones de los agentes y su orden de menor a mayor prioridad:

Explorando  
Comunicándose  
Limiando  
Regresando  
Explorando

### ***Propuesta de solución***

La solución propuesta consiste en un sistema multiagente que utiliza la arquitectura de subsunción para gestionar el comportamiento. Los agentes utilizan percepción local para identificar suciedad, evitar obstáculos, regresar a cargar (calculando la ruta de regreso óptima con BFS) y colaborar entre sí cuando se encuentran. La simulación permite evaluar el desempeño del sistema bajo distintos números de agentes y condiciones iniciales.

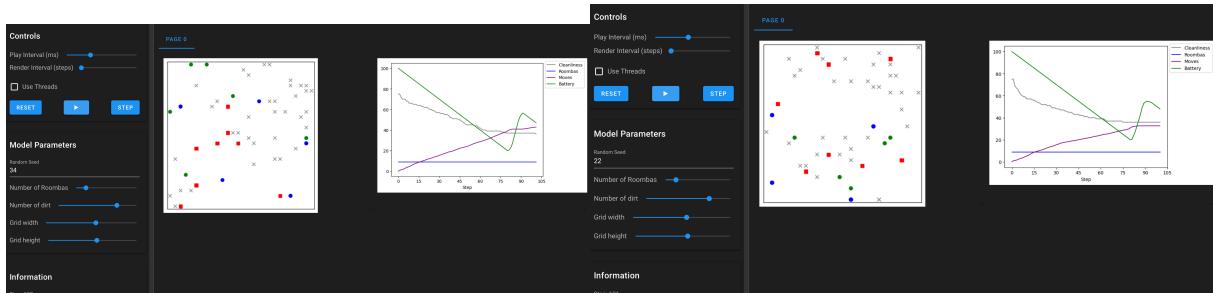
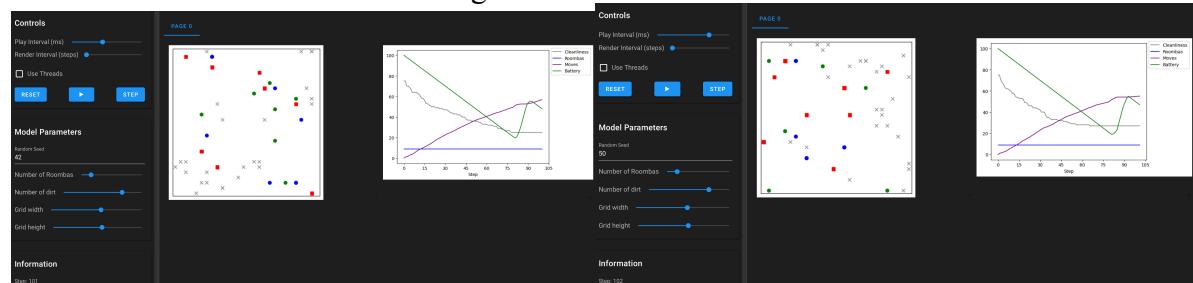
## Análisis de resultados

Para poder comprobar la funcionalidad de la solución propuesta se realizaron dos simulaciones con diferentes consideraciones.

### Simulación

Para la simulación 2, se tomaron en cuenta las consideraciones generales, los agentes empiezan en posiciones aleatorias, los agentes conocen solamente la posición de su estación de carga inicial, pero pueden cargarse en cualquier estación de carga, y se recopilan los datos solicitados por cada agente.

Las estadísticas recolectadas son las siguientes:



La gráfica de *Cleanliness* muestra una disminución progresiva de la suciedad conforme avanza la simulación. Sin embargo, la pendiente no es constante, lo que significa que existen momentos donde la limpieza se estanca temporalmente. Probablemente esto ocurre debido a la dispersión aleatoria de la suciedad, la distancia que los agentes deben recorrer para llegar la suciedad varía bastante (debido a que es random) y que no es la manera más óptima de recorrer el mapa.

La curva de los movimientos crece de manera continua debido a que los Roombas deben recorrer distancias más grandes para encontrar suciedad conforme pasa el tiempo, esto tiene como consecuencia un incremento de movimientos hacia las etapas finales, aun cuando la limpieza ya es alta. Mientras que, la curva de la batería refleja que los agentes identifican otras estaciones de carga, lo que refleja que la lógica de autonomía energética está funcionando como se esperaba.

## Conclusiones

En conclusión, el sistema multiagente desarrollado demostró ser bastante efectivo para la tarea de limpieza, ya que logró obtener una reducción progresiva y significativa de la suciedad en el ambiente mientras mantenía la viabilidad energética de los roombas. La arquitectura de subsunción implementada permitió manejar adecuadamente las prioridades entre limpieza, supervivencia y colaboración, lo que permitió que los agentes funcionaran eficientemente. Además, el sistema mostró ser consistente en desempeño independientemente de la semilla aleatoria utilizada. Por otro lado, la implementación de BFS para el regreso a estaciones de carga resultó crucial, generando rutas óptimas que minimizaron el consumo energético y previniendo muertes por agotamiento de batería. Sin embargo, se identificaron limitaciones importantes, como la tendencia de los agentes a permanecer cerca de las estaciones de carga y una exploración que podría optimizarse mediante algoritmos diferentes y más especializados. Para un trabajo futuro, me gustaría implementar mecanismos de comunicación a distancia, así como incorporar aprendizaje por refuerzo para optimizar decisiones de movimiento.