



**Tecnológico
de Monterrey**

Evidencia 1. Actividad Integradora

Modelación de sistemas multiagentes con gráficas computacionales

Kevin Martínez A01660520

Itthandehui Joselyn Espinoza Mazón A01734547

5 de Marzo de 2024

Parte 1: Sistemas MultiAgentes.

Diagrama de Clases

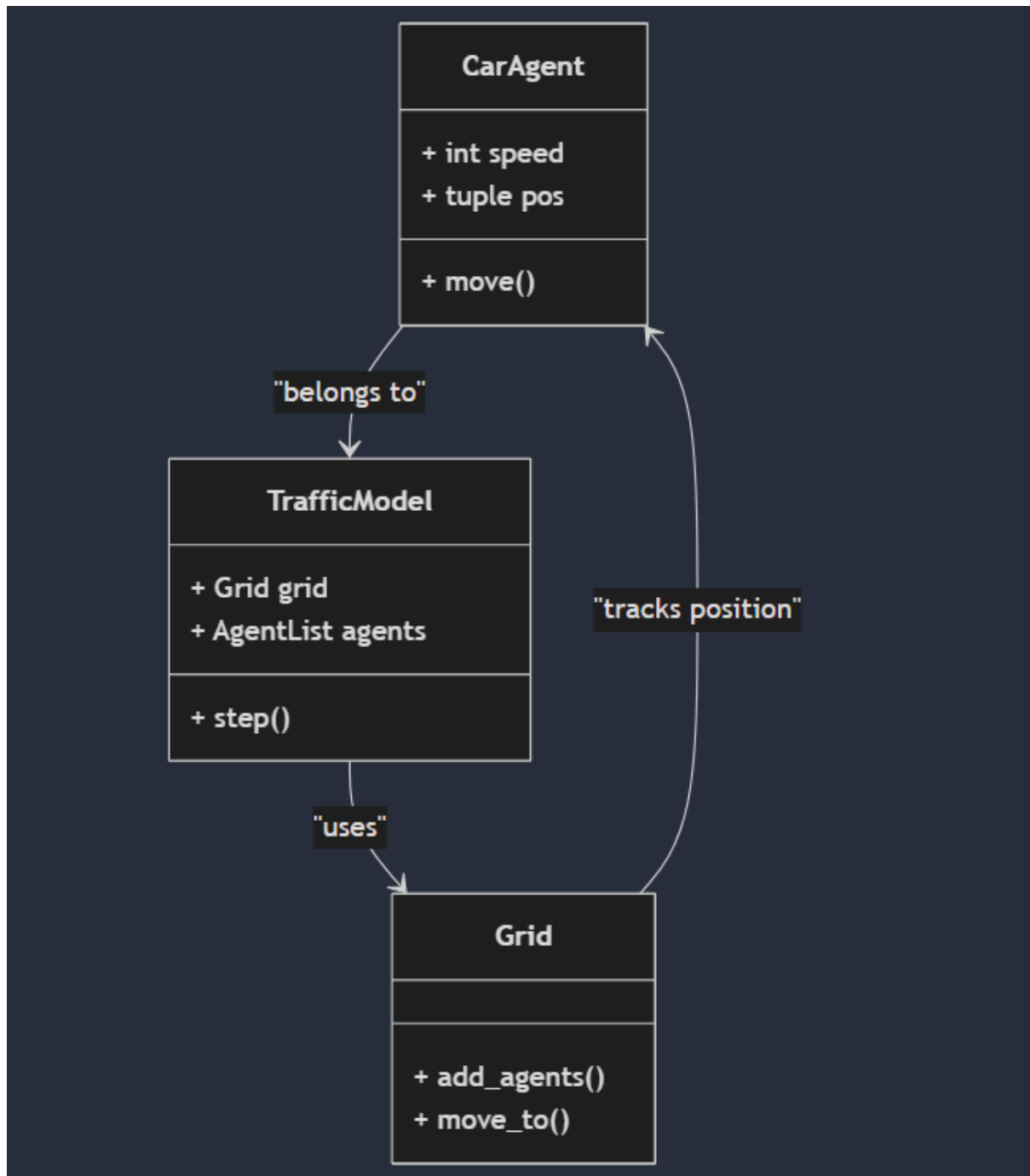


Diagrama de Secuencia

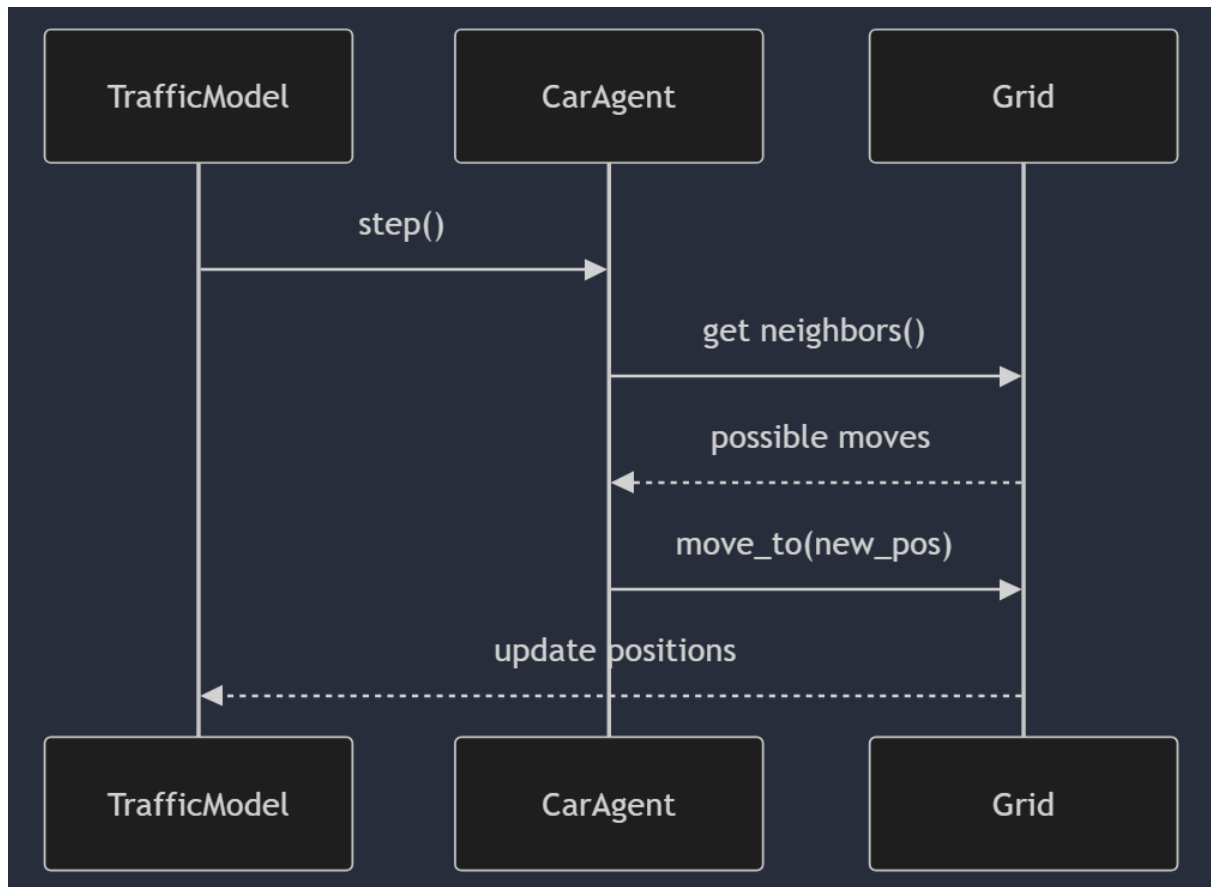
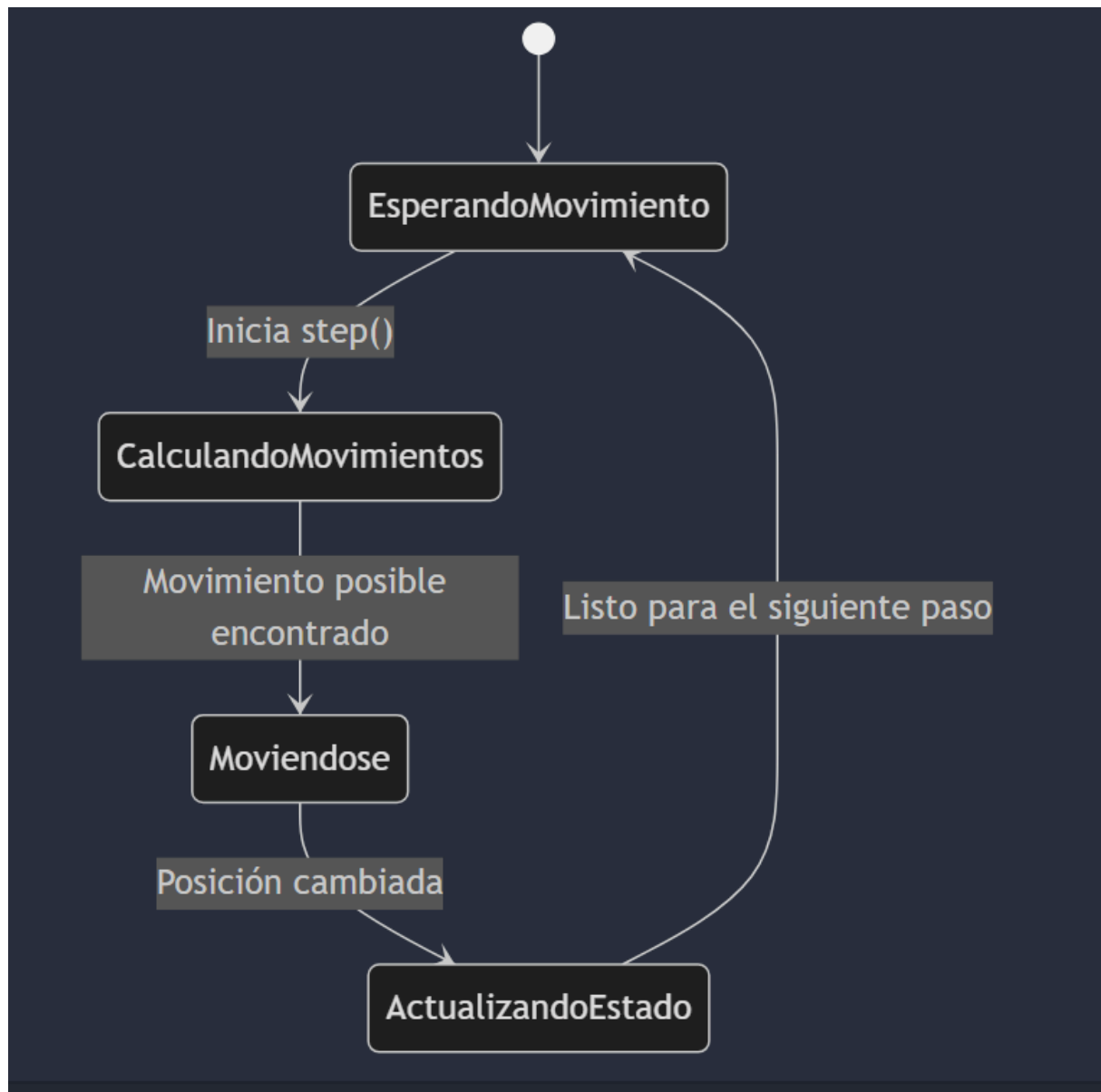
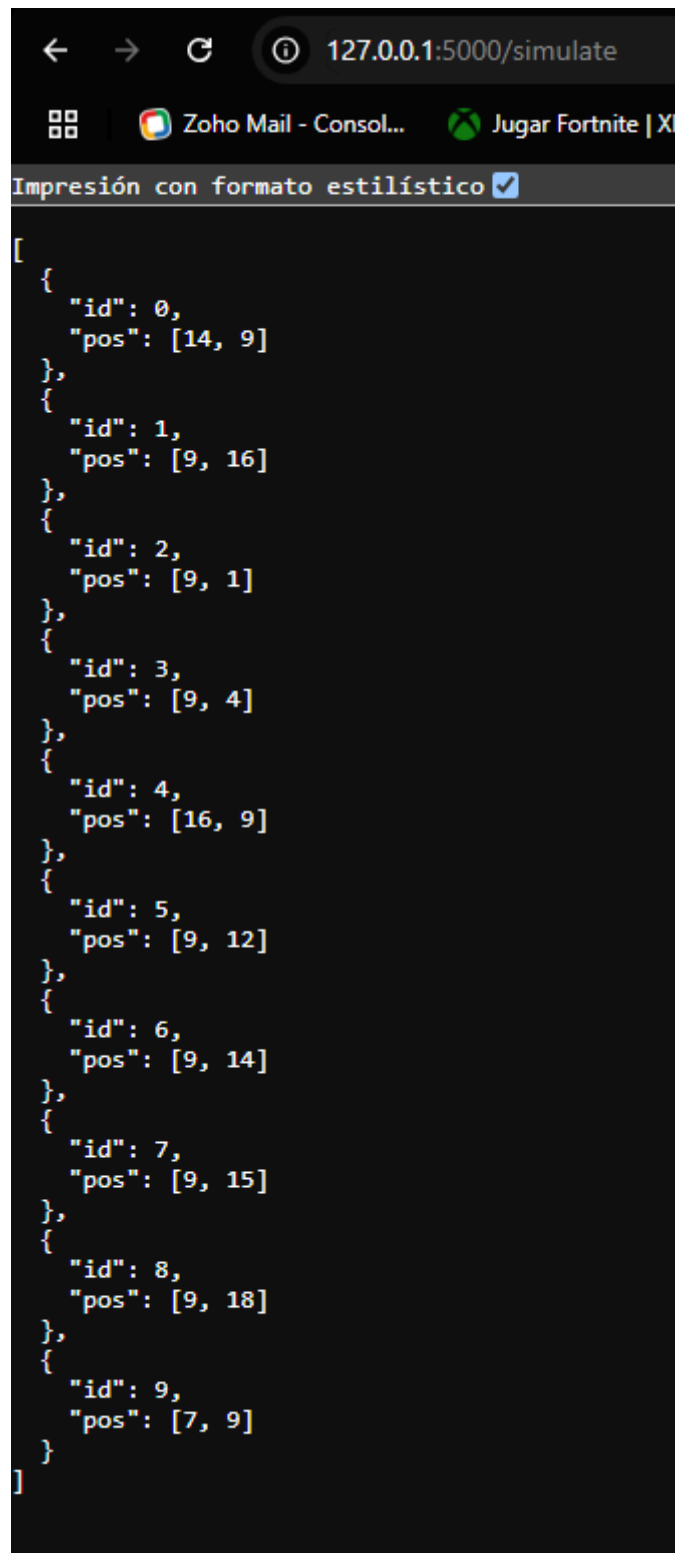


Diagrama de Protocolo de Interacción



Tiempo necesario hasta que los agentes completan la tarea

El tiempo necesario dependerá de la cantidad de pasos en la simulación. En este caso, cada iteración del ciclo de simulación representa un paso en el tiempo, y podemos ver que los agentes siguen moviéndose sin una condición clara de finalización.



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/simulate'. The browser has tabs for 'Zoho Mail - Consol...' and 'Jugar Fortnite | Xb...'. Below the browser window, there is a dark-themed code editor showing a JSON array of agent positions. The array contains 10 objects, each with an 'id' and a 'pos' (position) array. The positions are: [14, 9], [9, 16], [9, 1], [9, 4], [16, 9], [9, 12], [9, 14], [9, 15], [9, 18], and [7, 9].

```
[
  {
    "id": 0,
    "pos": [14, 9]
  },
  {
    "id": 1,
    "pos": [9, 16]
  },
  {
    "id": 2,
    "pos": [9, 1]
  },
  {
    "id": 3,
    "pos": [9, 4]
  },
  {
    "id": 4,
    "pos": [16, 9]
  },
  {
    "id": 5,
    "pos": [9, 12]
  },
  {
    "id": 6,
    "pos": [9, 14]
  },
  {
    "id": 7,
    "pos": [9, 15]
  },
  {
    "id": 8,
    "pos": [9, 18]
  },
  {
    "id": 9,
    "pos": [7, 9]
  }
]
```

Número de movimientos realizados por los agentes

Cada vez que un agente se mueve, se registra en la simulación. Basándonos en el log:

Todos los agentes están en constante movimiento (excepto los que no encuentran movimientos válidos).

Algunos agentes se quedan atrapados en posiciones sin salida y no pueden moverse más (⊘ Agente CarAgent (Obj 9) no encontró movimientos válidos en (9, 15)).

```
[{"id": 5, "pos": [9, 11]}, {"id": 6, "pos": [9, 12]}, {"id": 7, "pos": [9, 14]}, {"id": 8, "pos": [9, 19]}, {"id": 9, "pos": [8, 9]}]
127.0.0.1 - - [05/Mar/2025 17:53:43] "GET /simulate HTTP/1.1" 200 -
✓ Ejecutando un paso de la simulación...
✚ Moviendo autos en el grid...
✓ Agente CarAgent (Obj 2) se mueve de (17, 9) a (18, 9)
✓ Agente CarAgent (Obj 3) se mueve de (9, 17) a (9, 18)
✓ Agente CarAgent (Obj 4) se mueve de (9, 0) a (9, 1)
✓ Agente CarAgent (Obj 5) se mueve de (9, 3) a (9, 4)
⊘ Agente CarAgent (Obj 6) no encontró movimientos válidos en (19, 9).
✓ Agente CarAgent (Obj 7) se mueve de (9, 11) a (9, 10)
✓ Agente CarAgent (Obj 8) se mueve de (9, 12) a (9, 11)
✓ Agente CarAgent (Obj 9) se mueve de (9, 14) a (9, 15)
⊘ Agente CarAgent (Obj 10) no encontró movimientos válidos en (9, 19).
✓ Agente CarAgent (Obj 11) se mueve de (8, 9) a (9, 9)
📋 Datos actualizados: [{"id": 0, "pos": [18, 9]}, {"id": 1, "pos": [9, 18]}, {"id": 2, "pos": [9, 1]}, {"id": 3, "pos": [9, 4]}, {"id": 4, "pos": [19, 9]}, {"id": 5, "pos": [9, 10]}, {"id": 6, "pos": [9, 11]}, {"id": 7, "pos": [9, 15]}, {"id": 8, "pos": [9, 19]}, {"id": 9, "pos": [9, 9]}]
127.0.0.1 - - [05/Mar/2025 17:53:44] "GET /simulate HTTP/1.1" 200 -
✓ Ejecutando un paso de la simulación...
✚ Moviendo autos en el grid...
✓ Agente CarAgent (Obj 2) se mueve de (18, 9) a (17, 9)
✓ Agente CarAgent (Obj 3) se mueve de (9, 18) a (9, 17)
✓ Agente CarAgent (Obj 4) se mueve de (9, 1) a (9, 2)
✓ Agente CarAgent (Obj 5) se mueve de (9, 4) a (9, 3)
✓ Agente CarAgent (Obj 6) se mueve de (19, 9) a (18, 9)
⊘ Agente CarAgent (Obj 7) no encontró movimientos válidos en (9, 10).
✓ Agente CarAgent (Obj 8) se mueve de (9, 11) a (9, 12)
✓ Agente CarAgent (Obj 9) se mueve de (9, 15) a (9, 14)
✓ Agente CarAgent (Obj 10) se mueve de (9, 19) a (9, 18)
✓ Agente CarAgent (Obj 11) se mueve de (9, 9) a (8, 9)
📋 Datos actualizados: [{"id": 0, "pos": [17, 9]}, {"id": 1, "pos": [9, 17]}, {"id": 2, "pos": [9, 2]}, {"id": 3, "pos": [9, 3]}, {"id": 4, "pos": [18, 9]}, {"id": 5, "pos": [9, 10]}, {"id": 6, "pos": [9, 12]}, {"id": 7, "pos": [9, 14]}, {"id": 8, "pos": [9, 18]}, {"id": 9, "pos": [8, 9]}]
127.0.0.1 - - [05/Mar/2025 17:53:45] "GET /simulate HTTP/1.1" 200 -
✓ Ejecutando un paso de la simulación...
✚ Moviendo autos en el grid...
✓ Agente CarAgent (Obj 2) se mueve de (17, 9) a (16, 9)
✓ Agente CarAgent (Obj 3) se mueve de (9, 17) a (9, 16)
✓ Agente CarAgent (Obj 4) se mueve de (9, 2) a (9, 1)
✓ Agente CarAgent (Obj 5) se mueve de (9, 3) a (9, 4)
✓ Agente CarAgent (Obj 6) se mueve de (18, 9) a (17, 9)
✓ Agente CarAgent (Obj 7) se mueve de (9, 10) a (9, 9)
✓ Agente CarAgent (Obj 8) se mueve de (9, 12) a (9, 13)
✓ Agente CarAgent (Obj 9) se mueve de (9, 14) a (9, 15)
✓ Agente CarAgent (Obj 10) se mueve de (9, 18) a (9, 17)
✓ Agente CarAgent (Obj 11) se mueve de (8, 9) a (7, 9)
📋 Datos actualizados: [{"id": 0, "pos": [16, 9]}, {"id": 1, "pos": [9, 16]}, {"id": 2, "pos": [9, 1]}, {"id": 3, "pos": [9, 4]}, {"id": 4, "pos": [17, 9]}, {"id": 5, "pos": [9, 9]}, {"id": 6, "pos": [9, 13]}, {"id": 7, "pos": [9, 15]}, {"id": 8, "pos": [9, 17]}, {"id": 9, "pos": [7, 9]}]
127.0.0.1 - - [05/Mar/2025 17:53:46] "GET /simulate HTTP/1.1" 200 -
```

Posibles estrategias para mejorar la eficiencia

Para reducir el tiempo total y la cantidad de movimientos innecesarios, podemos considerar estrategias de optimización:

- **Implementar una meta específica:**

Si cada CarAgent tiene un destino (ejemplo: moverse a una coordenada específica), podrían optimizar su camino en lugar de moverse aleatoriamente.

- **Evitar bloqueos y movimientos innecesarios:**

Un algoritmo de evitación de colisiones permitiría a los autos tomar rutas alternativas en lugar de quedarse atrapados. Si un agente detecta que no tiene movimientos válidos, podría retroceder o buscar otra opción.

Parte 3: Integración

Diagrama de Clases

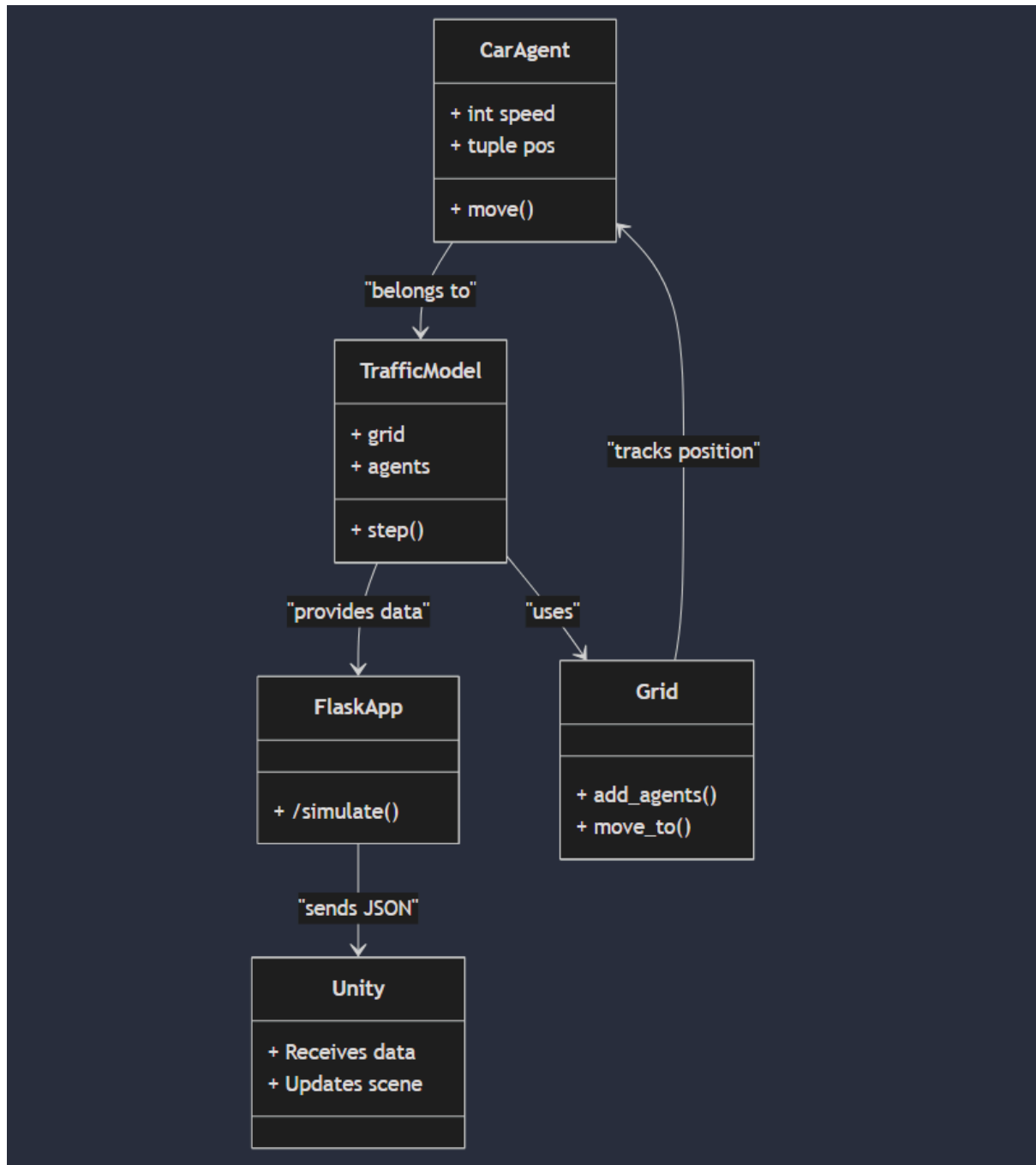


Diagrama de protocolo de interacción.

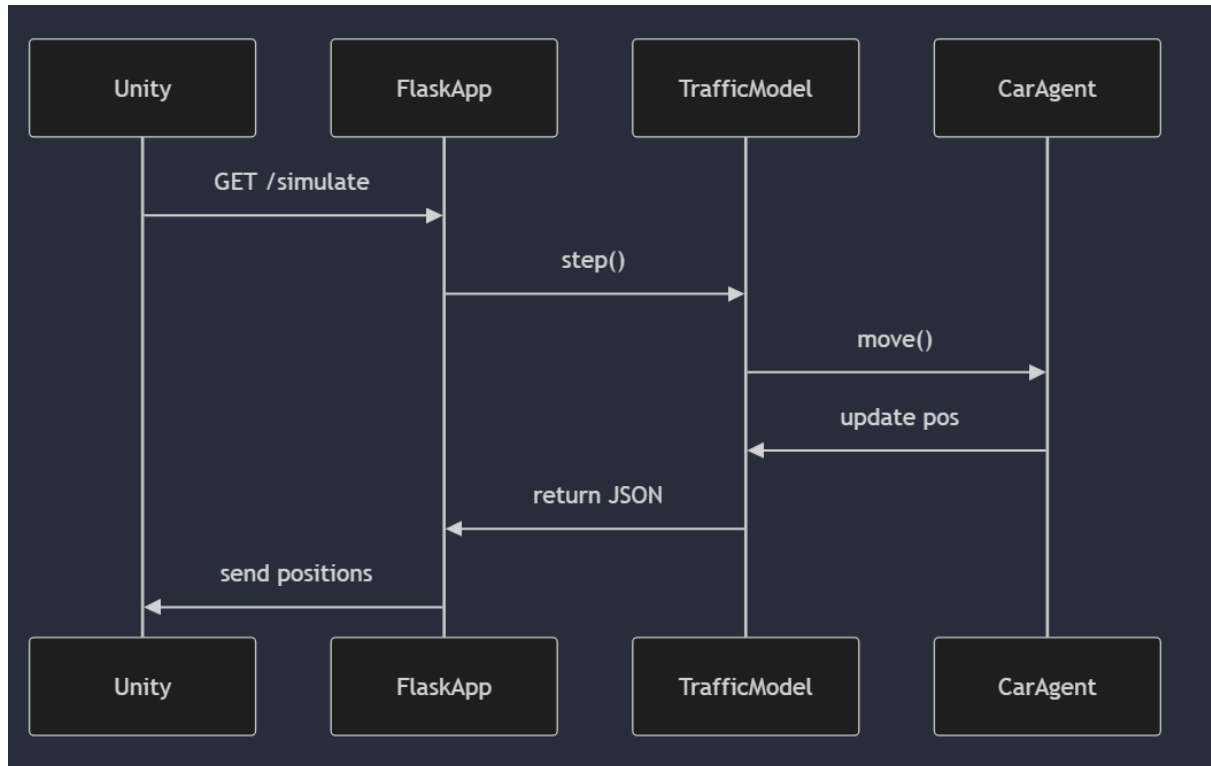
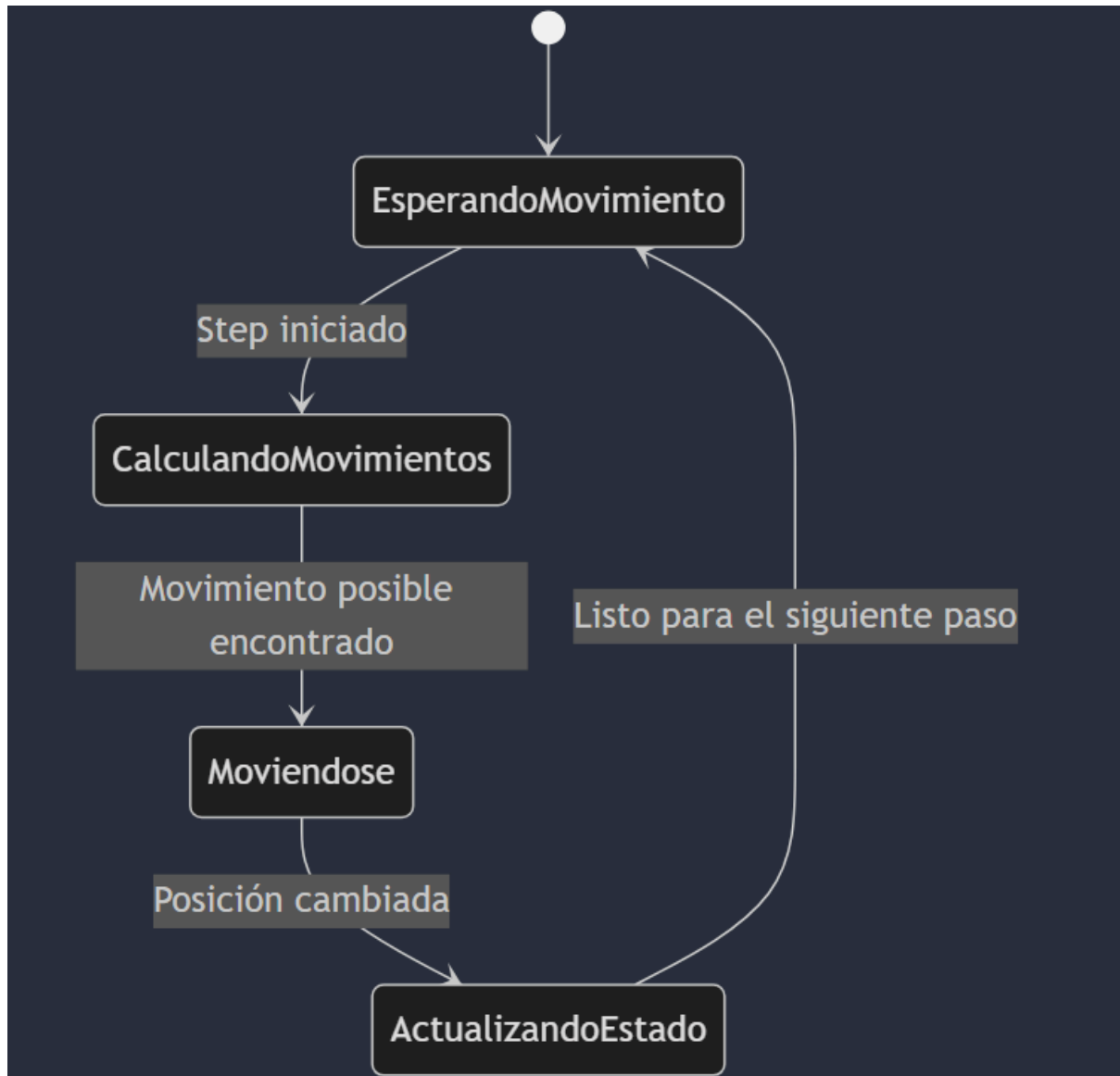


Diagrama de Secuencia



Herramientas utilizadas para la integración

- **Backend (Servidor Flask)**
 - Flask → Para recibir peticiones de Unity y gestionar la simulación con AgentPy.
 - AgentPy → Para modelar la simulación multiagente de tráfico.
- **Simulación**
 - AgentPy → Define los agentes (CarAgent), sus movimientos y la cuadrícula (Grid).
 - NumPy → Manejo de datos y cálculos matemáticos para los movimientos.
- **Frontend (Unity)**
 - C# + UnityWebRequest → Permite a Unity solicitar datos al servidor Flask (GET /simulate).
 - GameObjects en Unity → Representan visualmente a los CarAgent.
 - JSON Parsing → Unity interpreta la respuesta JSON del servidor y actualiza la posición de los autos.

Dificultades y Soluciones

Dificultad	Causa	Solución Implementada
Errores en neighbors().	CarAgent intentaba acceder a posiciones inexistentes en Grid.	Se implementó un cálculo manual de vecinos en vez de usar neighbors().
Los agentes se desactivaron demasiado pronto.	Se desactivan inmediatamente tras un error.	Se implementó un contador de errores (invalid_steps) antes de desactivar un agente.
Los autos no se movían en Unity	Unity no recibía correctamente la información JSON	Se verificó el formato de JSON y se imprimieron logs en Unity para depuración.

