UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería


# Sistema de semáforos inteligentes para la regulación de tránsito vehicular

Trabajo de graduación presentado por José Pablo Kiesling Lange para optar al grado académico de Licenciado en Ingeniería en Ciencias de la Computación y Tecnologías de la Información

Guatemala,

2025

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería


# Sistema de semáforos inteligentes para la regulación de tránsito vehicular

Trabajo de graduación presentado por José Pablo Kiesling Lange para optar al grado académico de Licenciado en Ingeniería en Ciencias de la Computación y Tecnologías de la Información

Guatemala,

2025

Vo.Bo.:

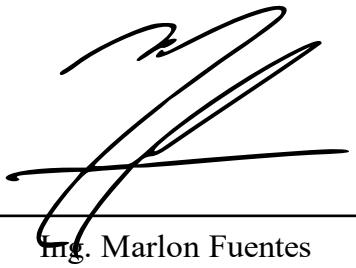
(f) _____

Ing. Alberto Suriano

Tribunal Examinador:

(f) _____

Ing. Alberto Suriano



(f) _____

Ing. Marlon Fuentes

Fecha de aprobación: Guatemala, 22 de noviembre de 2025.

# Índice

# Lista de figuras

# Lista de cuadros

# Lista de siglas

**A2C** Advantage Actor–Critic.

**ANADIE** Agencia Nacional de Alianzas para el Desarrollo de Infraestructura Económica.

**API** Application Programming Interface.

**ASCT** Adaptive Signal Control Technology.

**ATCS** Adaptive Traffic Control System.

**CEUR** Centro de Estudios Urbanos y Regionales.

**CTDE** Centralized Training with Decentralized Execution.

**DRL** Deep Reinforcement Learning.

**GAE** Generalized Advantage Estimation.

**GPU** Graphics Processing Unit.

**IPPO** Independent Proximal Policy Optimization.

**JICA** Japan International Cooperation Agency.

**LADOT** Los Angeles Department of Transportation.

**MAPPO** Multi-Agent Proximal Policy Optimization.

**MARL** Multi-Agent Reinforcement Learning.

**MUTCD** Manual on Uniform Traffic Control Devices.

**OSM** Open Street Map.

**PNUD** Programa de las Naciones Unidas para el Desarrollo.

**PPO** Proximal Policy Optimization.

**RL** Reinforcement Learning.

**SAT** Superintendencia de Administración Tributaria.

**SCATS** Sydney Coordinated Adaptive Traffic System.

**SCOOT** Split Cycle Offset Optimisation Technique.

**SDG** Stochastic Gradient Descent.

**SUMO** Simulation of Urban MObility.

**TraCI** Traffic Control Interface.

**USAC** Universidad de San Carlos de Guatemala.

**VAS** Vía Alterna del Sur.

# Resumen

En movilidad urbana, los Sistemas de Semáforos Inteligentes ajustaron fases en tiempo real para reducir demoras y mejorar la confiabilidad del viaje. Este trabajo implementó un Sistema de Semáforos Inteligentes basado en Aprendizaje por Refuerzo Multiagente usando el algoritmo Multi-Agent Proximal Policy Optimization (MAPPO). Se construyeron cuatro escenarios representativos de la Ciudad de Guatemala (Bulevar Los Próceres, Avenida Elena–Anillo Periférico, Calzada Atanasio Tzul y Avenida Reforma), parametrizados con criterios de ingeniería de tránsito. La política se optimizó con una función de recompensa cuyo objetivo era minimizar acumulación de retraso en el tránsito de la Ciudad de Guatemala.

En los resultados, el desempeño de los modelos entrenados mostraron convergencia estable y mejoras respecto a las situaciones actuales de coordinación vehicular. Dicha mejora se da gracias a que, promediando los cuatro entornos, el tiempo de viaje promedio descendieron en 73.25 %, el tiempo de espera promedio se redujo en 72.52 % y la velocidad promedio aumentó 13.57 %.

En conclusión, el Sistema de Semáforos Inteligentes diseñado con un algoritmo MAPPO interpreta los elementos de cada entorno evaluado teniendo un desempeño óptimo en ellos. Esto es porque capta dependencias de corredor y amortigua la no estacionariedad multiagente, priorizando la disipación de colas sobre picos de velocidad. Esto se traduce en progresiones más largas y menos bloqueos de tránsito, lo cual sugiere su implementación en entornos que posean una cantidad de semáforos y afluencia vehicular adecuadas para el uso del Sistemas de Semáforos Inteligentes en la Ciudad de Guatemala

# Abstract

In urban mobility, Intelligent Traffic Signal Systems adjust phases in real time to reduce delays and improve travel-time reliability. This work implements an Intelligent Traffic Signal System based on Multi-Agent Reinforcement Learning using the MAPPO algorithm. Four representative scenarios of Guatemala City (Bulevar Los Próceres, Avenida Elena–Anillo Periférico, Calzada Atanasio Tzul, and Avenida Reforma) were constructed and parameterized with traffic-engineering criteria. The policy was optimized with a reward function aimed at minimizing the accumulation of delay in the city's traffic.

In the results, the trained models exhibit stable convergence and improvements over current vehicular coordination setups. Specifically, averaging across the four environments, average travel time decreases by 73.25 %, average waiting time decreases by 72.52 %, and average speed increases by 13.57 %.

In conclusion, the Intelligent Traffic Signal System designed with MAPPO achieves strong performance across the evaluated environments because it captures corridor dependencies and mitigates multi-agent non-stationarity, prioritizing the dissipation of queues over peak speeds. This translates into longer green-wave progressions and fewer blockages, suggesting deployment in corridors with sufficient signal density and traffic demand suitable for Intelligent Traffic Signal Systems in Guatemala City.

# CAPÍTULO 1

## Introducción

El desarrollo de soluciones basadas en inteligencia artificial para la regulación del tránsito vehicular ha cobrado gran relevancia, especialmente en ciudades con una cantidad alta de vehículos. Dado esto, los sistemas de semáforos inteligentes han surgido como una alternativa para mejorar la movilidad urbana mediante el ajuste de tiempos en función del flujo vehicular.

Este trabajo propone el diseño e implementación de un simulador que permita evaluar la efectividad de un sistema de semáforos inteligentes en distintos escenarios urbanos. Para ello, se utilizarán modelos basados en datos reales y técnicas de aprendizaje por refuerzo y redes neuronales con el fin de optimizar la regulación del tráfico. A través de la simulación, se analizará el impacto de distintas configuraciones en la reducción de tiempos de espera y la mejora del flujo vehicular.

El proyecto busca generar una herramienta que facilite la toma de decisiones en la gestión del tránsito, permitiendo probar estrategias antes de su implementación en entornos reales. De esta manera, se espera que los resultados obtenidos sirvan como referencia para futuras aplicaciones en la planificación y optimización del tráfico en Guatemala.

# CAPÍTULO 2

## Justificación

El tránsito vehicular en Guatemala representa un desafío debido al aumento constante de vehículos y la falta de sincronización en los semáforos. Según datos de la Superintendencia de Administración Tributaria (SAT) en 2024 [1], el parque vehicular fue de 5,771,508 vehículos, el cual representa un crecimiento de 9 % respecto a la cantidad del año anterior. Este incremento, presenta problemas, ya que en 2024 el tiempo promedio que invertía una persona al día en el tráfico era entre 3 a 4 horas [2].

Es por lo que, según el Programa de las Naciones Unidas para el Desarrollo (PNUD) en 2024 [3] la Municipalidad de Guatemala busca implementar un Sistema de Semáforos Inteligentes como parte del Plan Maestro de Ciudades Inteligentes. Dicho proyecto tiene como fin mejorar la movilidad urbana en las 511 intersecciones semaforizadas, reduciendo así los tiempos de traslado vehicular en un 25 % [4]. Sin embargo, posterior a su instalación, se necesitará un periodo de prueba de 45 días para evaluar el sistema, lo cual puede resultar costoso en caso de que no se obtengan los resultados esperados.

Por consiguiente, este proyecto propone el diseño e implementación de un sistema de semáforos inteligentes en escenarios simulados para ajustar los tiempos de los semáforos. En función de la configuración del escenario, se evaluará la efectividad del sistema, permitiendo determinar la mejor solución global para todos los agentes involucrados y optimizando la regulación del tránsito vehicular. Se medirá el impacto del sistema en métricas clave como el tiempo medio de recorrido por vehículo, el número de vehículos trasladados por unidad de tiempo, el tiempo de espera en cola, y la cantidad de paradas promedio por trayecto.

Asimismo, el proyecto contribuirá a la validación previa de estrategias de movilidad, ayudando a reducir el tiempo de implementación, evitar inversiones innecesarias y proporcionar una herramienta de análisis que maximice el beneficio social, económico y ambiental del nuevo sistema de semáforos.

CAPÍTULO 3

# Objetivos

## 3.1. Objetivo general

Implementar un sistema de semáforos inteligentes basado en aprendizaje por refuerzo multiagente que permita optimizar la regulación del tránsito vehicular en distintos escenarios urbanos simulados y evaluar el impacto sobre el flujo vehicular.

## 3.2. Objetivos específicos

- Implementar entornos configurables que representen el comportamiento de los agentes involucrados en los escenarios de objeto de estudio parametrizando sus características.
- Construir un sistema de coordinación de semáforos inteligentes que optimice el flujo vehicular mediante técnicas de aprendizaje por refuerzo y redes neuronales.
- Evaluar el sistema de coordinación de semáforos inteligentes al compararlo con situaciones de escenarios base.

CAPÍTULO 4

# Marco teórico

## 4.1. Control de tráfico urbano y semaforización

La semaforización es un método de control del derecho de paso en intersecciones que regula los movimientos vehiculares y peatonales mediante asignaciones temporales de prioridad. Su objetivo es ordenar los flujos de tránsito para poder reducir conflictos y mejorar la seguridad. En la práctica, los semáforos combinan elementos físicos y lógicos que se diseñan con base en patrones de tránsito y la forma de la intersección [5].

En términos operativos, el Manual on Uniform Traffic Control Devices (MUTCD) dice que una intersección semaforizada se describe por las fases (conjuntos de movimientos que reciben verde simultáneamente) y los ciclos (tiempo total para completar todas las fases). Estos conceptos son estándar en la ingeniería de tránsito y se emplean para ajustar planes según el día, la hora y condiciones especiales [6].

El Highway Capacity Manual indica que la evaluación del desempeño de una intersección semaforizada se apoya en medidas como demora media por vehículo, nivel de servicio, longitud de cola y confiabilidad del tiempo de viaje. Esto permite ajustar tiempos y coordinar entre intersecciones usando metodologías para evaluar las intersecciones semaforizadas, permitiendo comparar esquemas de diseño bajo distintos escenarios de demanda [7].

### 4.1.1. Esquemas tradicionales

En la práctica se distinguen tres esquemas clásicos. El primero es el control de tiempo fijo, el cual mantiene constantes los tiempos de verde, amarillo y rojo dentro de cada plan horario. Este esquema es apropiado cuando los patrones de demanda son estables [8]

El segundo es el control actuado, en donde se usan detectores que permiten definir las fases según los patrones en el tránsito. Esto permite mejorar la eficiencia en condiciones especiales o en horarios con actividad diferente a la esperada. Para este esquema se describe

la configuración de detectores y los parámetros a usar [9].

El último es la coordinación arterial y de redes, que se caracteriza por sincronizar ciclos y fases para favorecer el transito en sentidos dominantes. En muchos sistemas se gestionan planes por horas o días. Este esquema tiene amplica documentación de diseño, implementación y evaluación de la coordinación y sus efectos en el tiempo de viaje [8].

### 4.1.2. Contexto urbano en Guatemala

La Ciudad de Guatemala tiene una red vial con un patrón radial anular con accesos desde municipios vecinos. El esquema de semaforización que usa es mixto, combinando control actuado y coordinación arterial. Esta estructura y su crecimiento ya era identificada por el Plan Maestro de Transporte Urbano elaborado con cooperación de Japan International Cooperation Agency (JICA) donde se describe la lógica metropolitana, la dependencia de corredores radiales y la necesidad de coordinación semafórica en ejes estratégicos. [10]

Como se puede apreciar en el Cuadro 1 se muestran datos de la Ciudad de Guatemala del Departamento de Movilidad Urbana de la Municipalidad de Guatemala en 2015 analizados por Centro de Estudios Urbanos y Regionales (CEUR)–Universidad de San Carlos de Guatemala (USAC) [11]. Estos accesos comprenden más del 45 % de circulación en el año de estudio, siendo estas rutas las más transitadas en la Ciudad de Guatemala bajo este estudio. Además, hay antecedentes que justifican que la experimentación y evaluación del los datos presentados se centre en intersecciones de dichos corredores, donde el potencial de reducción de demora y colas es más relevante [12]. Estos resultados se alinean con datos recolectados por Agencia Nacional de Alianzas para el Desarrollo de Infraestructura Económica (ANADIE) en 2024, los cuales se puede apreciar en la Figura 1.

Cuadro 1: Circulación diaria y porcentaje del total en la Ciudad de Guatemala (2015).

| Acceso | Circulación 2015 (unidades diarias) | Porcentaje del total en Ciudad de Guatemala (%) |
|---|---|---|
| Calzada Roosevelt | 258,806 | 12.5 |
| Calzada Raúl Aguilar Batres | 157,677 | 7.6 |
| Calzada de San Juan | 128,342 | 6.2 |
| Calle Martí–José Milla y Vidaurre | 92,629 | 4.5 |
| Bulevar Los Próceres | 80,856 | 3.9 |
| Bulevar El Naranjo | 78,019 | 3.8 |
| Avenida de Petapa | 64,548 | 3.1 |
| Avenida Hincapié | 49,960 | 2.4 |
| Bulevar La Pedrera | 21,278 | 1.3 |
| La Vía Alterna del Sur (VAS) | – | – |
| **Totales** | **932,115** | **45.3** |

### 4.1.3. Sistema de semáforos inteligentes

Un sistema de semáforos inteligentes se define como un conjunto de control semafórico capaz de medir el estado del tránsito en tiempo real, analizar su desempeño y ajustar los parámetros descritos anteriormente de según patrones. Este enfoque se conoce como Adap-

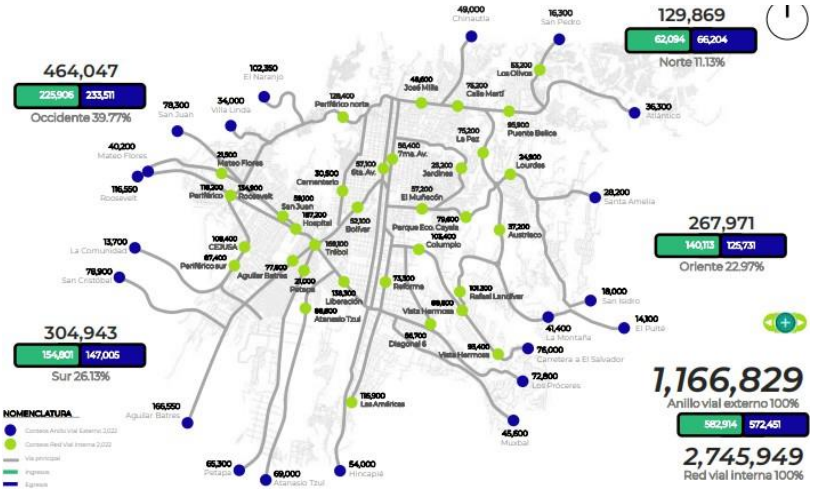



Figura 1: Distribución de vehículos en la Ciudad de Guatemala en 2024 [13]

tive Signal Control Technology (ASCT) y su objetivo es reducir demoras y evitar planes fijos desactualizados [14].

La forma en que operan los ASCT es ciclico donde los sensores recogen datos, el sistema evalúa patrones de flujo y aplica nuevos parámetros de temporización. En la práctica, existen plataformas consolidadas como Split Cycle Offset Optimisation Technique (SCOOT) y Sydney Coordinated Adaptive Traffic System (SCATS) , las cuales buscan ajustar de manera continua los parámetros con base en mediciones de flujo, colas y ocupación [15].

Los sistemas de semáforos inteligentes ASCT permiten que se pueda resolver usando Aprendizaje por Refuerzo, ya que estos sistemas ajustan las fases y los ciclos con base en patrones en tiempo real. Es por ello que, según estudios, se pueden usar datos para optimizar el sistema de semaforización al presentar mejoras frente a esquemas tradicionales. Además, la evidencia de campo con estos sistemas muestran mejoras, lo que justifica evaluar esta variante en contextos como la Ciudad de Guatemala [16]

## 4.2. Reinforcement Learning

El Aprendizaje por Refuerzo (Reinforcement Learning (RL)) es un tipo de Aprendizaje Automático donde el agente aprende por interacción. Como se muestra en la Figura 2, en cada paso de tiempo $t$, el agente observa un estado $s_t$ y selecciona una acción $a_t$ según una política $\pi(a \mid s)$; el entorno le entrega una recompensa $r_t$ y el agente transita a un nuevo estado $s_{t+1}$. Dicha política $\pi$ define la acción $a_t$ que debe tomar el agente en un estado $s_t$ en el tiempo $t$, y el objetivo principal de RL es encontrar la política óptima $\pi^*$ que maximiza la recompensa acumulada a lo largo del tiempo [17].

Para cumplir dicho objetivo, se usan las funciones valor $V_\pi$ o $Q_\pi$, las cuales estiman la recompensa esperada a largo plazo desde un estado determinado $s_t$ para $V_\pi$ y, además, tomando una acción determinada $a_t$ para $Q_\pi$. Estas funciones valor $V_\pi$ o $Q_\pi$ pueden repre-

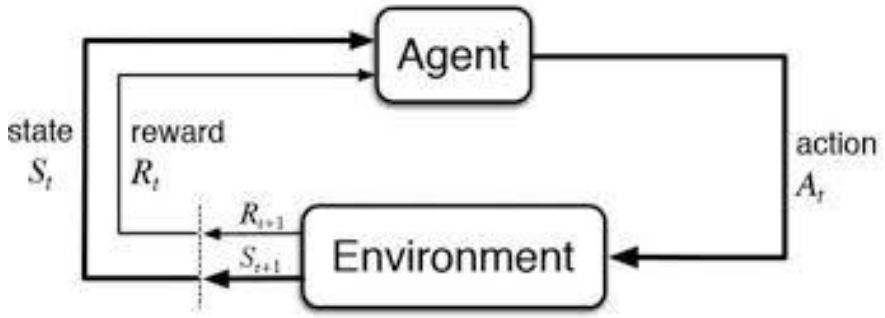



Figura 2: Ciclo de Aprendizaje por Refuerzo de un agente con el ambiente.
[18]

sentarse en forma de tabla cuando los espacios de estados $S$ y acciones $A$ son reducidos, o aproximarse cuando estas cantidades incrementan. Sin embargo, en ambientes aún más grandes y con espacios continuos, es preferible usar una aproximación de la política, donde esta se parametriza como $\pi_\vartheta(a \mid s)$, donde $\vartheta$ son dichos parámetros [19].

### 4.2.1. Aprendizaje por Refuerzo Multiagente

El Aprendizaje por Refuerzo Multiagente (Multi-Agent Reinforcement Learning (MARL)) busca resolver un sistema multiagente que, como se aprecia en la Figura 3, se compone de un entorno con diversos agentes que interactúan con él para lograr objetivos específicos. El objetivo principal de este tipo de RL es aprender la política óptima $\pi^*$ que maximiza la recompensa acumulada global. En el ciclo de entrenamiento, cada agente toma acciones individuales para generar una acción conjunta que cambia el estado del entorno. Los agentes reciben recompensas individuales y el nuevo estado del entorno tras la acción conjunta realizada. Los datos generados en cada paso de tiempo $t$ permiten la mejora de la política $\pi$ de cada agente [20].

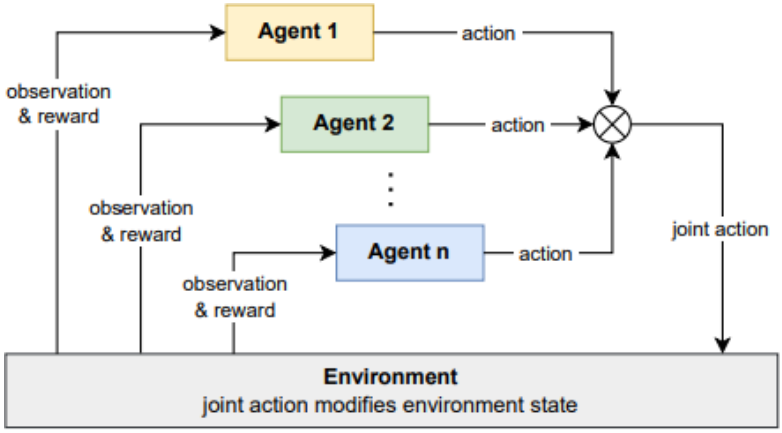



Figura 3: Ciclo de Aprendizaje por Refuerzo Multiagente con el ambiente.
[20]

### 4.2.2. Aprendizaje por Refuerzo Profundo

Como se mencionó anteriormente, en ambientes con espacios de estados $S$ y acciones $A$ grandes, se usan métodos de aproximación de políticas donde la política se parametriza como $\pi_\vartheta(a \mid s)$. Para optimizar la recompensa esperada, se deben hallar los parámetros $\vartheta$ que generen el mayor retorno a lo largo del tiempo. Esto se logra usando algoritmos de gradiente de política y, en Aprendizaje por Refuerzo Profundo (Deep Reinforcement Learning (DRL)), estos algoritmos se potencian mediante el uso de redes neuronales como aproximadores [21].

Entre los métodos de gradiente de política se encuentran REINFORCE y Actor–Critic. El primero se basa en métodos de Monte Carlo (sin requerir conocer el modelo del entorno, pero necesitando episodios completos) para estimar el retorno esperado. Por otro lado, el segundo tiene un actor que ajusta la política parametrizada $\pi_\vartheta(a \mid s)$ y un crítico que ajusta una función valor aproximada $\hat{V}_\pi$, estimada mediante *Temporal Difference* (sin necesidad de conocer el modelo del entorno y realizando *bootstrapping*) [19].

Entre los métodos de Actor–Critic, algunos de los más utilizados son el Actor–Critic con ventaja (Advantage Actor–Critic (A2C)) y la Optimización de Políticas Proximales (Proximal Policy Optimization (PPO)). Al poner estos algoritmos en práctica en un sistema de RL con un solo agente se obtuvieron los resultados que se muestran en la Figura 4. Como se aprecia, en el caso de REINFORCE la solución se alcanza en los episodios finales con mayor variación, mientras que los métodos Actor–Critic logran un rendimiento cercano al óptimo al converger a recompensas altas con menos pasos y menor varianza. Finalmente, en esta comparación práctica se observa que PPO tiende a aprender antes la solución que A2C [20].

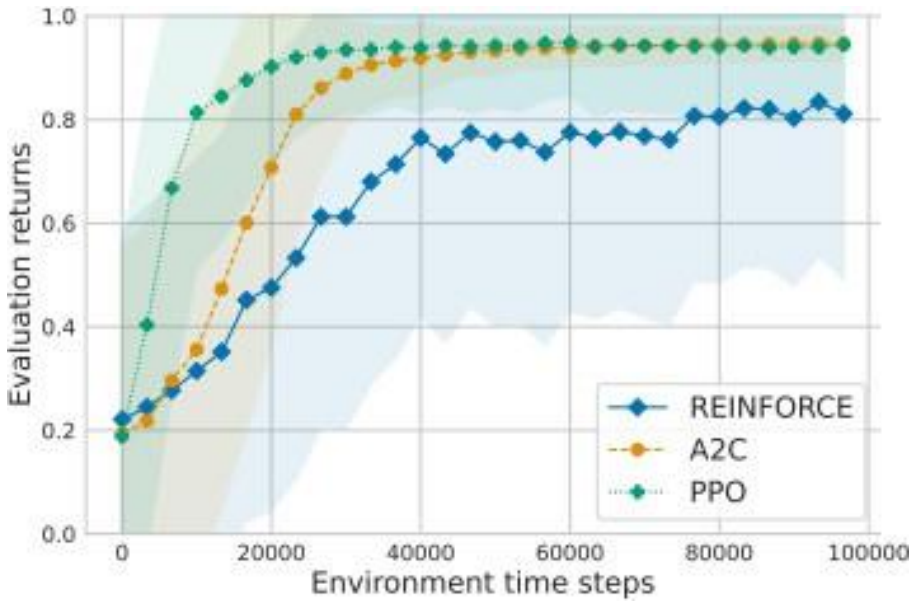



Figura 4: Curvas de aprendizaje para REINFORCE, A2C y PPO en un experimento RL con un solo agente con 100000 pasos.

[20]

### 4.2.3. Optimización de políticas proximales en Aprendizaje por Refuerzo Multiagente

Como se mencionó, PPO se apoya en el paradigma Actor-Critic y en el principio de mantener las actualizaciones de la política dentro de una región de confianza para evitar degradaciones bruscas del desempeño. La Figura 5 muestra esta arquitectura a alto nivel: un *actor* parametriza la política $\pi_\vartheta(a \mid s)$ y un *crítico* aproxima el valor $V_\phi(s)$. En PPO se optimiza un objetivo sustituto que limita la magnitud del cambio de probabilidad entre la política antigua $\pi_{\vartheta_{\text{old}}}$ y la actual $\pi_\vartheta$ mediante un recorte (clipping) [20]. Para una trayectoria $\tau$ y ventajas $\hat{A}_t$, el objetivo por paso $t$ es

$$L^{\text{CLIP}}(\vartheta) = \mathbb{E}_t\left[\min\left(r_t(\vartheta)\hat{A}_t,\ \text{clip}(r_t(\vartheta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right], \qquad r_t(\vartheta) := \frac{\pi_\vartheta(a_t \mid s_t)}{\pi_{\vartheta_{\text{old}}}(a_t \mid s_t)}.$$

El término de recorte con $\epsilon \in (0, 1)$ restringe la actualización efectiva, evitando pasos de política demasiado altos. El objetivo total añade una penalización de error de valor y una bonificación:

$$L^{\text{PPO}}(\vartheta, \phi) = \mathbb{E}_t\left[L^{\text{CLIP}}(\vartheta) - c_v\left(V_\phi(s_t) - \hat{V}_t\right)^2 + c_e\, \mathrm{H}\left[\pi_\vartheta(\cdot \mid s_t)\right]\right],$$

donde $c_v, c_e > 0$ ponderan la pérdida de valor y el término de entropía H. Para estimar ventajas de baja varianza se utiliza *Generalized Advantage Estimation (GAE)* (GAE-$\lambda$):

$$\hat{A}_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}, \qquad \delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t),$$

con $\gamma \in (0, 1)$ y $\lambda \in [0, 1]$. Es por ello, que PPO itera en ciclos de recolección de trayectorias completas con la política $\pi_{\vartheta_{\text{old}}}$, procesamiento de $\hat{A}_t$ (GAE) y con múltiples épocas de optimización del objetivo recortado sobre mini-batches con el fin de lograr una actualización de $\vartheta_{\text{old}} \leftarrow \vartheta$ [20].

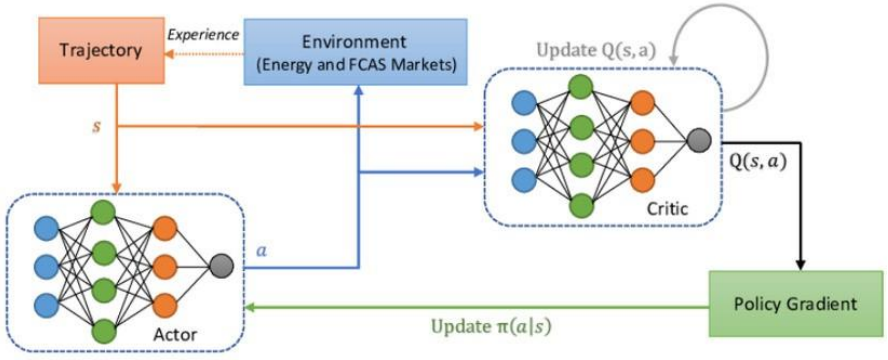



Figura 5: Arquitectura PPO con DRL
[22]

Por lo tanto, según Albrecht (2024) [20] los componentes clave de PPO son los siguientes:

- **Objetivo recortado ($\epsilon$).** Limita el cambio relativo de probabilidad $r_t(\vartheta)$ para actualizaciones conservadoras.

- **GAE-$\lambda$.** Estimador de ventajas con buen sesgo–varianza, controlado por $\lambda$.
- **Crítico $V_\phi$ y pérdida de valor.** Ancla temporal (*bootstrapping*) para reducir varianza.
- **Bonificación entrópica.** Mantiene exploración y evita colapso prematuro de la política.
- **Optimización por épocas y mini-batches.** Mejora eficiencia del uso de datos recolectados.

En el contexto de MARL, hay dos variantes de PPO que pueden ser usadas: Optimización de políticas proximales independientes (Independent Proximal Policy Optimization (IPPO)) y Optimización de políticas proximales multiagente (MAPPO). En el primero, cada agente es independiente, ya que tienen su propia política y no se comunican entre sí mientras aprenden y ejecutan. El problema es que, como se mencionó previamente, el objetivo principal de MARL es maximizar la recompensa acumulada global, pero al ser todos los agentes independientes cada uno solo maximiza su recompensa acumulada [23]. Por lo tanto, bajo el problema presentado, se tiene mayor ventaja si se usa MAPPO

### 4.2.4. Optimización de políticas proximales multiagente

Como se dijo, MAPPO extiende PPO al ajuste cooperativo multiagente bajo el esquema Centralized Training with Decentralized Execution (CTDE): durante el entrenamiento, los agentes comparten información para entrenar un *crítico centralizado* (que observa el estado/observaciones conjuntas y, opcionalmente, las acciones conjuntas), pero durante la ejecución cada agente actúa de forma descentralizada con su política local. La Figura 6 ilustra esta idea, con múltiples actores (uno por agente o compartido por grupos homogéneos) se entrenan con un único (o compartido) crítico que recibe información global.

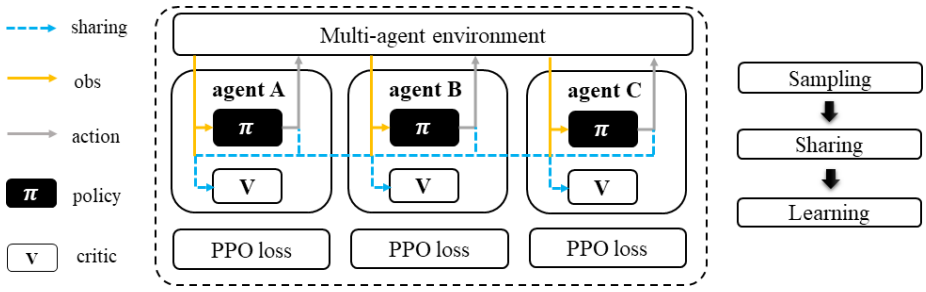



Figura 6: Arquitectura MAPPO
[22]

Bajo el problema del presente trabajo, los beneficios prácticos en redes urbanas incluyen: mejor coordinación entre intersecciones adyacentes, estabilidad de entrenamientoal amortiguar el ruido de recompensa con un crítico global y robustez ante patrones cambiantes al aprender representaciones que integran la dinámica de la red completa [23].

Según Amato (2024) [23], la forma en la que opera MAPPO incluye las siguientes fases:

1. *Recolección de datos.* Cada agente *i* interactúa con su entorno local siguiendo la política $\pi_{\vartheta^{(i)}}$. Se registra además un resumen global $g_t$ disponible solo en entrenamiento.
2. *Cálculo de ventajas centralizadas.* El crítico centralizado $V_\phi(g_t)$ evalúa el retorno esperado con él se evalúan ventajas $\hat{A}_t^{(i)}$ usando GAE-$\lambda$ para cada agente.
3. *Actualización PPO por agente.* Para cada política $\pi_{\vartheta^{(i)}}$ se maximiza el objetivo recortado usando las ventajas centralizadas $\hat{A}_t^{(i)}$. El crítico se entrena minimizando $V_\phi(g_t) - \hat{V}_t^{\ 2}$.
4. *Ejecución descentralizada.* En despliegue, cada agente usa solo su observación local $o_t^{(i)}$ para actuar sin necesitar $g_t$.

Por consiguiente, en un esquema de tránsito $g_t$ puede incluir colas por acceso y offsets entre intersecciones . Los actores descentralizados ajustan fase verde local, mientras el crítico centralizado aprende a valorar configuraciones que favorecen ondas verdes y evitan que se acumulen vehículos. Esto concreta el beneficio de MAPPO en contextos donde la coordinación es esencial para reducir demora promedio y longitud de cola.

# CAPÍTULO 5

# Metodología

Para poder llevar a cabo este trabajo de investigación, se realizaron seis etapas. Dichas etapas comprenden partes de investigación, experimentación y evaluación:

1. Selección de herramientas
2. Selección de escenarios
3. Construcción de entornos
4. Selección de hiperparámetros del modelo
5. Entrenamiento del modelo
6. Evaluación del modelo

Cada una de estas etapas se hizo tomando en cuenta el marco teórico previo, en el que se hace énfasis sobre los diferentes tipos de algoritmos para resolver problemas de RL. Como se presentó, se evidencia que el algoritmo de MAPPO es el adecuado para resolver el problema planteado en este trabajo, por lo que parte de la metodología planteada, se basa en el uso del mismo.

## 5.1. Selección de herramientas

En esta etapa se tuvo que investigar sobre las herramientas actuales para poder modelar y simular los entornos de tráfico de la Ciudad de Guatemala. Además, se hizo la investigación acerca de las herramientas que permitieran diseñar un modelo de RL que contemplara los elementos de tráfico urbano para lograr los objetivos planteados.

### 5.1.1. Herramienta de Modelación y Simulación

La herramienta que se usó para modelar y simular el tráfico urbano de la Ciudad de Guatemala fue Simulation of Urban MObility (SUMO). Esta herramienta open source permite simular entornos reales con vehículos individuales y semáforos. SUMO ofrece features para construir redes parametrizadas por medio de la generación y asignación de demanda, y recolectar datos como flujos, tiempos de viaje, colas, entre otros [24].

Dicha construcción de escenarios realistas es posible porque SUMO importa redes desde OSM, además, cuenta con cuenta con asistentes y herramientas para depurar geometrías, prioridades y semáforos. Esta cadena de herramientas facilita el proceso para crear intersecciones y zonas reales de la Ciudad de Guatemala [25].

Para este trabajo, un requisito clave es interactuar en tiempo real con la simulación para ver el desempeño de la política entrenada. SUMO expone Traffic Control Interface (TraCI), una interfaz cliente-servidor que permite durante la ejecución leer variables como colas y tiempos de espera como se puede apreciar en la Figura 7. Asimismo, permite actuar sobre el entorno como cambiar fases de semáforos. Además, SUMO mantiene un módulo específico para semáforos con funciones de consulta y modificación de estados, lo cual fue esencial para implementar y evaluar controladores basados en RL como MAPPO [26].


Figura 7: Ejecución de simulación con SUMO exponiendo variables en tiempo real.

Como alternativas consideradas, se encontró con la herramienta AnyLogic. Sin embargo, no resultó viable para este trabajo por razones de licencia y mantenimiento. AnyLogic no es open source y se distribuye bajo ediciones comerciales. Su edición gratuita impone límites que afectan directamente la investigación con RL (por ejemplo limita a <7 variables y <500 iteraciones).A su vez, limita el tiempo de simulación a 5 horas y su edición "University Researcher" es de pago, lo que dificulta la reproducibilidad y el despliegue abierto de resultados [27].

Asimismo, se encontró con la herramienta Flow [28], la cual no ha tenido releases desde 2019, lo que sugiere falta de mantenimiento y riesgo de incompatibilidades con versiones

actuales de librerías de RL.

Es por ello que SUMO se tomó como herramienta de modelación y simulación ya que es una plataforma integral para investigación en gestión de tráfico y control de intersecciones. Las capacidades descritas lo hacen adecuado para evaluar políticas de control en escenarios realistas como los de la Ciudad de Guatemala

### 5.1.2. Herramienta de entrenamiento de modelo

Para entrenar el conjunto de agentes representados en este trabajo como semáforos, se eligió Ray RLlib que es la librería de RL distribuida de Ray. RLlib provee implementaciones productivas de algoritmos que incluye PPO y una Application Programming Interface (API) multiagente con un modelo de ejecución. La documentación oficial detalla PPO en RLlib y su configuración declarativa mediante bloques que permite el uso de PPO como multiagente [29].

La API mencionada anteriormente está basada en RLModule (modelo/política entrenable) y EnvRunner (trabajador que interactúa con el entorno), lo que permite separar el entrenamiento del módulo de política y la interacción con el entorno. Esto simplifica la integración con wrappers personalizados, incluido el que provee SUMO conocido como SUMO-RL. Esta framework expone el control semafórico de SUMO como un entorno. Además, RLlib se integra de forma directa con entornos Gymnasium, lo cual permite acoplarse con SUMO-RL con mayor facilidad [30].

Por lo que, en conjunto, se usó RLlib para el entrenamiento multiagente, SUMO-RL como conexión con SUMO y Gymnasium como interfaz estándar de entornos.

## 5.2. Selección de escenarios

Para los escenarios se planeaba ver el desempeño del modelo de RL en las rutas presentadas en el Cuadro 1. Sin embargo, se encontró que los beneficios de la coordinación semafórica se maximizan en rutas con intersecciones próximas y cuando hay varios cruces en secuencia y semaforización. La razón es porque así se puede tener control sobre los vehículos entre semáforos con pérdidas mínimas [31]. Además, en evaluaciones en Adaptive Traffic Control System (ATCS) son más efectivas en rutas urbanas con múltiples señales cercanas y variabilidad de demanda, mientras que su ventaja disminuye cuando las intersecciones están muy separadas o existen tramos largos sin control a nivel [32]

Por lo tanto, un factor a considerar es el criterio de espaciamiento. Según Los Angeles Department of Transportation (LADOT), se recomienda coordinar señales cuando la separación entre intersecciones es menor a 800 metros, porque a esa escala todavía la semaforización es efectiva. Este espaciamiento se toma como condición para coordinar o incluso para no introducir nuevas señales basándose en la separación para una progresión eficiente [33].

Tomando lo anterior en cuenta, las rutas que se tomarán en cuenta serán aquellas que

presenten una densidad de semaforización que sea significativa para la aplicación de un modelo de RL que gestione el transito vehicular y que además hayan sido tomadas en cuenta en el plan de Sistema de Semáforos Inteligentes de la Municipalidad de Guatemala. Las rutas que se seleccionaron para este trabajo son las siguientes:

- Bulevar Los Próceres
- Avenida Elena - Anillo Periférico
- Calzada Atanasio Tzul
- Avenida Reforma

## 5.3. Construcción de entornos

Para la construcción de entornos, se usó el OSM Web Wizard que es una interfaz web incluida en SUMO que automatiza la creación de un escenario a partir de OSM. Como se puede apreciar en la Figura 8 esta herramienta permite localizar un área en el mapa, y colocar ciertos parámetros para construir un entorno. Para el entrenamiento, se configuró 1 hora de duración del entorno y con las opciones mostradas en la parte inferior de la misma imagen, solo se seleccionó Çar-only networkçon el fin de que tuviera mejor rendimiento mientras se entrenaba.

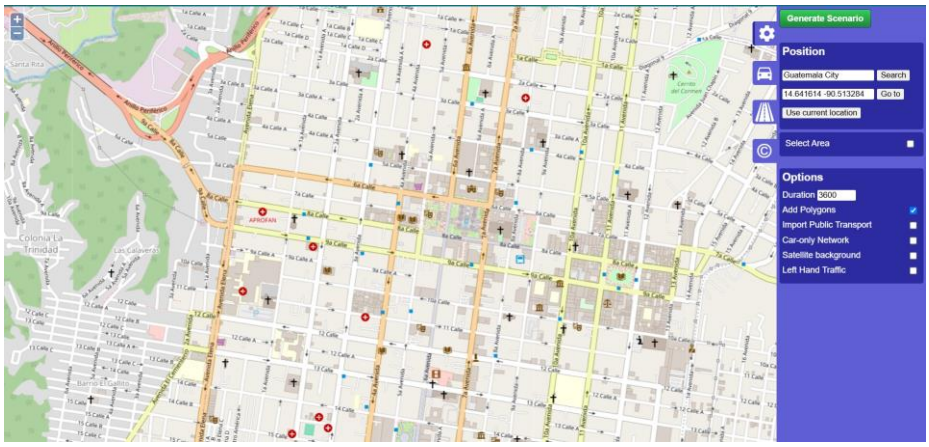



Figura 8: Pantalla principal de OSM Web Wizard

Asimismo, hay otros parámetros que se pueden configurar, los cuales se aprecian en la Figura 9. Dado que el trabajo se enfoca en los vehículos, solo se seleccionó esta opción. En el caso de las variables de este tipo de transporte, el Through Traffic Factor indica el porcentaje de tráfico que atraviesa el área y Count representa la cantidad de vehículos. Los datos ingresados en esta parte corresponden a los que se muestran en la Figura 1.

Con los entornos ya creados, se puede hacer la integración de ellos con el wrapper de SUMO previamente mencionado. Para ello, se hizo un archivo que se encarga de que el entorno sea compatible para entrenarse usando Ray. Dicho archivo tiene 3 responsabilidades:

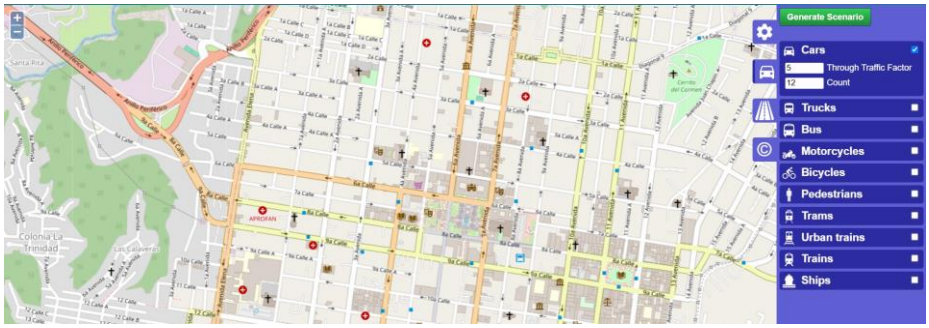



Figura 9: Configuración de generación de vehículos

1. **Parseo del ambiente**: Dado el entorno, se extraen la red y rutas del entorno. Esto se hace con referencia para que se tenga un mapeo entre cada elemento extraído y su representación en el entorno.
2. **Adaptación a SUMO-RL**: El wrapper filtra los argumentos necesarios para ajustarse a la firma real de SumoEnvironment que tiene el entorno de semáforos como MultiAgentEnv.
3. **Unificación de espacios**:
   - **Observación**. Se convierte cada observación en un vector para que se pueda trabajar.
   - **Acción**. Se normaliza la acción por agente al número de fases verdes reales y se evita un cambio de fase no permitido.

## 5.4. Selección de hiperparámetros del modelo

Una vez se tenían los escenarios definidos, se puede entrenar los modelos de RL para cada uno. No obstante, la integración con el wrapper mencionado requería parámetros para su uso. Estos parámetros son circunstanciales para poder entrenar un modelo óptimo y que la solución sea la mejor. Para ello, la selección de hiperparámetros se dividió en dos grupos: fijos fundamentados por otros estudios y buenas prácticas en MAPPO y el segundo grupo son valores sensibles que se exploraron para hacer un grid search.

Para el primer grupo indicado, se divividió nuevamente en dos subgrupos: los parámetros del entorno que sus valores pueden establecidos por regulaciones de tránsito o definidos por la librería, presentados en el Cuadro 2 y los demostrados para algoritmos MAPPO por previos estudios mostrados en el Cuadro 3.

En el primer caso, muchos de los parámetros son basados en regulaciones internacionales indicados por instituciones como el MUTCD y los otros parámetros del entorno son necesarios para su uso con el wrapper. Dichos parámetros son los siguientes:

- **Tiempo amarillo**: La MUTCD establece que el intervalo amarillo debe estar entre 3 y 6 s, dependiendo de velocidad y otras condiciones [34].

- **Tiempo mínimo verde**: El manual MUTCD indica que los valores mínimos típicos están entre 5 segundos y 15 segundos 5–15 [35].
- **Tiempo máximo verde**: El mismo manual afirma que longitudes de ciclo urbanas comunes son entre 60 segundos y 100 segundos. Además, se sugiere que el máximo verde no exceda en una razón de entre 1.25–1.5 respecto al el verde calculado [35].

Cuadro 2: Parámetros del entorno

| Parámetro | Valor |
|---|---|
| Tiempo amarillo | 3 segundos |
| Tiempo mínimo verde | 10 segundos |
| Tiempo máximo verde | 60 segundos |
| Semilla | 327 |
| Tiempo simulación | 1 hora |
| Frecuencia de decisión del agente | 1 segundo |

Para los parámetros del algoritmo MAPPO se definieron según estudios realizados. Estos parámetros si bien tienen relevancia en el desempeño de los modelos, no son los más circunstanciales, por lo cual se definieron como se menciona en el Cuadro 3. A continuación se explica la razón de su valor:

- **Gamma**: 0.99 es el valor estándar de PPO para tareas continuas con horizonte medio a largo. Esto permite conservar suficiente memoria del futuro sin volver inestable el aprendizaje [36].
- **Value function clipping**: En RLlib, este valor se usa para evitar actualizaciones desmesuradas del crítico. El valor por defecto es 10.0, y es el que suele recomendarse salvo que tengas señales de valor extremadamente grandes o pequeñas [29].
- **Tamaño de lote de entrenamiento**: PPO suele necesitar lotes relativamente grandes para estimar gradientes con baja varianza. Como guía pública, OpenAI Spinning Up usa 4,000 pasos por época en PPO vanilla [36].
- **Longitud de trozos de muestreo**: En RLlib, la longitud de trozos de muestreo tiene que ser coherente con el tamaño de lote de entrenamiento en una relación 100:1. Esto permite que haya suficiente horizonte para GAE/advantage estable, actualizaciones frecuentes y hace más sencillo cuadrar el tamaño de lote global [29].
- **Tamaño de Stochastic Gradient Descent (SDG) de minibatch**: PPO entrena varias épocas sobre el lote de entrenamiento partiéndolo en minibatches. Valores entre 64–256 son comunes: 256 aprovecha mejor Graphics Processing Unit (GPU) y reduce el ruido del gradiente sin perder demasiada diversidad por lote [29].
- **Número de iteraciones**: en RLlib, el número de iteraciones se calcula como la razón entre el número de pasos objetivo y el tamaño de lote de entrenamiento. Según

documentaçión, el número de pasos en PPO es 200,000, y al tener el valor de 4,000 de tamaño de lote de entrenamiento, se puede saber que se tendrá que hacer 50 iteraciones [37].

- **Número de épocas**: Es el valor por defecto en implementaciones maduras de PPO y funciona bien en la práctica. Por ejemplo, Stable-Baselines3 documenta 10 épocas [37].

Cuadro 3: Parámetros del algoritmo MAPPO según estudios

| Parámetro | Valor |
|---|---|
| Gamma | 0.99 |
| Value function clipping | 10.0 |
| Tamaño de lote de entrenamiento | 4,000 |
| Longitud de trozos de muestreo | 40 |
| Tamaño de SDG de minibatch | 256 |
| Número de iteraciones | 50 |
| Número de épocas | 10 |

Para el caso de los parámetros que se iban a definir por medio de Grid Search, se investigó los dos valores más óptimos para probar según docmmentación o estudios realizados. Dichos parámetros y sus valores probados se pueden apreciar en el Cuadro 4. Dichos parámetros son los siguientes:

- **Learning rate**: Según el paper de Schulman de PPO, el valor habitual es 3e-4 y 1e-4 añade estabilidad [38].
- **Clip**: 0.2 es el valor del paper y 0.1 hace el update más conservador si hay inestabilidad. [38].
- **GAE**: 0.95 es el valor por defecto en PPO y 0.98 reduce sesgo si la señal de valor es estable.[38].
- **Entropía**: Coeficiente de exploración comúnmente del orden 1e-2 o 5e-3 en trabajos de control continuo [38].

Por último, se tuvo que elegir la función de recompensa. La librería de sumo-rl presenta cuatro posibles funciones:

- **diff-waiting-time**: Cambio en el retraso acumulado de los vehículos entre dos pasos de simulación. Se premia cuando el retraso acumulado baja y castiga cuando sube. Es la recompensa por defecto en SUMO-RL [39].

Cuadro 4: Parámetros y sus valores probados en Grid Search

| Parámetro | Posible primer valor | Posible segundo valor |
|---|---|---|
| Learning rate | 3e-4 | 1e-4 |
| Clip | 0.2 | 0.1 |
| GAE | 0.95 | 0.98 |
| Entropía | 1e-2 | 5e-3 |

- **queue**: número total de vehículos detenidos en cola en los carriles que llegan al semáforo. Minimizarla reduce colas visibles y bloqueo de intersecciones [39].
- **average-speed**: Velocidad media de los vehículos en los carriles relevantes, maximizarla incentiva mantener el flujo optimizado[39].
- **pressure**: vehículos que se aproximan - vehículos que salen. Minimizar presión equilibra colas y es famosa por su buen desempeño y estabilidad a nivel de red.[39].

Dado los objetivos planteados en este trabajo de investigación, la función que tiene más relación con ellos es mejorar el cambio en el retraso acumulado de los vehículos entre dos pasos de simulación y minimizar el tiempo de espera total. Por consiguiente, la función será diff-waiting-time.

## 5.5. Entrenamiento del modelo

Una vez se tuvo los hiperparámetros por Grid Search y los previamente definidos, el entrenamiento se realizó, como se mencionó anteriormente, con MAPPO compartiendo política entre semáforos. Cada iteración consistió en recolectar trayectorias desde SUMO bajo la política actual, calcular ventajas con GAE y optimizar PPO durante varias épocas sobre el lote de entrenamiento. Se registraron pérdidas de la política, del valor y del total así como métricas de entorno y checkpoints periódicos.

Para acelerar el muestreo se habilitó paralelismo y múltiples entornos por runner. Se controló el tiempo de timeout de muestreo para evitar bloqueos cuando la demanda provocaba condiciones extremas de congestión.

En el entrenamiento final, se usó la duración del episodio para capturar efectos transitorios y estacionarios, y se utilizó early stopping empírico (monitoreo de retorno y estabilidad de métricas) para evitar sobre-entrenamiento.

## 5.6. Evaluación del modelo

En la evaluación se comparó la política aprendida contra líneas base (ciclos fijos y control actuado simple), manteniendo iguales escenarios y semillas. Para ello, se hicieron tres tipos de evaluaciones.

La primera fue a nivel visual. Esta es más intepretativa a nivel personal ya que permite ver el comportamiento de los semáforos una vez se les aplicó el modelo de RL. Esto por sí solo no indica ninguna métrica o valor medible, pero se puede apreciar la mejora a simple vista al comparar los mismos escenarios con la presencia o ausencia de un sistema de semáforos inteligentes.

La segunda es a nivel de métricas puntuales. Las métricas que se midieron fueron el tiempo medio de espera por vehículo, la velocidad promedio (m/s) y tiempo medio de viaje. Estas se pueden presentar individualmente y solo su valor para comparar rendimientos.

Finalmente, la última también fueron métricas, pero estas son presentadas en gráficas. Estas son referentes al desempeño del modelo y la recompensa que obtuvo mientras entrenaba y finalmente, gráficas comparativas del desempeño de todos los modelos con el fin de analizar su comportamiento en diferentes entornos.

CAPÍTULO 6

# Resultados

## 6.1. Hiperparámetros por Grid Search

Como se puede apreciar en el Cuadro 5, solo uno de los cuatro hiperparámetros mantuvo su valor por defecto (Learning rate), es decir, según la literatura o documentación de la librería, se decía que el valor por defecto tenía grandes resultados. Sin embargo, con el Grid Search se halló que tres de ellos: Clip, GAE y Entropía dieron un mejor resultado en la función de recompensa con los valores mostrados en dicho Cuadro.

Cuadro 5: Valores de los parámetros hallados por Grid Search

| Parámetro | Valor |
|---|---|
| Learning rate | 3e-4 |
| Clip | 0.1 |
| GAE | 0.98 |
| Entropía | 5e-3 |

## 6.2. Rendimiento de Sistema de semáforos inteligentes

### 6.2.1. Bulevar Los Próceres

Para el ambiente de Bulevar Los Próceres, se puede apreciar su representación se observa en la Figura 10, y el desempeño del modelo de dicho ambiente en la Figura 11. Este modelo tuvo una dinámica de aprendizaje limpia, ya que tras la exploración, su aprendizaje asciende hasta estabilizarse tal y como se vio en la Figura 4 con PPO. Bajo el esquema CTDE, esto sugiere que el crítico centralizado logró capturar los patrones de cola y los cuellos de botella locales, facilitando actualizaciones del actor gracias a los hiperparámetros presentados anteriormente.

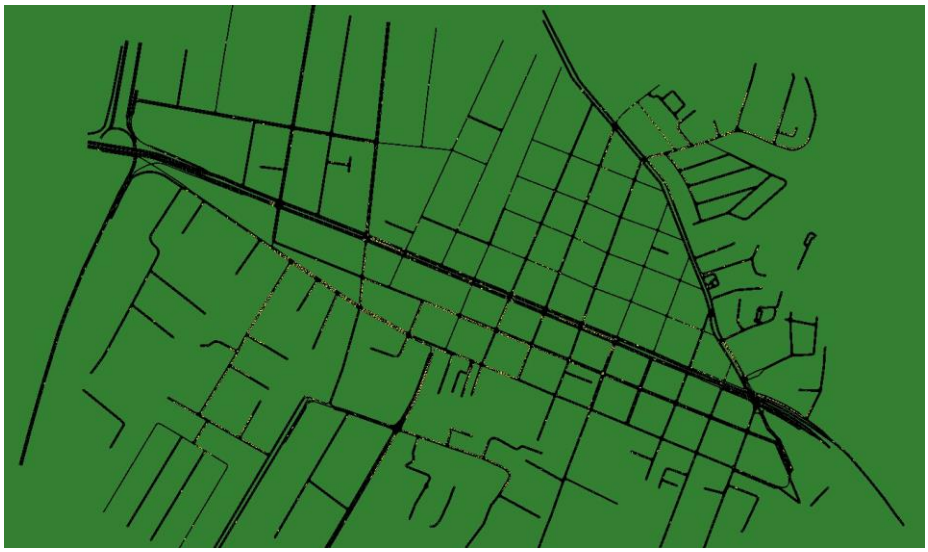


Figura 10: Representación de Bulevar Los Proceres en Sumo

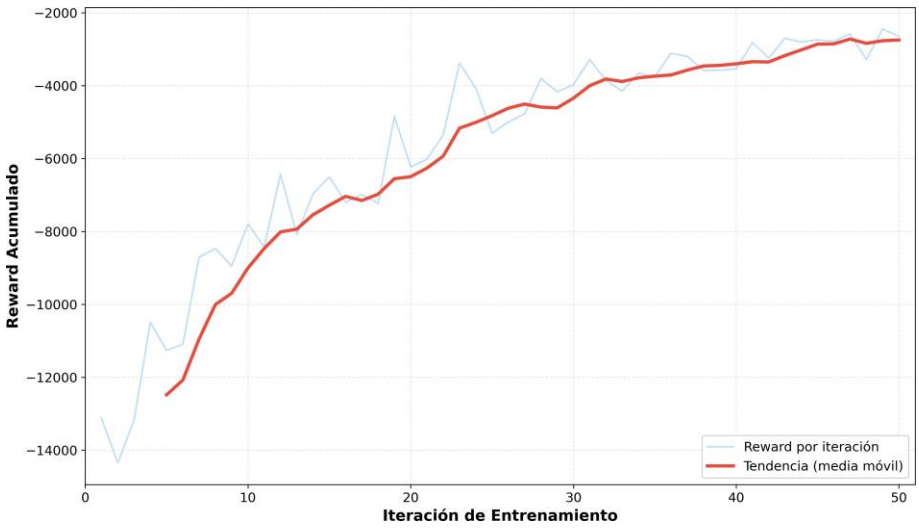



Figura 11: Curva de aprendizaje en Bulevar Los Próceres (media móvil en rojo).

En cuanto a métricas obtenidas, el éxito de la política aprendida se puede ver en el

Cuadro 6 que el tiempo de espera y del tiempo de viaje se redujo en 70 % y 77 % respectivamente. Además, tuvo un incremento de velocidad moderado de 14 %. Este comportamiento de grandes reducciones en esperas con un aumento de velocidad más discreto es consistente con la función de recompensa usada de diff-waiting-time. Esto es ya que el agente aprende a minimizar detenciones y a suavizar el flujo entre cambio de verdes.

Esto incluso se puede interpretar como que el controlador deja de depender de planes fijos y responde al estado en tiempo real, ajustando fases para reducir acumulación de retraso de vehículos.

Cuadro 6: Métricas comparativas: base vs. MAPPO (Próceres).

| **Métrica** | **Base** | **MAPPO** |
|---|---|---|
| Tiempo de Viaje Promedio (s) | 10 227.20 | 2 939.55 |
| Tiempo de Espera Promedio (s) | 9 417.19 | 2 135.64 |
| Velocidad Promedio (m/s) | 1.41 | 1.61 |

### 6.2.2. Avenida Elena - Anillo Periférico

Para el ambiente Avenida Elena - Anillo Periférico mostrado en la Figura 12, presenta un modelo con un entrenamiento graficado en la Figura 13 donde se puede ver una mayor varianza al inicio con una exploración más agresiva que se empieza a estabilizar alrededor de la iteración 25. A partir de ahí, la media móvil asciende y permanece en esos valores. Esto se puede analizar como que el crítico centralizado necesitó acoplarse a relaciones entre señales antes de considerar ventajas por cada semáforo agente. La razón principal de esto, puede ser por la presencia de calles secundarias y que el flujo se acumule en el Anillo Periférico y su intersección con la Avenida Elena.

A pesar de ello, el resultado en positivo, puesto que se muestra reducciones entre 70–71 % para tiempos de viaje y espera promedio, y un incremento de 12 % en velocidad. Como se mencionó anteriormente, este entorno presenta muchas calles secundarias, y la política parece haber priorizado la disipación de colas lo que se evidencia en menos detenciones.

Cuadro 7: Métricas comparativas: base vs. MAPPO (Elena–Periférico).

| **Métrica** | **Base** | **MAPPO** |
|---|---|---|
| Tiempo de Viaje Promedio (s) | 6 858.95 | 1 962.05 |
| Tiempo de Espera Promedio (s) | 6 312.19 | 1 877.87 |
| Velocidad Promedio (m/s) | 2.75 | 3.09 |

Figura 12: Representación de Avenida Elena - Anillo Periférico en Sumo

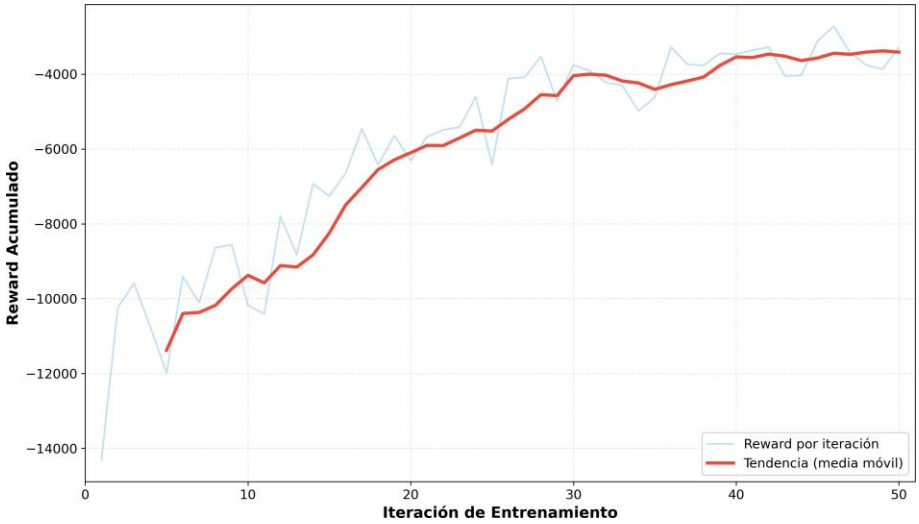



Figura 13: Curva de aprendizaje en Avenida Elena – Anillo Periférico.

### 6.2.3. Calzada Atanasio Tzul

En el caso del ambiente de la Calzada Atanasio Tzul cuya representación se muestra en la Figura 14 se desarrolló un modelo con una curva de entrenamiento mostrado en la 15. En dicho modelo aparece un “bache” en la recompensa alrededor de las iteraciones 20–22 seguido de recuperación y tendencia ascendente. Este comportamiento se da principalmente por dos razones. La primera es por el entorno en donde la amplitud del mismo y la inclusión de rutas como el Bulevar Liberación, permite esta forma. Asimismo, la segunda razón es por un comportamiento típico del aprendizaje multiagente, ya que al cambiar las políticas locales, varía la naturaleza del entorno de los demás ( y el crítico necesita recalibrar la estimación de valor global.

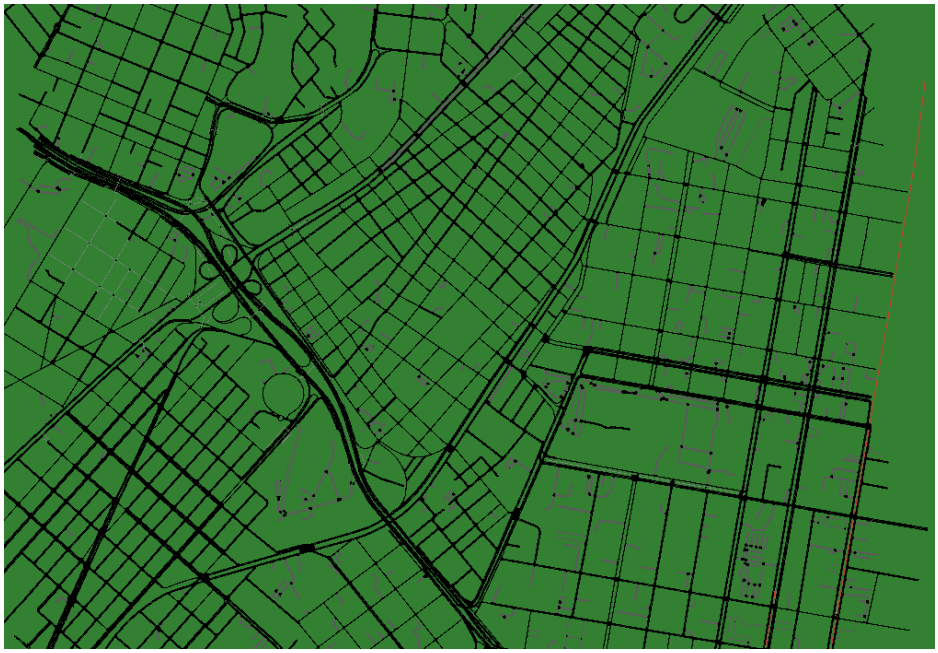


Figura 14: Representación de Calzada Atanasio Tzul en Sumo

No obstante, estos "baches"son tratados correctamente gracias al enfoque que presenta MAPPO cuando logra amortiguar bien este fenómeno. Esto se confirma puesto el sistema vuelve a ascender y cierra con mejoras entre 70–75 % en tiempo de viaje y espera promedio y más de 15 % en velocidad. Dado este entorno, se puede ver que los mayores beneficios se manifiestan no tanto en picos de velocidad como en la eliminación de paradas redundantes y en la contención de colas que bloquean cruces.

Cuadro 8: Métricas comparativas: base vs. MAPPO (Atanasio).

| **Métrica** | **Base** | **MAPPO** |
|---|---:|---:|
| Tiempo de Viaje Promedio (s) | 10 619.15 | 2 677.77 |
| Tiempo de Espera Promedio (s) | 9 603.40 | 2 828.66 |
| Velocidad Promedio (m/s) | 2.42 | 2.78 |

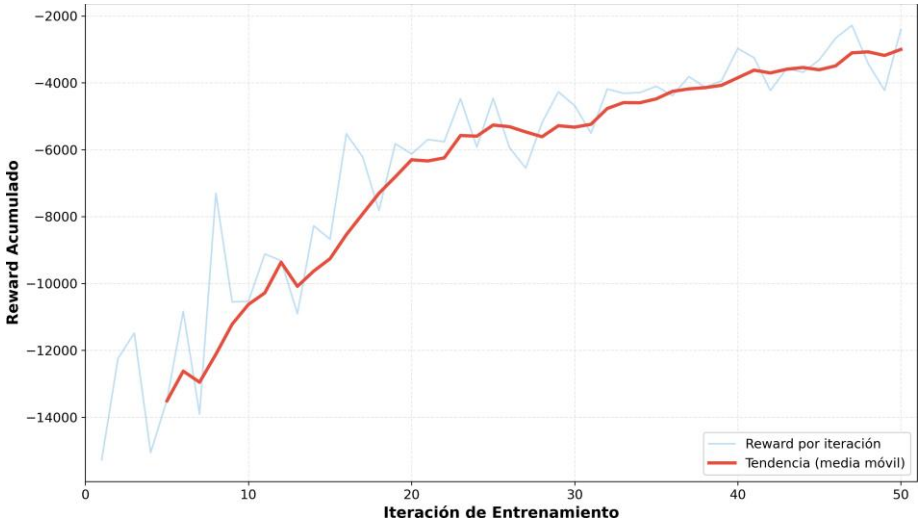



Figura 15: Curva de aprendizaje en Calzada Atanasio Tzul.

### 6.2.4. Avenida Reforma

Por último la Avenida Reforma tiene un entorno presentado en la Figura 16 y un modelo con su desempeño en la Figura 17 tiene curva más suave a partir de la iteración 15, con crecimiento constante. Dado que es el entorno con más semáforos principales, es el corredor con condiciones base más exigentes, este comportamiento sugiere que el crítico centralizado se beneficia de una señal global más informativa con colas u ocupación para valorar configuraciones que generan verde.

En cuanto a las métricas, el modelo nuevamente reduce el tiempo de viaje y espera promedio 74 % y 72 % respectivamente, y eleva la velocidad en 13 %. El patrón refuerza la hipótesis de que las ganancias absolutas son mayores en ambientes de alta demanda con muchos semáforos. Justamente es donde MAPPO despliega su ventaja sobre planes fijos o actuados no coordinados.

Cuadro 9: Métricas comparativas: base vs. MAPPO (Reforma).

| Métrica | Base | MAPPO |
|---|---|---|
| Tiempo de Viaje Promedio (s) | 16 328.63 | 4 199.74 |
| Tiempo de Espera Promedio (s) | 15 208.21 | 4 299.32 |
| Velocidad Promedio (m/s) | 1.01 | 1.14 |

### 6.2.5. Rendimiento general entre entornos

Como se pudo apreciar en todos los escenarios, la política MAPPO produce mejoras consistentes y de gran magnitud frente a las líneas base. Promediando los cuatro escenarios, se puede ver el desempeño de cada uno de ellos respecto el tiempo de viaje en la Figura 19

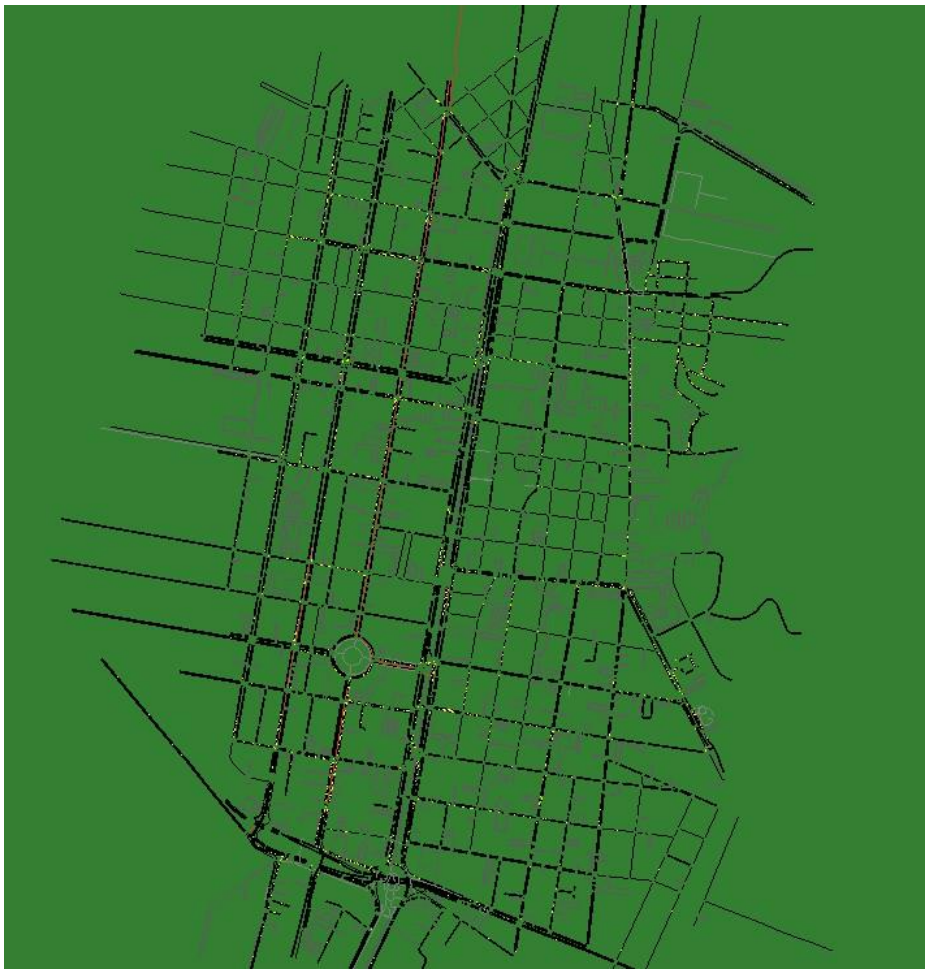


Figura 16: Curva de aprendizaje en Avenida Reforma.

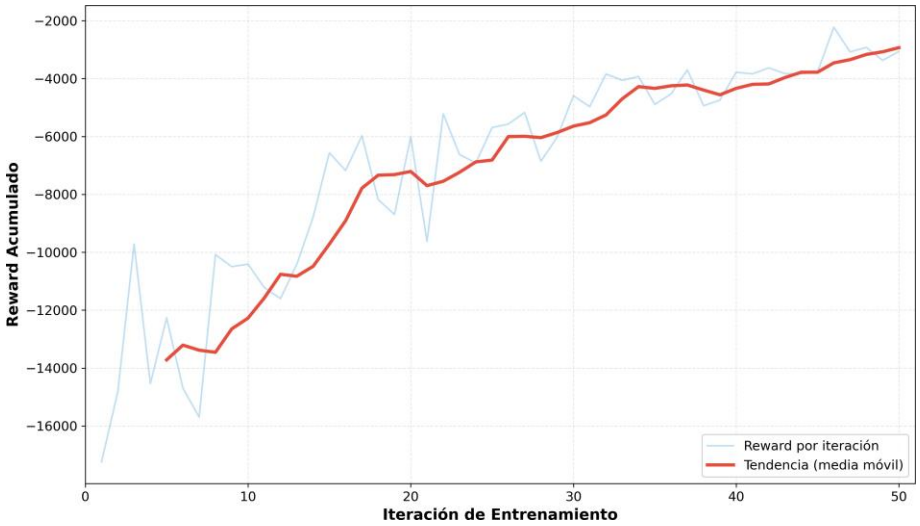



Figura 17: Curva de aprendizaje en Avenida Reforma.

en donde cae de 11 008.48 segundos a 2 944.78 segundos, es decir, que hubo una reducción de 73.25 %, mientras que el tiempo de espera en los cuatro escenarios presentado en el Cuadro 18 desciende de 10 135.25 segundos a 2 785.37 segundos, teniendo una reducción de 72.52 %. La velocidad promedio aumenta de (1.90 m/s a 2.16 m/s, incrementando un 13.57 %. Estas cifras replican el patrón de recortes de colas con aumentos de velocidad más moderados, lo que es coherente con el objetivo de la recompensa minimizar acumulación de retraso.

Desde la perspectiva de aprendizaje, las curvas de aprendizaje de los cuatro ambientes muestran una convergencia. La media móvil asciende de valores cercanos a $-14\,000$ hasta alrededor de $-3\,500$, indicando menor retraso acumulado paso a paso. En términos de arquitectura, el esquema CTDE de MAPPO explica la consistencia entre entornos, el crítico centralizado captura dependencias de corredor trata de forma correcta la no estacionariedad, mientras los actores locales ejecutan. Los hiperparámetros hallados favorecen actualizaciones y ventajas de baja varianza, lo que estabiliza el entrenamiento multiagente. El resultado es menores tiempos de viaje como menor dispersión.

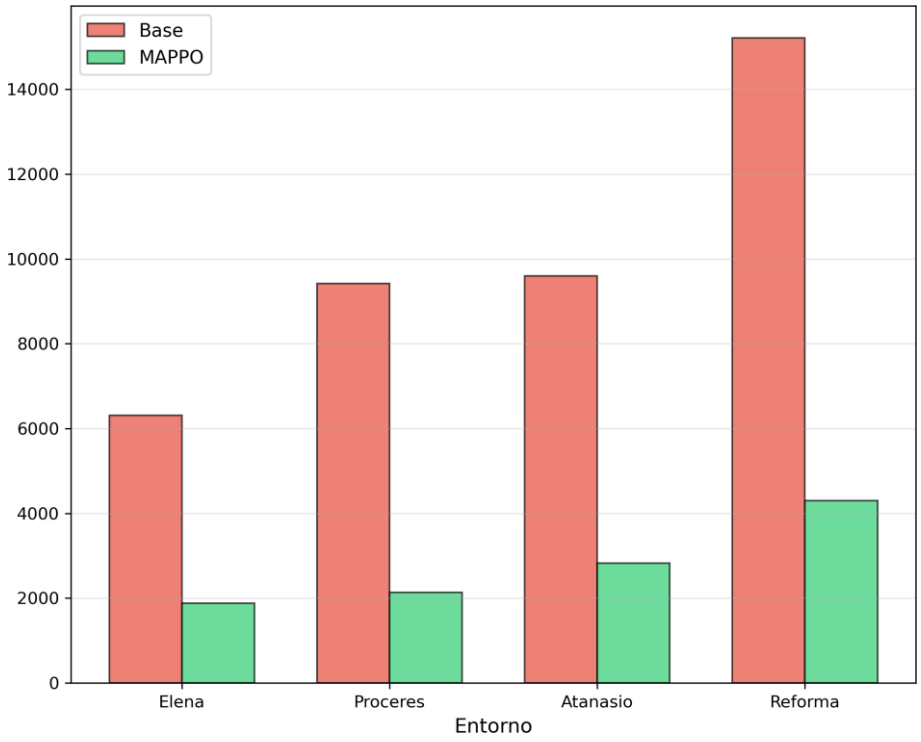



Figura 18: Comparación global de tiempo de espera promedio (base vs. MAPPO).

Finalmente, las ganancias más altas se da en corredores con congestión base elevada y densidad semafórica suficiente, tal y como sucedió en Avenida Reforma, donde la coordinación aprendida genera progresiones más largas y evita bloqueos. Este hallazgo sugiere una estrategia de despliegue de priorizar ejes de alta demanda con espaciamiento entre señales para maximizar beneficio. Además, el patrón de grandes reducciones en espera y aumentos moderados en velocidad indica que MAPPO ataca principalmente tiempos muertos de colas, lo que mejora también la confiabilidad del viaje.

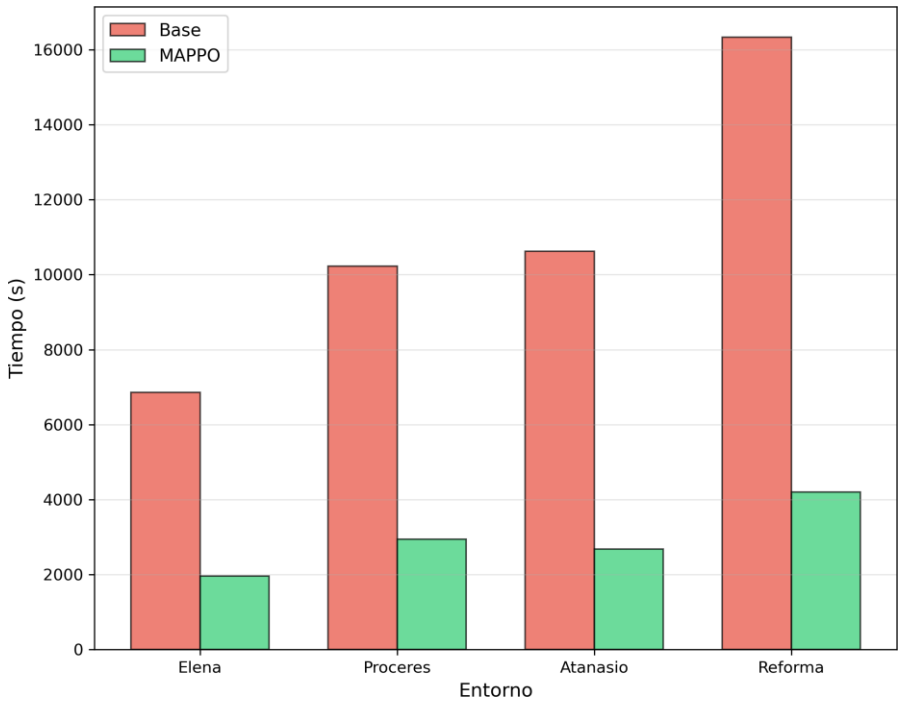



Figura 19: Comparación global de tiempo de viaje promedio (base vs. MAPPO).

CAPÍTULO 7

# Conclusiones

- Se implementó un Sistema de Semáforos Inteligentes basado en MARL y se demostró su impacto positivo sobre el flujo vehicular en entornos urbanos simulados. Esto permite que se pueda tomar en consideración la implementación en la vida real de este sistema dado su desempeño en los ambientes simulados.
- Se logró construir cuatro escenarios parametrizables (Bulevar Los Próceres, Avenida Elena–Anillo Periférico, Calzada Atanasio Tzul y Avenida Reforma), los cuales por su cantidad de semáforos y afluencia vehicular son representativos del tráfico de la Ciudad de Guatemala y aplicables para un sistema de semáforos inteligentes.
- Se desarrolló un algoritmo MAPPO basado en aprendizaje por refuerzo con uso de redes neuronales para usarse en un sistema de semáforos inteligentes. Se investigó y realizó técnicas como grid search para usar los hiperparámetros que cumplieran las reglas de tránsito y tuvieran la mejor recompensa acumulada.
- Con el Sistema de Semáforos Inteligentes, se logró mejorar las condiciones de tráfico en los escenarios implementados, reduciendo tiempo promedio de viaje en 73 %, tiempo promedio de espera en 72 % y aumentando la velocidad promedio en un 13 %.

# CAPÍTULO 8

## Recomendaciones

- Realizar un trabajo complementario de generación de entornos representativos con el fin de tener más control sobre la adición o reducción de elementos de tránsito tentativos que no estén actualmente.
- Considerar otro tipo de agentes en los entornos, tales como personas y transporte público con el fin de evaluar el rendimiento del sistema de semáforos inteligentes en esos casos.
- Tener acceso a datos actualizados y representativos de la Ciudad de Guatemala para generar afluencia vehicular similar a situaciones más recientes.
- Entrenar el modelo MAPPO en cómputo con mayor capacidad para evaluar más combinaciones de hiperparámetros y más variedad en valores de los mismos.
- Realizar técnicas como fine tunning en los modelos si se quieren usar en entornos similares en los cuales no fueron entrenados.

# CAPÍTULO 9

## Bibliografía


[1] Superintendencia de Administración Tributaria (SAT). (2024). Parque Vehicular por Departamento. URL: https://portal.sat.gob.gt/portal/estadisticas-indicadores-tributarios/#1506924610997-43de15cc-b0d8.

[2] Llamas, A. (2024). ¿Cuántas horas al día pasa un guatemalteco en el tránsito? URL: http://soy502.com/articulo/cuantas- horas- pasa- guatemalteco- transito-101566#:~:text=La%20autoridad%20de%20tr%C3%A1nsito%20revel%C3%B3,horas%2C%20de%20lunes%20a%20viernes..

[3] PNUD. (2024). En el marco del proyecto MuniJoven, la Municipalidad de Guatemala y el PNUD en Guatemala continúan cooperación en la implementación del sistema de semáforos inteligentes en la Ciudad de Guatemala. URL: https://www.undp.org/es/guatemala/noticias/en-el-marco-del-proyecto-munijoven-la-municipalidad-de-guatemala-y-el-pnud-en-guatemala-continuan-cooperacion-en-la-implementacion.

[4] Municipalidad de Guatemala. (2024). Semáforos inteligentes. URL: https:// www.muniguate.com/blog/2024/02/22/semaforos-inteligentes/.

[5] Roess, R., Prassas, E. y McShane, W. (2019). Traffic Engineering.

[6] U.S. Department of Transportation, Federal Highway Administration. (2023). Manual on Uniform Traffic Control Devices for Streets and Highways. URL: https://mutcd.fhwa.dot.gov/pdfs/11th_Edition/mutcd11thedition.pdf.

[7] Transportation Research Board. (2016). Highway Capacity Manual, Sixth Edition: A Guide for Multimodal Mobility Analysis.

[8] U.S. Department of Transportation, Federal Highway Administration. (2015). Traffic Signal Timing Manual.

[9] Urbanik, Tom et al. (2015). Signal Timing Manual.

[10] Japan International Cooperation Agency (JICA). (1992). Master Plan Study on the Comprehensive Urban Transportation System in Guatemala Metropolitan Area: Executive Summary. URL: https://openjicareport.jica.go.jp/pdf/10974442.pdf.

[11] Centro de Estudios Urbanos y Regionales (CEUR), Universidad de San Carlos de Guatemala. (2016). La infraestructura vial en el Área Metropolitana de Ciudad de Guatemala. Exposición de resultados de investigación: «Análisis de variables sobre los principales accesos a Ciudad de Guatemala». URL: https ://ceur .usac .edu .gt/ eventos/Metropolitanas/Presentaciones/05- Analisis- de- variables- sobre-los-principales-accesos-a-CDG.pdf.

[12] Carvajal, M. y Argueta, J. (2025). "Análisis de movilidad urbana con datos geoespaciales en tiempo real y el modelo de gravedad".

[13] Agencia Nacional de Alianzas para el Desarrollo de Infraestructura Económica (ANADIE). (2024). Desarrollo e infraestructura.

[14] U.S. Department of Transportation, Federal Highway Administration. (2017). Adaptive Signal Control Technology (ASCT). URL: https://www.fhwa.dot.gov/innovation/everydaycounts/edc-1/asct.cfm.

[15] Fehon, K. (2023). Adaptive Traffic Signals: Overview.

[16] Wei, H. et al. (2021). "Recent Advances in Reinforcement Learning for Traffic Signal Control: A Survey of Models and Evaluation". URL: https://doi.org/10 .1145 / 3447556.3447565.

[17] Russell, S. y Norvig, P. (2020). Artificial Intelligence: A Modern Approach.

[18] Gonzalez-Santocildes, A., Vazquez, J. y Eguíluz, A. (2024). "Adaptive Robot Behavior Based on Human Comfort Using Reinforcement Learning".

[19] Sutton, R. y Barto, A. (2018). Reinforcement Learning: An Introduction.

[20] Albrecht, S., Christianos, F. y Schäfer, L. (2024). Multi-Agent Reinforcement Learning: Foundations and Modern Approaches.

[21] Morales, M. (2020). Grokking Deep Reinforcement Learning.

[22] Unnikrishnan, A. (2024). Financial News-Driven LLM Reinforcement Learning for Portfolio Management.

[23] Amato, C. (2024). A First Introduction to Cooperative Multi-Agent Reinforcement Learning.

[24] Eclipse Foundation y DLR - Institute of Transportation Systems. (2025). SUMO User Documentation. URL: https://sumo.dlr.de/docs/.

[25] Eclipse Foundation y DLR - Institute of Transportation Systems. (2025). Tools/Routes. URL: https://sumo.dlr.de/docs/Tools/Routes.html.

[26] Eclipse Foundation y DLR - Institute of Transportation Systems. (2025). TraCI — Traffic Control Interface. URL: https://sumo.dlr.de/docs/TraCI.html.

[27] The AnyLogic Company. (2025). AnyLogic. URL: https://www.anylogic.com/.

[28] Flow Project. (2019). Flow. URL: https://github.com/flow-project/flow.

[29] Ray Project. (2025). RLlib Algorithms. URL: https://docs.ray.io/en/latest/rllib/rllib-algorithms.html.

[30] Ray Project. (2025). RLlib — New API Stack: Migration Guide. URL: https://docs.ray.io/en/latest/rllib/new-api-stack-migration-guide.html.

[31] California Department of Transportation (Caltrans). (2023). Feasibility and Comparison Study of Adaptive Traffic Control Systems.

[32] Stevanovic, A. (2010). Adaptive Traffic Control Systems: Domestic and Foreign State of Practice. URL: http://onlinepubs.trb.org/onlinepubs/nchrp/nchrp_syn_403.pdf.

[33] Los Angeles Department of Transportation (LADOT). (2020). Traffic Signal Warrant. URL: https://ladot.lacity.gov /sites/default/files/documents /traffic-signal-warrant-rev-08.10.20.pdf.

[34] Yu, L et al. (2004). Guidebook on Determining Yellow and Red Intervals to Improve Signal Timing Plans for Left-Turn Movements.

[35] Bonneson, J. et al. (2011). Traffic Signal Operations Handbook, Second Edition. URL: https://static.tti.tamu.edu/tti.tamu.edu/documents/0-6402-P1.pdf.

[36] OpenAI. (2025). Implementación de PPO. URL: https://spinningup.openai.com/en/latest/_modules/spinup/algos/tf1/ppo/ppo.html.

[37] Stable-Baselines3 Developers. (2025). PPO — Stable-Baselines3 Documentation. URL: https://stable-baselines3.readthedocs.io/en/master/modules/ppo.html.

[38] Schulman, J et al. (2017). "Proximal Policy Optimization Algorithms". URL: https://arxiv.org/abs/1707.06347.

[39] Alegre, L. SUMO-RL — Documentation. URL: https://lucasalegre.github.io/sumo-rl/.