

**Instituto Tecnológico y de Estudios Superiores de Monterrey**



**Modelación de sistemas multiagentes con gráficas computacionales**

**GRUPO 570**

**MAESTROS:**

Mauricio Bezares Peñúñuri

Iván Axel Dounce Nava

**Actividad integradora : Movilidad Urbana**

**EQUIPO**

Ithandehui Joselyn Espinoza	A01734547
Sergio Jiawei Xuan	A01784695
Kevin Alfredo Martínez Catalán	A01660520

**FECHA**

# Diseño del Sistema Multiagente para la Simulación de Tráfico

Nuestra simulación de tráfico se basa en un sistema multiagente, en el cual diferentes tipos de agentes interactúan para replicar dinámicas del tráfico urbano. Los agentes tienen roles específicos y toman decisiones en función de reglas establecidas y datos del entorno.

## 1. CarAgent (Agente Vehículo)

El **CarAgent** representa los automóviles dentro de la simulación, desplazándose a lo largo de una cuadrícula y siguiendo reglas de tráfico predefinidas. Puede cambiar de dirección, detenerse ante semáforos o congestión, y acudir a estaciones de servicio cuando el nivel de combustible es bajo.

- **Estados posibles:**

MOVING: Se está desplazando a su destino.

WAITING: Detenido por un semáforo en rojo o sin movimiento válido.

REFUELING: En una gasolinera, recargando gasolina.

- **Acciones posibles:**

Moverse en una dirección válida.

Detenerse si hay un semáforo en rojo.

Cambiar de dirección si hay un bloqueo.

Dirigirse a una GasStationAgent si el nivel de gasolina es bajo.

## Descripción Refinada

Para evaluar la efectividad de la simulación, definimos métricas clave para cada agente:

- **Indicadores de desempeño:**

- Tiempo promedio en llegar a su destino (cuanto menos, mejor).
- Número de paradas en semáforos (menos paradas indica mejor flujo).
- Cantidad de recargas de gasolina (demasiadas recargas indican rutas ineficientes).

- **Éxito:**

Un CarAgent es exitoso si llega a su destino en el menor tiempo posible sin quedarse sin gasolina.

## **2. TrafficLightAgent (Agente Semáforo Inteligente)**

El TrafficLightAgent regula el flujo de tráfico en intersecciones controlando los cambios de luz en los semáforos. Su comportamiento se basa en un algoritmo de aprendizaje por refuerzo, lo que le permite optimizar los tiempos de espera y mejorar la circulación en función de la congestión detectada.

- **Estados posibles:**

RED: Luz roja, los autos deben detenerse.

GREEN: Luz verde, los autos pueden avanzar.

YELLOW: Cambio de rojo a verde o viceversa.

- **Acciones posibles:**

Mantener luz actual (rojo o verde).

Ajustar el tiempo de cambio según la congestión detectada.

Optimizar tiempos de espera para mejorar el flujo de tráfico.

### **Descripción Refinada**

Para evaluar la efectividad de la simulación, definimos métricas clave para cada agente:

- **Indicadores de desempeño:**

- Tiempo total de espera de los autos en la intersección.
- Número de cambios de luz por ciclo (demasiados cambios pueden ser ineficientes).
- Reducción de congestión en intersecciones.

- **Éxito:**

El TrafficLightAgent es exitoso si reduce el tiempo de espera promedio de los autos y mantiene el tráfico en movimiento sin atascos.

### **3. GasStationAgent (Agente Gasolinera)**

Actúa como un punto estratégico en la simulación donde los **CarAgents** pueden detenerse para recargar combustible cuando su nivel es bajo. Funciona como un **nodo en la cuadrícula**, regulando el flujo de autos que requieren abastecimiento y evitando congestiones en su área.

- **Estados posibles:**

AVAILABLE: Gasolinera disponible.

BUSY: Gasolinera ocupada con un auto recargando.

- **Acciones posibles:**

Permitir que un CarAgent recargue gasolina.

Administrar tiempos de recarga..

### **Descripción Refinada**

Para evaluar la efectividad de la simulación, definimos métricas clave para cada agente:

- **Indicadores de desempeño:**

- Número de autos que necesitan recarga (si es muy alto, los autos pueden no estar optimizando rutas).
- Tiempo promedio de recarga.

- **Éxito:**

La GasStationAgent es exitosa si los autos se recargan eficientemente y no hay largas filas de espera.

## **Descripción final del ambiente**

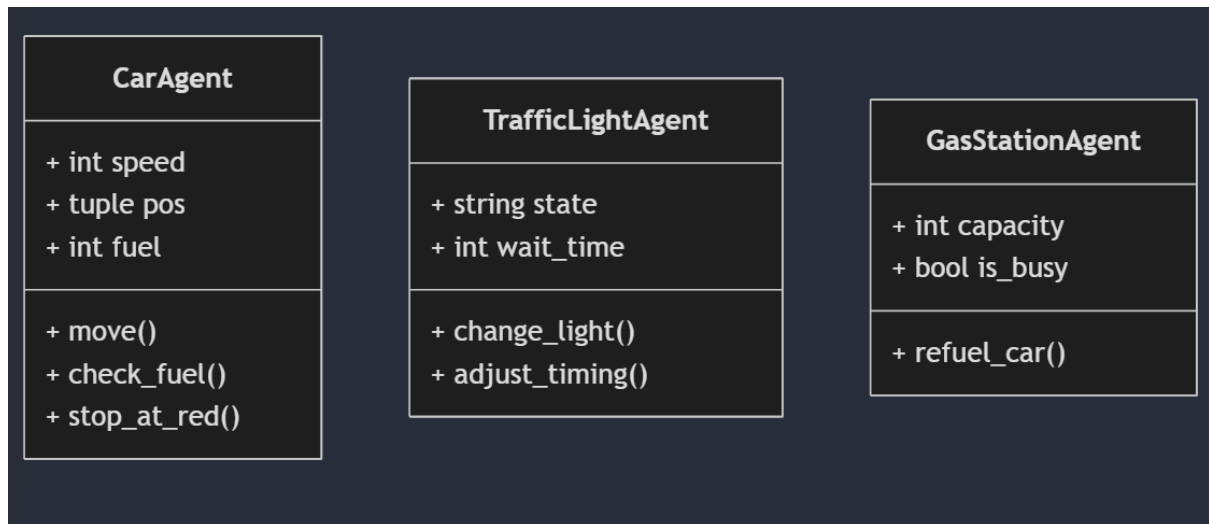
La simulación se desarrolla en un espacio urbano cuadriculado con diferentes elementos:

- **Características del entorno:**
  - Grid de 20x20 celdas que representa la ciudad.
  - Calles donde los CarAgent pueden moverse siguiendo reglas de tráfico.
  - Intersecciones con semáforos, gestionadas por TrafficLightAgent.
  - Gasolineras distribuidas estratégicamente para que los autos recarguen.

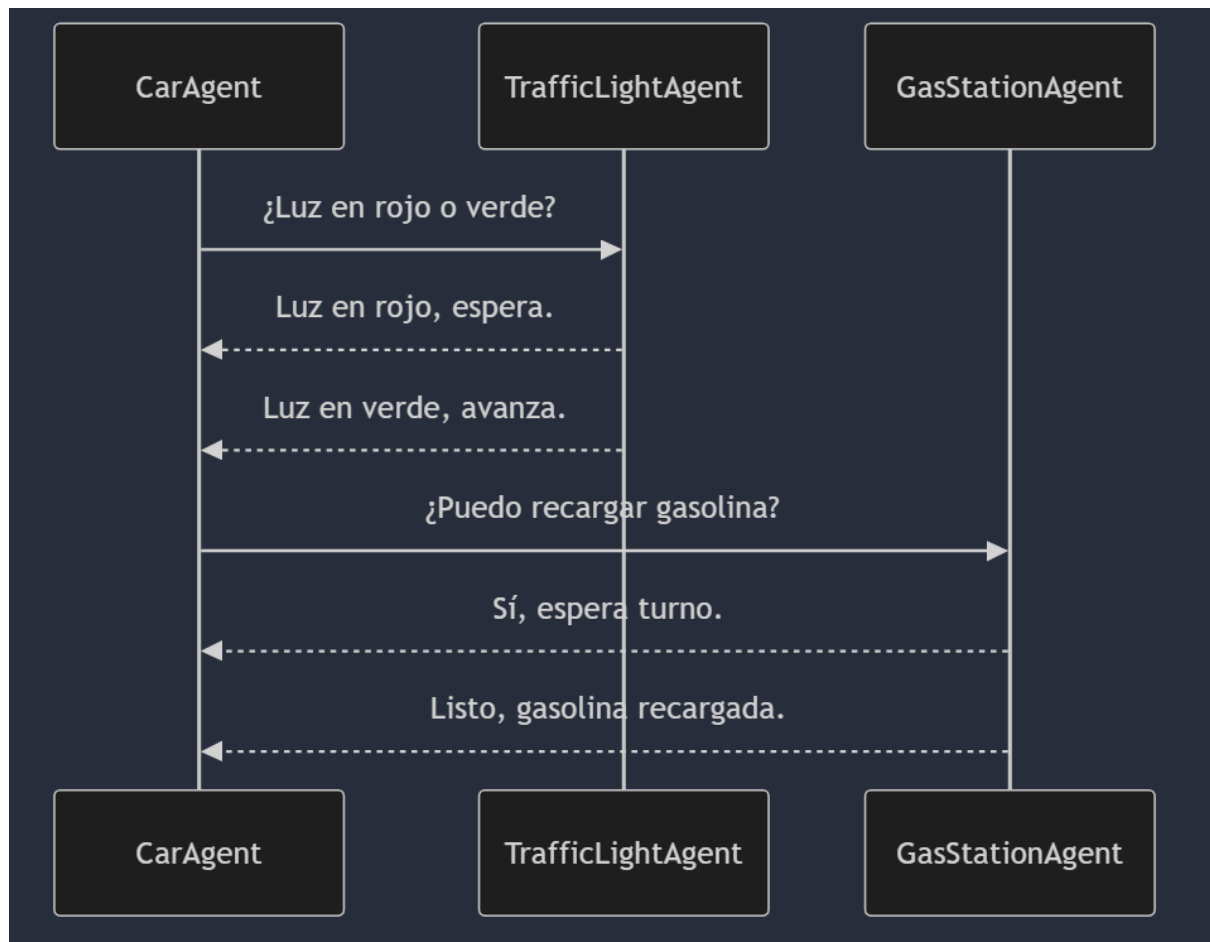
### **Reglas del tráfico:**

- Los autos deben respetar los semáforos (TrafficLightAgent).
- Si un auto se queda sin gasolina, debe dirigirse a la GasStationAgent.
- Los autos intentan encontrar la mejor ruta posible sin quedar atrapados en el tráfico.

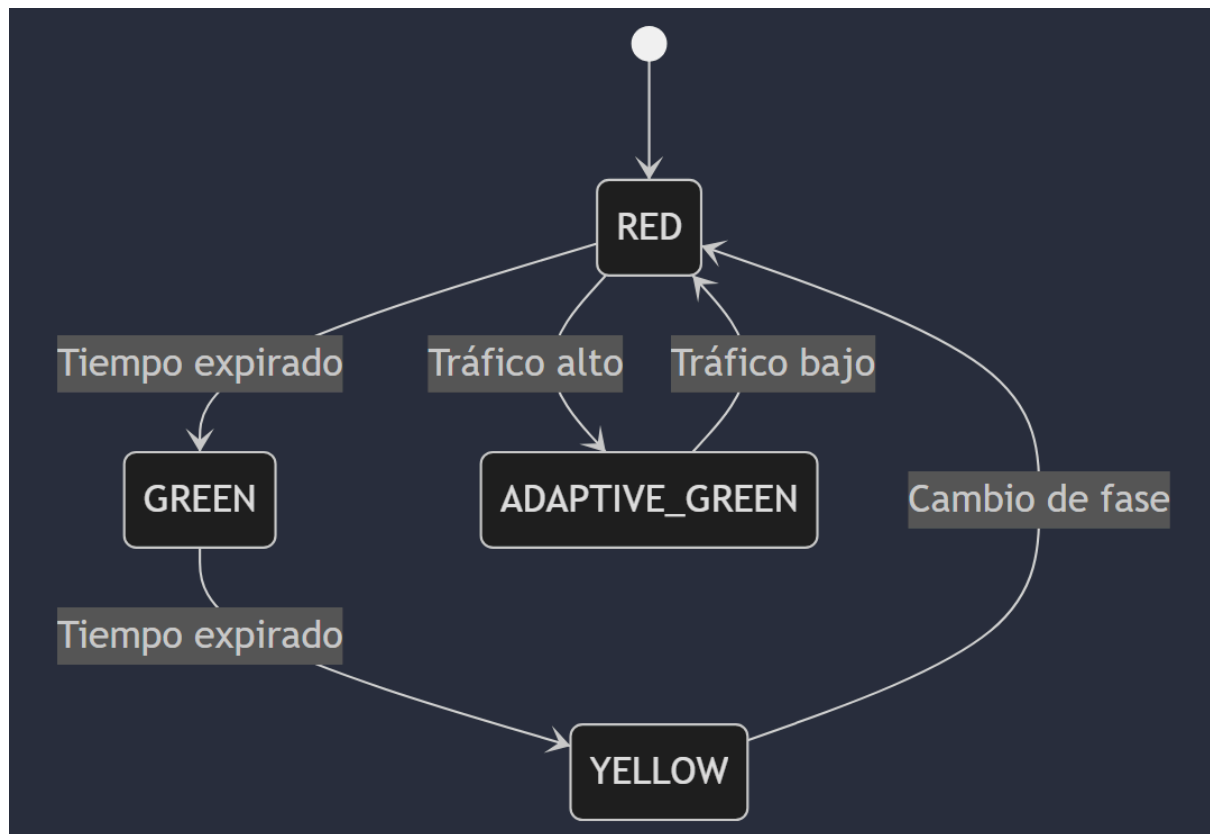
## UML de Clases de Agente



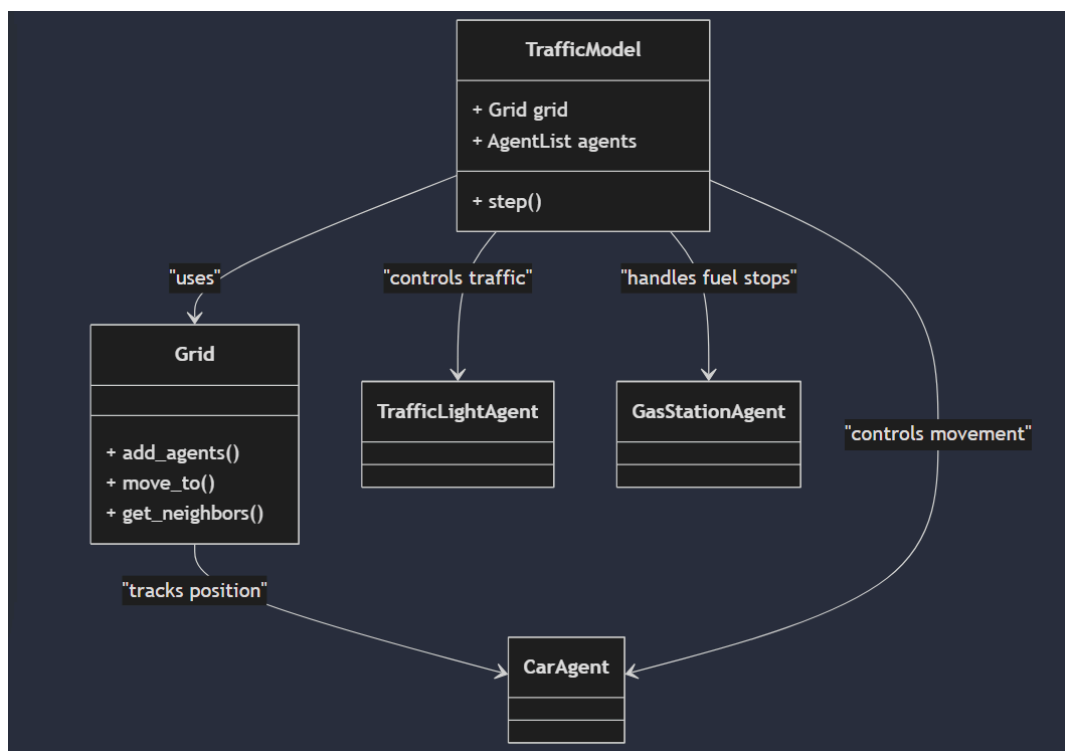
## UML de Protocolos de Interacción



## UML de Estados



## UML de Clases



## Propuesta de Aprendizaje Multiagente



## Objetivo del aprendizaje

El aprendizaje en el sistema multiagente estará centrado en el CarAgent (Agente Vehículo) , específicamente en la decisión de cuándo y dónde recargar combustible. Se busca optimizar el comportamiento de los vehículos para que predigan cuándo deben dirigirse a una estación de servicio antes de quedarse sin gasolina , evitando bloqueos o retrasos en el tráfico.

## Situación de aprendizaje

- Cada CarAgent tiene un nivel de gasolina que disminuye conforme se mueve.
- Si el nivel de gasolina es bajo, el agente debe decidir cuándo ir a la GasStationAgent .
- Inicialmente, los autos pueden esperar hasta que el combustible esté casi agotado antes de reaccionar, pero con el aprendizaje , aprenderán a ir con anticipación para evitar quedarse varados.

## Payoff Matrix

Los autos (CarAgent) deben decidir cuándo ir a una gasolinera basándose en su nivel de combustible y disponibilidad de estaciones.

<b>CarAgent / GasStationAgent</b>	<b>Ir a cargar (Estación libre)</b>	<b>Ir a cargar (Estación ocupada)</b>
Cargar temprano	(-1, 2) (pierde algo de tiempo, pero no se queda sin gasolina)	(-3, -2) (pierde mucho tiempo en espera)
Cargar tarde	(3, -1) (usa más su gasolina, pero llega a tiempo)	(-5, -5) (se queda sin gasolina, el peor caso)

- Si el auto va a cargar gasolina cuando hay estaciones disponibles, es bueno (+2 para la gasolinera, -1 para el auto por la pérdida de tiempo).
- Si el auto espera demasiado y la estación está ocupada, hay un castigo alto (-3 a -5).
- Si logra maximizar el uso de su gasolina sin quedarse sin combustible, obtiene una gran recompensa (+3).
- El aprendizaje buscará la estrategia que minimice el riesgo de quedarse sin gasolina mientras reduce el tiempo perdido en carga.

### **Justificación del Modelo de Aprendizaje (MxN)**

Juego 2x2 → Funciona bien para una decisión simple (ir o no a cargar gasolina).

Juego MxN → Si tenemos varias gasolineras, podríamos ampliar el aprendizaje para que el auto escoja la mejor estación disponible considerando:

- Distancia a cada gasolinera.
- Número de autos en espera en cada estación.
- Nivel de gasolina actual.
- Si hay varias estaciones disponibles, el auto aprenderá a elegir la mejor en cada caso.

### **Mecanismo de Aprendizaje Escogido**

Usaremos Q-Learning (Aprendizaje por Refuerzo) para que los CarAgent aprendan a optimizar su decisión de recarga.

#### **● Estructura del Aprendizaje:**

##### **1. Estados (s)**

- Nivel de gasolina (Bajo, Medio, Alto).
- Disponibilidad de gasolineras (Libre, Ocupada).
- Distancia a la gasolinera (Cerca, Lejos).

## 2. Acciones (a)

- Esperar para recargar.
- Ir a la gasolinera más cercana.
- Buscar otra gasolinera con menos autos.

## 3. Recompensas (R)

- Positiva (+3): Si llega a la gasolinera justo a tiempo y sin congestión.
- Neutra (-1): Si carga demasiado pronto y pierde tiempo.
- Negativa (-5): Si se queda sin gasolina antes de llegar.

## Plan de trabajo

	Tareas	Fecha	Intervalo de esfuerzo
Kevin	<ul style="list-style-type: none"><li>● Implementación de los agentes inteligentes que controlan el tráfico.</li><li>● Programación del comportamiento de los vehículos en función de su entorno.</li><li>● Integración del aprendizaje multiagente para optimizar las decisiones.</li></ul>	10/03/2025	Alta
Sergio	<ul style="list-style-type: none"><li>● Expansión del entorno:</li><li>● Mayor detalle en la ciudad (calles, intersecciones, señalización).</li><li>● Implementación de gasolineras en el</li></ul>	10/03/2025	Media-Alta

	mapa.		
Joselyn	<ul style="list-style-type: none"> <li>• Creación de modelos 3D originales en Maya , incluyendo:</li> <li>• Semáforos</li> <li>• Personas para poblar la ciudad.</li> <li>• Edificio para mejorar la ambientación.</li> <li>• Exportación e integración de los modelos en Unity .</li> </ul>	10/03/2025	Media-Alta