



Simulación de Tráfico Urbano con Sistemas Multiagente

Campus Santa Fe

Emiliano Deyta Illescas | A01785881

Erick Alonso Morales Dieguez | A01029293

Mariano Carretero Fuentes | A01029708

Jueves 4 de Diciembre 2025

Prof. Octavio Navarro Hinojosa

Prof. Gilberto Echeverría Furió

1. Introducción

1.1 Contexto del Problema

La movilidad urbana se entiende como la capacidad que tienen las personas para desplazarse de un punto a otro dentro de una ciudad, aprovechando la infraestructura disponible para realizar actividades económicas, sociales y personales. Cuando esta movilidad se apoya casi exclusivamente en el uso del automóvil particular, se genera una fuerte dependencia al auto que impacta directamente en la calidad de vida de los habitantes.

En México, el uso del automóvil ha crecido de manera acelerada en las últimas décadas. Los Kilómetros-Auto Recorridos (VKT) se han triplicado, pasando de 106 millones en 1990 a 339 millones en 2010. Este incremento se asocia con problemas como smog, incremento de accidentes viales, enfermedades respiratorias y congestión vehicular.

El reto de movilidad urbana que se aborda en este proyecto consiste en proponer una solución que reduzca la congestión vehicular mediante la simulación gráfica del tráfico en una ciudad. Para ello se modela un sistema multiagente donde los vehículos son agentes autónomos que se mueven en una red vial, deben respetar reglas de tránsito, evadir obstáculos y evitar generar cuellos de botella.

1.2 Importancia de Modelar Sistemas Multiagente

Dentro de la inteligencia artificial, un agente se define como un sistema computacional situado en un ambiente, capaz de actuar de manera autónoma para alcanzar objetivos delegados. En un sistema multiagente (MAS), múltiples agentes coexisten, toman decisiones locales, perciben parcialmente el entorno y sus acciones generan efectos colectivos que no se pueden explicar solo desde el comportamiento de un agente aislado.

En el contexto del tráfico vehicular urbano:

- Cada vehículo se modela como un agente autónomo que percibe su entorno local (tipo de celda, estado de semáforo, presencia de otros autos) y escoge una acción de acuerdo con su objetivo principal: llegar a su destino.
- La interacción entre muchos agentes en un ambiente dinámico y no determinista (otros vehículos, semáforos, spawns aleatorios) hace que el comportamiento global del sistema sea difícil de analizar con modelos puramente analíticos.
- Los sistemas multiagente permiten estudiar cómo reglas locales simples (situación → acción) pueden producir patrones globales de flujo, congestión o bloqueo.

1.3 Comportamientos Emergentes en Ambientes Complejos

Un comportamiento emergente es una propiedad global del sistema que surge de la interacción entre agentes y ambiente, y que no puede atribuirse directamente a un solo agente. En la simulación propuesta se observan comportamientos emergentes típicos de sistemas complejos:

- **Congestión vehicular:** al compartir recursos limitados (calles discretas con capacidad unitaria), la acumulación de vehículos en intersecciones produce colas y reducción del flujo.
- **Ondas de tráfico:** cuando un vehículo se detiene (por semáforo o por otro auto), la detención se propaga hacia atrás, generando “olas” de frenado.
- **Deadlocks:** configuraciones en las que varios vehículos se bloquean mutuamente, especialmente en intersecciones con flujos cruzados y alta densidad.
- **Transición de flujo laminar a turbulento:** con baja densidad de vehículos el flujo es relativamente regular; al aumentar la densidad se observan patrones caóticos y zonas de atasco.

Estos fenómenos no son programados explícitamente, sino que emergen de las políticas locales de los agentes y de la estructura del ambiente.

1.4 Alcances y Propósito del Proyecto

Este proyecto implementa una simulación de tráfico urbano utilizando el framework Mesa para sistemas multiagente y una visualización 3D basada en modelos geométricos y matrices de transformación (traslación, escala, rotación) para posicionar los autos, edificios y semáforos dentro de la escena.

Además, se emplea un modelo de iluminación local en tiempo real para resaltar el estado de los semáforos y mejorar la percepción del entorno (por ejemplo, utilizando componentes ambiente, difusa y especular al estilo del modelo de Phong).

Los propósitos principales son:

- Modelar el comportamiento de vehículos en una red vial urbana discreta derivada de un mapa real.
- Analizar el impacto de diferentes parámetros de comportamiento (probabilidad de choque, olvidar ruta, ignorar semáforos, zigzag) en el desempeño global del sistema.
- Comparar conductores “normales” (agentes con comportamiento más racional y deliberativo) contra conductores “drunk”, con componentes que degradan su racionalidad.

- Identificar configuraciones críticas que favorecen o degradan la movilidad urbana y, a partir de ello, extraer conclusiones útiles para el diseño de políticas o mecanismos de control.

2. Problema y Propuesta de Solución

2.1 Descripción del Problema

El problema central que se modela es la congestión vehicular en una ciudad cuya movilidad depende en gran medida del automóvil particular. Desde la perspectiva de agentes, se trata de un ambiente donde:

- Existe alta densidad de vehículos desplazándose en un espacio discreto y limitado.
- Hay múltiples orígenes y destinos, lo que implica diversidad de rutas posibles.
- Intersecciones reguladas por semáforos actúan como recursos críticos para la coordinación de los flujos.
- Los conductores presentan comportamientos heterogéneos: algunos siguen las reglas de manera racional; otros actúan de forma más reactiva y errática.

Se busca que los agentes vehículos alcancen sus destinos evitando colisiones, respetando (en la mayoría de los casos) los semáforos y reduciendo la congestión generada por ellos mismos.

2.2 Limitaciones y Desafíos del Entorno

Limitación	Descripción
Espacio discreto	Grid de 36x35 celdas basado en un mapa de una ciudad.
Direccionalidad	Calles unidireccionales con flujo predefinido.
Capacidad limitada	Una celda solo admite un vehículo a la vez.
Semáforos binarios	Estados verde/rojo sin estados intermedios.

Spawn points fijos Vehículos solo aparecen en bordes del mapa.

Desde el punto de vista de clasificación de ambientes, se trata de un entorno parcialmente accesible, no determinista, no episódico, dinámico y discreto.

2.3 Justificación del Enfoque Multiagente

El enfoque multiagente resulta adecuado porque:

- **Descentralización:** ningún agente tiene una visión completa del sistema; cada vehículo percibe solo información local y toma decisiones autónomas, lo que se alinea con la definición de agentes inteligentes con autonomía y racionalidad limitada.
- **Escalabilidad:** el modelo permite agregar o quitar vehículos sin cambiar la lógica central del ambiente, solo modificando parámetros como el intervalo de spawn.
- **Realismo:** en la vida real los conductores no conocen el estado global del tráfico; operan con percepciones locales y reglas simples (seguir calle, respetar semáforo, evitar chocar).
- **Emergencia:** el tráfico surge de la combinación de muchos comportamientos reactivos y deliberativos simples, tal como se ejemplifica en arquitecturas reactivas donde conductas locales generan patrones complejos.

2.4 Propuesta General de Solución

La solución implementada se organiza en términos de tipos de agentes y un modelo de ciudad:

- **Agentes Car:** vehículos “normales” que combinan deliberación y reactividad. Utilizan A* para calcular una ruta hacia su destino, respetan la direccionalidad de las calles, verifican el estado de los semáforos y evitan colisiones siempre que es posible.
- **Agentes drunkDriver:** variantes de los Car que introducen comportamientos erráticos (ignorancia de semáforos, olvidos de ruta, movimiento en zigzag y probabilidad aumentada de colisión).
- **Agentes Traffic_Light:** semáforos que alternan entre verde y rojo siguiendo un ciclo fijo de steps.
- **Modelo CityModel:** integra el mapa de la ciudad, administra la creación y eliminación de vehículos, gestiona los semáforos y recolecta métricas de desempeño global.

La visualización gráfica 3D coloca cada agente en el espacio mediante transformaciones geométricas y un esquema de iluminación que permite distinguir claramente el estado de los semáforos y la posición relativa de los vehículos en el entorno urbano.

2.5 Objetivos Principales del Modelo

- **Throughput:** maximizar el número de vehículos que llegan a su destino por unidad de tiempo (steps).
- **Eficiencia:** lograr que al menos 85 % de los vehículos generados alcancen su destino.
- **Resiliencia:** analizar cómo cambia el desempeño del sistema ante la introducción de conductores erráticos.
- **Análisis paramétrico:** identificar, mediante experimentos sistemáticos, qué combinaciones de parámetros afectan más la movilidad.

3. Diseño del Agente

3.1 Tipo de Agente

Agente Car (Normal)

- **Tipo:** agente híbrido (reactivo + deliberativo).
 - **Componente deliberativa:** planificación de ruta mediante A*, considerando el mapa como un grafo de celdas transitables.
 - **Componente reactiva:** respuestas inmediatas ante cambios en el entorno (semáforo rojo, vehículo adelante, colisión).

Agente drunkDriver

- **Tipo:** agente predominantemente reactivo.
 - Hereda la estructura básica de Car, pero introduce probabilidades para ignorar semáforos, olvidar la ruta y realizar movimientos aleatorios.

Desde la teoría de agentes inteligentes, ambos tipos comparten:

- **Racionalidad limitada:** seleccionan acciones que maximizan su medida de desempeño con base en percepciones locales y conocimiento parcial del entorno.
- **Autonomía:** deciden por sí mismos sin intervención externa directa en cada step.
- **Reactividad:** perciben cambios (como un semáforo que cambia a rojo) y responden oportunamente.

- **Proactividad:** en el caso de Car, toman iniciativa para acercarse al destino, no solo reaccionan a estímulos.

3.2 Objetivo del Agente

Agente	Meta Principal	Condición de Éxito
Car	Llegar al destino asignado	<code>self.cell == self.destination</code>
drunkDriver	Llegar al destino (con ruido)	Misma condición, pero con trayecto degradado por comportamientos erráticos
Traffic_Light	Regular el flujo vehicular	Alternar estados según ciclo configurado

La medida de desempeño se centra en llegar al destino, minimizar el tiempo de viaje y evitar colisiones, en línea con la noción de racionalidad que evalúa al agente por la calidad de sus resultados según una métrica de desempeño.

3.3 Modelo PEAS del Agente

Car

- **P – Performance (Desempeño)**
 - Llegar exitosamente al destino.
 - Minimizar el número de steps empleados.
 - Evitar colisiones.
- **E – Environment (Entorno)**
 - Grid discreto 36x35.
 - Calles unidireccionales.
 - Semáforos con ciclos de 10 steps.
 - Otros vehículos como obstáculos dinámicos.
 - Edificios como obstáculos estáticos.
- **A – Actuators (Actuadores)**

- Moverse a una celda adyacente (arriba, abajo, izquierda, derecha) respetando la dirección de la calle.
- Detenerse (esperar por semáforo o por tráfico).
- Removerse del sistema al alcanzar el destino.
- **S – Sensors (Sensores)**
 - Tipo de celda actual (calle, semáforo, destino, edificio).
 - Estado del semáforo asociado a la intersección.
 - Presencia de vehículos en celdas adyacentes.
 - Coordenadas del destino objetivo.

3.4 Reactividad

El comportamiento reactivo se implementa como reglas situación → acción, típicas de agentes reactivos:

Para Car:

- **Si** el semáforo actual está en rojo
→ **Entonces** el agente espera (no se mueve en ese step).
- **Si** la celda siguiente en la ruta está ocupada por otro vehículo
→ **Entonces** espera o, en casos específicos, se activa la lógica de choque con cierta probabilidad `crash_prob`.

Para `drunkDriver` la reactividad se modifica por probabilidades:

- **Si** semáforo en rojo
→ con probabilidad `ignore_light_prob` lo ignora y avanza, en otro caso se detiene.
- **Si** se activa el modo “olvidar ruta” (`forget_route_prob`)
→ el agente deja de seguir el plan deliberativo y realiza movimientos aleatorios durante varios steps.

3.5 Proactividad

Los agentes muestran proactividad en los siguientes aspectos:

- Cálculo previo de la ruta mediante A* antes de iniciar el movimiento.
- Replanificación de la ruta cuando la trayectoria actual se vuelve inválida (por cambios en el entorno).
- Búsqueda alternativa de caminos cuando la ruta directa se encuentra bloqueada.

3.6 Racionalidad

La racionalidad del agente se evalúa respecto a la medida de desempeño definida: maximizar la probabilidad de llegar al destino de forma eficiente y segura.

- Car intenta seguir la ruta óptima calculada por A* y respetar las restricciones del entorno, procurando minimizar tiempo y evitar colisiones.
- drunkDriver conserva la misma meta, pero su racionalidad se ve deteriorada por decisiones probabilísticas que lo alejan del comportamiento óptimo (por ejemplo, olvidar la ruta o moverse en zigzag).

Los criterios de decisión incluyen:

1. Priorizar el progreso hacia el destino.
2. Respetar direcciones de las calles y estados de semáforos.
3. Reducir la probabilidad de colisión frente a otros vehículos.

3.7 Comportamientos y Reglas Situación → Acción

Car (Normal)

Situación	Acción
No hay ruta calculada	Calcular A* hacia el destino.
Semáforo en rojo	Esperar (waiting_at_light = True).
Vehículo en la celda siguiente	Esperar.
Vehículo adelante + crash_prob	Provocar choque y entrar en modo de recuperación (10 steps).
Ruta disponible y libre	Avanzar a la siguiente celda del path.

Llegó al destino

Removerse del sistema.

drunkDriver

Situación

Acción

random_steps_remaining > 0

Moverse a un vecino aleatorio.

Tirada < forget_route_prob

Activar modo random durante 3–5 steps.

Semáforo rojo + tirada < ignore_light_prob

Ignorar semáforo y avanzar.

Movimiento normal

Aplicar zigzag con cierta probabilidad.

4. Arquitectura de Subsunción

4.1 Justificación Teórica

La arquitectura de subsunción es un ejemplo clásico de arquitectura de agentes reactivos basada en múltiples comportamientos simples, organizados en capas con distintas prioridades. Las capas más bajas tienen mayor prioridad y pueden inhibir o suprimir las capas superiores.

Este enfoque es adecuado para el tráfico urbano porque:

- Los comportamientos de “sobrevivencia” (como evitar colisiones) deben tener prioridad absoluta sobre otros objetivos como seguir la ruta.
- Permite agregar comportamientos más complejos (por ejemplo, zigzag o ignorar semáforos en los drunkDriver) sin reescribir los comportamientos básicos.

4.2 Capas Definidas y Propósito

Representación de capas:

CAPA 3: Navegación Deliberativa ← A* pathfinding

Planificar ruta al destino.

CAPA 2: Seguimiento de Reglas ← Respeto de semáforos y direcciones

Obedecer señales de tráfico.

CAPA 1: Evitación de Colisiones ← Detección de vehículos vecinos

No chocar con otros.

CAPA 0: Movimiento Básico ← Motor de movimiento

Ejecutar el desplazamiento seleccionado.

4.3 Prioridades Entre Comportamientos

Prioridad	Comportamiento	Descripción
1 (Alta)	Evitar colisión	Siempre verificar antes de moverse.
2	Respetar semáforo	A menos que el agente sea drunkDriver que decide ignorar.
3	Seguir ruta A*	Navegación hacia destino.
4 (Baja)	Movimiento por defecto	Seguir dirección de calle si no hay ruta.

4.4 Mecanismo de Supresión e Inhibición

La inhibición de capas se implementa de forma explícita en el método `step()` del agente, utilizando retornos tempranos:

```
def step(self):
```

```
    # CAPA 1: Recuperación de crash (inhibe todo)
```

```
    if self.crash_recovery_steps > 0:
```

```

self.crash_recovery_steps -= 1
return # SUPRIME todas las capas inferiores

# CAPA 2: Llegada a destino (inhibe movimiento)
if self.cell == self.destination:
    self.remove()
    return # SUPRIME movimiento

# CAPA 3: Movimiento normal
self.move_along_path()

```

Cada condición de nivel alto (por ejemplo, estar en recuperación de un choque o haber llegado al destino) detiene la ejecución de comportamientos inferiores como navegación o movimiento básico.

4.5 Ejemplos de Situaciones

Ejemplo 1: Coche normal en semáforo rojo con vehículo adelante

- Capa 3 (Navegación A*): indica avanzar.
- Capa 2 (Semáforo): detecta rojo → decide esperar.
- Capa 1 (Colisión): en este caso no llega a evaluarse porque ya se decidió esperar.

Resultado: el coche se detiene en el semáforo.

Ejemplo 2: drunkDriver que olvida la ruta

- Capa 3: existe un path calculado.
- Comportamiento estocástico: se activa forget_route.
- Capa 2: no hay semáforo restrictivo.
- Capa 1: no hay vehículo inmediato que provoque colisión.
- Capa 0: ejecuta movimiento aleatorio (zigzag o vecinos aleatorios).

Resultado: el vehículo se desorienta temporalmente, recorriendo trayectorias poco eficientes.

5. Características del Ambiente

5.1 Clasificación del Ambiente

Característica	Clasificación	Justificación
Accesibilidad	Parcialmente accesible	Los agentes ven solo celdas adyacentes y su semáforo.
Determinismo	No determinista	La acción de moverse puede verse afectada por otros agentes (ocupación de celdas).
Episodicidad	No episódico	Las decisiones actuales afectan el estado futuro del sistema.
Dinamismo	Dinámico	El ambiente cambia mientras los agentes deciden, debido a otros vehículos y al ciclo de semáforos.
Continuidad	Discreto	Grid de celdas y tiempo en steps discretos.

Estas dimensiones corresponden al marco teórico de clasificación de ambientes para agentes.

5.2 Representación del Ambiente en el Modelo

- **Mapa base:** archivo 2024_base.txt con un grid de 36x35 caracteres.
- **Símbolos principales:**
 - $> < ^ \vee$: calles con dirección (derecha, izquierda, arriba, abajo).
 - S s : semáforos (diferente estado inicial).
 - # : edificios/obstáculos.
 - D : celdas de destino.
- La interpretación del mapa se apoya en un diccionario (mapDictionary.json) que asigna propiedades a cada tipo de celda.

En la parte gráfica 3D, cada celda se asocia a modelos geométricos (autos, semáforos, edificios) posicionados mediante matrices de transformación: traslación para ubicar el objeto en el grid, escala para ajustar su tamaño al entorno, y rotación para alinearlo con la dirección de la calle.

5.3 Agentes y Entidades Presentes

Tipo	Clase	Cantidad	Móvil
Vehículo normal	Car	Variable (spawn dinámico)	Sí
Conductor errático	drunkDriver	Variable (según normal_spawn_ratio)	Sí
Semáforo	Traffic_Light	Fijo (definido en el mapa)	No
Calle	Road	Fijo	No
Edificio	Obstacle	Fijo	No
Destino	Destination	Fijo	No

5.4 Restricciones y Reglas del Entorno

- **Unidireccionalidad:** las calles solo permiten movimiento en la dirección indicada por el símbolo del mapa.
- **Capacidad unitaria:** cada celda puede contener a lo sumo un vehículo.
- **Spawn en bordes:** los vehículos aparecen únicamente en las calles del perímetro del mapa.
- **Semáforos síncronos:** todos los semáforos comparten un ciclo fijo (10 steps).
- **Destinos fijos:** las celdas D son estáticas durante toda la simulación.

5.5 Dinámica del Sistema y Fuentes de Incertidumbre

Las principales fuentes de incertidumbre son:

- **Spawn estocástico:** tanto el punto de aparición del vehículo como su destino se seleccionan de forma aleatoria entre un conjunto de celdas válidas.
- **Tipo de conductor:** con probabilidad `normal_spawn_ratio` se genera un `Car` y, en caso contrario, un `drunkDriver`.
- **Comportamientos erráticos:** los `drunkDriver` toman decisiones basadas en probabilidades (`forget_route_prob`, `ignore_light_prob`, etc.).
- **Orden de ejecución:** en cada step, el orden en que se actualizan los agentes puede variar, lo que introduce no determinismo adicional.

6. Evaluación del Modelo

6.1 Objetivos de la Evaluación

- Determinar la eficiencia del sistema de tráfico bajo diferentes configuraciones de parámetros.
- Identificar qué parámetros tienen mayor impacto en la movilidad (eficiencia, throughput, tiempo hasta meta).
- Comparar explícitamente las configuraciones con conductores normales vs escenarios con creciente proporción de `drunkDriver`.
- Establecer límites operativos donde el sistema deja de ser funcional (eficiencia muy baja o tiempos excesivos).

6.2 Métricas Utilizadas

Métrica	Fórmula	Descripción
Eficiencia	$\text{cars_arrived} / \text{cars_spawned}$	Porcentaje de vehículos que llegan a destino.
Throughput	$\text{cars_arrived} / \text{total_steps}$	Vehículos procesados por unidad de tiempo.
Steps hasta meta	Steps hasta que eficiencia $\geq 85\%$	Velocidad global del sistema.

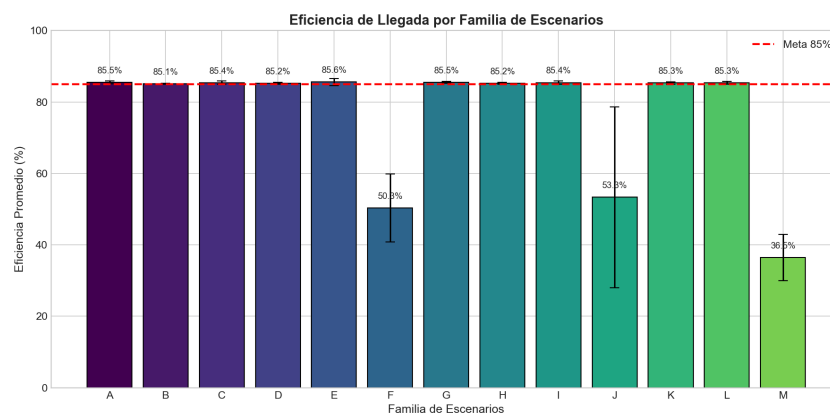
6.3 Parámetros Evaluados

Parámetro	Valores Probados	Variable
Spawn Interval	{1, 3, 5, 7}	spawn_interval
Normal Crash Prob	{0, 0.35, 0.7}	normal_crash_prob
Proporción Normales	{1.0, 0.75, 0.5}	normal_spawn_ratio
Drunk Crash Prob	{0, 0.35, 0.7}	drunk_crash_prob
Ignorar Semáforos	{0, 0.35, 0.7}	drunk_ignore_light_prob
Olvidar Ruta	{0, 0.35, 0.7}	drunk_forget_route_prob
Zigzag	{0, 0.35, 0.7}	drunk_zigzag_intensity

Total de escenarios evaluados: **116**.

6.4 Resumen General de Resultados

6.4.1 Eficiencia por Familia de Escenarios



Análisis por familia:

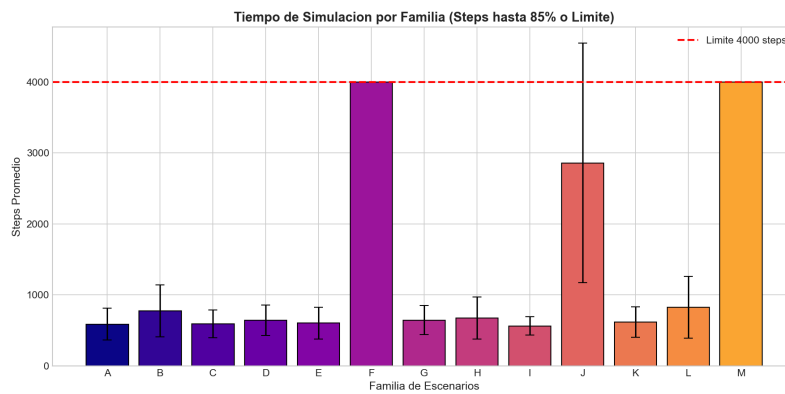
Familia	Eficiencia Media	Observaciones
A	85.5 %	Escenario base: solo coches normales.
B	85.1 %	La probabilidad de crash en normales casi no afecta.
C	85.4 %	25 % de drunk sin comportamiento errático crítico.
D	85.2 %	Crash en drunk tiene impacto mínimo.
E	85.6 %	Ignorar semáforos produce efecto neutro o ligeramente positivo.
F	50.3 %	Olvidar ruta tiene impacto severo.
G	85.5 %	Zigzag con efecto despreciable.
H	85.2 %	50 % drunk con crash.
I	85.4 %	50 % drunk ignorando luces.
J	53.3 %	50 % drunk olvidando ruta.
K	85.3 %	50 % drunk con zigzag.

L 85.3 % Combinación crash normal + drunk.

M 36.5 % Peor caso: todos los factores activos.

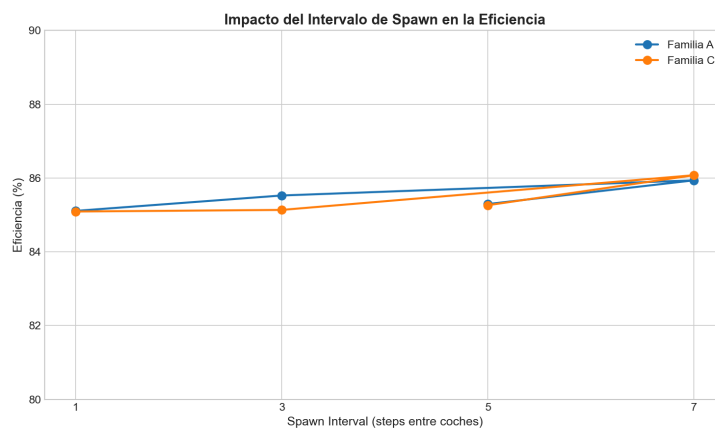
Hallazgo principal: la probabilidad `forget_route_prob` es el parámetro con impacto más negativo en la eficiencia del sistema. El resto de parámetros (`crash`, `ignorar semáforos`, `zigzag`) mantienen la eficiencia cercana a ~85 %.

6.4.2 Tiempo de Simulación por Familia



Las familias **F**, **J** y **M** alcanzan el límite de 4000 steps sin lograr el umbral de 85 % de eficiencia, lo que confirma que el comportamiento de olvidar ruta genera degradación prolongada en el sistema.

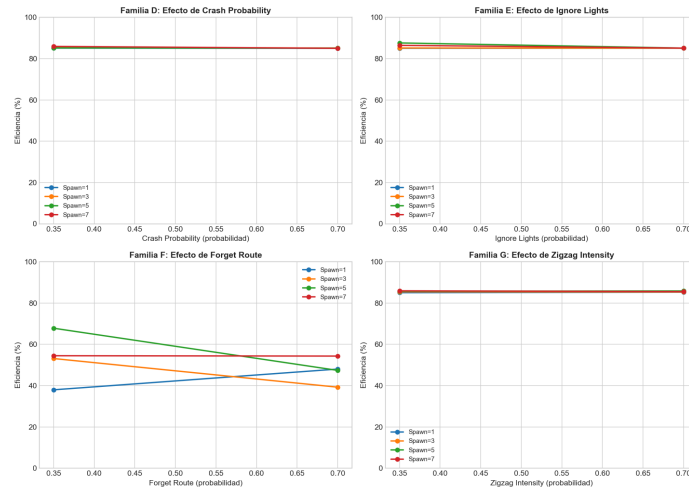
6.4.3 Impacto del Spawn Interval



- Intervalos de spawn pequeños (1) generan mayor densidad de tráfico, pero mientras no se active `forget_route`, el sistema mantiene eficiencia alta.

- Con intervalos grandes (7), la densidad baja y el tráfico es más fluido, pero la eficiencia sigue cercana a 85 % en escenarios sin forget_route.

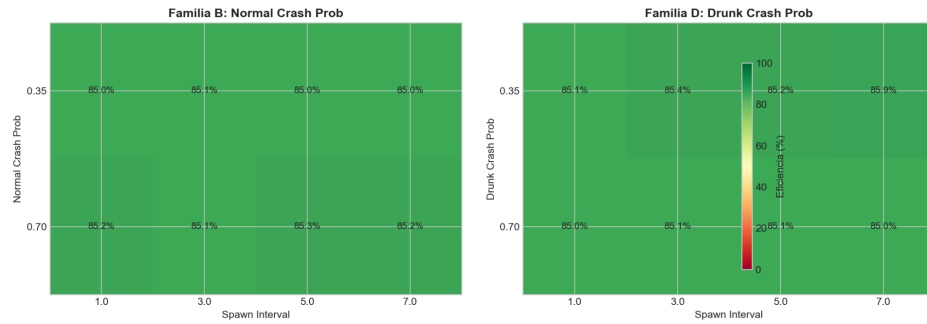
6.4.4 Comparación de Comportamientos Drunk



Comportamiento Impacto en Explicación Eficiencia

Crash	Bajo (~0 %)	Los choques generan pausas locales y temporales.
Ignorar luces	Bajo (~0 %)	En algunos casos incluso reduce espera innecesaria.
Forget Route	Alto (-30 % a -50 %)	Los vehículos se desorientan y ocupan espacio sin avanzar.
Zigzag	Muy bajo (~0 %)	El movimiento lateral no impide alcanzar destino.

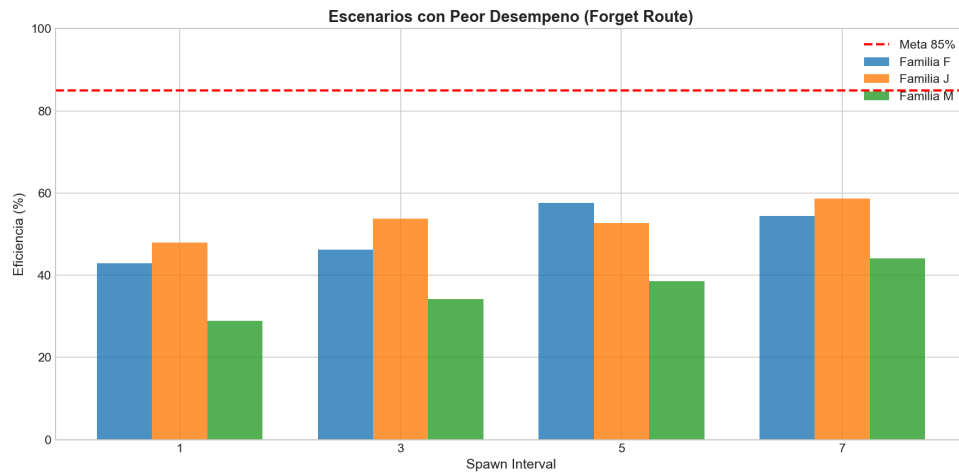
6.4.5 Análisis de Probabilidad de Crash



Los mapas de calor muestran que incrementar la probabilidad de crash (para normales y drunk) tiene efecto limitado en la eficiencia general:

- Las colisiones son eventos locales.
- El tiempo de recuperación (10 steps) es corto.
- Otros vehículos pueden rodear al vehículo en recuperación.

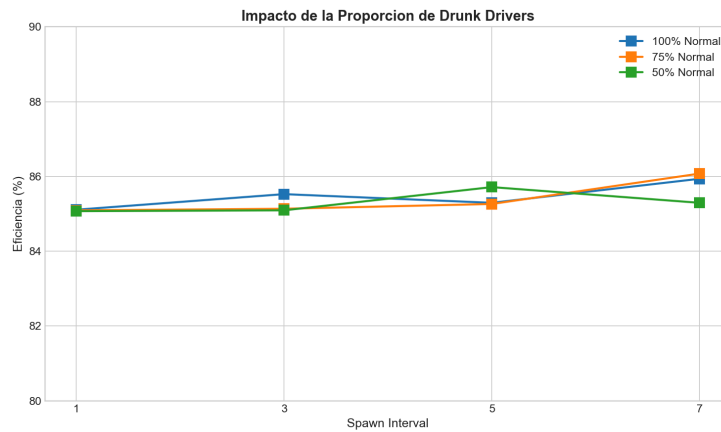
6.4.6 Escenarios con Peor Desempeño



Las familias de peor desempeño (**F**, **J**, **M**) comparten el hecho de tener `forget_route_prob > 0`:

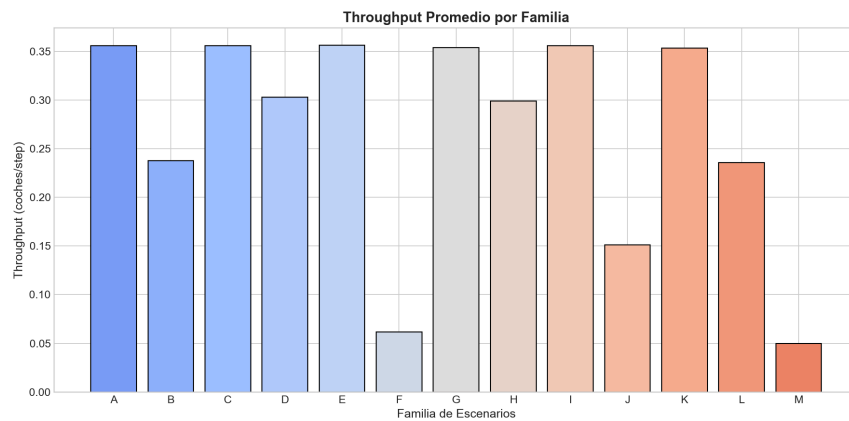
- Familia F: 75 % normales, `forget_route = {0.35, 0.7}`.
- Familia J: 50 % normales, `forget_route = {0.35, 0.7}`.
- Familia M: todos los parámetros en 0.35.

6.4.7 Impacto de la Proporción de Drunk Drivers



Cuando `forget_route = 0`, la proporción de `drunkDriver` no modifica significativamente la eficiencia, incluso con porcentajes altos. Esto muestra que el sistema es robusto a ciertos tipos de ruido siempre que los agentes mantengan una ruta clara hacia sus metas.

6.4.8 Throughput del Sistema



El throughput (vehículos/step) se mantiene relativamente estable en todas las familias excepto en aquellas donde aparece `forget_route`, donde disminuye considerablemente debido a que los vehículos permanecen más tiempo dentro del sistema.

6.4.9 Tabla Resumen de Resultados

Familia	Eficiencia Media	Eficiencia Min	Eficiencia Max	Steps Promedio
A	85.5 %	85.1 %	85.9 %	587
B	85.1 %	85.0 %	85.3 %	775
C	85.4 %	85.1 %	86.1 %	589
D	85.2 %	85.0 %	85.9 %	642
E	85.6 %	85.0 %	87.6 %	601
F	50.3 %	38.0 %	67.8 %	4001
G	85.5 %	85.0 %	85.9 %	642
H	85.2 %	85.0 %	85.9 %	674
I	85.4 %	85.1 %	86.3 %	563
J	53.3 %	28.2 %	85.7 %	2860
K	85.3 %	85.0 %	86.0 %	615

L	85.3 %	85.0 %	86.3 %	827
M	36.5 %	28.9 %	44.1 %	4001

7. Conclusiones

7.1 Funcionamiento Global del Sistema

El sistema multiagente implementado logra simular de manera adecuada el tráfico vehicular urbano en un ambiente discreto, dinámico y parcialmente accesible. De acuerdo con las métricas definidas:

- El modelo es robusto: bajo configuraciones normales alcanza de forma consistente eficiencias superiores al 85 %.
- La navegación basada en A* proporciona rutas razonablemente óptimas dentro del mapa discreto, aun cuando el ambiente es no determinista.
- La combinación de semáforos, restricciones de capacidad por celda y reglas locales de los agentes produce un flujo de tráfico realista en términos cualitativos.

7.2 Comportamientos Emergentes Observados

Se observaron varios comportamientos emergentes, consistentes con sistemas complejos de agentes:

- **Congestión en intersecciones:** múltiples vehículos detenidos por semáforos generan colas visibles en el grid.
- **Propagación de detenciones:** una detención se propaga hacia atrás, generando ondas de tráfico.
- **Auto-regulación del tráfico:** en muchas configuraciones el sistema encuentra un equilibrio dinámico donde la entrada de nuevos vehículos se compensa con el número de vehículos que salen.
- **Degradación por desorientación:** cuando los agentes olvidan su ruta, se generan bloqueos y congestiones persistentes que reducen drásticamente la eficiencia global.

7.3 Limitaciones del Modelo

Limitación	Impacto	Posible Mejora
Grid discreto	Movimientos poco naturales	Migrar a espacio continuo o celdas más finas.
Semáforos síncronos	Ciclos no optimizados por flujo	Diseñar semáforos adaptativos.
Una celda = un carro	Subestima capacidad real	Modelar múltiples carriles o ancho de vía.
A* estático	No considera tráfico dinámico	Usar algoritmos adaptativos (D*, etc.).
Sin comunicación V2V	Agentes completamente aislados	Introducir mensajes entre agentes.

7.4 Posibles Mejoras

- **Semáforos adaptativos:** ajustar los ciclos de luz según la densidad local de vehículos.
- **Rutas alternativas dinámicas:** replanificación más frecuente en respuesta a congestiones.
- **Comunicación entre agentes:** permitir que los vehículos compartan información sobre bloqueos o accidentes.
- **Múltiples carriles:** modelar mejor la estructura vial real.
- **Agentes prioritarios:** introducir vehículos de emergencia con reglas especiales (pueden ignorar ciertos semáforos).

7.5 Conclusión Final

El análisis paramétrico muestra que el parámetro más crítico es la probabilidad de olvidar ruta (`forget_route_prob`). Cuando los agentes pierden su plan deliberativo y actúan únicamente de forma reactiva y aleatoria:

- Se incrementa el tiempo de permanencia de los vehículos en el sistema.
- Se forman bloqueos en intersecciones y zonas de alta densidad.
- El throughput disminuye y la eficiencia puede caer por debajo del 50 %.

En cambio, comportamientos como colisionar ocasionalmente, ignorar algunos semáforos o moverse en zigzag tienen un impacto mucho menor, ya que:

- Son eventos locales con recuperación rápida.
- No impiden que el agente conserve una ruta clara hacia su meta.
- El sistema tiene suficiente redundancia espacial para absorber estas perturbaciones.

Desde una perspectiva de movilidad urbana, esto sugiere que los comportamientos que generan desorientación o pérdida del objetivo (por ejemplo, distracciones graves del conductor) pueden tener un impacto sistémico más fuerte que eventos puntuales como pequeños choques o maniobras arriesgadas aisladas. Reducir la probabilidad de “perder la ruta” en sentido literal o figurado se vuelve clave para mantener la eficiencia de la red de tráfico.