

## 

## 

## **Reto: Modelación de sistemas multiagentes con gráficas computacionales**

Alejandro Melendez Torres - A00832494

Thomas Freund Paternostro - A00831997

Mauricio Cantú Torres - A01633805

Patricio Mendoza Pasapera A00830337

Jueves 18 de Agosto de 2022 Monterrey, N.L.

Diseño de un sistema multiagente

# Índice

Contenido

[Índice](#_heading=h.gjdgxs) **2**

[**1. Introducción**](#_heading=h.ktl6cwe5lavv) **3**

[**2. Créditos**](#_heading=h.3dy6vkm) **3**

[**3. Contexto y Problema**](#_heading=h.3nf89oeq4h2v) **3**

[**4. Objetivos generales**](#_heading=h.3ukwbterpw3s) **3**

[**5. Restricciones**](#_heading=h.tdgwgw9p0qi0) **3**

[**6. Historias de usuario**](#_heading=h.xtxm07siw14f) **4**

[6.1 Funcionales](#_heading=h.bchku5wzktt) 4

[6.2 No funcionales](#_heading=h.j13u99zdwxed) 4

[**7. Descripción del sistema Multiagente**](#_heading=h.t9ro865y5r6a) **4**

[7.1 Modelo de los Agentes](#_heading=h.bz2xyktfurgd) 4

[7.2 Modelo del escenario](#_heading=h.5v206iblzhq3) 5

[7.3 Modelo de la Interacción](#_heading=h.pu9oznalct0o) 5

[**8. Modelación 3D**](#_heading=h.wfhkwea94sso) **5**

[8.1 Descripción de la escena a modelar](#_heading=h.tpy55uxntcnh) 5

[8.2 Descripción de Componentes Gráficos](#_heading=h.obhdxlvmwwgm) 5

[8.3 Descripción de Prefabs](#_heading=h.igk5wsf43wx8) 6

[8.4 Descripción de Scripts](#_heading=h.lt9wj827ohm4) 6

[**9. Entregables de administración de proyecto**](#_heading=h.xkxadafxbp1u) **6**

[**10. Referencias**](#_heading=h.qjdss5edg9x) **6**

# 

# 1. Introducción

La situación problema es poder proponer una mejora al tránsito en una intersección muy transitada tanto por coches como por personas. Al introducir un agente inteligente que en este caso sería un semáforo. Esta intersección ha tenido muchos problemas de ineficacia en el ámbito de movilidad en los últimos meses, por lo que nos contrataron a nosotros para hacer una simulación del lugar y determinar qué es lo que ocasiona estos atascamientos y, si se puede, hacer alguna mejora. Para realizar esto, por cuestión de practicidad y tiempo, se realizará una simulación donde se podrá apreciar cómo este sistema afecta al tráfico de la intersección.

# 2. Créditos

Alejandro Melendez Torres Diseñador de gráficos

Mauricio Cantú Torres Negociador

Thomas Freund Paternostro SCRUM Master

Patricio Mendoza Pasapera Programador

# 3. Contexto y Problema

La movilidad urbana es un factor fundamental para el desarrollo de cualquier ciudad, sin embargo, a medida que aumenta el número de población en las ciudades, se vuelve más complicado gestionar y optimizar el proceso de movilidad en las ciudades. Pero, también han surgido tecnologías que permiten la generación de propuestas que buscan mejorar la fluidez del movimiento de las personas dentro de una ciudad, independientemente del medio de transporte que decidan utilizar. Para este reto, se busca mejorar la manera que interactuan los diferentes agentes al introducir cierta inteligencia al semaforo una intersección de modo que los semáforos cambien de color en base a diferentes factores. Al hacer que este pueda determinar el tiempo de cruce más óptimo basado en la cantidad de peatones y/o de vehículos en los pasos peatonales y en las intersecciones respectivamente. Con el objetivo de optimizar la movilidad de los coches y los peatones que están presentes en la intersección.

# 4. Objetivos generales

Queremos demostrar que al introducir un semáforo inteligente en este sistema; se pueda eficientar la movilidad de una intersección en por lo menos un 20% en tiempos. Al hacer que este pueda determinar el tiempo de cruce más óptimo basado en la cantidad de peatones y/o de vehículos en los pasos peatonales y en las intersecciones respectivamente. Podremos determinar el porcentaje de eficiencia al comparar a nuestro agente inteligente con un semáforo que controla el flujo, tanto del tráfico peatonal como del tráfico vehicular, con un tiempo constante.



# 5. Restricciones

Dado que la intersección tiene forma de cruz, los coches tendrán la capacidad de dar vueltas y transitar “libremente” por la simulación.

El proyecto se restringe al uso de Unity (para poder graficar y simular el reto), Python (para escribir el programa multiagente), MESA (librería de python para agentes), C# (lenguaje principal en unity), Visual Studio Code (el IDE), entre otras en un plazo máximo de 5 semanas.

Estamos usando una metodología SCRUM para poder realizar la planeación, ejecución y replanteamiento constante y ágil.

Sin embargo, sí existen ciertas restricciones en nuestro modelo que vale la pena mencionar. Dichas restricciones estarán clasificadas por agente y se describen a continuación:

**Coches:**

* Bajo ninguna circunstancia podrán dar vuelta hacia la derecha (relativa a la calle en la que se encuentren).
* Solo los coches en el carril izquierdo pueden dar vuelta a la izquierda. Los demás vehículos tienen que ir recto.
* Todos los coches irán a la misma velocidad.
* Los coches no pueden cruzar un semáforo en rojo (como lo harían algunas personas en la vida real).
* Un coche esperando a que el semáforo se ponga en verde tiene que estar esperando en el lugar designado ANTES del cruce peatonal.

**Peatones:**

* Los peatones sólo pueden cruzar la calle usando los cruces peatonales (caminar sobre ellos todo el tiempo).
* Ningún peatón puede estar en la calle cuando el semáforo correspondiente a su cruce se ponga en verde.
* Dado a que, gracias al diseño de nuestra intersección, un peatón puede llegar a su destino cruzando solo una vez la calle; la simulación estará limitada a que un peatón solo pueda cruzar la calle una vez y no tomar varios cruces para llegar a su destino.

**Semáforos:**

* El estado de un semáforo (verde y rojo) estará determinado por el tiempo y por la presencia de peatones esperando en la calle.
* Si no hay peatones esperando, el semáforo se pondrá en verde.
* Si hay peatones, empezará un temporizador para regular el estado.
* Un semáforo solo puede tener un estado a la vez.

**Modelo:**

* La simulación tendrá un número máximo de agentes en la pantalla (por definirse).
* La simulación durará un tiempo determinado y luego se podrá volver a repetir.

# 6. Requerimientos

## 6.1 Requerimientos funcionales

**Semáforo Inteligente:**

1. El semáforo tiene en cuenta el tiempo transcurrido donde el tiempo del estado rojo y verde varía dependiendo de la cantidad de peatones y vehículos esperando y el estado amarillo es siempre constante.
2. El semáforo inteligente cambia de color dependiendo del numero de personas, vehiculos y el tiempo transcurrido.
3. El semáforo puede cambiar a 3 colores distintos (verde, amarillo y rojo).

**Carros:**

1. Un carro puede acelerar si su semáforo está en verde y este tiene que moverse de manera aleatoria de un punto A a un punto B
2. Cuando el coche llega a su punto B, este se destruye.
3. Si un carro está acelerando y está en el carril correcto, podrá dar vuelta a la izquierda.
4. Los vehículos empiezan a desacelerar, cuando el semaforo cambia de verde a amarillo, cuando este cambie a rojo.

**Peatones:**

1. Los peatones caminan del punto A al punto B cuando está en rojo el semáforo.
2. Los peatones dejan de caminar y esperan en su cruce cuando el semáforo cambia a verde.
3. Los peatones sólo cruzan en el paso peatonal.

## 6.2 Requerimientos no funcionales

**Semáforo:**

1. El color verde indica que los carros pueden cruzar y los peatones deben de parar.
2. El color rojo indica que los carros deben de parar y los peatones pueden cruzar.
3. El semáforo es alto.
4. El samaforo es de color negro.

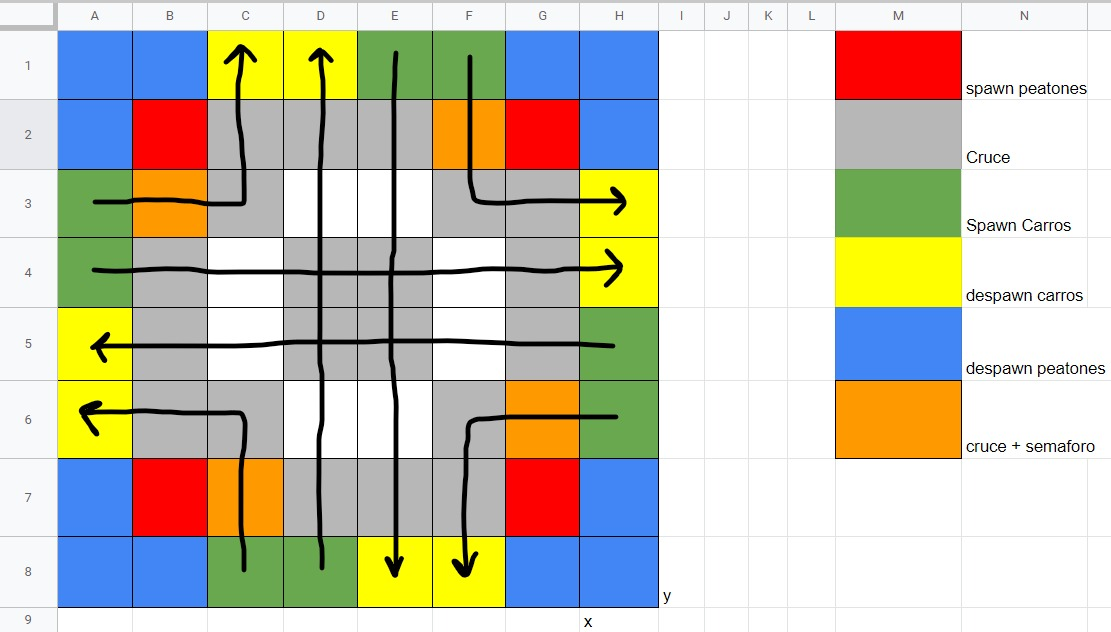
**Carros:**

1. Todos los carros presentes en la simulación tendrán diferentes tamaños.
2. Todos los carros presentes en la simulación tendrán diferentes colores.

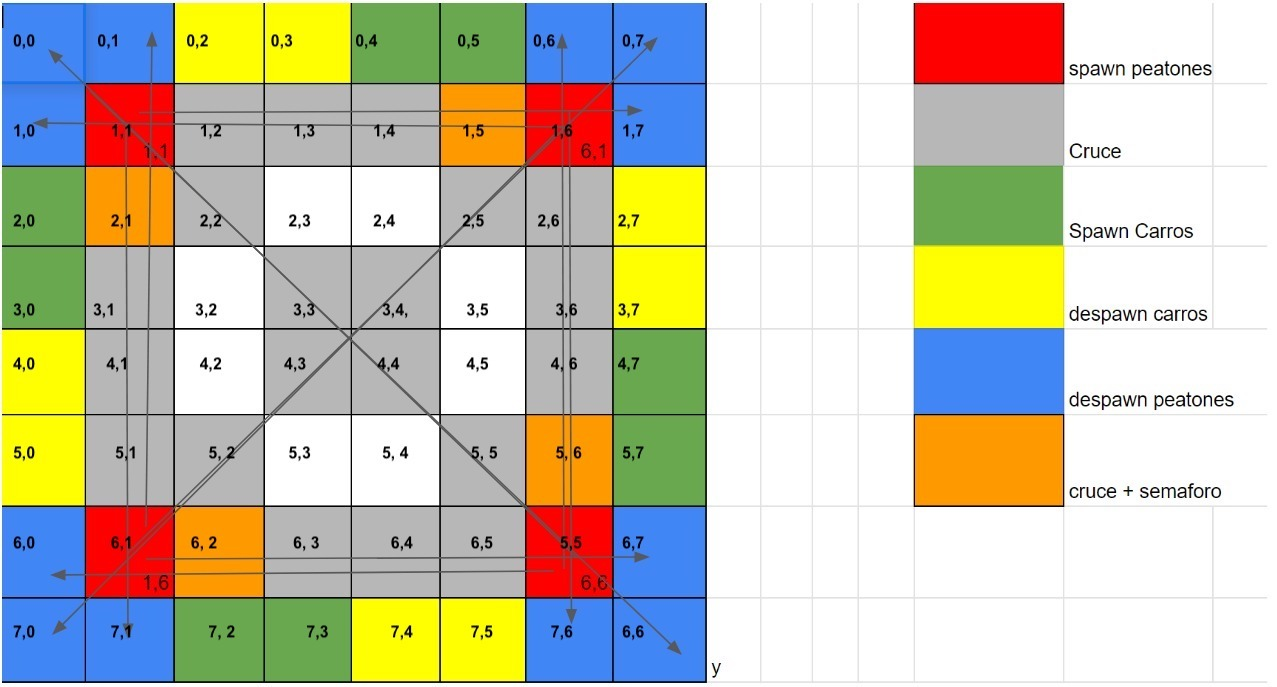
**Peatones:**

1. Los peatones son tanto hombres como mujeres.
2. Los peatones tienen diferentes alturas y prendas.

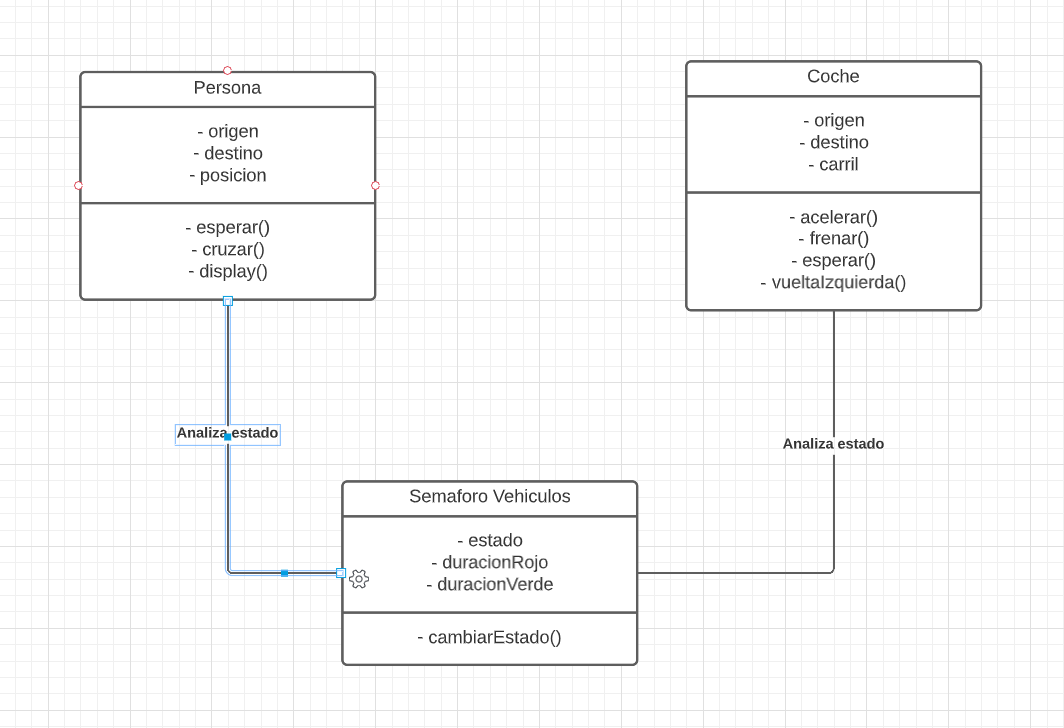
# 7. Descripción del sistema Multiagente



**Movimiento del coche**



**Movimiento del Peatón**



**Diagrama UML**

### Diagrama de estados

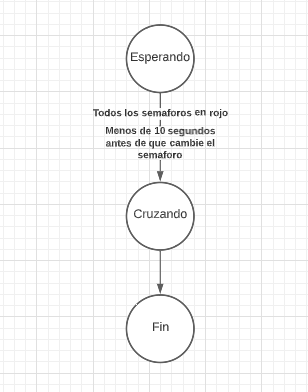
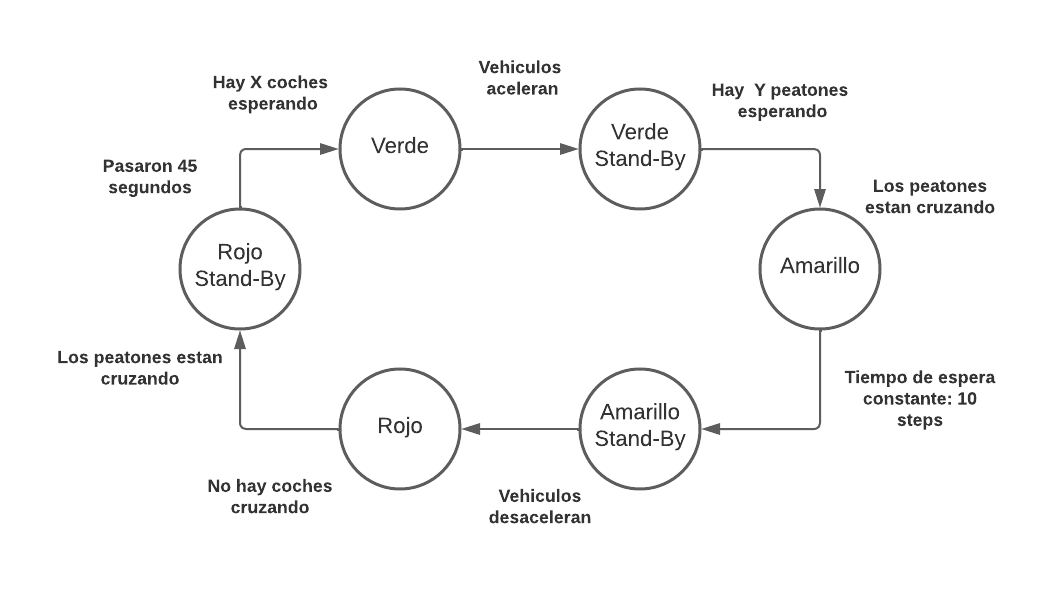


Diagrama de estado del peatón



**Diagrama de estado del coche**



**Diagrama de estado del semáforo inteligente.**

## 7.1 Modelo de los Agentes

**Semáforo**

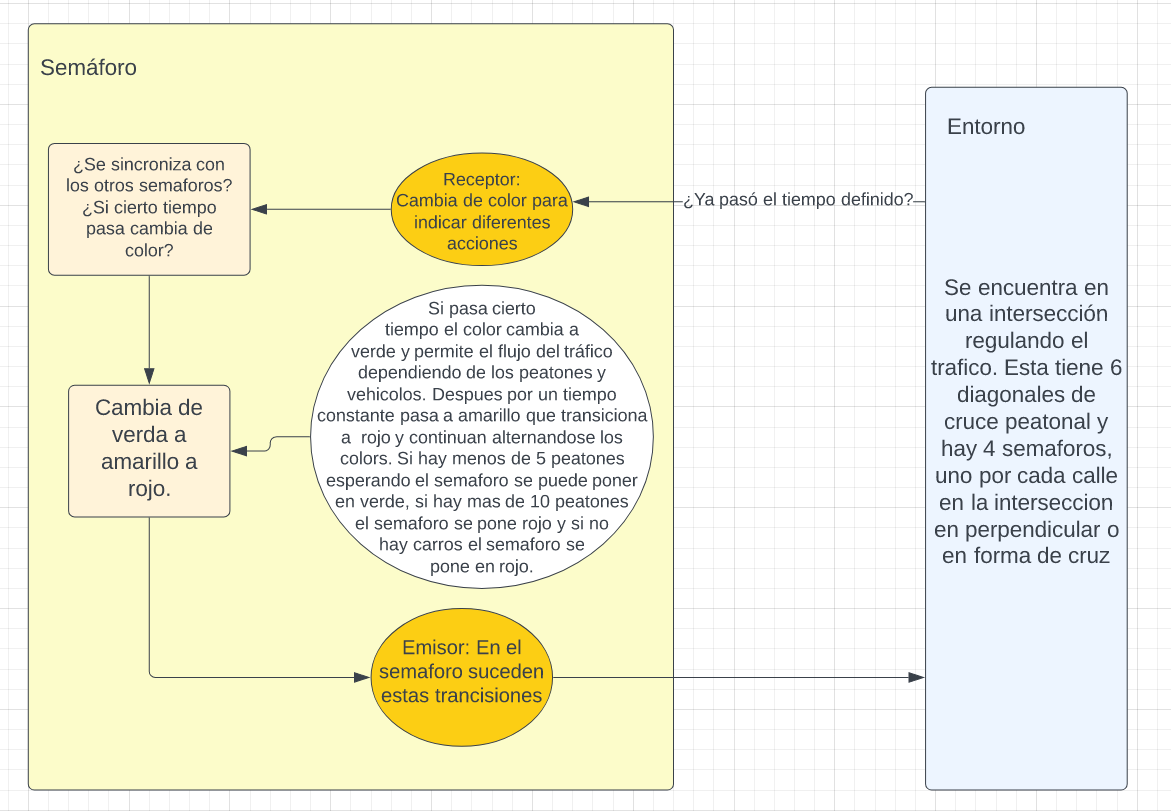
1. Creencias: Cambiar a color verde o color rojo
2. Planes: Detectar el número de peatones, coches y el tiempo (steps) transcurrido desde el último cambio de color para tomar una decisión
3. Cooperación: Analizar las acciones de los otros agentes para determinar el color del semáforo y sincronizarse con los otros semáforos.
4. Aprendizaje: Optimizar los cambios de color en base al movimiento del resto de los agentes presentes en la escena.

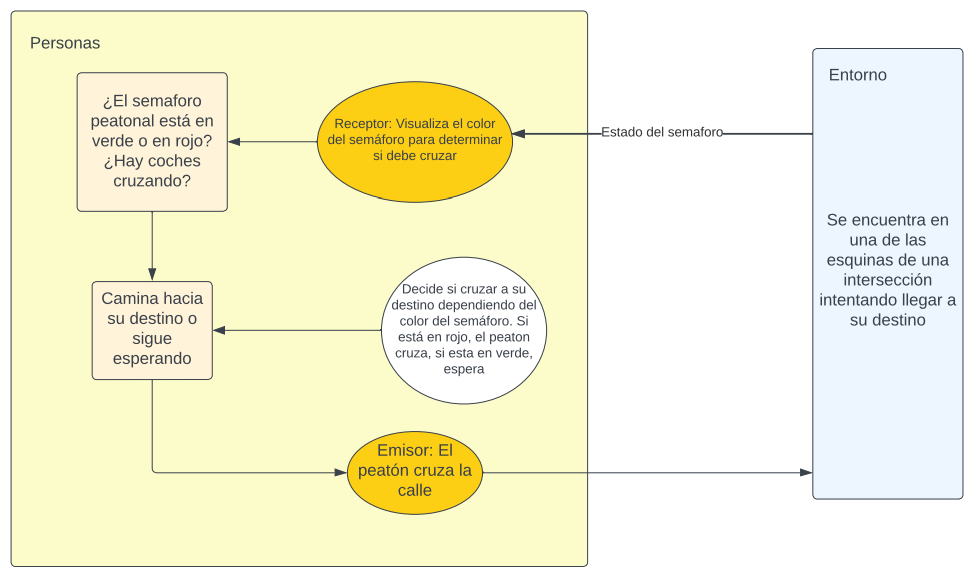
**Carros**

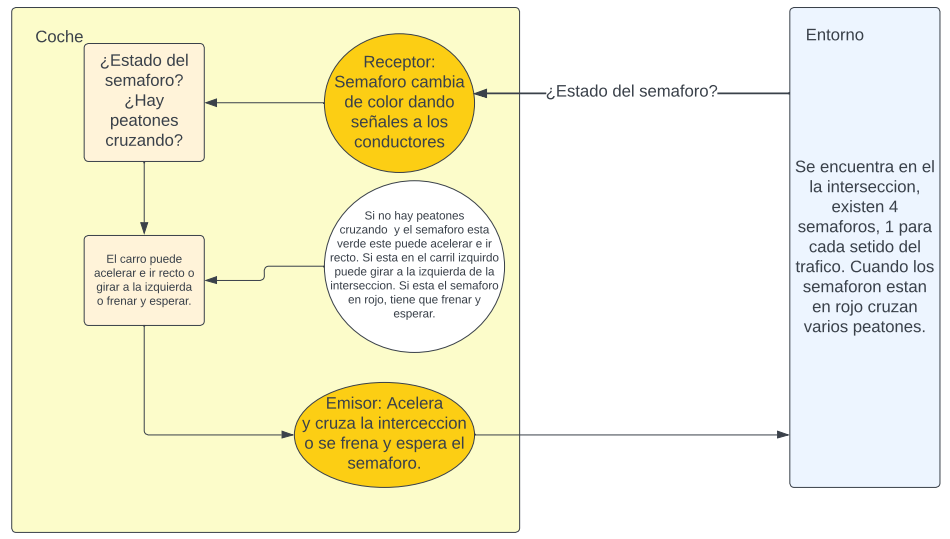
1. Creencias: Transportar a sus pasajeros de un punto A a un punto B, esto lo hace acelerando y respetando las indicaciones del semáforo.
2. Planes: Estar analizando el semáforo, si este se encuentra en verde podrá acelerar, de lo contrario se frenará.
3. Cooperación: Analizar el semáforo de tal manera que pueda seguir sus condiciones de transporte.
4. Aprendizaje: Aprender cuanto se tarda el semáforo en cambiar de color.

**Peatones**

1. Creencias: Poder caminar de un punto A a un punto B cuando el semáforo está en rojo para los vehículos.
2. Planes:Estar analizando el semáforo, si este se encuentra en rojo para los coches.
3. Cooperación: Analizar el semáforo de tal manera que pueda cruzar el paso peatonal.
4. Aprendizaje: Aprender cuanto se tarda un peatón en cruzar una intersección para poder mejorar los tiempos con los demás agentes.





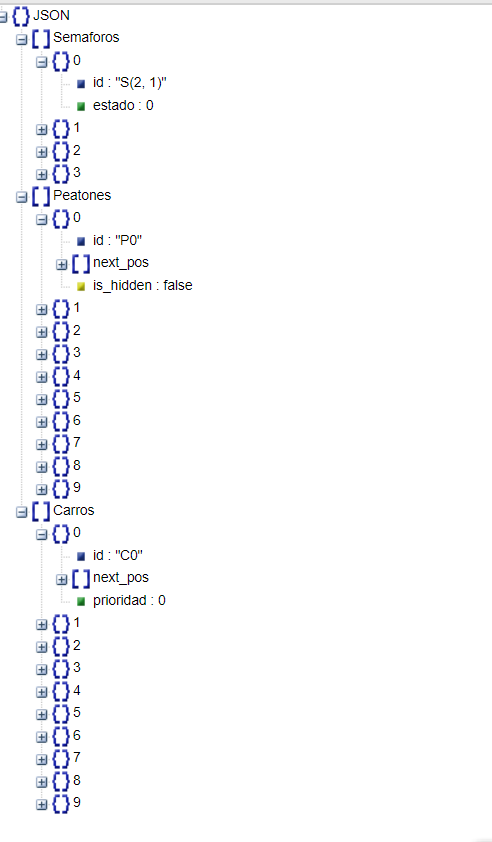


## 7.2 Modelo del escenario

Nuestro ambiente dentro de la simulación es discreto, pues se utilizan steps como método de manejo de tiempo. Está desarrollado de modo que con cada step se mantenga igual que en el step anterior, es decir, el escenario es estático. Únicamente cambian las trayectorias de los coches y los peatones, ya que son aleatorias, por lo cual el modelo es estocástico. A su vez, dado que vamos a manejar diferentes tipos de agentes, nuestro modelo sería multiagente.

Los semáforos alternan entre tres estados distintos: verde, que indica que pueden pasar los vehículos, amarillo que permite indicar que los vehiculos ya tendran que para de cruzar en la transicion de verde a rojo; el rojo que indica que los vehículos tienen que detenerse antes de la línea peatonal de las calles. Los vehículos avanzan dependiendo del estado que indican los semáforos y estos están sincronizados de tal manera que cada cierto periodo de tiempo, o si se detecta cierto número de peatones, se permite el paso a dos calles y después cambia de verde a amarillo y despues rojo, dejando que los peatones puedan cruzar entre los seis pasos peatonales y después los vehículos de las calles contrarias podrán circular. Los vehículos que se encuentran en el carril izquierdo de cada calle deben girar a la calle izquierda adyacente y solo estos pueden hacer esto. Los coches que están en el carril derecho tienen que ir recto y no pueden cruzar al carril izquierdo. Los peatones esperan a que el semáforo esté en rojo para los vehículos y por este periodo de tiempo estos pueden cruzar los pasos peatonales.

## 7.3 Modelo de la Interacción

****

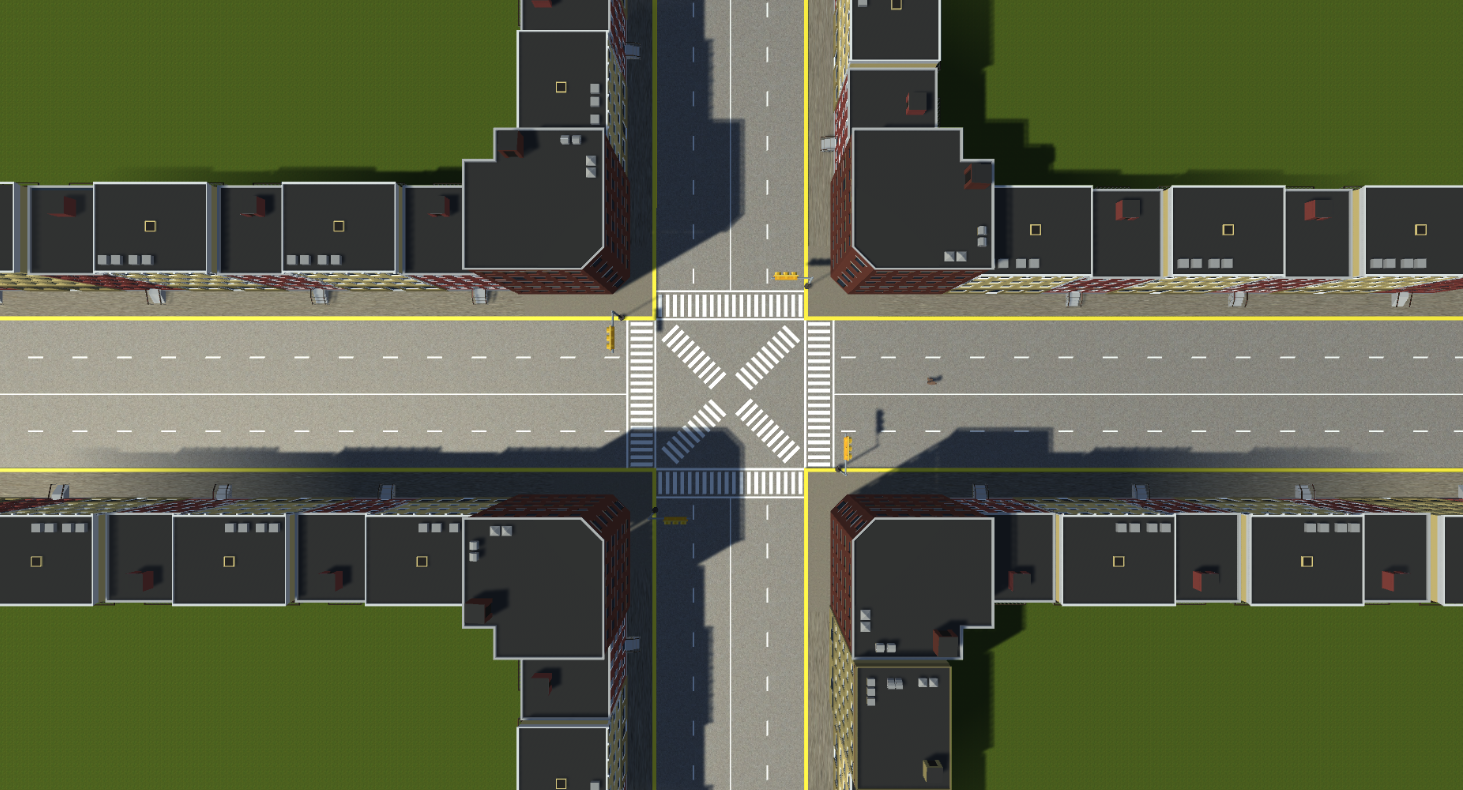
**Esto es un ejemplo de lo que recibimos:**

{"Semaforos": [{"id": "S(2, 1)", "estado": 1}, {"id": "S(1, 5)", "estado": 0}, {"id": "S(6, 2)", "estado": 0}, {"id": "S(5, 6)", "estado": 1}], "Peatones": [{"id": "P0", "next\_pos": [6, 1], "is\_hidden": false}, {"id": "P1", "next\_pos": [1, 1], "is\_hidden": false}, {"id": "P2", "next\_pos": [6, 1], "is\_hidden": false}, {"id": "P3", "next\_pos": [6, 6], "is\_hidden": false}, {"id": "P4", "next\_pos": [6, 1], "is\_hidden": false}, {"id": "P5", "next\_pos": [1, 1], "is\_hidden": false}, {"id": "P6", "next\_pos": [6, 1], "is\_hidden": false}, {"id": "P7", "next\_pos": [6, 6], "is\_hidden": false}, {"id": "P8", "next\_pos": [1, 6], "is\_hidden": false}, {"id": "P9", "next\_pos": [6, 1], "is\_hidden": false}], "Carros": [{"id": "C0", "next\_pos": [5, 0], "prioridad": 0}, {"id": "C1", "next\_pos": [0, 3], "prioridad": 0}, {"id": "C2", "next\_pos": [2, 7], "prioridad": 0}, {"id": "C3", "next\_pos": [3, 7], "prioridad": 0}, {"id": "C4", "next\_pos": [7, 4], "prioridad": 0}, {"id": "C5", "next\_pos": [3, 7], "prioridad": 0}, {"id": "C6", "next\_pos": [5, 0], "prioridad": 0}, {"id": "C7", "next\_pos": [2, 7], "prioridad": 0}, {"id": "C8", "next\_pos": [4, 0], "prioridad": 0}, {"id": "C9", "next\_pos": [0, 3], "prioridad": 0}]}

# 8. Modelación 3D

## 8.1 Descripción de la escena a modelar

Se observa una intersección perpendicular desde un ángulo superior. La intersección conecta cuatro calles. En cada calle se encuentra un semáforo colocado del lado izquierdo. También, en esta intersección hay dos pasos peatonales horizontales, verticales y diagonales. En esta intersección interactúan semáforos, peatones y vehículos.



## 8.2 Descripción de Componentes Gráficos

| **Nombre** | Simple Urban Buildings Pack 1 |
| --- | --- |
| **Descripción** | Edificios que se pueden colocar en la intersección para poder adornar el entorno. |
| **Imagen** |  |
| **Autor/Fuente** | <https://assetstore.unity.com/packages/3d/environments/urban/simple-urban-buildings-pack-1-33563> |

| **Nombre** | Tarbo - Trafic Lights |
| --- | --- |
| **Descripción** | Distintos modelos de semáforos para nuestra intersección. |
| **Imagen** |  |
| **Autor/Fuente** | <https://assetstore.unity.com/packages/3d/environments/urban/tarbo-city-traffic-lights-pack-free-154053> |

| **Nombre** | Hand Painted Grass Texture |
| --- | --- |
| **Descripción** | Material de zacate para rellenar. |
| **Imagen** |  |
| **Autor/Fuente** | <https://assetstore.unity.com/packages/2d/textures-materials/floors/hand-painted-grass-texture-78552> |

| **Nombre** | Asphalt materials |
| --- | --- |
| **Descripción** | Material de asfalto para poder simular las calles y banquetas. |
| **Imagen** |  |
| **Autor/Fuente** | <https://assetstore.unity.com/packages/2d/textures-materials/roads/asphalt-materials-141036> |

## 8.3 Descripción de Prefabs

| **Nombre** | Simple Cars Pack |
| --- | --- |
| **Descripción** | Distintos Prefabs de Vehículos |
| **Imagen** |  |
| **Autor/Fuente** | <https://assetstore.unity.com/packages/3d/vehicles/land/simple-cars-pack-97669> |

| **Nombre** | Distant Lands Free Characters |
| --- | --- |
| **Descripción** | Distintos Prefabs de peatones para los |
| **Imagen** |  |
| **Autor/Fuente** | <https://assetstore.unity.com/packages/3d/characters/distant-lands-free-characters-178123> |

## 8.4 Descripción de Scripts

Indicar el uso de cada script y en caso de estar reutilizando código agregar la fuente.

| **Nombre** | cameraController |
| --- | --- |
| **Descripción** | Script habilita las cámaras de la simulación. |
| **Interacciones** | no tiene ninguna interacción. |
| **Autor/Fuente** | Alejandro Melendez Torres - Thomas Freund Paternostro |

| **Nombre** | agentController |
| --- | --- |
| **Descripción** | Script que recibe los datos del servidor, los procesa y manda instrucciones a cada uno de los agentes. |
| **Interacciones** | recibe un json del servidor, este lo procesa y manda instrucciones a los agentes. |
| **Autor/Fuente** | Alejandro Melendez Torres - Thomas Freund Paternostro |

| **Nombre** | classCarro |
| --- | --- |
| **Descripción** | Script que permite mover el carro. |
| **Interacciones** | Recibe datos del agentController (next position) |
| **Autor/Fuente** | Alejandro Melendez Torres - Thomas Freund Paternostro |

| **Nombre** | classPeaton |
| --- | --- |
| **Descripción** | Script que permite mover el peatón. |
| **Interacciones** | Recibe datos del agentController (next Position) |
| **Autor/Fuente** | Alejandro Melendez Torres - Thomas Freund Paternostro |

| **Nombre** | classSemaforos |
| --- | --- |
| **Descripción** | Script que permite cambiar de color el semaforo |
| **Interacciones** | Recibe datos del agentController(estado del semáforo) |
| **Autor/Fuente** | Alejandro Melendez Torres - Thomas Freund Paternostro |

# 9. Entregables de administración de proyecto

**Liga al Product Backlog:** [TheBoyz' Product Backlog](https://docs.google.com/spreadsheets/d/1wilUf0yTLrOjkHVpsFj1sMqlY97nFiWbMK9hU_QGCg8/edit?usp=sharing)

**Liga a la carpeta de drive:** [**https://drive.google.com/drive/u/1/folders/1He-wDTQrRjV1f\_MEU-tCJmuzG2LLWHWX**](https://drive.google.com/drive/u/1/folders/1He-wDTQrRjV1f_MEU-tCJmuzG2LLWHWX)

**Liga al Google Colab:**

[**https://colab.research.google.com/drive/17xOGjNMKMVihoTahXPvRzjkFMKqG5pVg?usp=sharing**](https://colab.research.google.com/drive/17xOGjNMKMVihoTahXPvRzjkFMKqG5pVg?usp=sharing)

# 10. Reflexiones

**Alejandro:**

Al empezar esta unidad de formación yo tenía en mente una simulación sencilla y con 1 solo agente, al estar viendo los distintos temas empecé a tener distintas ideas de como pudiera mejorar la simulación. Al terminar este proyecto me sobresale mucho la comunicación de los distintos componentes (python a unity), si uno fallaba todo el sistema dejaba de funcionar.

**Mauricio:**

A lo largo del semestre pudimos formar parte del proceso de realizar una simulación de una intersección vial en la que interactúan carros y personas que buscan llegar a su destino. Nuestra meta final era simular la diferencia entre utilizar un semáforo tradicional y un semáforo inteligente. Ahora que hemos concluido el reto, me doy cuenta que la mejor forma de aprender un concepto es por medio de la práctica en un problema real, como fue el caso de este reto.

**Patricio:**

Al comenzar esta unidad de formación, no tenía idea de qué esperar sobre ella pues el nombre “multiagentes y gráficas computacionales” no me decía mucho. Pero a lo largo de estas cinco semanas he aprendido muchísimo acerca de multiagentes y sobre como modelar en Unity. También algo que me gusto aprender fue como crear servidores en nuestro localhost y pasar datos en tiempo real a estos.

**Thomas:**  A lo largo de este bloque pude entender varios conceptos importantes. Principalmente sobre la teoría de multiagentes y como estos son importantes dado que se basan en principios y temas reales que consecuentemente si terminan interactuando y creando como vivimos como sociedad. El objetivo fue poder mejorar la eficiencia del cruce vehicular y peatonal y llegamos a exceder el objetivo inicial de que por sí ya era bastante ambicioso. Considero que también este reto me sirvió para poder crear marcos comparativos y recrear sistemas cerrados que nos permitan entender en la práctica lo que aprendimos en la materia.

# 10. Referencias

Unity. (2021, 16 mayo). *Unity Asset Store - The Best Assets for Game Making*. Unity Asset Store. Recuperado 6 de septiembre de 2022, de https://assetstore.unity.com/