



# Tecnológico de Monterrey

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

***Campus Estado de México***

***Modelación de sistemas multiagentes con gráficas computacionales***

Evidencia 2. Avances y presentación del reto

TC2008B

Grupo: 301

Equipo: 4

**Integrantes del equipo:**

*Aleny Sofia Arévalo Magdaleno | A01751272*

*Luis Humberto Romero Pérez | A01752789*

*Valeria Martínez Silva | A01752167*

*Pablo González de la Parra | A01745096*

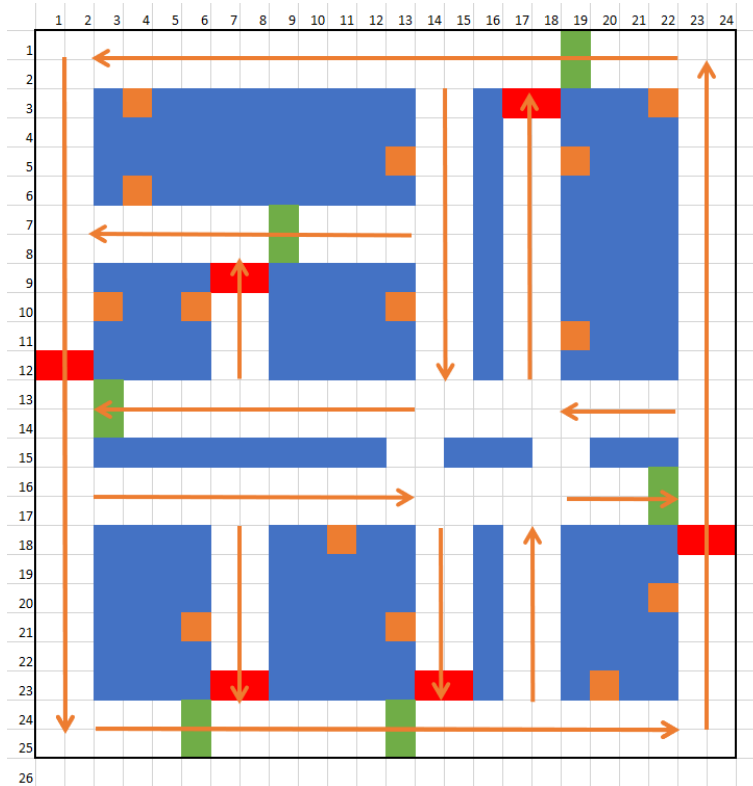
**Fecha de entrega:**

**2 de Diciembre de 2022**

## Repositorio de Github

<https://github.com/pablogonzalezdelaparra/MovilidadUrbana>

### Descripción del medio ambiente



De acuerdo con los requerimientos de la situación problema, el ambiente que elegimos desarrollar simula el tránsito vehicular en una ciudad ficticia en México. Esta consiste en una calle principal la cual rodea el perímetro de la ciudad, y una rotonda en el centro que dirige y habilita el tránsito a la calles centrales. A continuación se mencionan ciertas características relevantes del ambiente:

- 1 cuadrícula de 24x25
- 1 calle principal
- 216 edificios (celdas de obstáculo)
- 14 destinos (celdas de destino)
- 1 rotonda
- 12 semáforos (24 celdas)
- 6 intersecciones con semáforos o 12 pares de semáforos
- $k$  cantidad de carros iniciales
- 1 carro cada  $n$  tiempo (o step)

El diseño de nuestro ambiente se encuentra construido con el fin de promover la interacción entre agentes “Carro” y agentes “Semáforo”. Esto debido a que, la lógica principal que queremos simular y modelar, son los embotellamientos o la carga vehicular pesada entre estos agentes de manera constante. Aunado a esto, estos agentes nos permiten modelar un sistema capaz de implementar una optimización del problema planteado. A continuación se listan ambos agentes.

- Agente “Carro”

- **Agente “Semáforo”**

De acuerdo con nuestra implementación, además de los agentes “Semáforo”, existen diversos agentes (de MESA) los cuales no poseen ninguna funcionalidad, debido a su falta de autonomía. Por lo que, a diferencia de otros, los agentes “Carro” poseen dos maneras de integrarse al sistema. Se mencionan a continuación:

- Se pueden agregar o generar  $k < 22$  agentes “Carro” antes de comenzar la simulación
- Se pueden ir agregando o generando 1 agente “Carro” cada  $n$  tiempo (o step)

Esto es debido a que, la implementación del ambiente que decidimos utilizar solamente permite la generación de agentes “Carro” dentro de los estacionamientos (o destinos), al igual que en las celdas sobre las esquinas de la cuadrícula, con el fin de simular de manera más realista los casos que implican salir de un estacionamiento con una gran carga vehicular, al igual que incorporarse a una avenida o calle principal.

Por lo que, los agentes “Carro” pueden ser agregados a la cuadrícula siempre y cuando exista al menos celda vacía y disponible para ejecutar la acción. Aunado a esto, la cantidad de carros que se pueden agregar cada  $n$  tiempo también puede ser modificada si se requiere.

Aunque las características y detalles de ambos agentes se mencionan a continuación, es importante comentar que la interacción entre ambos agentes es mutua, y que no existe ninguna variable externa que afecte o influya en el comportamiento de ambos. Al enfocarse de manera más específica en las intersecciones controladas por señales de semáforos inteligentes, estas presentan un elemento importante para su funcionamiento.

A diferencia de las demás celdas de tránsito, en las 4 celdas de intersección (pasando ambos semáforos), estas promueven la toma de decisión sobre cuál es el camino correcto que el agente “Carro” debe de tomar con el fin de llegar a su destino lo más rápido posible. En temas de los semáforos, su funcionalidad al igual que su lógica interna se describe a lo largo de este entregable. A continuación se describen las características del ambiente.

- **Mayormente inaccesible**

El ambiente es mayormente inaccesible debido a que el agente no tiene acceso al estado completo o total del ambiente. Es decir, se requiere de un estado interno con el fin de poder tomar una decisión de acción. Dentro de nuestro ambiente se considera que este es mayormente inaccesible debido a la limitación que poseen los sensores de ambos agentes principales. En el caso del agente “Carro”, esto es notable cuando se analiza que este solo puede observar las celdas a su alrededor (o vecinos). Es decir, cualquier otro aspecto del ambiente se encuentra oculto para la toma de decisiones y el conocimiento que posee. Cabe recalcar que, aunque posee una serie de celdas la cual determinan la mejor ruta para llegar a su destino, los sensores de este se limitan a sus vecinos.

En el caso del agente “Semáforo”, este posee un poco más de información sobre su entorno, debido a la comunicación directa que tiene entre agentes del mismo tipo, y los agentes “Carro”. Y aunque conozca el estado y ciertas características de los otros agentes, las secciones de las calles e incluso ciertas intersecciones se encuentran ocultas a estos, limitando el razonamiento al momento de tomar decisiones. Por lo que, aunque no exista un promedio exacto de la accesibilidad de nuestro ambiente, se puede estimar que se encuentra entre un rango de 45% a un 55%.

- **Mayormente determinista**

El ambiente es mayormente determinista debido a que el estado actual y las acciones realizadas por el agente llegan a determinar el estado siguiente del ambiente. Es decir, se puede estar al tanto del mayor número de aspectos accesibles del ambiente. Esto es notable cuando se observa el comportamiento y la interacción que poseen ambos agentes. En el caso del agente “Carro”, la posición actual, al igual que la presencia de otros agentes a su alrededor determina de manera objetiva su siguiente movimiento. Esto considerando factores como su destino, la ruta que tiene, la prioridad en los cruces, entre otros.

En el caso del agente “Semáforo”, el incremento en el número de carros en su fila afecta directamente su comportamiento con el fin de intentar disminuir el tránsito vehicular. Es considerado mayormente determinista debido a la interacción directa, o causalidad que existe entre los agentes. Es importante recalcar que, al no tener en cuenta la posición o estado del ambiente cuando se generan o eliminan agentes “Carro”, se trabaja con un cierto grado de incertidumbre, por lo que no puede ser considerada completamente determinista.

- **No episódico**

El ambiente es no episódico debido a que las acciones del agente no se dividen en episodios o ciclos previamente establecidos. Es decir, el agente tiene que pensar de manera adelantada para actuar. En este caso, el comportamiento de los agentes al igual que su interacción con el ambiente no se encuentran ligados de manera episódica. Esto debido a que las acciones que se realizan dependen completamente de las acciones producidas anteriormente. Dentro de nuestro ambiente, las acciones de un agente “Carro” dependen mayormente en qué dirección tiene que avanzar de acuerdo a la ruta calculada en cada turno, mientras que en los agentes “Semáforo”, el cambio en su estado (color) depende completamente del estado anterior, al igual que factores previamente mencionados como el número de carros en su fila, con el fin de alternar entre ambos.

Esto es incluso más notable cuando se analizan los agentes “Carro”. Debido a que si su comportamiento fuese episódico, no tendrían la necesidad de analizar y decidir sobre su entorno (o movimiento) cada que avancen una celda (porción de una calle), o incluso cuando tengan que evitar chocar. Los agentes deben de pensar por adelantado con el fin de determinar su comportamiento.

- **Dinámico**

El ambiente es dinámico debido a que existe la posibilidad de que el ambiente presente modificaciones mientras que el agente delibera. Es decir, el ambiente cambia con el paso del tiempo. Debido a la naturaleza de ciertas simulaciones, a veces se considera a un ambiente como estático debido a la posibilidad de manipular el tiempo dentro de esta. En nuestro caso, nuestro ambiente es dinámico debido a que este cambia respecto al tiempo.

Por lo que mientras los agentes deciden su siguiente movimiento (o deliberan), el ambiente se encuentra cambiando debido a la deliberación de los demás agentes que lo constituyen. Por ejemplo, cuando un agente “Carro” se encuentra decidiendo su siguiente movimiento, o analizando si tiene la prioridad mayor con el fin de avanzar a una celda, los agentes “Semáforo” ya tuvieron la posibilidad de cambiar su estado (o recalcular el siguiente), lo que modifica el comportamiento de estos agentes en cada turno.

- **Discreto**

El ambiente es discreto debido a que existe un número limitado de acciones y percepciones. Es decir, es posible listar o enumerar estos hechos en el ambiente. Esto es más notable en simulaciones las cuales prohíben la utilización de rangos (valores continuos) para determinar una o muchas variables. Dentro de nuestro ambiente, este se considera discreto debido a que se pueden conocer todas las acciones o percepciones que un agente pueda llegar a ejecutar. En el caso de los agentes “Carro”, se conocen los posibles movimientos que cualquier agente de este tipo puede llegar a realizar, como es el avanzar, detenerse, salir y entrar a un estacionamiento (o destino).

Por otra parte, los agentes “Semáforo”, al poseer solamente una acción que cambie su entorno que es el alterar su estado (color), es más notable observar que estas acciones se pueden listar de manera concisa. Aunado a esto, debido a la naturaleza objetiva de nuestro ambiente, no existen variables externas las cuales afecten o añadan un grado de complejidad mayor a la cantidad de percepciones que pueda llegar a poseer cada agente.

### **Explicación detallada de los agentes**

Como se mencionó anteriormente, existen 2 agentes principales dentro de nuestra simulación. El agente “Carro” y el agente “Semáforo”. Cabe mencionar que, aunque en MESA se denominan agentes a elementos propios del ambiente como los edificios, calles y estacionamientos, estos no son agentes en el sistema debido a su falta de autonomía. A continuación se describen las características y detalles de ambos.

- **Agente “Carro”**

El agente “Carro” (o denominado simplemente carro para simplificar), es uno de los agentes responsables en la simulación del tráfico vehicular dentro de nuestra situación problema. Todos los carros comienzan en un lugar de estacionamiento o esquina de la cuadrícula, con el fin de simular de manera realista su trayecto. Estos solo pueden aparecer en una celda de acuerdo a las siguientes condiciones.

- Si la celda está vacía (es decir, no hay otro carro ocupando ese lugar).
- Si no hay un carro en la primera celda (al salir), el cual su destino sea ese mismo lugar que su celda (cuando sale del estacionamiento).

Con esto no solo garantizamos que existan carros atrapados en estos lugares de destino, sino que también permite demostrar la utilización de prioridades dependiendo del lugar en el que un carro se encuentre, explicadas más adelante. Aunque los carros puedan aparecer una celda, primero se busca que aparezcan en una celda de estacionamiento. Si esto no es posible, el carro intenta nuevamente aparecer en una celda de la esquina. Lo primero que realiza un carro antes de salir de su celda inicial, es calcular la ruta más corta que tiene que tomar para llegar a su destino. Esto lo realiza por medio del algoritmo A\* (o Astar), el cual retorna una lista de tuplas, cada una con la coordenada siguiente a la que se tiene que mover. El algoritmo de búsqueda se forma a partir de lo siguiente.

- $h : |x_1 - x_2| + |y_1 - y_2|$ 
  - Calcula el costo del posible movimiento obteniendo la distancia Manhattan entre el punto actual y el destino.

- $g\_score: \geq 1$ 
  - Determina la distancia entre la celda elegida y el inicio
- $f\_score: g + h$ 
  - Valor que determina el mejor movimiento.

Al realizar este cálculo, el carro siempre elige el primer valor que se devuelve (sobre la lista de tuplas o coordenadas), el cual representa el primer paso que tiene que tomar para llegar a su destino. Decidimos utilizar el A\* debido a la efectividad de la función heurística para calcular la distancia entre celdas de una cuadrícula. Aunado a esto, este no solo nos permite calcular una ruta hacia el destino, sino que esta ruta sea la más rápida posible. Existen varias razones por las que un carro puede o no avanzar a esta siguiente celda. Se mencionan a continuación.

1. Hay un carro en la celda siguiente a la que se quiere avanzar.
2. Se encuentra encima de un semáforo el cual está en rojo.
3. Hay un carro en la celda siguiente en la que se quiere avanzar.
4. La celda siguiente está apartada por un carro con mayor prioridad.

Esta última se refiere a la prioridad que se le otorga a los carros de acuerdo con la posición en la que se encuentran. Se listan a continuación.

- Prioridad máxima (3): El carro aparta una celda mientras se encuentra queriendo salir de un estacionamiento.
- Prioridad media (2): El carro aparta una celda mientras se encuentra en la calle principal (la calle que rodea la ciudad por fuera).
- Prioridad baja (1): El carro aparta una celda mientras se encuentra en las calles centrales.

Cuando dos o más carros, apartan una misma celda a la que se quieren (o deben) mover, estas prioridades determinan a quien se le tiene que dar permiso, y quien se tiene que esperar al siguiente movimiento. Entre más alta la prioridad, mayor decisión. Aunado a esto, este comportamiento se asemeja a la vida real, cuando un carro se quiere incorporar a avenidas principales, entre otros casos. En el caso de que existan carros con la misma prioridad queriendo moverse a la misma celda, se le da el paso al primero que la apartó.

Otra característica importante de los carros es que estos siguen la dirección o el flujo de la calle. Por ejemplo, si la celda tiene la dirección “ $\wedge$ ”, la única posibilidad del carro, es aumentar su  $y$  siempre y cuando el camino se encuentre libre. Esto cambia al entrar a las intersecciones y al momento de formarse (que se explica más adelante) .

Las intersecciones (o los espacios entre semáforos, y dentro de la rotonda ) tienen la dirección “ $\times$ ”. Esto significa que el carro puede moverse a cualquier celda a su alrededor. Aunque esta implementación suene un poco ambigua, cabe recalcar que, como un carro siempre se encuentra siguiendo el camino más corto hacia su destino (debido al A\*), se utilizan estas intersecciones estrictamente para cambiar de carril.

Esto es de suma importancia porque, si un carro se encuentra en el carril exterior, y quiere llegar a un estacionamiento que se encuentra en el carril interior, puede utilizar la intersección con el fin de acercarse y finalmente dar vuelta a su destino. Otro de los problemas con los que nos enfrentamos fueron los embotellamientos dentro de estas intersecciones, debido a los cambios constantes de carril entre carros. Esto lo solucionamos de la siguiente manera.

Si el “carro 1” aparta una celda, y el “carro 2” está parado en esa celda apartada, y el “carro 2” también está apartando la celda en la que se encuentra el “carro 1”. Uno de los dos decide moverse a una celda disponible (la cual puede no llegar a ser parte de la ruta más corta definida por el A\* con el fin de aliviar la congestión y permitir el flujo de vehículos. Y finalmente, el carro que se tuvo que mover a una celda disponible, vuelve a calcular la ruta más corta para llegar a su destino, regresando a su trayecto normal sin ningún problema.

Estas técnicas que evitan la congestión vehicular son esenciales para nuestra simulación, debido a que podemos agregar una cantidad mayor de carros, garantizando su funcionamiento correcto en cualquier escenario. Aunque ya se mencionó el caso de los semáforos en rojo, es importante recalcar que un coche se detiene siempre que el semáforo se encuentra en rojo, y avanza cuando este cambia a verde. El objetivo es disminuir en cantidad y presencia las filas que se crean de los carros esperando avanzar por el semáforo, con la lógica de los agentes “Semáforo”, explicado a continuación.

Finalmente, cuando los carros tienen a su destino como vecino directo (normalmente a sus lados), estos entran al estacionamiento y en el siguiente paso desaparecen. Estos no desaparecen al instante con el fin de demostrar que se han estacionado, simulando una vez más un comportamiento cercano a la realidad. Una de las mejoras implementadas para esta última entrega consiste en el cambio de carril debido al tráfico vehicular. Previamente se comentó que un carro solo puede cambiar carril, o sentido de celda cuando se encuentra en una intersección. Sin embargo, con el fin de optimizar el tránsito entre estos agentes, un carro puede cambiar de carril cuando existan uno o muchos carros en su siguiente celda. Es decir, cuando el carro se encuentra formado.

Este cambio de carril permite que los carros no se formen en una fila, sino que utilicen dos filas (o par de celdas) con el fin de reducir la longitud de carros y su presencia en otras calles. De igual manera, se implementó que un carro pueda salir de su trayectoria (o del camino delimitado por el A\* siempre y cuando su trayectoria actual se encuentre ocupada por un gran número de carros en su camino. Estos dos últimos cambios optimizan de manera exponencial la eficiencia en el tránsito vehicular, debido a que ambos se encargan de poder utilizar el mayor número de celdas posibles, con el fin de compactar a los agentes y así evitar que el tráfico en una sección se pase o desborde a cualquier otra sección.

Otra optimización que se realizó en comparación con el entregable anterior fue que la trayectoria que toma cada carro con el fin de llegar a su destino de manera más rápida y eficiente se calcula sólo cuando se realiza un cambio. De manera más detallada, cuando un carro aparece, a este se le asigna un propiedad con el camino completo para llegar a su destino, lo que permite utilizar esta variable en todas las funciones, eliminando la necesidad de llamar a la función A\* cada turno, por cada carro. Aunado a esto, cuando un carro requiere cambiar de carril debido a la congestión vehicular, se vuelve a calcular la trayectoria una sola vez, y se actualiza la propiedad del agente. Con esto garantizamos una menor exigencia en términos de procesamiento en cada turno del sistema.

### **Agentes “Semáforo”**

Los agentes “Semáforo” (o denominado semáforo para simplificar) es el segundo agente que interactúa y afecta nuestro ambiente. Los semáforos están encargados de optimizar el tráfico vehicular con el fin de evitar y disminuir embotellamientos, al igual que permitir el flujo

constante de tráfico en las intersecciones. A continuación se explica el comportamiento de estos agentes, al igual que su funcionalidad de manera detallada.

Los sensores de estos agentes no solamente permiten un mayor rango de detección sobre su ambiente, sino que al igual promueven una comunicación activa y colaborativa con los demás agentes de este tipo.

Antes de comenzar la simulación, estos agentes aparecen siempre en coordenadas o posiciones fijas de acuerdo con el diseño del mapa. Los semáforos (como en la vida real) tienen una dirección, lo que significa que regulan el tráfico solamente de la dirección o calle que tienen asignada. Y es esta dirección la que permite la cooperación y coordinación sobre el tráfico vehicular. Existen dos tipos de semáforos. Se mencionan a continuación.

- Semáforo vertical
  - Regula la circulación de vehículos izquierda o derecha de este.
- Semáforo horizontal
  - Regula la circulación de vehículos arriba o abajo de este.

Es importante distinguir entre estos tipos de estos semáforos con el fin de mantener un equilibrio entre ambos. Esto porque, al iniciar la simulación (por el momento), todos los semáforos, sin importar su tipo comienzan en rojo, sin embargo en el primer paso calculan el estado que mayor beneficie al tráfico vehicular, el cual se explica a continuación. Estas direcciones o tipos de semáforos son de suma importancia para el funcionamiento de las intersecciones.

El comportamiento de los agentes en la situación problema, consiste en que los semáforos cuenten o detecten el número de carros formados, y con esto determinar si conviene para el tránsito vehicular, ponerse en verde o rojo. Mientras que entre las parejas de semáforos, cuadrantes y el mapa completo exista siempre un porcentaje de 50% verdes y 50% rojos. A continuación se explica el procedimiento.

- 2 semáforos (celdas) comparten un número único de parejas.
- 4 semáforos (celdas) comparten un número único de cuadrante.
- Semáforos con el mismo número de pareja, cuentan y suma el número de carros en ambas filas
- Semáforos con el mismo número cuadrante, comparan la cantidad de carros de su pareja con la otra pareja
- La pareja con mayor número de carros se pone en verde, mientras que la pareja con el menor número de carros se pone en rojo

Esta lógica permite que las 2 parejas de semáforos en una misma intersección nunca tengan el mismo color (todos verdes o todos rojos). Esto es algo crucial no solamente debido a que este comportamiento no sucede en la vida real, sino que también este aumentaría la presencia de accidentes entre cada intersección. Sin embargo, la funcionalidad de contar los carros formados en su fila puede sonar un poco ambiguo. A continuación se explica el proceso.

- Se calcula la dirección del semáforo de acuerdo a su tipo.
  - Semáforo vertical: Derecha o izquierda.
  - Semáforo horizontal: Arriba o abajo.
- Se cuenta el número de carros en la fila o columna sobre la dirección de la vialidad, hasta encontrar un edificio o estacionamiento, o hasta llegar a un límite de celdas calle la cual puede ser modificado.



- Se suman el número de carros de la misma pareja.
- Se obtiene la cantidad de carros de la mitad de una intersección.
- Se repite el proceso desde el primer paso para la otra pareja

Como se pudo observar, existe una comunicación directa entre los semáforos con el fin de optimizar el tránsito para cualquier cantidad de carros que pueda llegar a presentarse. Una de las implementaciones o mejoras que se implementaron en este entregable consiste en el tiempo que los semáforos mantienen su estado. Esto debido a que, consideramos dos aspectos importantes al realizar la implementación final.

1. Para simular la vida real, los semáforos no pueden mantener un mismo estado el doble o triple de tiempo que su contraparte.
2. Los semáforos requieren cambiar constantemente de estado con el fin de evitar un embotellamiento mayor en una de sus dos filas, por el tiempo de espera.

Como se comentó previamente, aunque los semáforos decidan el tiempo en el que deben de mantener su estado, el cálculo del tiempo se calcula de la siguiente manera.

- $número\_de\_pasos = floor((carros\_por\_fila / total\_de\_carros) * n)$ 
  - Donde  $n$  es el número de pasos para volver a calcular los estados del semáforo

De acuerdo con nuestra implementación, cuando un carro posee más carros que su pareja, a este se le asignan  $k$  segundos sobre un límite de  $n$  pasos. Es decir, si un semáforo tiene 7 carros, mientras que el otro tiene 12, se le asigna el siguiente valor de verde al último semáforo.

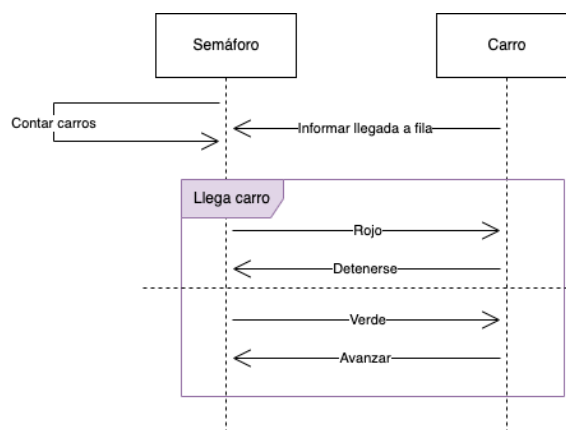
- $(12/19)*10 = 6$

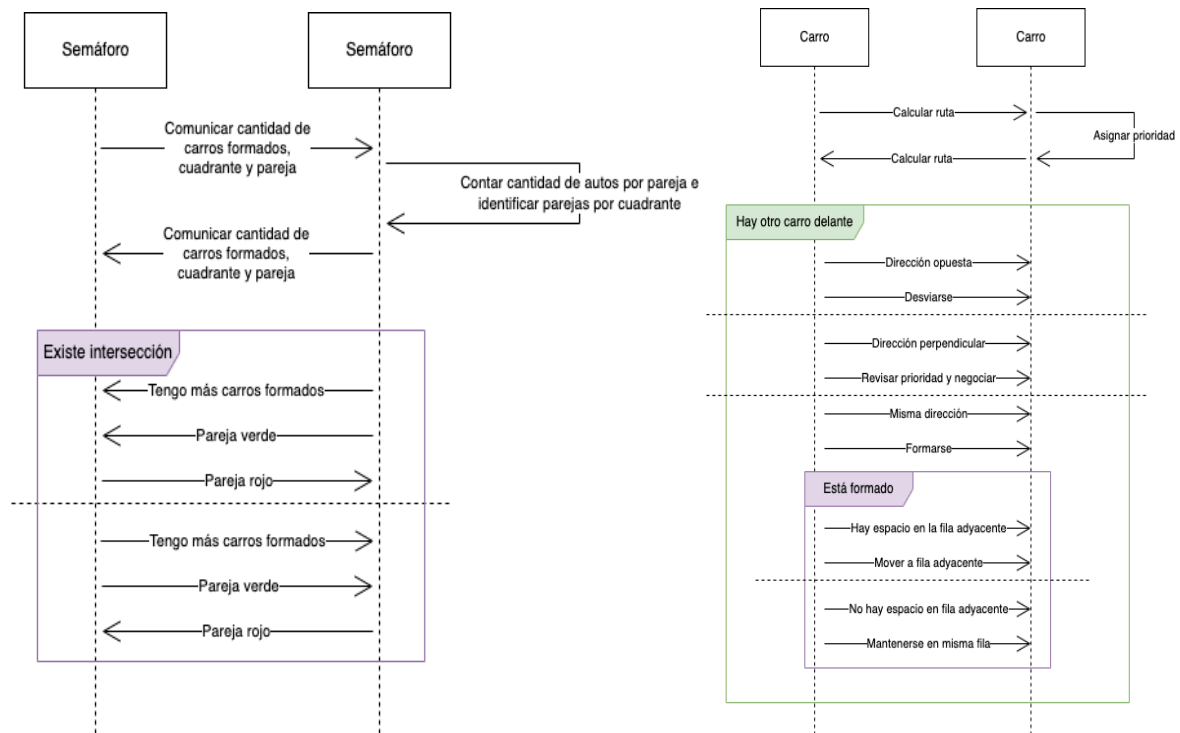
En este caso, determinamos que el último semáforo tiene un porcentaje de 63%, mientras que el otro tiene el restante 37%. Por lo que, entre 10 segundos, el último semáforo se mantendrá prendido 6 segundos, mientras que el otro utiliza el tiempo restante. Este valor de 10 segundos puede ser modificado, con el fin de probar distintos tiempos de activación y determinar el más apropiado de acuerdo al número de carros. Este cálculo se realiza cada  $n$  segundos con el fin de ahorrar procesamiento, al igual que evitar el comportamiento que los semáforos cambien de estado cada segundo, presentando un comportamiento intermitente.

## Diagramas de agente

Carro		Semáforo	
Grupo: Grupo de carros Rol: Conductor		Grupo: Grupo de semáforos Rol: Controlador	
Eventos: Semáforo rojo Semáforo verde Detección de carro Detección de obstáculos Celda de intersección		Servicio: Controlar tránsito vehicular	
Evento - Acción: Semáforo rojo: Detenerse Semáforo verde: Seguir Detección de carro: Negociar Detección de obstáculos: Esquivar Celda de intersección - Calcular posición (orientación) de destino		Protocolo: Control vehicular óptimo	
		Eventos: Llegada de auto (cambio en el número de autos en la fila) Cambio de estado en otro semáforo	
		Metas: Controlar y optimizar el tránsito vehicular Plan: Coordinar estado (color) de todos los semáforos Acciones: Cambiar estado (color) Comunicar con otros semáforos (pareja y cuadrante) Contar número de carros en la fila Planificar sistema óptimo de luces	
		Conocimiento: Cantidad de carros en su fila (formados) Estado (color) de todos los semáforos	

## Diagramas de protocolos de interacción





**Vídeo describiendo el proceso de instalación (desde cero), configuración (si así se requiere) y ejecución de la simulación**

Link de Google Drive:

<https://drive.google.com/file/d/1uRKgEApJAt6t68O0JjxohsZSbwxwVZR/view?usp=sharing>

Link de Youtube: <https://youtu.be/qSi-w2LYf4c>

**Vídeo describiendo la funcionalidad y ejecución de simulación**

Link de Google Drive:

<https://drive.google.com/file/d/1gLPM7o5gxMZayPIWvziM2JNia90K5PXp/view?usp=sharing>

Link de Youtube: <https://youtu.be/vao5CNTz0x8>