

RETO SMA: CIUDAD INTELIGENTE

Equipo 3:
Diego Zurita
Julián Cisneros
Luis David Maza
Luis Martínez

SISTEMAS MULTIAGENTES

01

Un sistema compuesto por múltiples agentes, que son entidades autónomas con la capacidad de tomar decisiones y actuar de manera independiente.

02

Los agentes pueden interactuar entre sí y con su entorno.

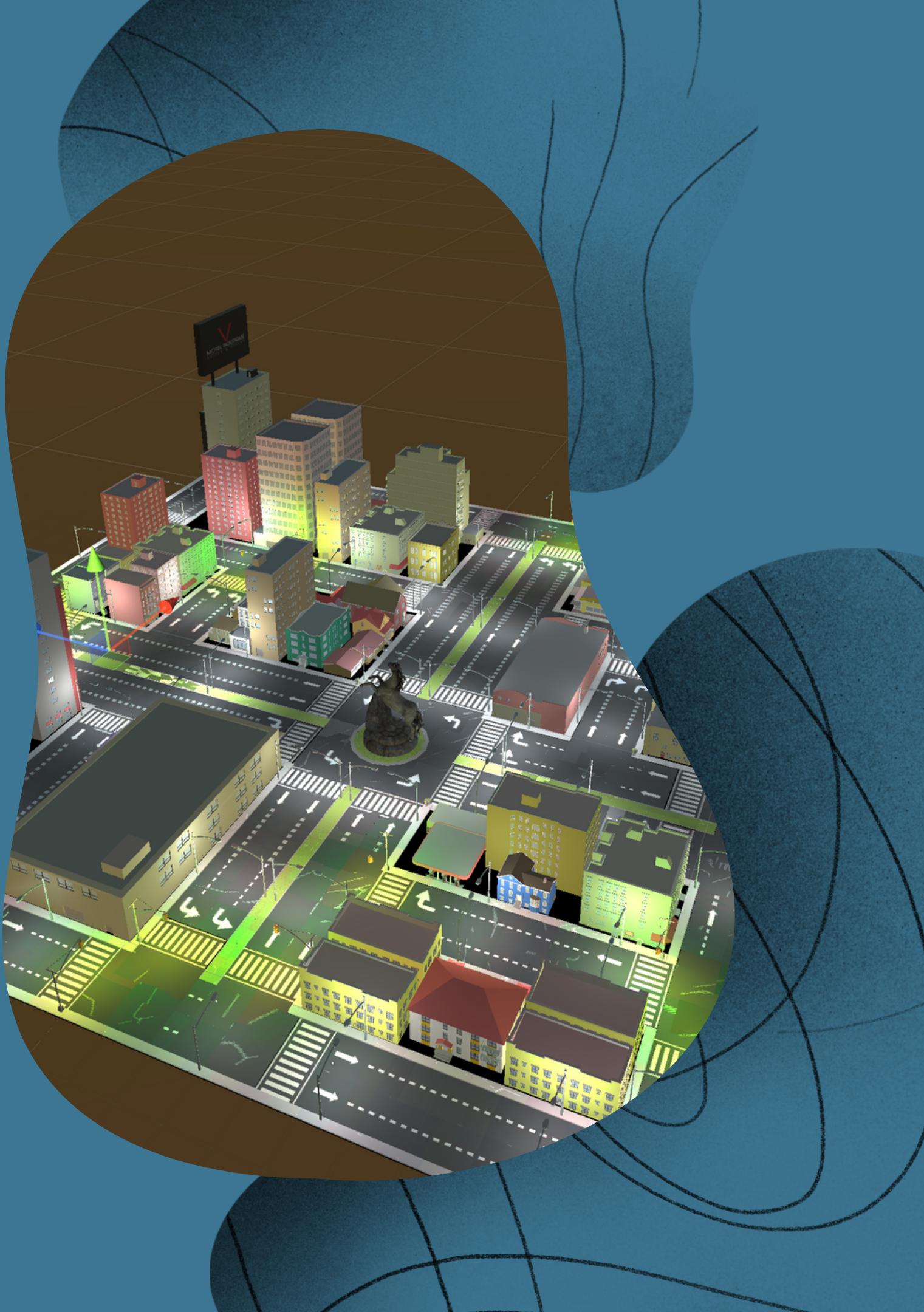
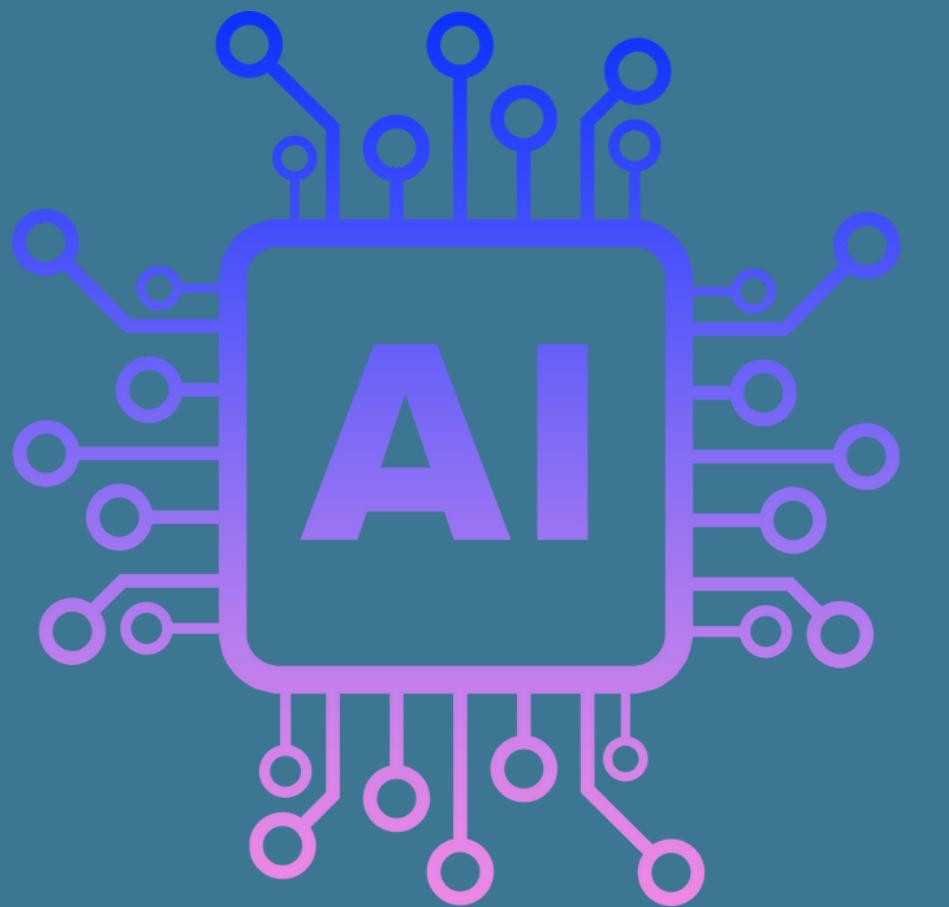
03

Estos sistemas manejan tareas complejas, a menudo mediante la colaboración y la competencia entre los agentes.

RETO

Mejorar la Movilidad Urbana en México

- Objetivo: Reducir la congestión vehicular en las ciudades mexicanas.
- Método: Utilizar un sistema de multiagentes para simular el tráfico de manera gráfica.



PLAN DE TRABAJO

Responsable	Nombre de la Actividad	Fecha de Entrega	Esfuerzo Realizado (Esperado)
JuliánLuis DavidLuis FernandoDiego	Hacer los Diagramas de clase de agentes involucrados.	7/Noviembre/2023	Media
JuliánLuis DavidLuis FernandoDiego	Hacer los Diagramas de protocolos.	7/Noviembre/2023	Media
JuliánLuis DavidLuis FernandoDiego	Hacer la Entrega de la Revisión 1.	7/Noviembre/2023	Poca
JuliánLuis DavidLuis FernandoDiego	Prototipo de los Agentes.	10/Noviembre/2023	Poca
JuliánLuis DavidLuis FernandoDiego	Creación de los Agentes de manera gráfica.	15/Noviembre/2023	Media
JuliánLuis DavidLuis FernandoDiego	Reuniones de Equipo.	7/Noviembre/2023 9/Noviembre/2023 14/Noviembre/2023 19/Noviembre/2023 24/Noviembre/2023 2023	Mucha
JuliánLuis DavidLuis FernandoDiego	Realizar documentaciones del avance.	7/Noviembre/2023 9/Noviembre/2023 14/Noviembre/2023 19/Noviembre/2023 24/Noviembre/2023 2023	Media
JuliánLuis DavidLuis FernandoDiego	Programación de los Agentes.	21/Noviembre/2023	Mucha
JuliánLuis DavidLuis FernandoDiego	Hacer Entrega Final del Reto	24/Noviembre/2023	Mucha

DEFINICIÓN DE AGENTES

Agentes propuestos para el reto

Coché

Grupo Medios de transporte
Rol: Medio de transporte personal

Eventos

Circular por la ciudad
Seguir indicaciones de semáforo
Seguir indicaciones del oficial de tránsito
Detectar su estacionamiento final
Detectar otro coche

Eventos - Acción

Circular por la ciudad-> El coche avanza por la calle siguiendo los sentidos de las mismas
Seguir indicaciones de semáforo-> El coche se detiene si está en rojo o sigue si está en verde
Seguir indicaciones del oficial de tránsito -> El coche sigue las indicaciones del oficial de tránsito si el semáforo está descompuesto
Detectar su estacionamiento final-> El coche entra al estacionamiento
Detectar otro coche-> El coche se detiene

Semáforo

Grupo: Tránsito

Rol: Organizador de movimiento de medios de Transporte

Eventos

Detecta grupo de medios de transporte

Detecta medios de transporte

Detecta una señal de otro Semáforo

Da indicaciones a los agentes del grupo medio de transporte

Un personaje de tránsito le da órdenes

Eventos - Acción

Detecta grupo de medios de transporte -> Pone luz verde y da prioridad al grupo de medios de transporte con más miembros o al azar si la cantidad es la misma.

Detecta una señal de otro Semáforo -> La toma en cuenta para tomar decisiones de qué luz poner y cuánto tiempo

Un personaje de tránsito le da órdenes -> Le da prioridad a las órdenes

Oficial de Tránsito

Grupo: Tránsito

Rol: Organizador de Medios de Transporte

Eventos

Detenerse

Detectar medios de transporte

Detecta una señal de Semáforo

Detecta algún caso concurrido

Eventos - Acción

Detenerse -> Se detiene en un lugar seguro

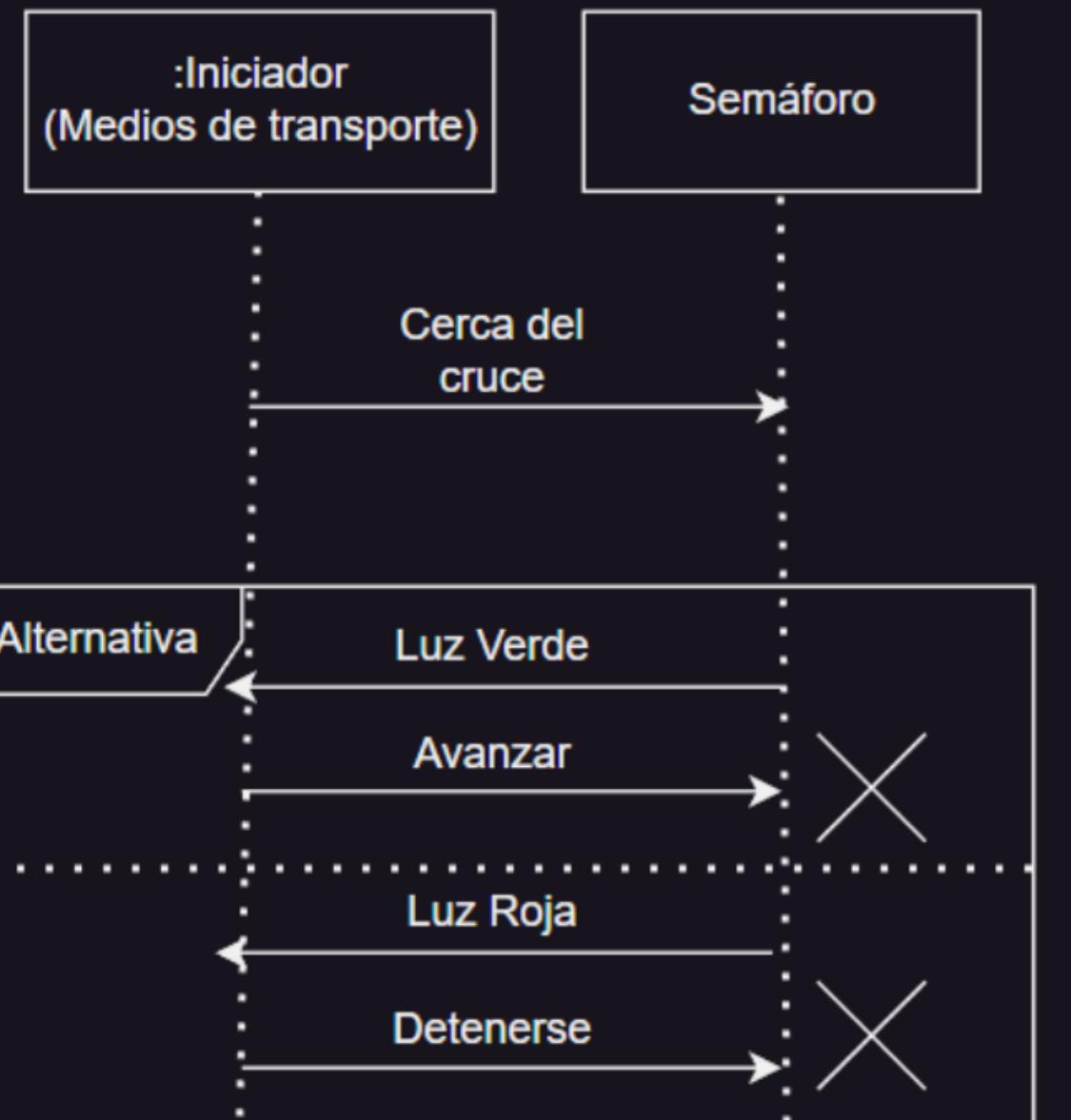
Detectar medios de transporte-> Da prioridad a una calle para que estos avancen más tiempo

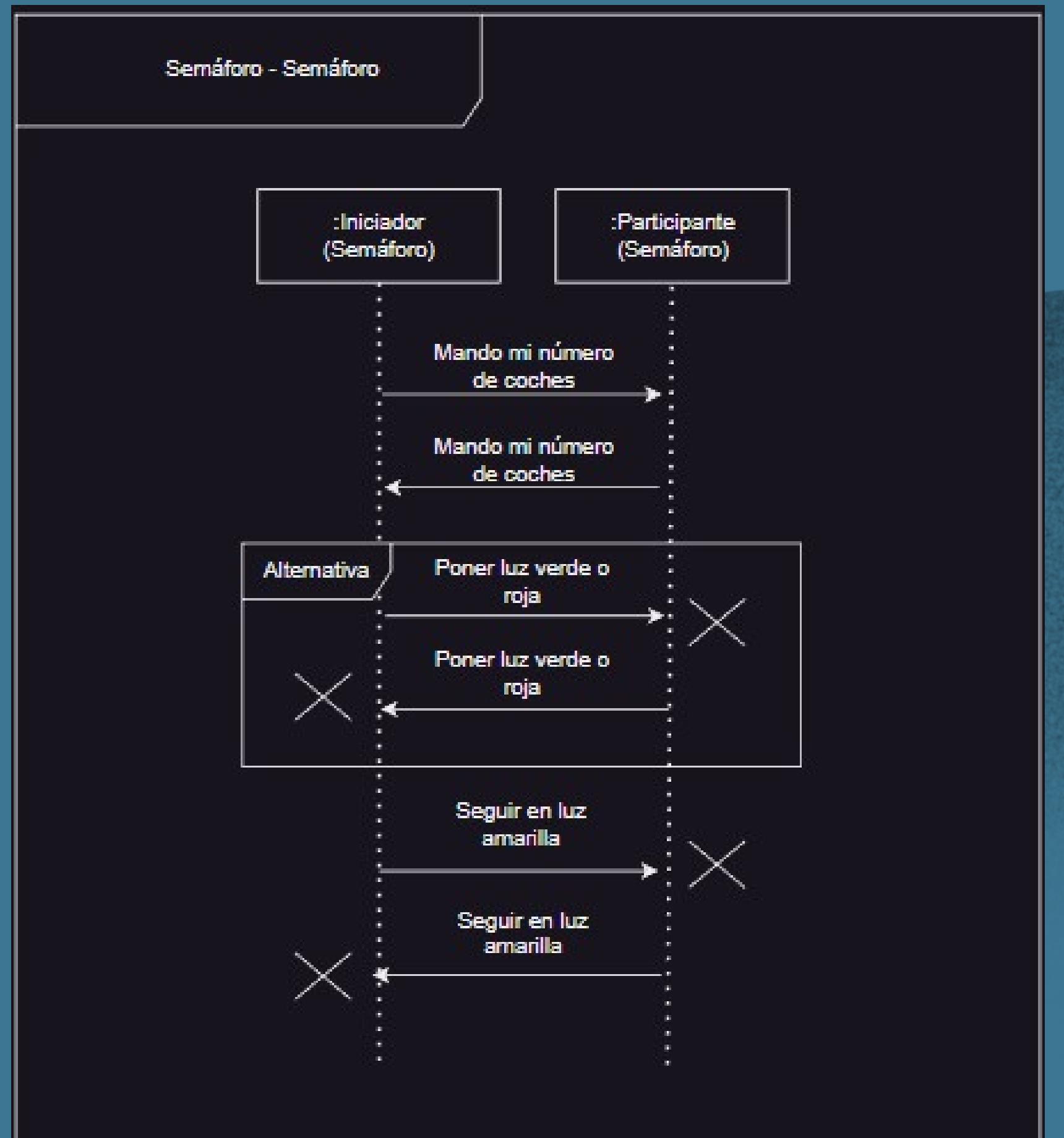
Detecta una señal de Semáforo -> Se mueve a la ubicación del Semáforo y da indicaciones.

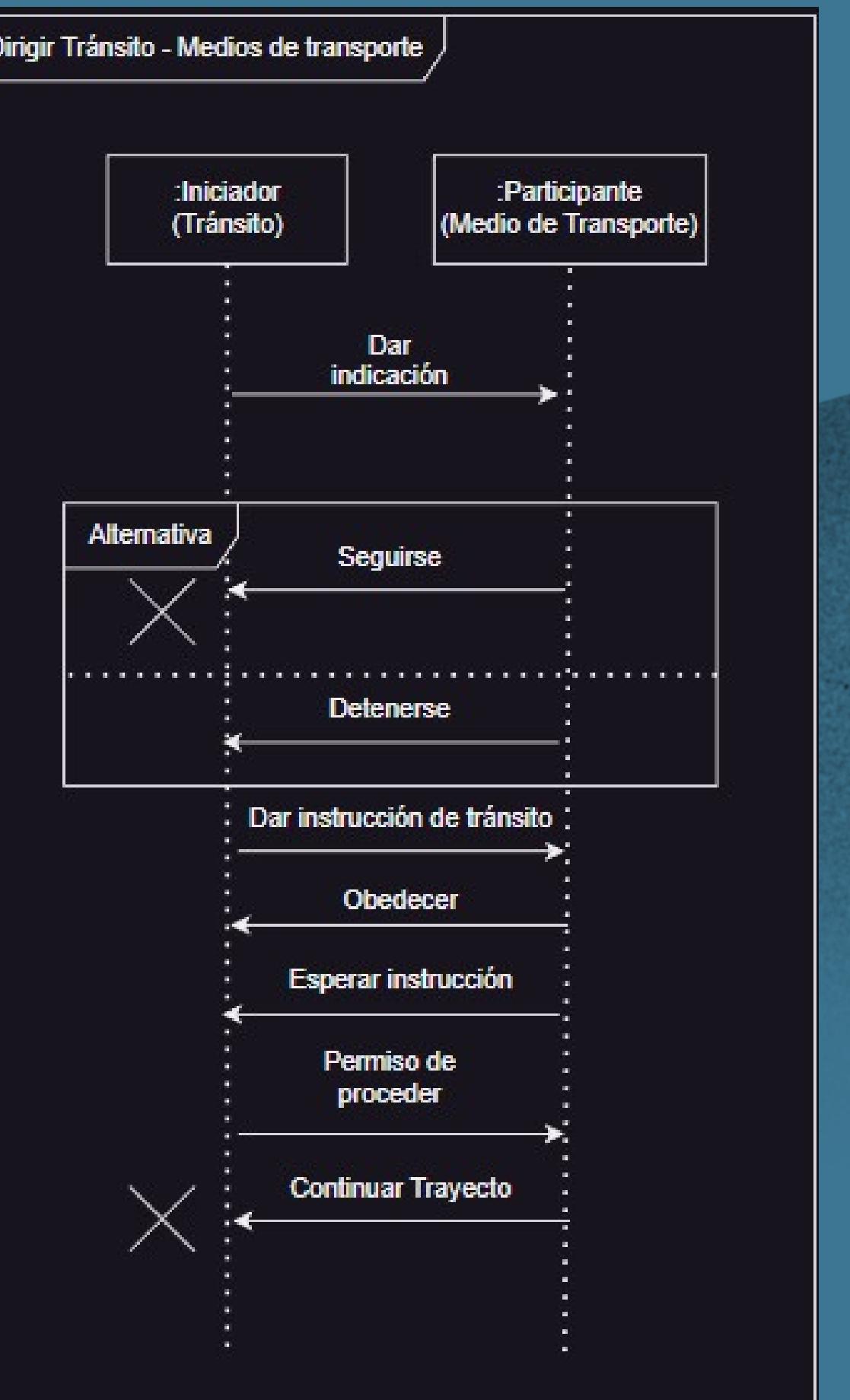
Detecta algún caso concurrido -> Da indicaciones a los agentes del grupo medio de transporte.

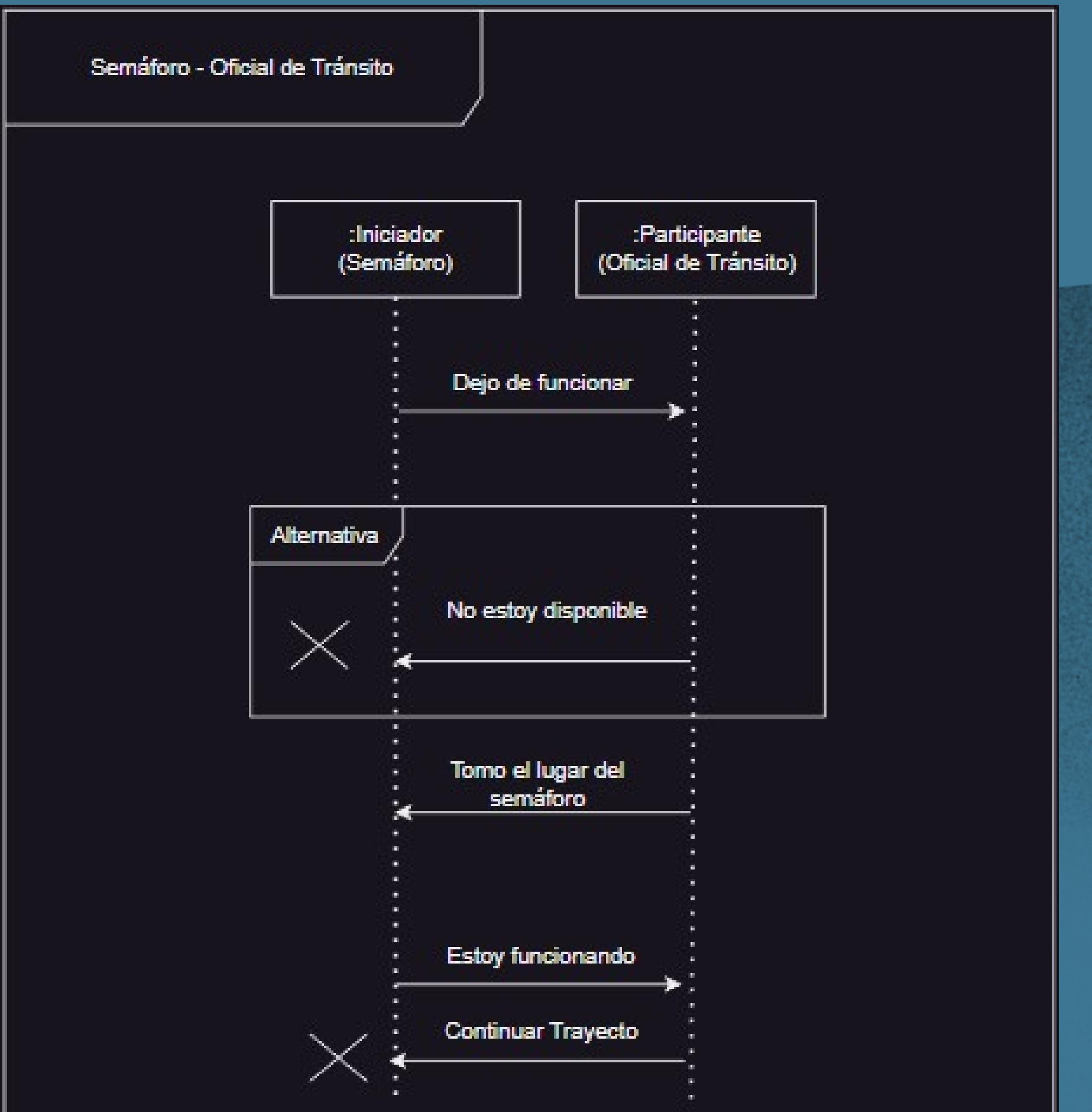
DIAGRAMAS

Semáforo - Medios de transporte









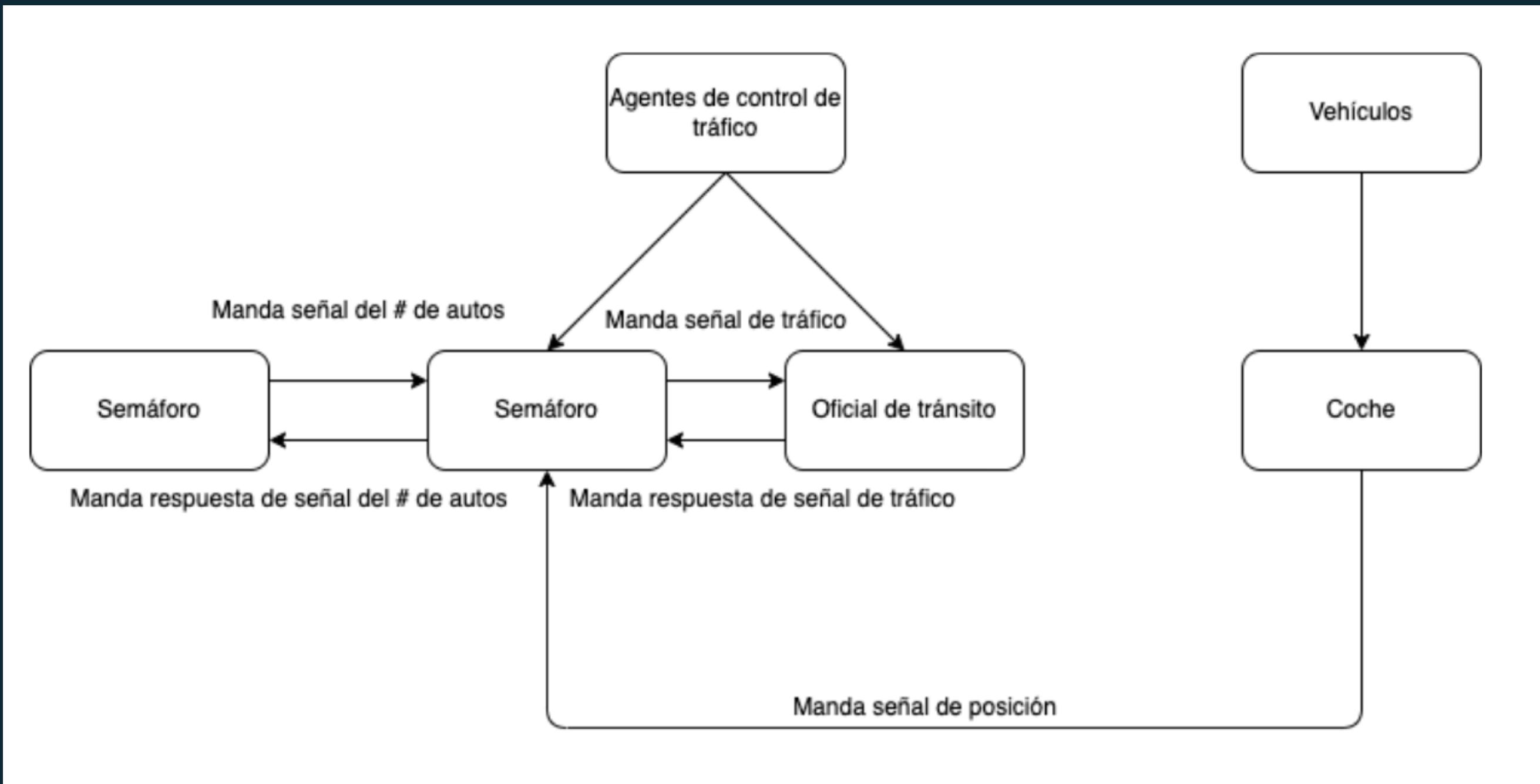


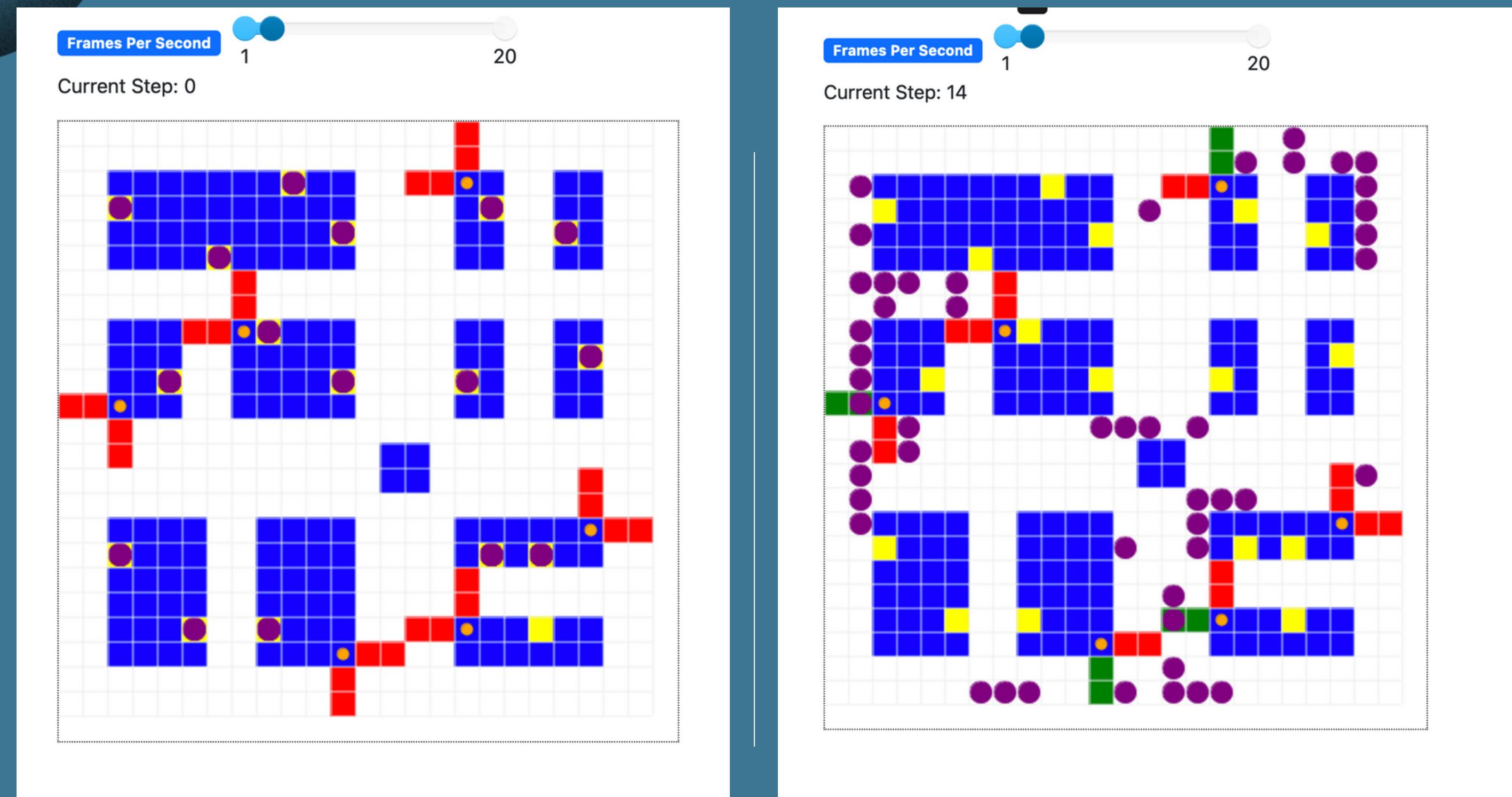
Diagrama Ontología 2:

- Ciudad:
 - Construcción:
 - Edificio
 - Glorieta
 - Calle:
 - Estacionamiento
 - Posición
 - Vehículos:
 - Coche
 - Posición actual
 - Inicio
 - Destino
 - Control de tráfico:
 - Semáforo
 - Verde
 - Rojo
 - Oficial de Tránsito
 - Cambiar Verde
 - Cambiar Rojo

Término	Significado
Verde	El vehículo tiene permitido avanzar
Rojo	El vehículo no tiene permitido avanzar
Cambiar Verde	El oficial de tránsito cambia el semáforo a estado “VERDE”
Cambiar Rojo	El oficial de tránsito cambia el semáforo a estado “ROJO”
Posición Actual	Indica el lugar actual del agente
Inicio	Inicia la posición de inicio del agente
Destino	Inicia la posición final del agente

GRAFICAS Y MESA

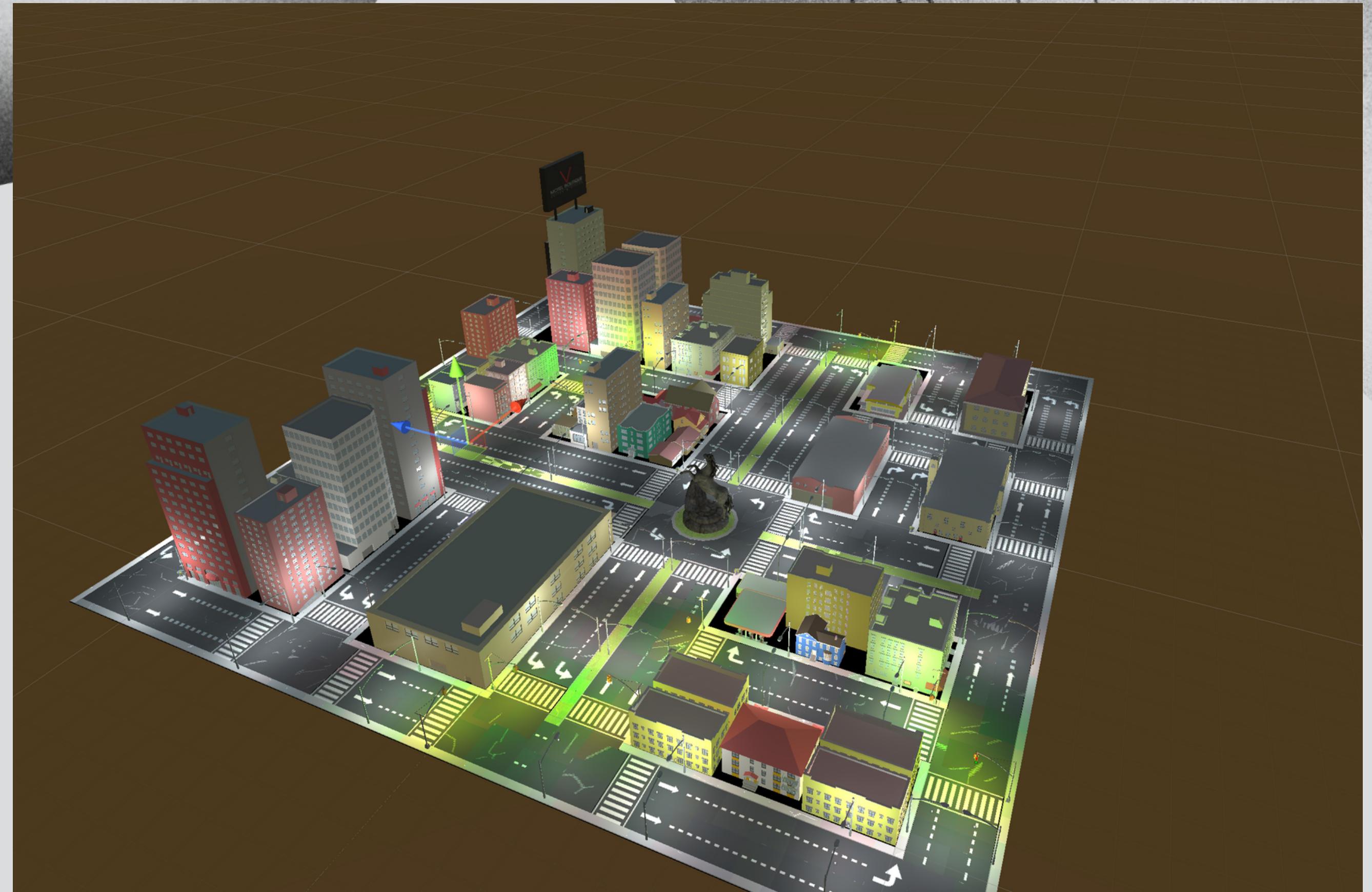
SIMULACIÓN MESA



CARROS



CIUDAD



IMPLEMENTACIÓN

SERVIDOR FLASK

```
from flask import Flask, redirect, url_for, request #type: ignore
from ciudad import CiudadModel
import json

app = Flask(__name__)

# Recurso prueba
# http://localhost/jsonResult
@app.route('/jsonResult', methods=['GET'])
def jsonResult():
    modelo.step()
    positions = modelo.crearJson()
    return json.dumps(positions)

if __name__ == '__main__':
    modelo = CiudadModel()
    app.run(host='0.0.0.0', port=8001, debug=True)
```

UNITY

```
[System.Serializable]
public class Auto
{
    public int id;
    public int posX;
    public int posY;
    public int firstpx;
    public int firstpy;
    public int destinationpx;
    public int destinationpy;
    public int step;
    public int PosSX;
    public int PosSY;
    public GameObject carro;
    public List<Vector3> Origcar;
    public float pivoteX;
    public float pivoteY;

    public void CambiarPosicionAPosS()
    {
        PosSX = posX;
        PosSY = posY;
    }
    public void inicializar_carro(GameObject carroN)
    {
        carro = carroN;
    }
}

[System.Serializable]
public class Semaforo
{
    public int id;
    public string color;
    public bool descompuesto;
    public int step;
}

[System.Serializable]
public class TrafficData
{
    public List<Auto> Autos;
    public List<Semaforo> Semaforos;
}
```

```
IEnumerator GetTrafficDataCoroutine(string uri)
{
    using (UnityWebRequest webRequest = UnityWebRequest.Get(uri))
    {
        yield return webRequest.SendWebRequest();

        if (webRequest.isNetworkError || webRequest.isHttpError)
        {
            Debug.LogError("Error: " + webRequest.error);
        }
        else
        {
            if (cont1 == 0)
            {
                ProcessJson_2(webRequest.downloadHandler.text);
            }
            else
            {
                ProcessJson(webRequest.downloadHandler.text);
            }
        }
    }
}

private void ProcessJson_2(string jsonString)
{
    TrafficData data = JsonUtility.FromJson<TrafficData>(jsonString);
    foreach (var autoData in data.Autos)
    {
        int idBuscado = autoData.id; // Cambia esto al id que deseas buscar
        foreach (Auto auto in autos)
        {
            if (auto.id == idBuscado)
            {
                auto.posX = 119 - (autoData.posX * 10);
                auto.posY = 119 - (autoData.posY * 10);
                auto.step = autoData.step;
                break;
            }
            else
            {
                Debug.Log("Auto no encontrado");
            }
        }
    }
    datoslistos_2();
}
```

CONCLUSIONES