

Instituto Tecnológico y de Estudios Superiores de Monterrey



**Tecnológico  
de Monterrey**

**Modelación del sistema multiagente para simular una intersección controlada por  
señales de semáforos inteligentes**

TC 2008 B: Modelación de sistemas multiagentes con gráficas computacionales

Profesor: Edgar Covantes Osuna

18 de agosto del 2021

Integrantes:

Fernando Doddoli Lankenau

A00827038

Cristóbal Alberto Escamilla Sada

A00827074

Juan Manuel Gómez Ortiz

A00828010

Andrés Guerra Ochoa

A00828452

## **Conformación del Equipo**

### **Cristóbal A. Escamilla Sada**

Fortalezas Soft: Optimista, Trabajador, Bueno trabajando con equipos

Fortalezas Hard: Hábil en la ejecución de algoritmos eficientes utilizando C++.  
Conocimiento de bases de datos SQL y programación en Java.

Áreas de oportunidad: En algunos casos tiendo a ser impuntual y desorganizado; es algo que tengo que mejorar. En cuanto a la programación me gustaría ampliar mi conocimiento de Python ya que no practico mucho el lenguaje.

Expectativas del bloque: Principalmente espero que con este bloque me abra las puertas a poder trabajar con la inteligencia artificial en un futuro. Estoy entusiasmado por aprender sobre los multiagentes así como sobre la matemática detrás de las gráficas computacionales. Esta es una combinación de disciplinas poderosas con las que podré elaborar grandes proyectos en un futuro. Personalmente me encuentro en una etapa de exploración ya que pronto tendré que elegir mi especialidad de carrera. Espero con la ayuda de este bloque poder tomar una decisión fundamentada.

### **Fernando Doddoli Lankenau**

Fortalezas Soft: Disciplina, Creatividad, Ejecución

Fortalezas Hard: Buen conocimiento de React JS, HTML, y CSS. Al igual que frameworks gráficos como Bootstrap.

Áreas de oportunidad: Mejoramiento en conocimientos relacionados con estructura de datos, optimización de algoritmos, y sintaxis de python.

Expectativas del bloque: A través de este bloque espero incrementar mis conocimientos de sistemas multi-agentes hechos con python y unity. Al final del bloque espero poder desarrollar un sistema gráfico multi-agente complejo de manera individual para simular algún tema de mi interés.

### **Andrés Guerra Ochoa**

Fortalezas Soft: Creativo, persistente, disciplinado.

Fortalezas Hard: Desarrollo de Videojuegos en Unity. Conocimientos de Python, C++ y Matlab.

Áreas de oportunidad: Mejorar conocimientos en el Desarrollo Web (HTML y CSS)

Expectativas del bloque: A través de este bloque me gustaría mejorar mis conocimientos de Python y aprender a usar el framework Mesa para poder desarrollar un sistema gráfico multi-agente. Tengo expectativas altas sobre este bloque debido a que se ve que el contenido es muy interesante al igual que el reto en el que voy a poder participar junto con mi equipo.

### **Juan Manuel Gómez Ortiz**

Fortalezas Soft: Dedicado, creativo, motivado

Fortalezas Hard: Desarrollo Web (HTML y CSS). Conocimientos de Python, C++ y Matlab.

Áreas de oportunidad: Mejorar conocimientos en estructura de datos, MySQL y Unity.

Expectativas del bloque: Durante este bloque espero aprender más sobre los sistemas multi-agentes. También espero mejorar mis habilidades con Python y Unity. Estoy emocionado de participar en este bloque y trabajar junto con mi equipo en el reto.

### **Expectativas del equipo:**

Estamos muy emocionados por formar parte de este bloque debido a que el contenido se ve muy interesante y retador. Esperamos mejorar nuestros conocimientos para poder solucionar el reto de la mejor forma posible. Somos un equipo muy comprometido que siempre busca formas de mejorar, y daremos nuestro mejor esfuerzo a través de las diferentes etapas del reto.

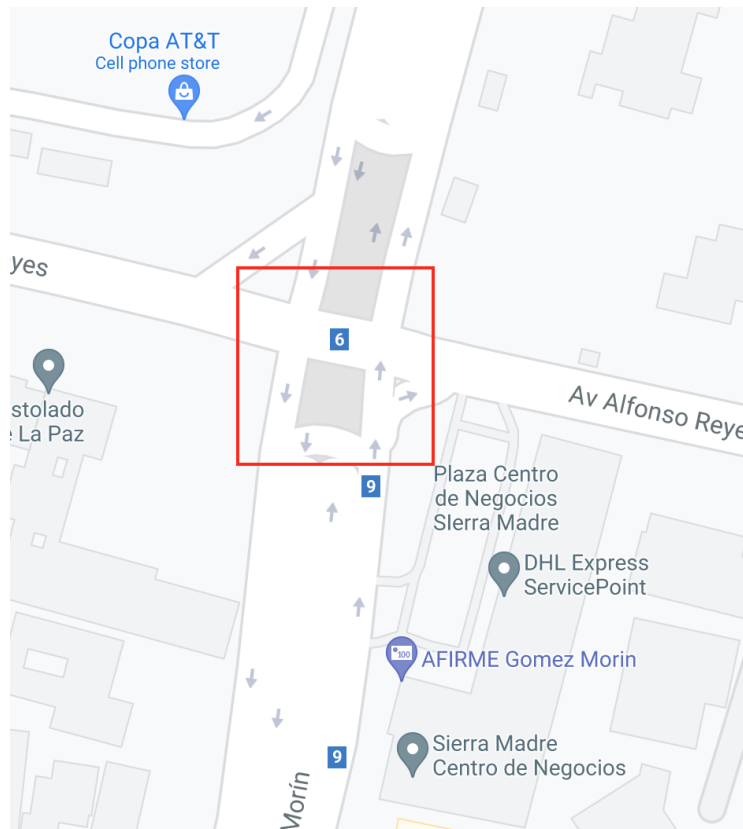
### **Creación de herramientas de trabajo colaborativo:**

Se creó un repositorio en Github llamado [Proyecto-Movilidad-Urbana](#) y una carpeta compartida en Google Drive para colaborar en los archivos necesarios. También se creó chat vía Whatsapp mediante el cual estaremos estableciendo comunicación como equipo.

## Situación Problema

### Descripción del Reto a Desarrollar

Con el objetivo de reducir la congestión vehicular y mejorar la movilidad urbana en México, desarrollaremos una sistema multi-agente donde se modela una intersección cuyo tráfico se controla mediante cuatro semáforos inteligentes. De esta forma, un semáforo podrá determinar el momento y duración de la luz verde. Así evitando posibles accidentes durante el cruce y mejorando la eficiencia vial. Esta simulación gráfica será basada en la actual intersección que conecta a la avenida de Gómez Morín con la avenida Alfonso Reyes, situada en el municipio de San Pedro Garza García. Se seleccionó esta intersección porque hay horas del día donde no hay carros y como conductor te tienes que esperar hasta 2 minutos para que el semáforo correspondiente se ponga en verde.



## Modelo

Las siguientes tres condiciones son necesarias para modelar el sistema:

- Mientras no haya un vehículo cercano, el semáforo estará en luz amarilla.
- Cuando un vehículo se acerque a la intersección, enviará un mensaje con el tiempo estimado de arribo.
- El semáforo dará luz verde al semáforo más cercano y establecerá un programa de luces a partir de ese punto para el resto de los vehículos.

Dicho esto, se necesitarán tres agentes reactivos simples diferentes para modelar el sistema: Carro, Semáforo, y CarrilIntersección, los cuales operarán en la siguiente simulación:

	0	1	2	3	4	5	6	7	8	9
0					A	↑				
1					↓	↑				
2					↓	↑				
3				S1	↓	↑	S3			
4	←	←	←	←			←	←	←	C
5	D	→	→	→			→	→	→	→
6				S4	↓	↑	S2			
7					↓	↑				
8					↓	↑				
9					↓	B				

*Imagen #1: Representación gráfica del modelo.*

En este gráfico representativo, las celdas rosas son semáforos, las grises son carriles, y las negras-grises son agentes de carrilIntersección.

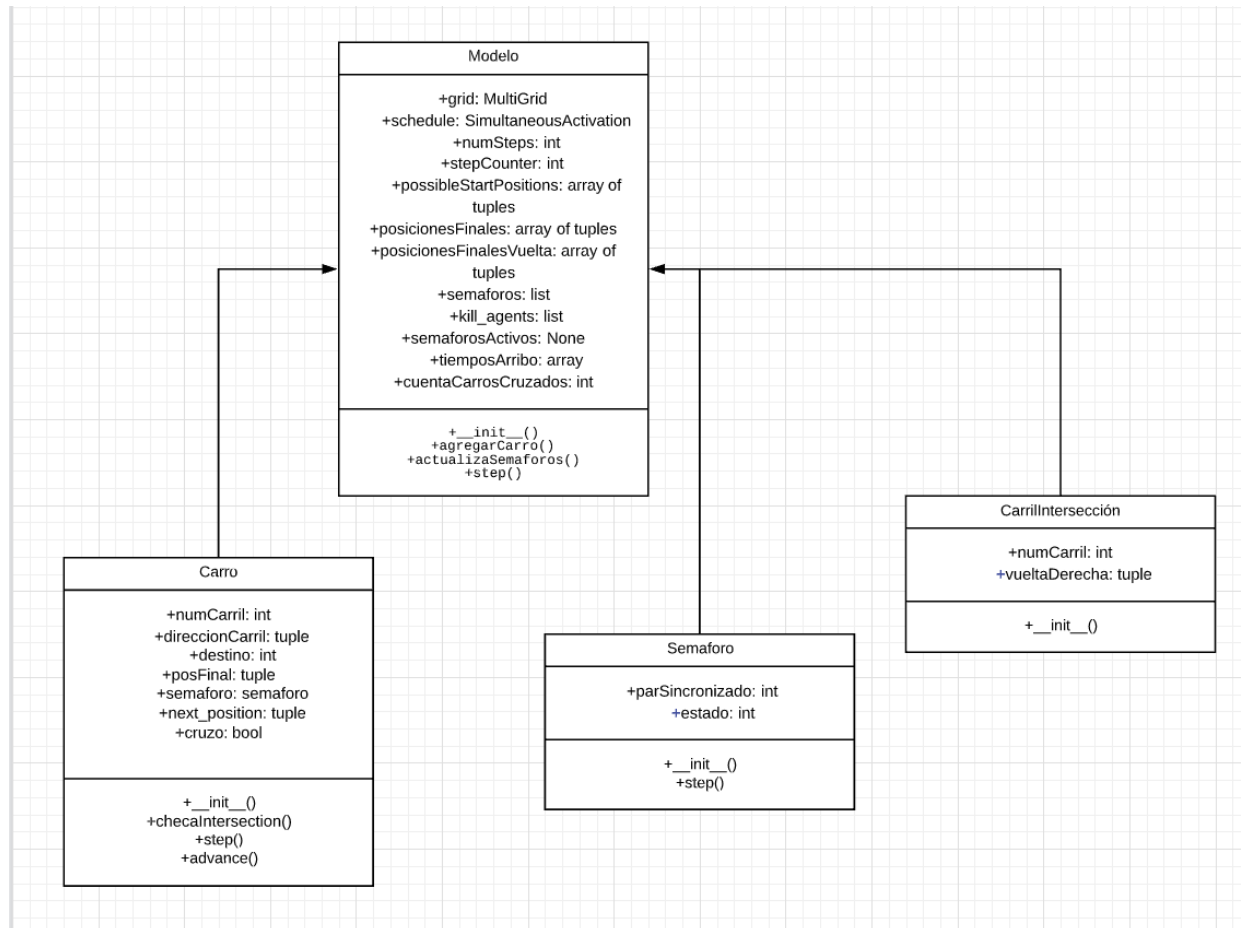
Adicionalmente, en la intersección existen cuatro carriles con direcciones diferentes: ‘A’, ‘B’, ‘C’ y ‘D’. Existe un semáforo por carril. El semáforo ‘S1’ controla el flujo de los carros en el carril ‘A’, el semáforo ‘S2’ controla el flujo de los carros en el carril ‘B’, el semáforo ‘S3’ controla el flujo de los carros en el carril ‘C’ y el semáforo ‘S4’ controla el flujo de los carros en el carril ‘D’. Al llegar al cruce, los carros tienen dos posibles direcciones independientemente del carril en el que se encuentren. La primera posibilidad es seguir derecho en su carril y la segunda posibilidad es dar vuelta a la derecha (perspectiva del carro). Esta lógica se expresa en la siguiente tabla:

<b>Carril Actual</b>	<b>Semáforo Responsable</b>	<b>Posible siguiente carril</b>	<b>Posible siguiente carril</b>
A	S1	A (seguir derecho)	C (vuelta a la derecha)
B	S2	B (seguir derecho)	D (vuelta a la derecha)
C	S3	C (seguir derecho)	B (vuelta a la derecha)
D	S4	D (seguir derecho)	A (vuelta a la derecha)

Dicho esto, los agentes interactúan de la siguiente manera para modelar el sistema. El agente de tipo carro se moverá en uno de los cuatro carriles de la intersección y al llegar al cruce, si el estado del semáforo es verde y no hay carros enfrente avanzará. El CarrilDeIntersección determinará la cómo se moverá el carro al estar en la intersección (vuelva a la derecha o seguir derecho). Finalmente, el modelo utilizará una variable de semáforosActivos que guardará el valor de 0 y 1 dependiendo de en qué carril se puso el carro. Esta variable se utilizará para cambiar del estado de los semáforos con valorParSincronizado igual al semáfosActivos. Lo cual permitirá cambiar la luz del semáforo

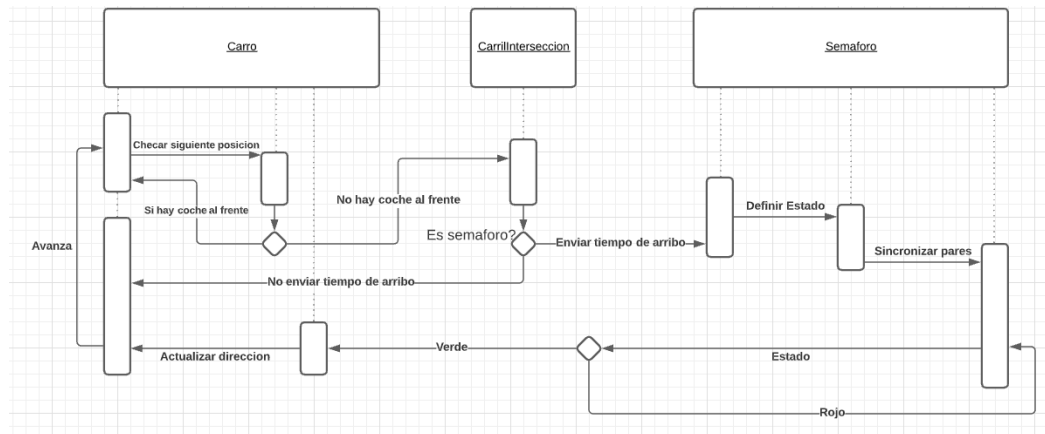
de forma correcta y mantener un programa eficiente que permita a los semáforos sincronizarse correctamente.

A continuación, se muestra gráficamente el funcionamiento de los diferentes tipos de agentes involucrados a través del uso de diagramas UML de clase.



*Imagen #2: Diagramas de Clase para los Agentes Involucrados*

Para que el modelo funcione correctamente, es necesario que cada uno de estos agentes tenga un comportamiento de involucramiento grupal. Cada agente tiene un tipo de repercusión en otro agente, y esto es lo que trae como consecuencia un modelo real y coordinado. Para representar este tipo de interacciones, utilizaremos un diagrama de secuencia UML



*Imagen #3: Diagrama de Secuencia entre agentes.*

En este diagrama (Imagen #3) podemos observar cómo se comportan los agentes del modelo, cuáles son los mensajes que se envían entre sí, y cuales son las respuestas que obtienen de sus solicitudes correspondientes. El carro por ejemplo, necesita estar constantemente consultando si su casilla de enfrente es una intersección, para que de esa forma pueda analizar cuál es el estado del semáforo y poder decidir si avanzar o no. Estos comportamientos suceden de manera simultánea e iterativa, por lo que esta representación aplica para cada agente, ya sea del mismo tipo o no, que hace parte del modelo.