

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus CEM (Edo. de México)

**Revisión 3 - Avance al 60%**

Renato Garcia Moran A01799387

Fabrizio Martinez Chavez A01749680

Ignacio Solís Montes A01751213

Kevin Santiago Castro Torres A01798925

Roger Vicente Rendón Cuevas A01749992

Modelación de sistemas multiagentes con gráficas computacionales

Gpo 302

21 de noviembre de 2024

**Descripción del ambiente:**

En un sistema multiagentes que involucra un automóvil inteligente, un sistema de semáforos y la inclusión de peatones que caminan por las banquetas, todos los agentes son reactivos. El ambiente está diseñado para garantizar la seguridad y eficiencia tanto del flujo vehicular como del tránsito peatonal.

Por su parte, los semáforos adaptan su comportamiento en función de la posición de los peatones y los vehículos, activando luces peatonales cuando es seguro cruzar.

El ambiente presenta las siguientes cinco características:

1. **Observabilidad:** Los sensores del sistema, incluidos los semáforos y el automóvil especial, detectan la posición de los vehículos y peatones, así como el momento en que cae la noche, permitiendo tomar decisiones en tiempo real.
2. **Accesibilidad:** Los agentes tienen acceso directo a información clave del entorno, como la ubicación de los peatones, la trayectoria de los vehículos y la hora del día, a través de sensores y comunicación entre ellos.
3. **Determinismo:** El ambiente es parcialmente determinista, ya que las acciones del sistema están regidas por reglas predefinidas, aunque puede haber incertidumbre debido a factores como cambios en el tráfico o el comportamiento de los peatones.
4. **Dinamicidad:** El ambiente cambia constantemente con el movimiento de vehículos y peatones, lo que afecta las señales y decisiones.
5. **Discretización:** Las acciones y decisiones ocurren en pasos discretos, con actualizaciones periódicas para garantizar una coordinación eficiente entre los agentes.

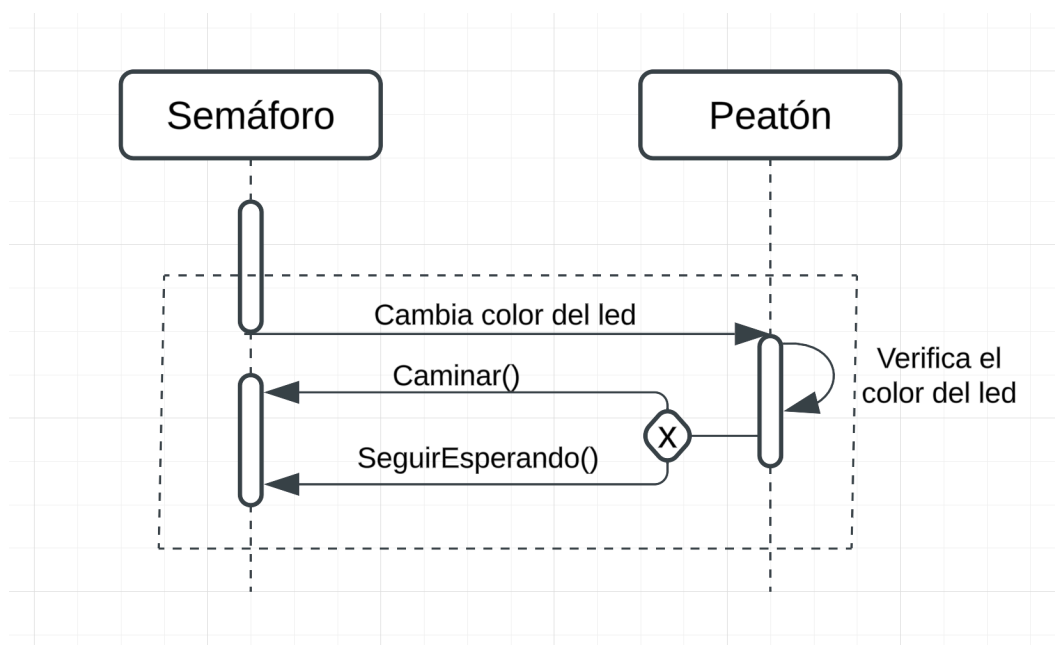
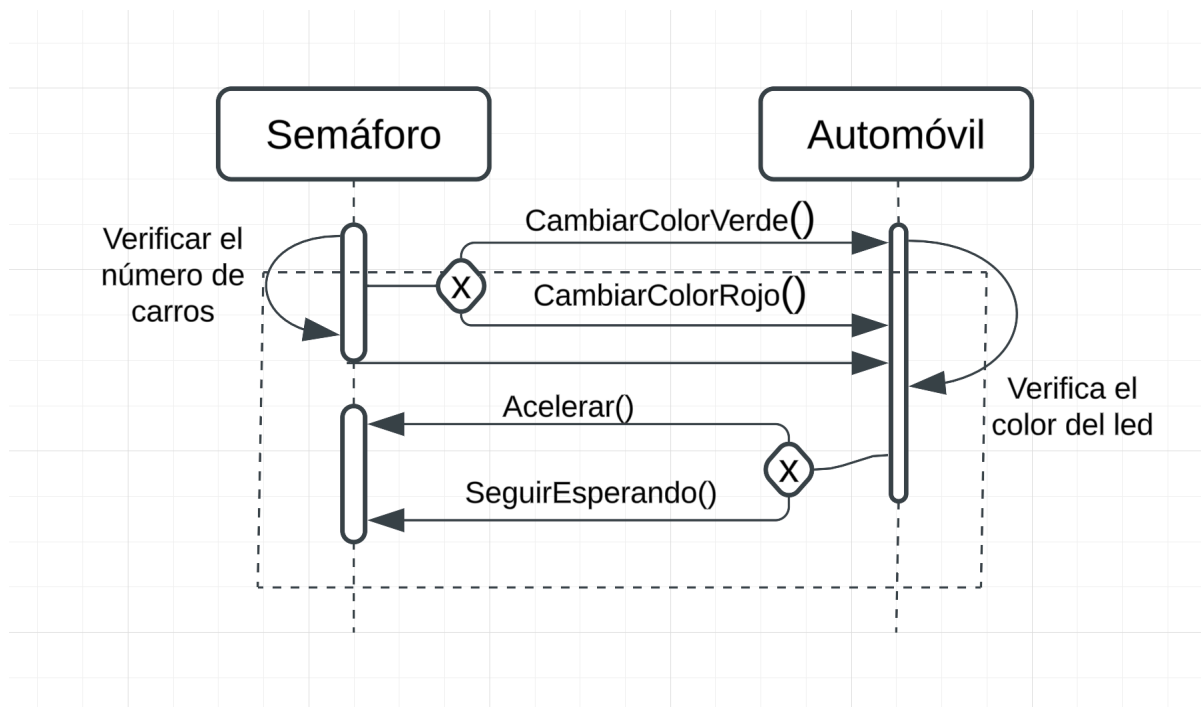
Este ambiente interactivo y reactivo asegura que los semáforos, el automóvil especial y los peatones trabajen en conjunto para mantener la seguridad y la fluidez en el sistema.

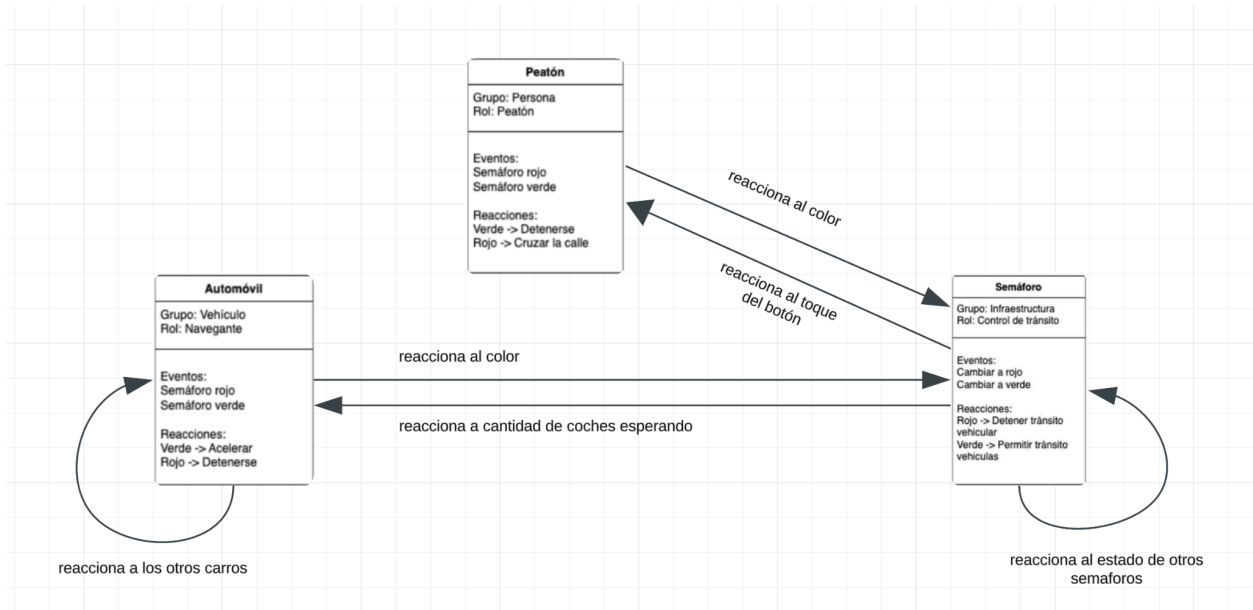
**Diagramas AUML:**

<b>Automóvil</b>
Grupo: Vehículo Rol: Navegante
Eventos: Semáforo rojo Semáforo verde  Reacciones: Verde -> Acelerar Rojo -> Detenerse

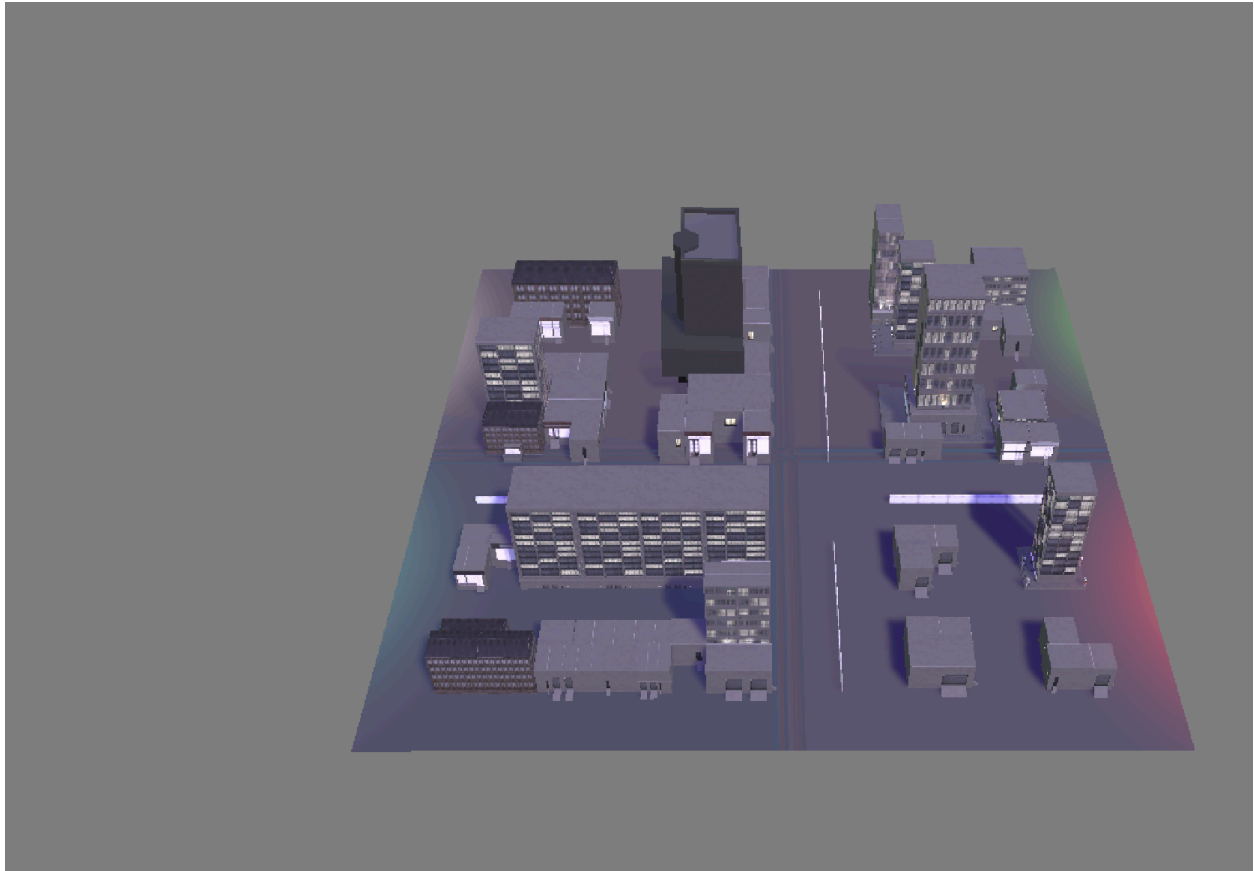
<b>Peatón</b>
Grupo: Persona Rol: Peatón
Eventos: Semáforo rojo Semáforo verde  Reacciones: Verde -> Detenerse Rojo -> Cruzar la calle

<b>Semáforo</b>
Grupo: Señalamiento Rol: Control de Tráfico
Eventos: Luz verde Luz Roja  Reacciones: Luz verde -> Avanzar Luz roja -> Detenerse

**Diagramas de interacción:**



### Avances de Gráficas Computacionales:



La imagen de arriba muestra el mapa del reto desarrollado en Unity, el cual incluye todos los edificios del mapa de la Actividad Integradora. En su estado actual, el mapa está listo para servir como base para la simulación de los agentes. Solo queda implementar el código en C# necesario para que los autos puedan desplazarse.

### Avances de SMA:

Se mejoró el movimiento haciendo un grafo dirigido con lista de adyacencia para manejar el movimiento de los automóviles, para la búsqueda del camino más corto entre el punto de inicio del agente y su objetivo final se utilizó la búsqueda en anchura, aprovechando las características de este algoritmo que garantiza encontrar el camino más corto para un grafo  $G=(V,E)$  en un tiempo lineal de  $O(V+E)$ , respetando los lineamientos de tránsito como el uso correcto de una glorieta o rotonda para hacer cambios de ruta válidos:

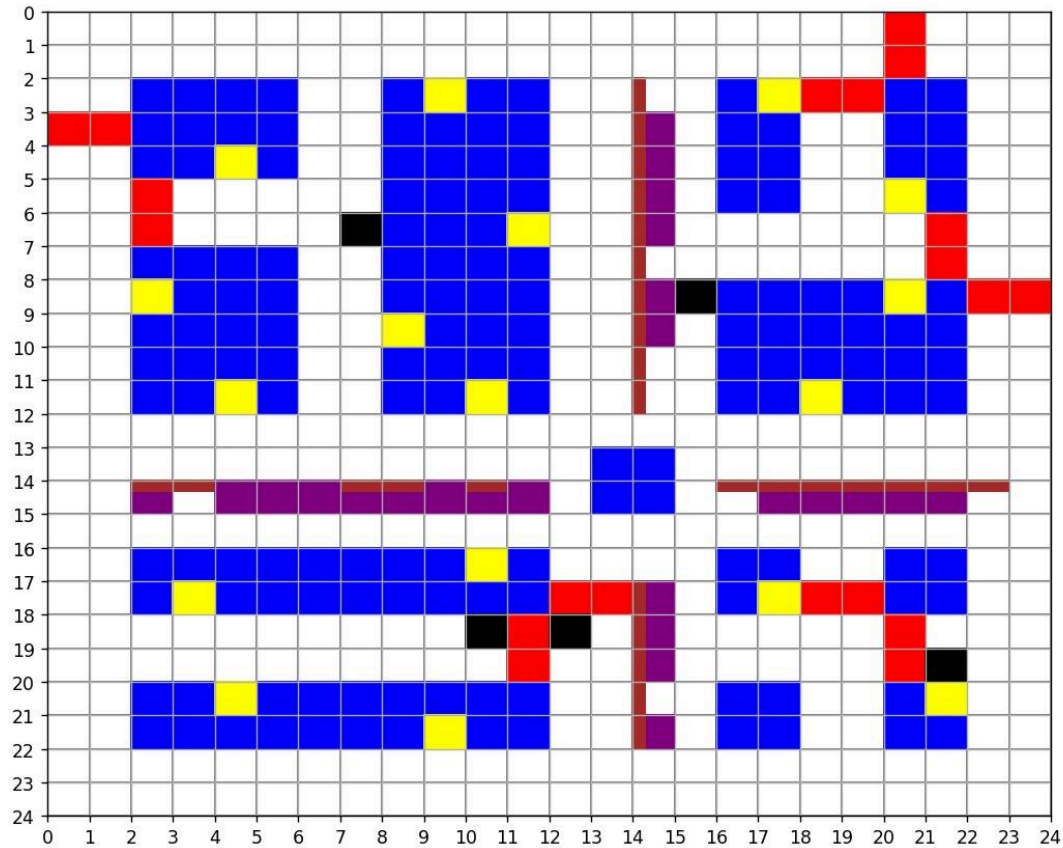
```
Camino desde (4, 20) hasta (9, 2): [(4, 20), (4, 19), (5, 19), (6, 19), (7, 19), (8, 19), (9, 19), (10, 19), (11, 19), (12, 19), (12, 20), (12, 21), (12, 22), (13, 22), (14, 22), (15, 22), (15, 21), (15, 20), (15, 19), (15, 18), (15, 17), (15, 16), (15, 15), (15, 14), (15, 13), (15, 12), (15, 11), (15, 10), (15, 9), (15, 8), (15, 7), (15, 6), (15, 5), (15, 4), (15, 3), (15, 2), (15, 1), (14, 1), (13, 1), (12, 1), (11, 1), (10, 1), (9, 2)]
```

Esto permite un comportamiento más inteligente de los agentes automóviles y permite manejar un mejor flujo de tráfico para los agentes semáforos, dando una mayor facilidad a los agentes peatones en la futura mejora del código.

Para los futuros agentes en proceso de planeación esto supone un avance que permite simplemente

Figure 1

— □



La imagen anterior muestra el progreso actual en el desarrollo del agente peatón, diseñado para desplazarse de manera limitada por los camellones dentro del entorno simulado. Este avance constituye una base funcional importante, aunque aún no se ha integrado completamente en el sistema principal. Esto se debe a que su desarrollo se realizó en paralelo a dos tareas clave: la mejora del agente vehículo y la implementación de grafos dirigidos en el mapa.

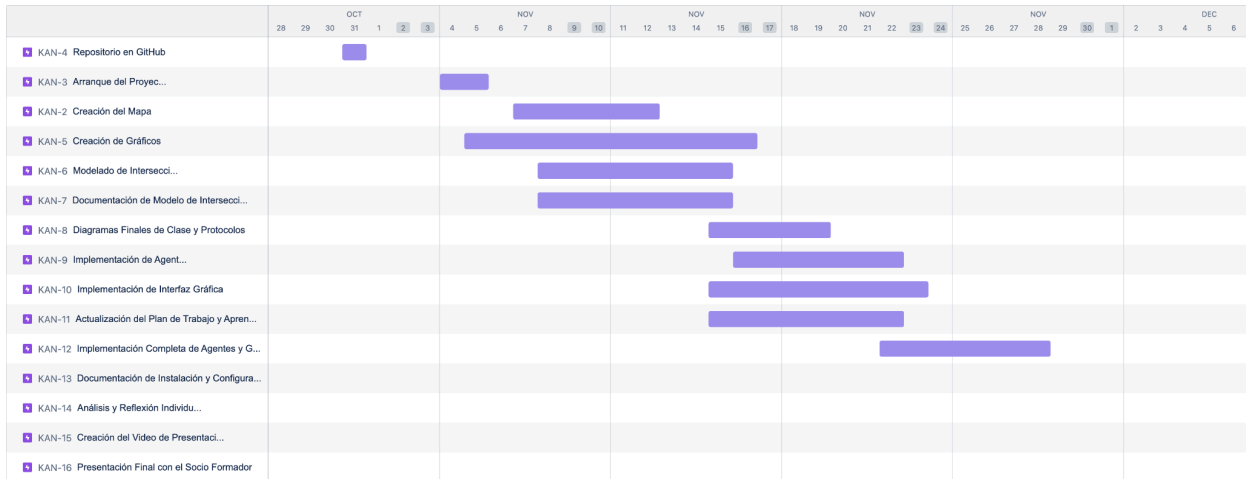
A pesar de no estar integrado, el trabajo realizado en esta etapa no es en vano. La lógica y los fundamentos del agente ya están desarrollados, incluyendo comportamientos básicos como el movimiento restringido. Este avance no solo reduce significativamente el tiempo y la complejidad que suponía su implementación inicial, sino que también permite aprovechar la estructura de los grafos para facilitar su futura integración con el mapa final.

**Pasos restantes para la integración del agente peatón:**

Para finalizar la implementación del agente peatón, será necesario completar las siguientes tareas:

- 1. **Incorporar los grafos en su lógica de movimiento:**  
El agente peatón debe utilizar los grafos dirigidos para determinar las rutas que puede seguir dentro del entorno. Esto incluye definir los puntos donde puede cruzar calles, siguiendo las restricciones de seguridad y optimizando su trayectoria hacia el destino.
- 2. **Desarrollar la interacción con los semáforos:**  
Es crucial implementar la "charla" que tendrá el agente peatón con el agente semáforo. El peatón deberá solicitar permiso para cruzar calles, y el semáforo responderá dependiendo de su estado (verde o rojo). Durante el cruce, el semáforo deberá garantizar que el paso del peatón sea seguro, bloqueando temporalmente el flujo vehicular hasta que el cruce haya concluido.

**Plan de trabajo:**



Los siguientes pasos son necesarios para alcanzar el 100% del proyecto:

**Actividades pendientes**

- 1. **Incorporar dos agentes adicionales para la noche**
  - **Responsables:** Fabrizio, Roger, Renato
  - **Inicio:** 23 de noviembre de 2024
  - **Finalización estimada:** 25 de noviembre de 2024
  - **Esfuerzo estimado:** 12 horas
  - **Descripción:** Crear la lógica de los dos nuevos agentes, integrarlos en el sistema y probar su interacción con el entorno.
- 2. **Implementar ciclo de día y noche**



- **Responsable:** Ignacio
  - **Inicio:** 24 de noviembre de 2024
  - **Finalización estimada:** 26 de noviembre de 2024
  - **Esfuerzo estimado:** 8 horas
  - **Descripción:** Modificar el mapa de Unity para reflejar transiciones dinámicas entre día y noche.
3. **Añadir assets decorativos**
- **Responsables:** Kevin, Ignacio
  - **Inicio:** 25 de noviembre de 2024
  - **Finalización estimada:** 27 de noviembre de 2024
  - **Esfuerzo estimado:** 10 horas
  - **Descripción:** Añadir assets visuales para mejorar la experiencia del usuario, incluyendo elementos urbanos y decorativos.
4. **Integrar completamente el agente peatón**
- **Responsables:** Roger, Fabrizio, Renato
  - **Inicio:** 26 de noviembre de 2024
  - **Finalización estimada:** 28 de noviembre de 2024
  - **Esfuerzo estimado:** 12 horas
  - **Descripción:** Ajustar la lógica del agente peatón e integrarlo completamente en el sistema, asegurando su correcto funcionamiento en el entorno.
5. **Separar y finalizar la conexión del peatón con Unity**
- **Responsables:** Kevin
  - **Inicio:** 27 de noviembre de 2024
  - **Finalización estimada:** 28 de noviembre de 2024
  - **Esfuerzo estimado:** 6 horas
  - **Descripción:** Finalizar la conexión del agente peatón con Unity y realizar pruebas para garantizar que las simulaciones funcionen correctamente.

El enfoque iterativo del proyecto nos permitirá cumplir con estos objetivos, garantizando la interacción correcta entre los agentes y un comportamiento coherente del sistema en cada iteración.

### **Justificación del avance al 60% del proyecto**

El avance actual corresponde al **60%** del proyecto, basado en los siguientes logros:

- **Mapa en Unity base completado:** El diseño del mapa en Unity, que incluye todos los elementos esenciales, ha sido terminado. Este mapa cumple con los requisitos mínimos para considerarse un entregable funcional, proporcionando la base para integrar y probar los agentes.

- **Implementación inicial de agentes en Python:** Se han desarrollado los agentes básicos necesarios para la simulación, permitiendo probar rutas, comportamientos y su interacción con el entorno. Además, se ha avanzado significativamente en la creación de la base del agente que facilita la creación de otro agente con características similares que funcionará como peatón especializado, aumentando la complejidad y el realismo del sistema.
- **Puente Python-Unity funcional:** La conexión entre el sistema multiagente desarrollado en Python y el entorno gráfico de Unity está completamente desarrollada y probada. Esto garantiza que no habrá contratiempos relacionados con la comunicación entre ambos componentes durante las etapas finales de integración.

#### **Actividades restantes:**

Aunque el proyecto muestra un avance significativo, todavía quedan por implementar elementos clave, como la iluminación dinámica de día y noche, la incorporación de assets decorativos para mejorar el entorno visual y la integración final del agente peatón en la simulación.

El 60% refleja el progreso sustancial en las bases del proyecto, cubriendo las etapas críticas de diseño, implementación inicial y pruebas de comunicación. Las actividades restantes se centran principalmente en la integración final, ajustes visuales y la incorporación de agentes adicionales, las cuales se pueden completar dentro del tiempo disponible, asegurando la entrega de un sistema multiagente funcional y bien integrado.

#### **Aprendizaje Adquirido:**

El proyecto ha proporcionado un entendimiento profundo de los sistemas multiagentes, su modelado en Python y la integración con simulaciones gráficas. Entre los aprendizajes más significativos destacan:

- **Sistemas Multiagentes (SMA):** Se adquirió experiencia en la creación de agentes reactivos que interactúan en un ambiente dinámico, incluyendo la programación de comportamientos autónomos y la resolución de conflictos en las decisiones.
- **Modelado en Python:** Desarrollo de agentes autónomos que utilizan estructuras de datos como grafos para calcular rutas óptimas en un entorno definido.
- **Integración de Python y Unity:** Construcción de un puente de comunicación para sincronizar datos entre el SMA en Python y el motor gráfico Unity.
- **Modelado 3D:** Uso de herramientas como Blender y Unity para diseñar el entorno gráfico y exportarlo eficientemente al simulador.
- **Optimización de algoritmos:** Cambio del algoritmo de búsqueda A\* a BFS, mejorando el rendimiento en grafos dirigidos.