Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

Sistemas multiagentes

Módulo de Reto:

Avance de proyecto

Equipo #4

Carlos Francisco Sanchez Llanes - A01741201

Andrés Eduardo Gomes Brandt - A01781321

Saúl Roberto Orozco Villanazul - A00831554

Fernando Adrián Fuentes Martínez - A01028796:

23- Enero- 2025

Simulación multiagentes en movilidad urbana

La simulación multiagentes ofrece una solución innovadora para abordar los complejos desafíos de la movilidad urbana moderna:

• Modelado realista del comportamiento individual de vehículos, peatones y semáforos.

• Evaluación de diferentes configuraciones de infraestructura

• Optimización de flujos de tráfico en tiempo real.

Objetivos de investigación del proyecto

Objetivo general: Optimizar el flujo vehicular en intersecciones urbanas mediante simulación multiagentes, identificando las configuraciones óptimas de infraestructura y control de tráfico.

Objetivos específicos:

- Determinar, mediante la utilización de dos modelos distintos de movimiento peatonal, cuál termina siendo más óptimo para distintos tipos de tráfico. Para lograr esto, se emplea de un A* y de una heurística, donde el A* esperará a que un obstáculo dinámico en su ruta se mueva para poder continuar, mientras que el que utiliza la heurística intentará utilizar nuevos caminos dependiendo de sus obstáculos.
- Graficar en medida de steps las observaciones anteriores y en función de la cantidad de peatones que se encuentran en un momento dado de la simulación, cuanto tardan los peatones que usan la ruta de A* vs cuánto tardan los peatones que usan la heurística
- En caso de una muy grande congestión, donde los peatones ya no estén llegando a sus destinos, borrar todos los peatones que existen después de una cantidad determinada de pasos, contar cuántos de los peatones que lograron llegar a su destino usan heurística y cuántos A* y contar cuantos peatones no pudieron llegar a su destino

Agentes (definición formal)

1. Obstacle (Obstáculo)

- **Rol**: Representa un obstáculo que bloquea caminos en el entorno.
- **Atributos**:
  - Sin atributos específicos, solo ocupa una posición en el grid.
- **Comportamiento**:
  - Pasivo. Su función es evitar que otros agentes ocupen su posición.

---

2. Spawn (Punto de aparición)

- **Rol**: Representa un punto de generación para peatones.
- **Atributos**:
  - Sin atributos específicos, define las posiciones donde se generan peatones.
- **Comportamiento**:
  - Pasivo.

---

3. Building (Edificio)

- **Rol**: Representa un edificio en el entorno.
- **Atributos**:
  - Sin atributos específicos.
- **Comportamiento**:
  - Pasivo.

---

4. Stoplight (Semáforo)

- **Rol**: Controla el flujo de vehículos y peatones en intersecciones.
- **Atributos**:
  - state (booleano): Indica el estado actual del semáforo (verde o rojo).
  - change_time (entero): Tiempo restante antes de cambiar de estado.
  - myRoads (lista): Lista de caminos asociados a este semáforo.
  - pos (cadena): Posición o configuración inicial del semáforo.
- **Comportamiento**:
  - Cambia su estado cada change_time pasos.
  - Detiene o permite el paso de vehículos y peatones según su estado.

---

5. Road (Camino)

- **Rol**: Representa un tramo de camino donde circulan vehículos y peatones.
- **Atributos**:
  - dir (dirección): Dirección del camino (e.g., "UP", "DOWN", "+").
  - paso_peatonal (booleano): Indica si el camino tiene un paso peatonal.
  - stop (booleano): Indica si el camino está detenido por un semáforo.
- **Comportamiento**:
  - Pasivo, pero afecta el movimiento de vehículos y peatones según sus atributos.

---

6. Peatón (Peatón)

- **Rol**: Representa a un peatón que se mueve hacia un objetivo siguiendo una de dos implementciones.
- **Atributos**:
    - goal (posición): Destino del peatón.
    - route (lista): Destino del peatón.
    - movement (bool): tipo de movimiento
- **Comportamiento**:
    - Sigue una ruta o se mueve mediante una heurística dependiendo de su setup por el ambiente
    - Se mueve siguiendo pasos peatonales y respetando semáforos.
    - Su generación se controla mediante agentes Spawn.

---

7. Car (Coche)

- **Rol**: Representa un coche que circula por las calles.
- **Atributos**:
    - dir (dirección): Dirección actual del coche.
    - prev_dir (dirección): Dirección anterior del coche.
- **Comportamiento**:
    - Calcula su próximo movimiento basado en:
        - Direcciones posibles.
        - Estado de los semáforos.
        - Presencia de obstáculos u otros agentes.
    - Si no puede avanzar, el coche puede detenerse o ser removido del modelo.

Ambiente

El entorno de la simulación está definido como una ciudad, generada a partir de un archivo de texto que luego es convertido a una matriz y posteriormente a un grid que toma como dimensiones las filas y columnas de dicha matriz. Este ambiente tiene las siguientes características :

Accesible: El ambiente es accesible ya que los peatones poseen conocimiento completo sobre el ambiente, siendo la posición de los distintos agentes que se encuentran en el grid (lo cual es cierto exclusivamente para los otros coches) la única cosa que desconocen

No determinístico: Debido a varios factores, entre los cuales podemos mencionar la aleatoriedad de la decisión en los destinos y las rutas, así como la gran cantidad de distintas rutas que se pueden tomar para llegar a los distintos destinos, el ambiente no es determinístico

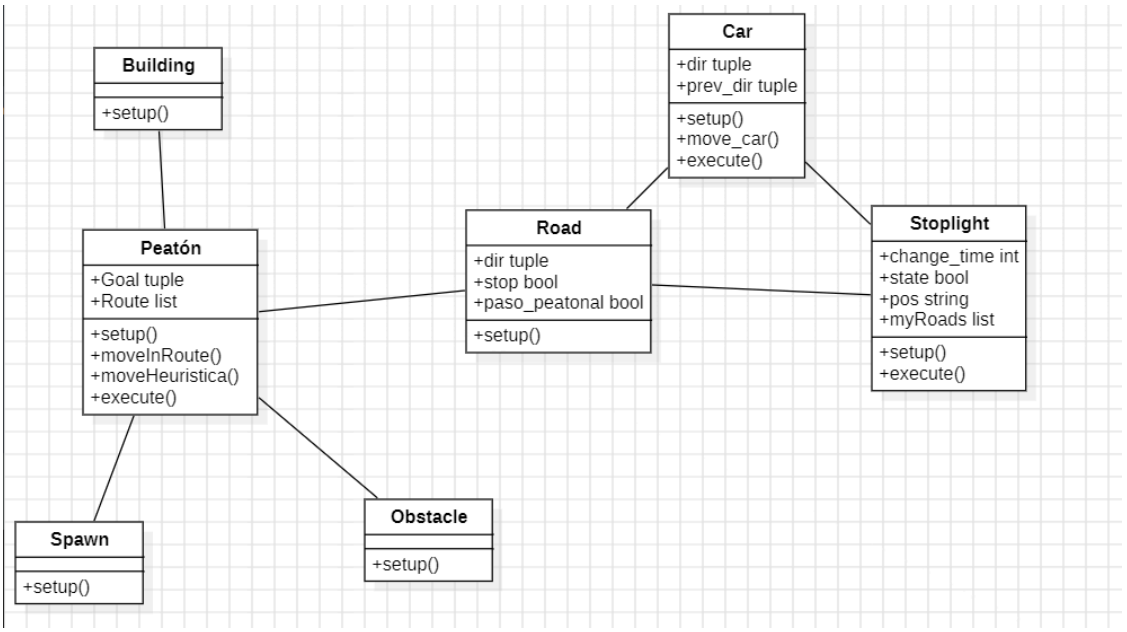
No Episódico: El ambiente no es episódico debido a que los semáforos son dependientes del estado en el que se encontraban en episodios anteriores para tener un conteo de cuando deberían cambiar de estado.

Estático: Debido a que toda la comunicación y manipulación que se realiza dentro del ambiente se realiza entre los agentes. De haberse implementado algunos de los agentes como parte del ambiente, se consideraría como un ambiente dinámico

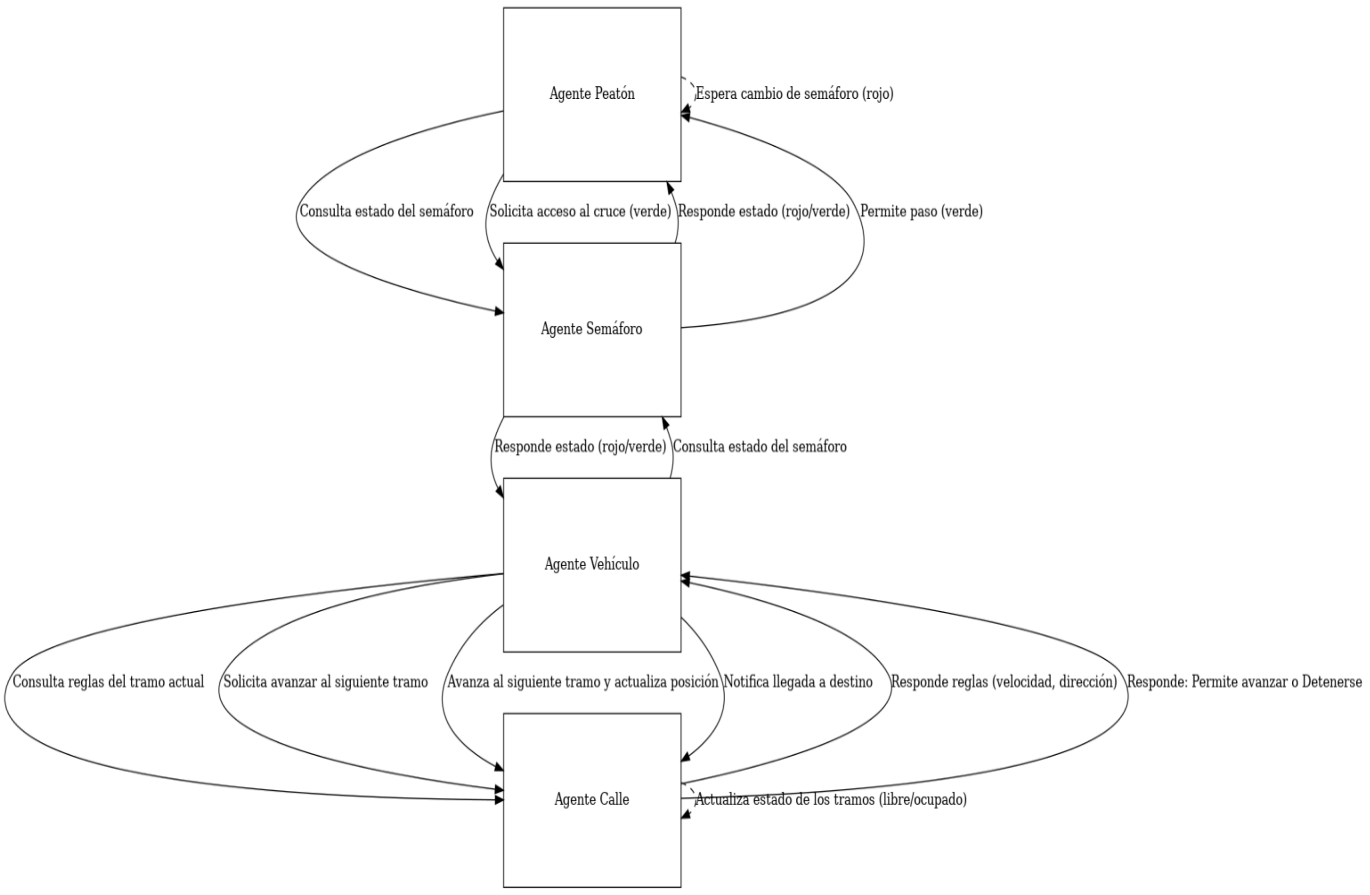
Discreto: Todos los valores que se trabajan dentro del ambiente son discretos al ser un grid.

Diagramas

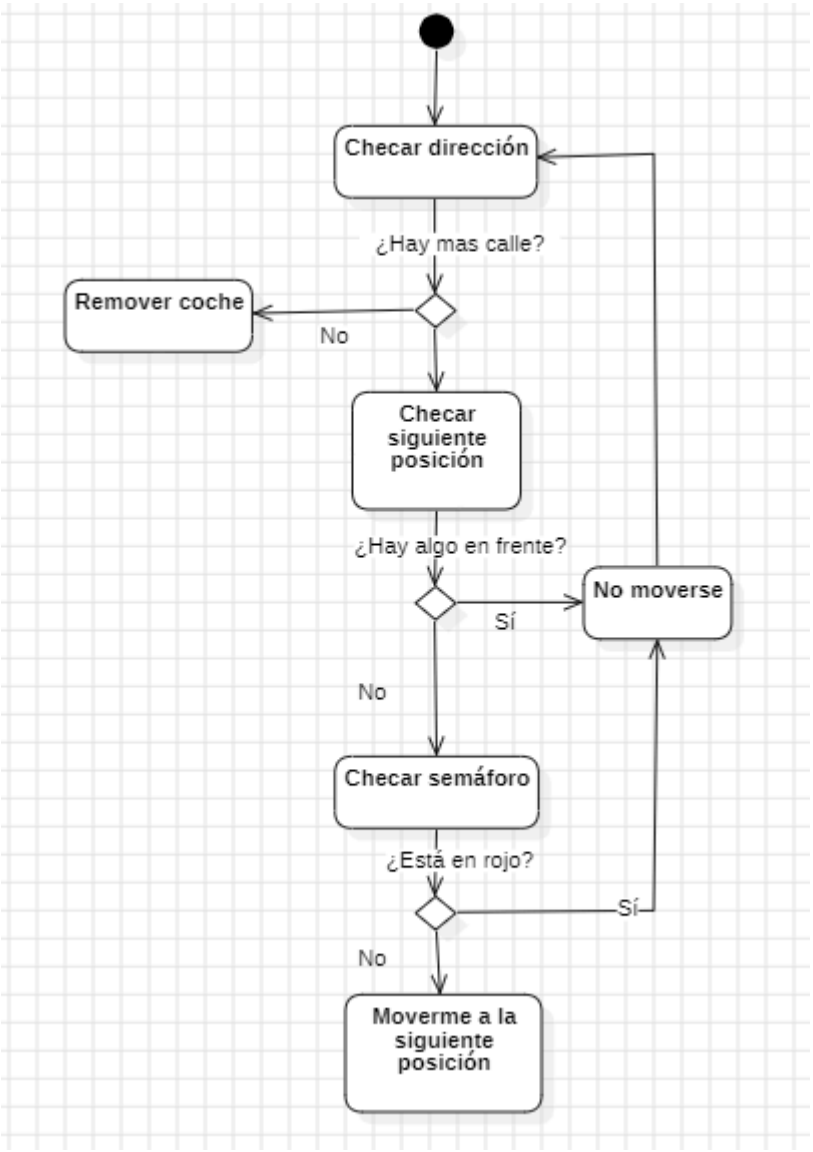
Diagrama de clases de los agentes



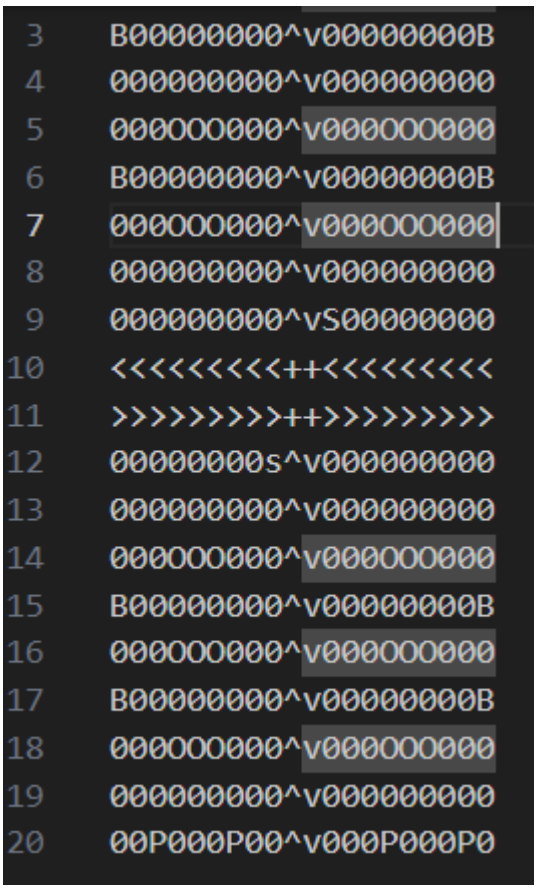
Protocolos de interacción



Movimiento de coche:



Base del mapa del modelo:



P: Spawn points

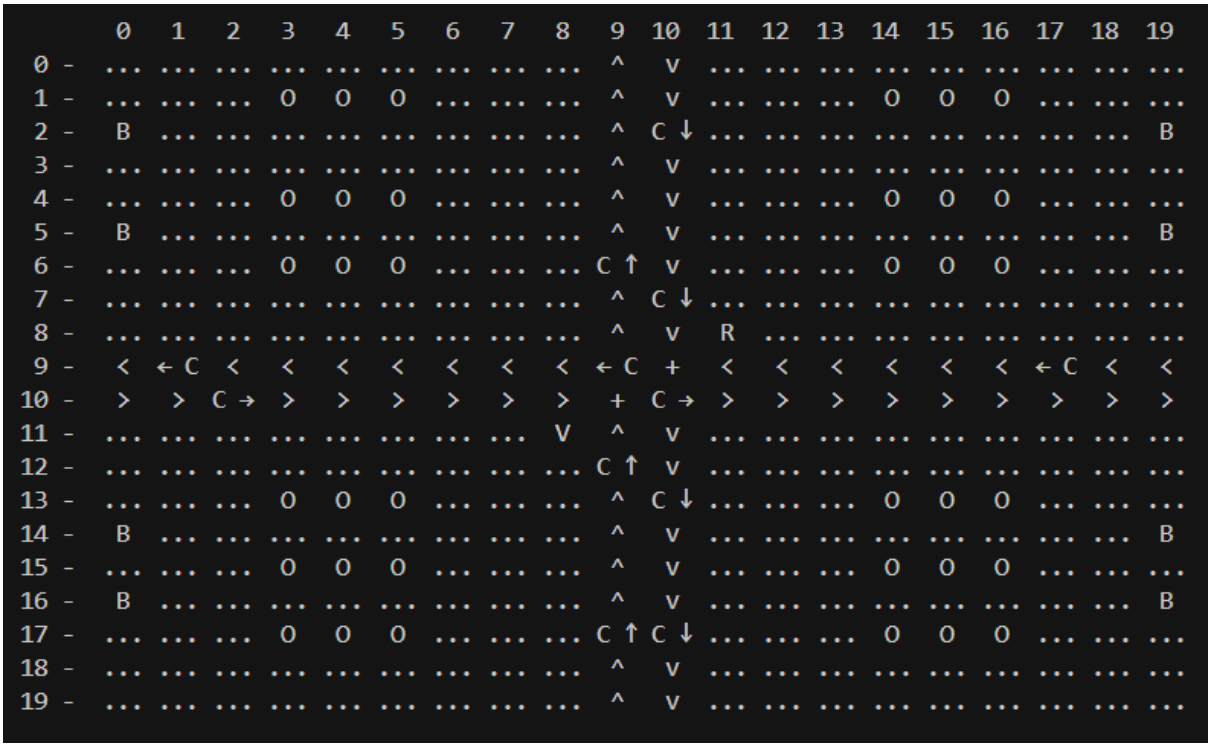
+,<,>,v,^: calles

O: Obstáculos

S, s: Semáforos

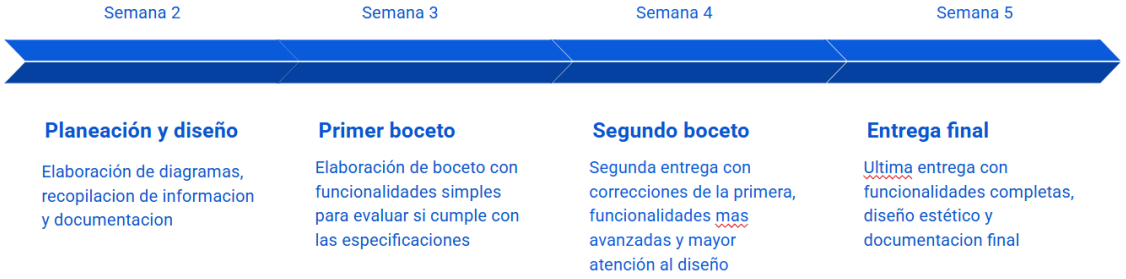
B: Edificios

Otro print del modelo, note que los spawn points son invisibles en esta representación y las C representan coches.



Plan de trabajo

Planeador de milestones en tiempo



Herramientas y plataformas

Herramientas	Plataformas
Repositorio remoto	Github
Planeador de tareas	Jira
Asistente	ChatGPT
Redaccion de documentacion	Google Docs
Lenguaje de programacion	Python

Entorno de programacion	Unity