



**Tecnológico  
de Monterrey**

**Modelación de sistemas multiagentes con gráficas  
computacionales (Gpo 1)**

**Profesor Luis Palomino Ramírez  
Profesor Omar Mendoza**

**ACTIVIDAD INTEGRADORA**

**Ariana Lisel Ayaviri Clavel A01706629**

En este avance de nuestro entregable final, propondremos una simulación en una de las avenidas que en el reto final será un cruce de ambas avenidas, donde los autos tendrán la capacidad de frenar cuando un semáforo esté en rojo, seguir cuando esté en verde y no chocar los unos con los otros.

En el archivo carMove.html diseñamos una ciudad con carros utilizando modelos obj, pero que finalmente no pudimos modelar en esta ocasión, porque tardaba como un minuto en cargar y el movimiento era lento así que para este entregable decidimos utilizar el index.html que es una versión simplificada.

## Entorno de Multiagentes

city:

- num\_cars:
- num\_trafficLights
- M (grid width)
- N (grid height)

Car:

- Start\_point
- end\_point
- x
- y
- z
- velocity
- angle

TrafficLight:

- state
- red\_time
- yellow\_time
- green\_time
- traffic\_direction

NUESTRO ARCHIVO JSON QUE MODIFICA EL ESTADO DE LOS SEMAFOROS SE VE ASI:

```
{
  "0": [
    {
      "state": 0,
      "id": 0
    },
    {
```

```

    "state": 0,
    "id": 1
  }
],
"1": [
  {
    "state": 0,
    "id": 0
  },
  {
    "state": 0,
    "id": 1
  }
],
"2": [
  {
    "state": 0,
    "id": 0
  },
  {
    "state": 0,
    "id": 1
  }
],
"3": [

```

```
{
  "state": 0,
  "id": 0
},
{
  "state": 0,
  "id": 1
}
]
```

NUESTRO ARCHIVO JSON CON LAS POSICIONES DE LOS CARROS

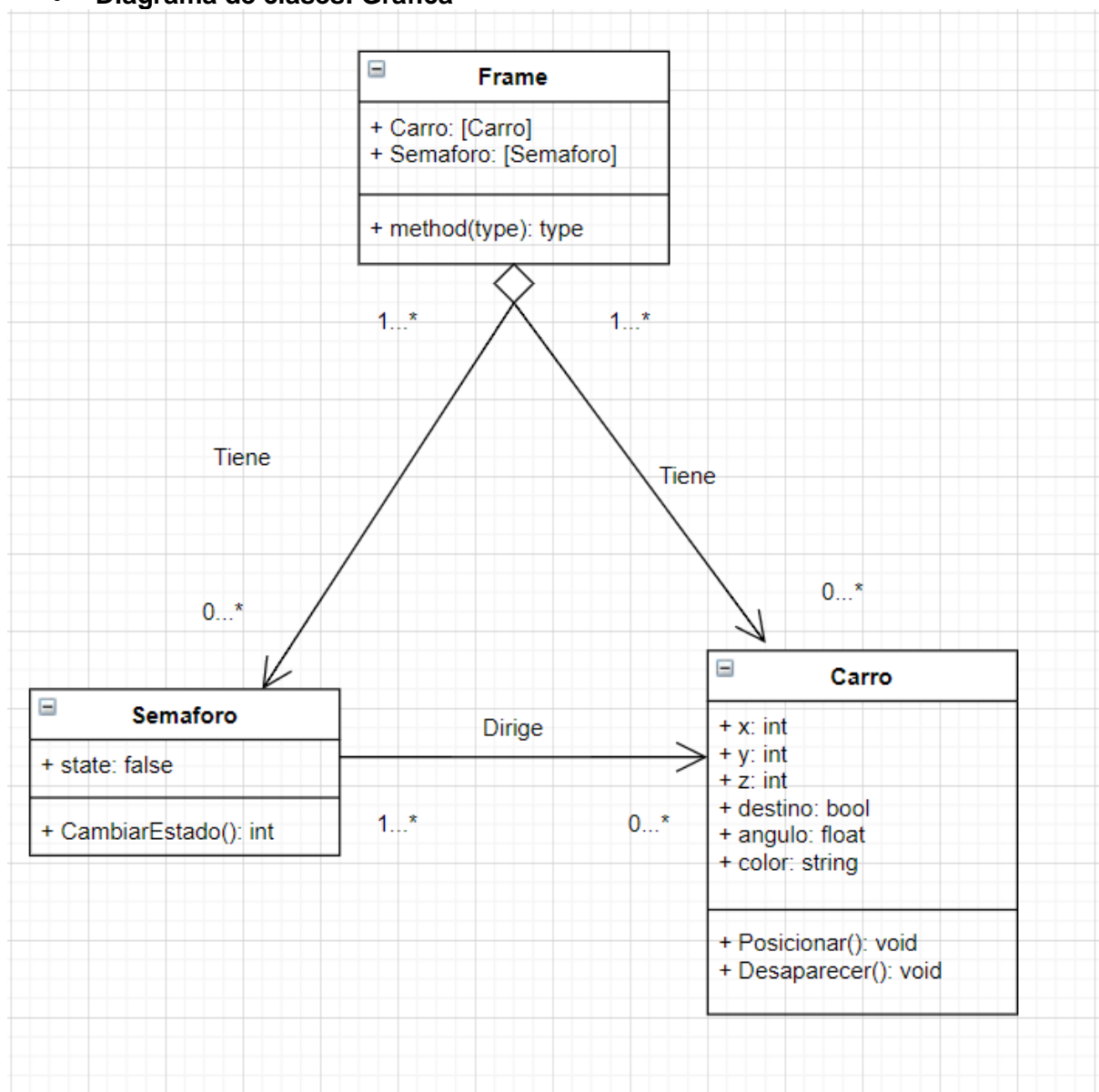
```
{
  "0": [
    {
      "x": 1.800000000000000114,
      "z": -7.832952712292837,
      "id": 0
    },
    {
      "x": 1.800000000000000114,
      "z": -15.928267033240843,
      "id": 1
    },
    {
      "x": 1.800000000000000114,
```

```
    "z": -22.645498049804644,  
    "id": 2  
  },  
  {  
    "x": 1.800000000000000114,  
    "z": -29.856287912442923,  
    "id": 3  
  },  
  {  
    "x": -1.800000000000000114,  
    "z": 7.922249961666012,  
    "id": 4  
  },  
  {  
    "x": -1.800000000000000114,  
    "z": 16.733671444011804,  
    "id": 5  
  },  
  {  
    "x": -1.800000000000000114,  
    "z": 26.03094165887586,  
    "id": 6  
  },  
  {  
    "x": -1.800000000000000114,
```

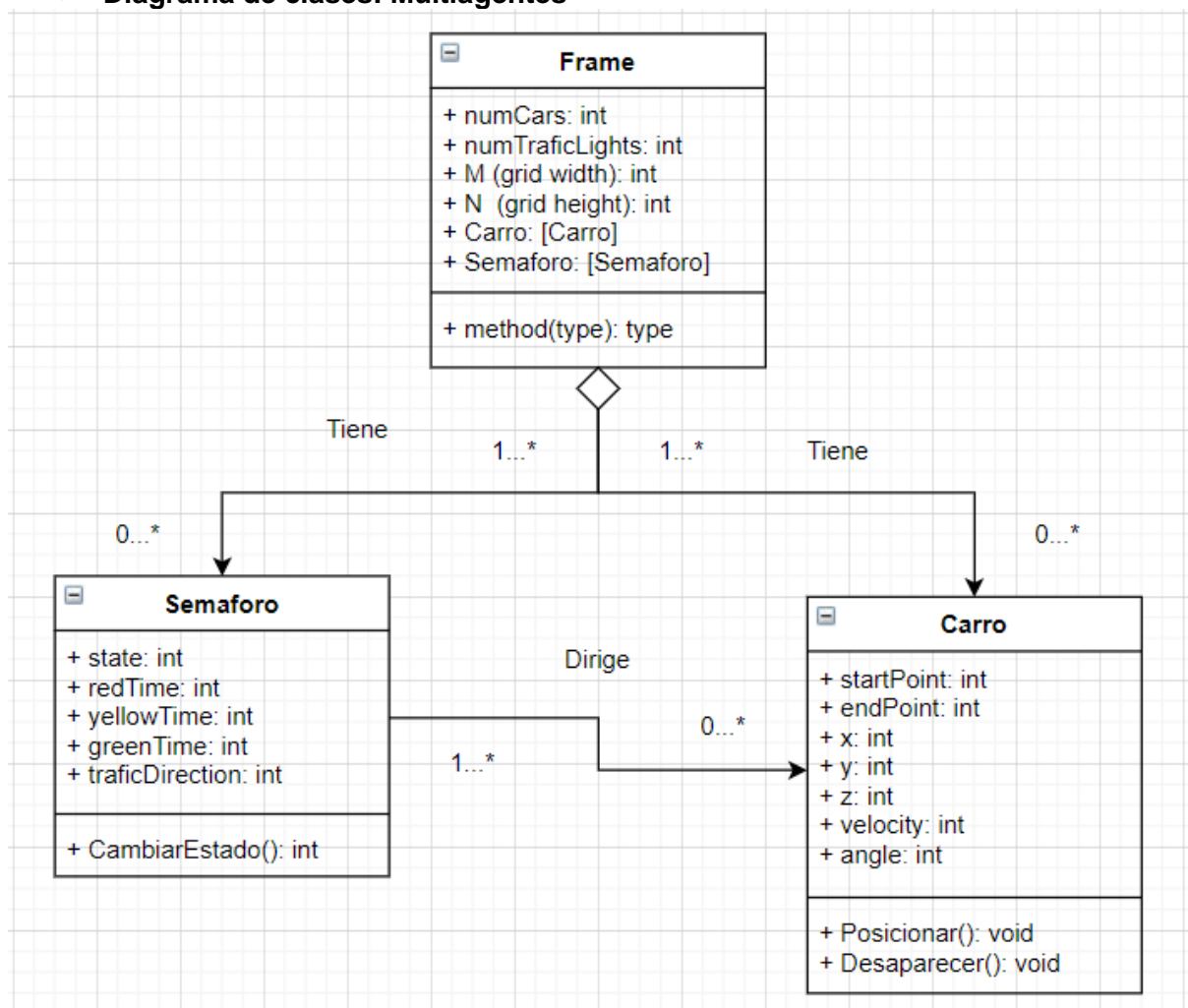
```
"z": 34.016086612575975,  
"id": 7  
}  
],
```

## Diagrama de Clases

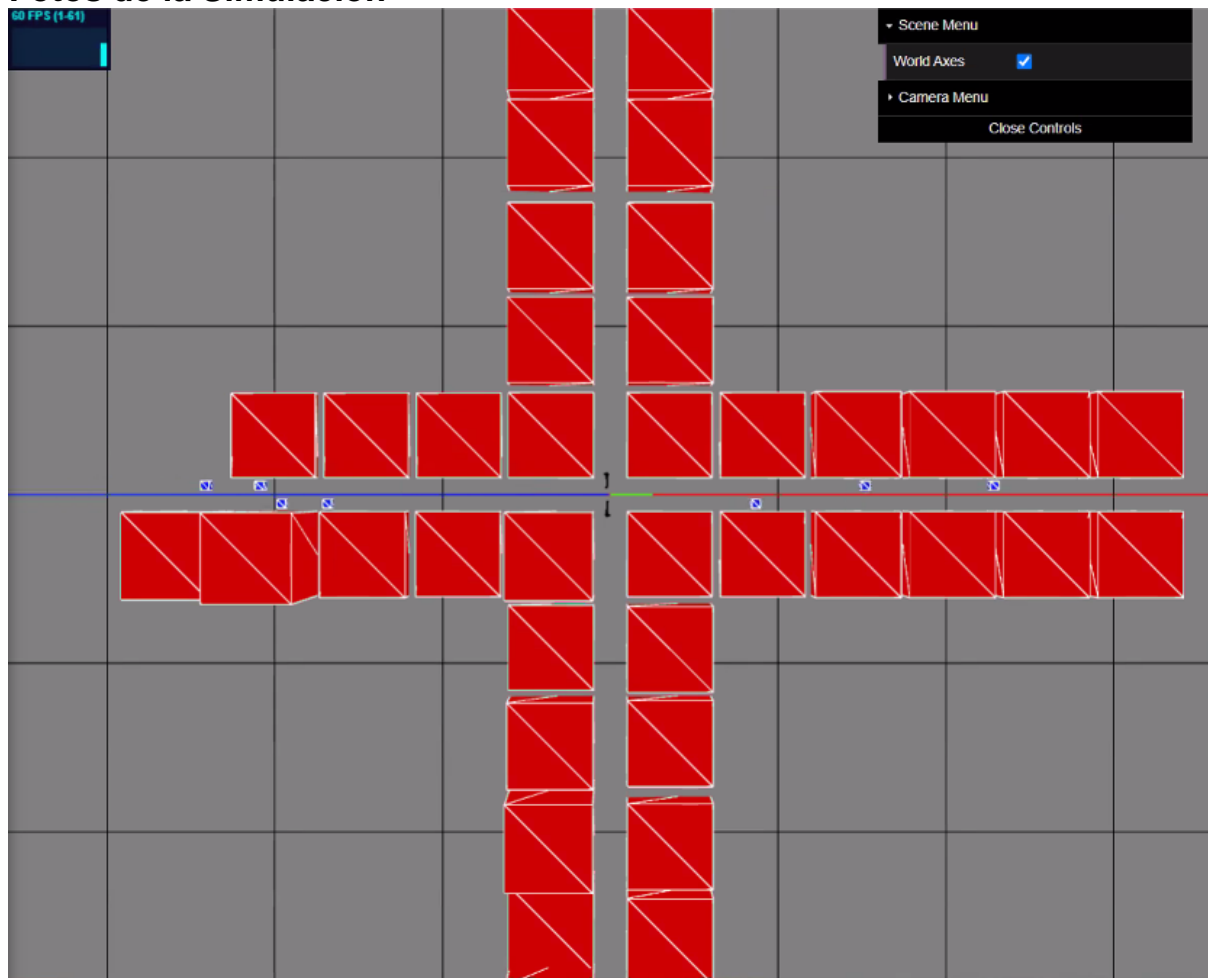
- Diagrama de clases: Grafica



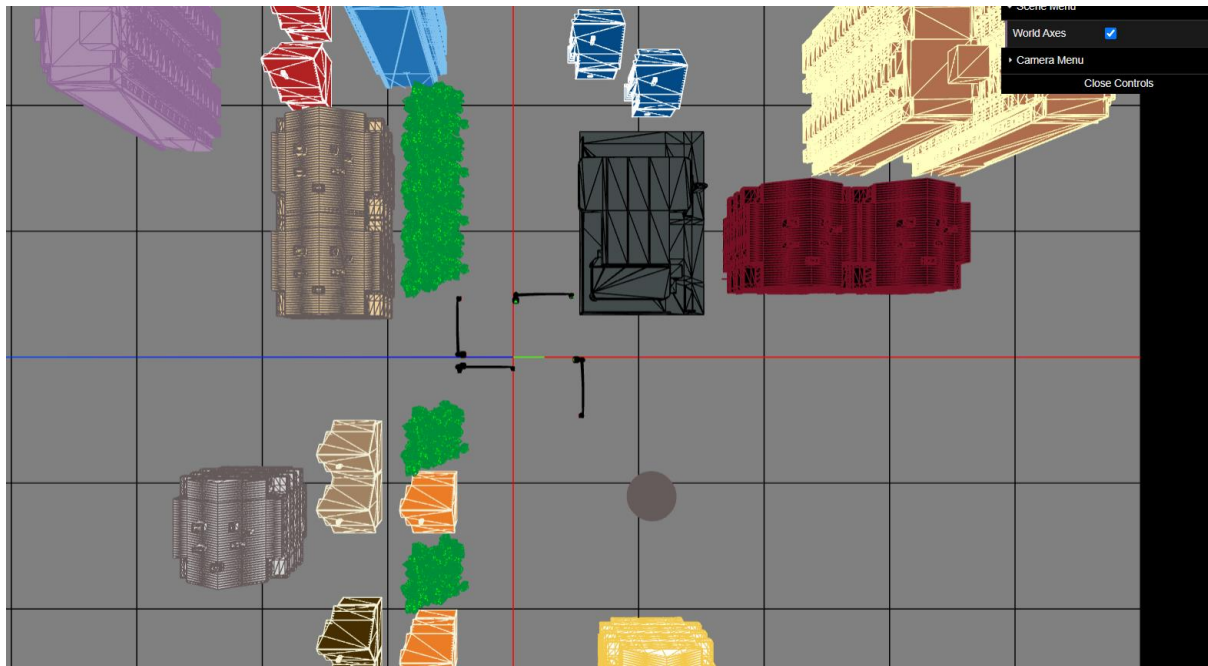
- Diagrama de clases: Multiagentes



## Fotos de la Simulación







Por ahora los semáforos dependen únicamente del parámetro del tiempo que se le asigne a cada estado del semáforo (rojo, amarillo y verde). Para el reto final el objetivo es que puedan medir la cantidad de tráfico y saber cuando ponerse en rojo o verde.

Por otro lado, los carros pueden saber cuando tienen vecinos cerca para respetar sus espacios y no chocar, también parar según el color del semáforo. En una futura implementación también esperamos que los carros puedan lograr hacer cruces en dos carriles cruzados de ambas avenidas, probablemente hacer cambios de carril y otros aspectos.

Los carros también tienen la capacidad de actualizar sus velocidades dependiendo de la lejanía de los vecinos, para acelerar cuando esté despejado y disminuir la velocidad cuando tienen a alguien cerca.

Implementamos funciones para acelerar y desacelerar y que la animación se vea mas realista, y tuvimos que adaptar un poco a los espacios. De hecho, cuando intentamos hacerlo con la ciudad ya armada el carro era muy grande y se chocaban.

En la parte de graficas pusimos un objeto cámara para poder moverlo a través de un menú y poder ver la escena desde distintas perspectivas, utilizamos un modelo muy básico de cubos para representar la ciudad y los autos para hacer la animación más eficiente y esperamos para el reto poder implementarlo en la ciudad.

Para la parte de los semáforos , les pusimos focos con esferas para lograr que cambien de color con el THREE group.