



**Tecnológico  
de Monterrey**

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

***M5. Revisión de avance 2***

***Profesores:***

Raul V. Ramirez Velarde

Iván Axel Dounce Nava

***Presentan:***

Salvador Rodríguez Paredes A01704562

Luis Adrian Abarca Gomez A01798043

Yudith Korina Hernández Palacios A00834231

**Fecha de entrega:**

10 de marzo del 2024

## ***REVISIÓN DE AVANCE 1***

### ***Fortalezas y área de oportunidad:***

***Yudith:*** Entre mis soft skill se encuentran aspectos como la creatividad, resiliencia, pensamiento crítico y adaptabilidad. Por otro lado, mis hard skill se encuentran enfocadas en el manejo de C#, Unity y Matlab, por lo que las áreas de oportunidad que observo se encuentran relacionadas al desarrollo de agentes inteligentes en Python y el tema de poder crear sistemas multiagentes.

***Salvador:*** El trabajo bajo presión, desarrolló web, la comunicación y el trabajo en equipo considero que son de mis principales fortalezas, respecto a mis áreas de oportunidad puedo encontrar el manejo y uso adecuado de algoritmos y estructuras de datos.

***Luis:*** Buen manejo y conocimiento utilizando las herramientas de Unity, modelación de escenarios y manejo de instancias para múltiples eventos que ocurran al mismo tiempo o reaccionen a una acción específica, facilidad de manejo de C#.

### ***Expectativas y lo que esperamos lograr:***

***Yudith:*** Las expectativas que tengo del curso es poder aprender más acerca de la modelación de gráficas en Unity y todo lo relacionado con este motor de desarrollo ya que es algo que me interesa mucho y quiero seguir especializándome en ello. Por otra parte, también espero comprender lo que son los sistemas multiagentes, sus ventajas y para qué sirven, así como aprender a desarrollar e implementar dicho sistema en Python para retomar el manejo de este lenguaje y ampliar mi percepción de lo que se puede lograr con el uso del mismo.

***Salvador:*** Me gustaría que este curso me brindara las herramientas suficientes para lograr desarrollar sistemas multiagentes y empaparme un poco con la construcción de gráficas computacionales, en particular el área de los videojuegos me gusta bastante pero ver que se pueden ocupar en otros ámbitos como la movilidad urbana también me llama la atención.

**Luis:** En este curso, buscó desarrollar habilidades sólidas en el manejo experto de Unity, incluyendo la modelación de escenarios y el manejo de instancias para eventos concurrentes o reacciones específicas. Mi objetivo es aprender a controlar y asignar eficientemente los espacios de estacionamiento disponibles en una zona urbana. Deseo crear una simulación que evite que los conductores circulen sin rumbo en busca de estacionamiento, utilizando mi conocimiento en C# para lograrlo. Quiero dominar la capacidad de crear sistemas inteligentes que gestionen de manera efectiva los espacios, mejorando así la experiencia de estacionamiento en la ciudad.

### ***Compromisos para lograrlo como equipo:***

Lo que esperamos lograr al final de este reto como equipo es ser capaces de resolver el reto de movilidad a través del desarrollo de un código en Python en conjunto con la implementación en Unity.

Para esto nos comprometemos a:

- Trabajar colaborativamente
- Hacer uso de la comunicación efectiva
- Apoyarnos mutuamente
- Enfocarnos en soluciones
- Perpetuar el aprendizaje continuo.

### ***Reto a desarrollar:***

Gestionar y asignar de manera eficiente los espacios de estacionamiento en una determinada área urbana tiene como objetivo principal evitar la circulación innecesaria de vehículos en busca de lugares para estacionarse. Este enfoque busca mitigar la congestión vehicular y

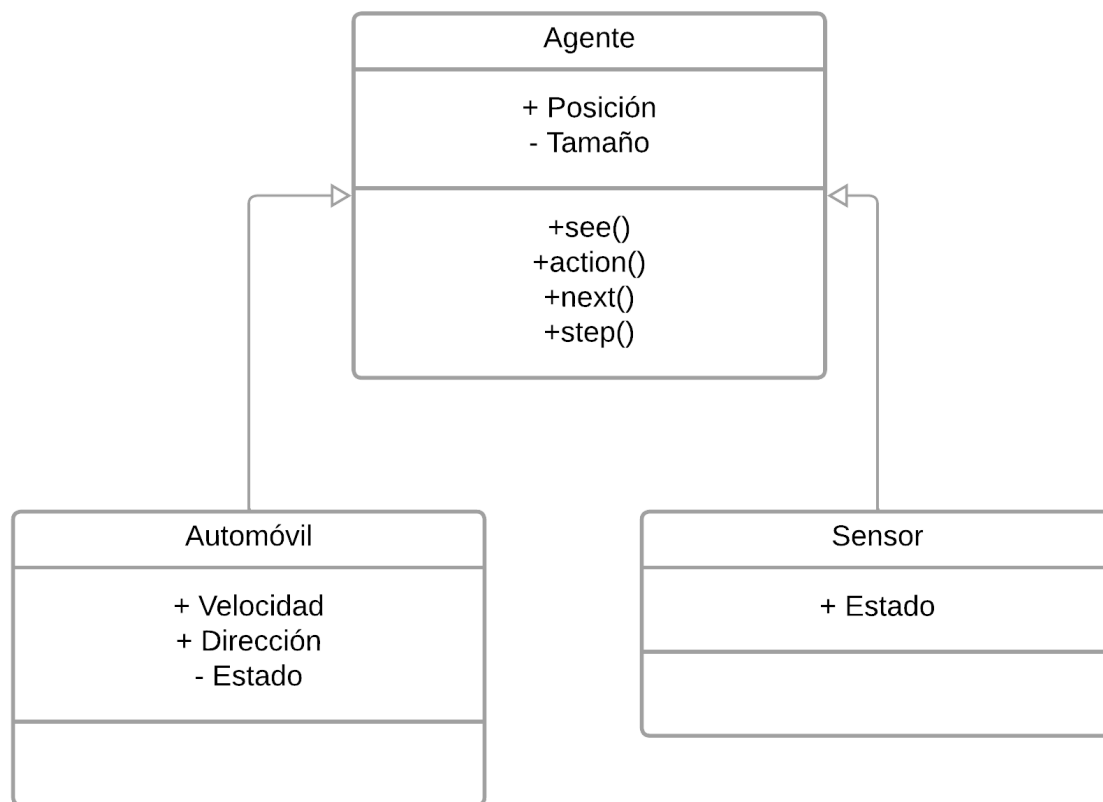
mejorar la fluidez del tráfico al asegurar una distribución ordenada y controlada de los espacios de estacionamiento disponibles. Este proceso contribuye a optimizar la utilización de recursos urbanos y promover una movilidad más eficiente al minimizar la búsqueda de estacionamiento, reduciendo así el tiempo y la energía dedicados a esa tarea.

### ***Agentes involucrados:***

- ➔ Agente que modele el comportamiento de un automóvil.
- ➔ Agente que modele los sensores de los lugares de estacionamiento.

### ***Diagramas:***

Diagrama de clase presentando los distintos agentes involucrados.



### ***Descripción del comportamiento esperado para los agentes:***

En el contexto del control y asignación de espacios de estacionamiento, consideramos utilizar un paradigma de razonamiento híbrido que combine elementos de razonamiento reactivo y deliberativo.

Esto debido a que gracias al razonamiento reactivo los agentes serían capaces de utilizarlo para detectar rápidamente la disponibilidad de espacios de estacionamiento y tomar decisiones inmediatas en función de dicha información. Por ejemplo, al detectar un automóvil que busca estacionamiento, un agente podría reaccionar asignando un espacio disponible cercano. Por otra parte, la aplicación de un razonamiento deliberativo por parte de los agentes permitiría planificar de manera más estratégica la asignación de espacios de estacionamiento. Por ejemplo, considerando la distribución general de vehículos en la zona y optimizando la asignación para minimizar el tiempo que los autos pasan buscando estacionamiento.

En conjunto, este enfoque híbrido permitiría a los agentes tomar decisiones rápidas en situaciones inmediatas y asimismo planificar de manera más estratégica para mejorar la eficiencia general del sistema de estacionamiento ya que además, los agentes deben ser capaces de coordinarse entre sí para evitar conflictos y maximizar la utilización de los espacios disponibles. Aclarando otros aspectos, de cómo deseamos abordar la problemática, podemos especificar que la entrada al estacionamiento será libre y por lo tanto sin cobro, asimismo habrá peatones como agentes implícitos en la problemática.

### ***Plan de trabajo y aprendizaje:***

Hemos usado una matriz con fechas, considerando las dos semanas que restan del bloque. Dividimos las tareas restantes como tres bloques principales por lo que las trabajaremos de esta manera, dedicando media semana a cada uno de los objetivos por realizar, aumentando progresivamente el número de horas dedicadas.

<b>Objetivo</b>	<b>Fecha de inicio de trabajo</b>	<b>Fecha de conclusión</b>	<b>Horas dedicadas por día</b>
<b><i>Comunicación entre agentes</i></b>	Domingo 4 de Marzo, 2024	Jueves 7 de Marzo, 2024	1-2 hora de trabajo al día
<b><i>Toma de decisiones multiagente</i></b>	Viernes 8 de Marzo, 2024	Lunes 11 de Marzo, 2024	3-4 de hora de trabajo
<b><i>Integración del reto</i></b>	Martes 12 de Marzo, 2024	Jueves 14 de Marzo, 2024	2+ horas de trabajo al día

## ***REVISIÓN DE AVANCE 2***

### ***Estrategia elegida para implementar***

Solución que permita a los semáforos coordinar sus tiempos para reducir la congestión de un cruce y que los peatones puedan cruzar la calle de forma segura.

### ***Agentes involucrados y forma de interacción:***

#### **→ Vehículos:**

- Tipo de Agente: Práctico
- Funcionalidad: Calcula la ruta más eficiente y se adapta al tráfico.
- Interacción: Solicita paso al controlador de tráfico y cruza el cruce cuando se le indica.

#### **→ Semáforos:**

- Tipo de Agente: Híbrido (Reactivo)
- Funcionalidad: Cambia su estado según las condiciones del tráfico y coordina con otros semáforos.
- Interacción: Recibe mensajes del controlador de tráfico para cambiar su estado y coordinar con otros semáforos.

#### **→ Controlador de Tráfico:**

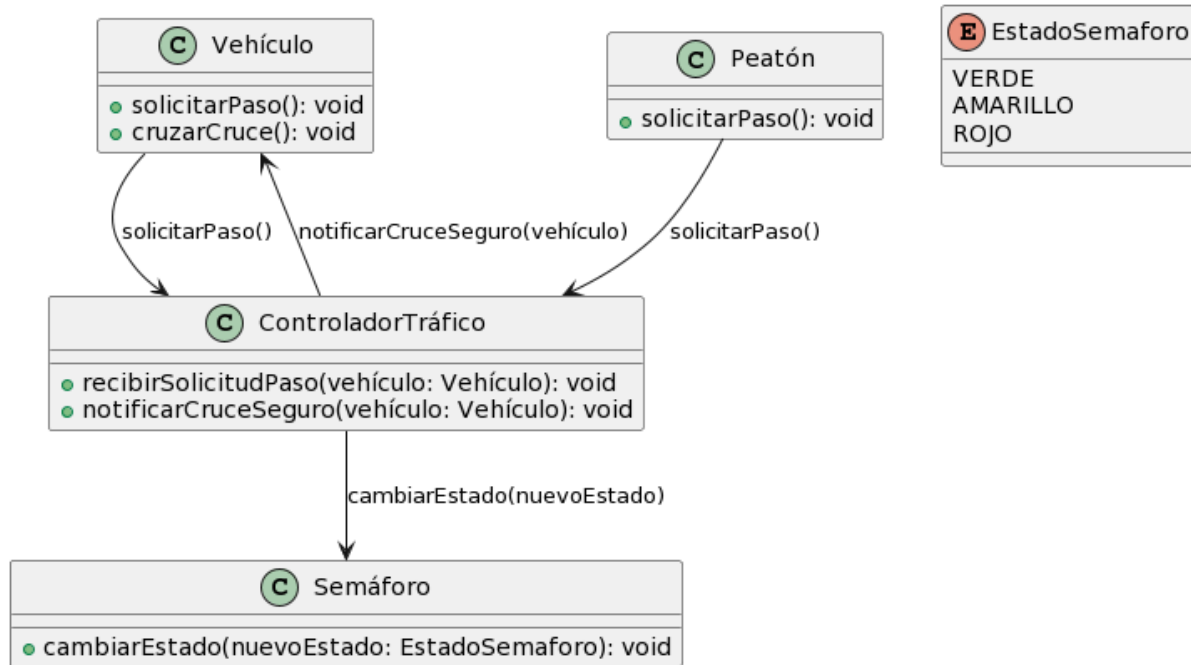
- Tipo de Agente: Deductivo
- Funcionalidad: Toma decisiones basadas en reglas y conocimiento previo.
- Interacción: Recibe solicitudes de paso de vehículos, coordina con los semáforos y comunica a los vehículos cuando pueden cruzar.

#### **→ Peatones:**

- Tipo de Agente: Práctico
- Funcionalidad: Deciden cuándo cruzar una calle de forma segura.

- Interacción: Solicitan paso al controlador de tráfico y cruzan cuando se les indica.

### Diagrama UML actualizado



Nota: se utilizó la herramienta online [plantuml.com](http://plantuml.com) para generar el diagrama de clases de agentes a través de un código donde se declaran los atributos y funciones de cada clase.

### Código en plantuml para la visualización del diagrama de clases de agentes

```

@startuml
class Vehículo {
    + solicitarPaso(): void
    + cruzarCruce(): void
}

class Semáforo {
    + cambiarEstado(nuevoEstado: EstadoSemaforo): void
}

class ControladorTráfico {
    + recibirSolicitudPaso(vehículo: Vehículo): void
    + notificarCruceSeguro(vehículo: Vehículo): void
}

class Peatón {
    + solicitarPaso(): void
}
  
```



}

```
enum EstadoSemaforo {  
    VERDE  
    AMARILLO  
    ROJO  
}
```

```
Vehículo --> ControladorTráfico : solicitarPaso()  
ControladorTráfico --> Semáforo : cambiarEstado(nuevoEstado)  
ControladorTráfico --> Vehículo : notificarCruceSeguro(vehículo)  
Peatón --> ControladorTráfico : solicitarPaso()
```

@enduml

### ***Protocolos de interacción (Contract NET)***

Inicio del Protocolo:

- Un vehículo envía un mensaje al controlador de tráfico solicitando paso en un cruce.
- El controlador de tráfico recibe la solicitud y notifica a los semáforos del cruce sobre la solicitud de paso.

Oferta de Servicio:

- Cada semáforo evalúa la solicitud de paso y responde al controlador de tráfico con una oferta de servicio (permitir o denegar el paso).

Selección de Oferta:

- El controlador de tráfico recibe las ofertas de servicio de los semáforos y selecciona la oferta más adecuada (por ejemplo, el semáforo más cercano y con menor congestión).

Confirmación de Servicio:

- El controlador de tráfico confirma al vehículo la disponibilidad de paso y coordina el cambio de estado del semáforo seleccionado.

Ejecución del Servicio:

- El semáforo cambia su estado a verde y permite el paso del vehículo.
- El vehículo cruza el cruce de manera segura.

Fin del Protocolo:

- El controlador de tráfico finaliza el protocolo y queda a la espera de nuevas solicitudes de paso.

### ***Enlace a Github:***

Se mantiene el mismo repositorio donde todos los integrantes del equipo documentamos nuestros avances y haremos aportaciones código para poder completar de manera satisfactoria el reto: <https://github.com/itschavaa/TC2008B>