

# Simulación Multiagente de Movilidad Urbana

**Joaquín Badillo Granillo**

Escuela de Ingeniería y Ciencias  
Tecnológico de Monterrey  
Campus Santa Fe

[A01026364@TEC.MX](mailto:A01026364@TEC.MX)

**Pablo Bolio Pradilla**

Escuela de Ingeniería y Ciencias  
Tecnológico de Monterrey  
Campus Santa Fe

[A01782428@TEC.MX](mailto:A01782428@TEC.MX)

## Resumen

**Palabras Clave:** Simulación, Multiagentes, Agentes Reactivos, Movilidad

## 1. Introducción

El rápido crecimiento urbano ha dado como resultado altas densidades poblacionales. En zonas centralizadas como la Ciudad de México, aunque la cercanía de los servicios es muy alta, la cantidad de vehículos que se trasladan por las calles ha hecho que el tiempo para llegar a dichos servicios sea muy largo en comparación con su distancia.

### 1.1 Propuesta de Solución

En este proyecto se consideró el problema de la movilidad urbana a partir de una simulación multiagente utilizando la librería de mesa ([Kazil, Masad, y Crooks, 2020](#)). En esta simulación, los agentes toman decisiones con un cierto grado de autonomía a partir de un conjunto de reglas finitas y sencillas, como el respetar las señales de tránsito (en particular los semáforos) y la navegación a partir de las rutas más cortas para alcanzar un destino.

Es entonces de nuestro interés observar si a partir de estas sencillas reglas y limitando el entendimiento que tienen los agentes del ambiente a pequeñas vecindades, pueden emerger comportamientos egoístas que resulten en congestionamientos. Asimismo, se planteó una simulación en la que es posible modificar la afluencia de agentes, de tal forma que se puedan observar los límites que tiene el diseño de una ciudad tras un incremento en la densidad poblacional.

## 2. Diseño de Agentes

Para que los agentes puedan interactuar con su entorno, se representaron calles, edificios, destinos, semáforos y vehículos como agentes.

Tanto los semáforos como los vehículos son capaces de cambiar su estado, pero únicamente los segundos tienen un objetivo: que es llegar su destino. Por esta razón, el diseño que es de nuestro interés es el de los vehículos, el cual se describe a detalle a continuación.

## 2.1 Capacidad Efectora (Actuadores)

Las acciones que puede realizar un vehículo están limitadas a:

1. Esperar.
2. Moverse.
3. Calcular una ruta.
4. Llegar a destino.

Esperar, moverse y llegar a su destino, son acciones sencillas y fáciles de implementar; mientras que el calculo de rutas permite una mayor libertad creativa. En esta simulación, los agentes calculan rutas a partir del algoritmo de A\*, usando generalmente el cuadrado de la norma  $\ell^2$  como función de costos y la norma  $\ell^1$  como heurística. En caso de recalculer una ruta, un agente puede además usar la información de sus vecinos más próximos para incrementar el costo entre celdas.

## 2.2 Percepción

Los vehículos son únicamente capaces de observar una vecindad en un radio de  $\sqrt{2}$  unidades (usando la norma  $\ell^2$  como métrica). No obstante cuentan con un conocimiento previo de las calles que les permite calcular rutas eficientes (asumiendo que no hay otros vehículos).

## 2.3 Proactividad

## 2.4 Métricas de Desempeño

Si fuera de nuestro interés determinar el desempeño individual de un agente, una métrica relevante sería la cantidad de episodios que le tomó llegar a su destino, aunque debido a que la distancia que tienen que recorrer varía es aún mejor evaluarlos individualmente como

$$\frac{\text{Episodios Transcurridos}}{\langle \text{Origen, Destino} \rangle_{\ell^1}},$$

donde  $\langle \mathbf{x}, \mathbf{y} \rangle_{\ell^1}$  representa la distancia Manhattan entre 2 puntos  $\mathbf{x}$ ,  $\mathbf{y}$ .

No obstante, como es de nuestro interés determinar el rendimiento de los agentes a un nivel global, es más razonable evaluar el rendimiento que tienen de forma colaborativa y como responden a cambios en el flujo de entrada de agentes al sistema. Por esta razón se decidió que la simulación se detuviera cuando no se pudiera ingresar más agentes en ninguna de las esquinas del modelo (que es donde aparecen los nuevos agentes) y se midió el rendimiento para un flujo de entrada particular como el número de episodios que podían transcurrir sin llegar a la condición de cierre.

$$R(\phi_e) = \text{Episodios Totales}$$

El flujo de entrada en la simulación, se representó como un parámetro de línea de comandos para el servidor, que determina el número de episodios que debe transcurrir antes de volver a agregar agentes.

### 3. Arquitectura de Subsunción

	<b>Evento</b>	<b>Condiciones</b>	<b>Acción</b>
4	Nada	Nada	Calcular ruta
3	Seguir ruta	Celda disponible (es su destino)	Llegar a destino
2	Seguir ruta	Celda disponible	Moverse siguiendo ruta
1	Seguir ruta	Celda ocupada por otro coche	Calcular ruta (actualizando costos para las celdas vecinas)
0	Seguir ruta	En un cruce con semáforo en estado rojo	Esperar

Tabla 1: Arquitectura de subsunción para los coches

### 4. Características del Ambiente

### 5. Conclusiones

## Referencias

Kazil, J., Masad, D., y Crooks, A. (2020). Utilizing python for agent-based modeling: The mesa framework. En R. Thomson, H. Bisgin, C. Dancy, A. Hyder, y M. Hussain (Eds.), *Social, cultural, and behavioral modeling* (pp. 308–317). Cham: Springer International Publishing.