



Tecnológico
de Monterrey

*Modelación de sistemas multiagentes con
gráficas computacionales*

M5. Revisión de avance 1

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:

29 de febrero del 2024

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.

Repository de Github:

Se generó el siguiente 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>

Asimismo, se generó un grupo de whatsapp con el objetivo de permitir la comunicación efectiva entre los distintos participantes del equipo y se establecieron horarios de reuniones a través de la plataforma de zoom para poder trabajar colaborativamente en tiempo real.

Reto a desarrollar:

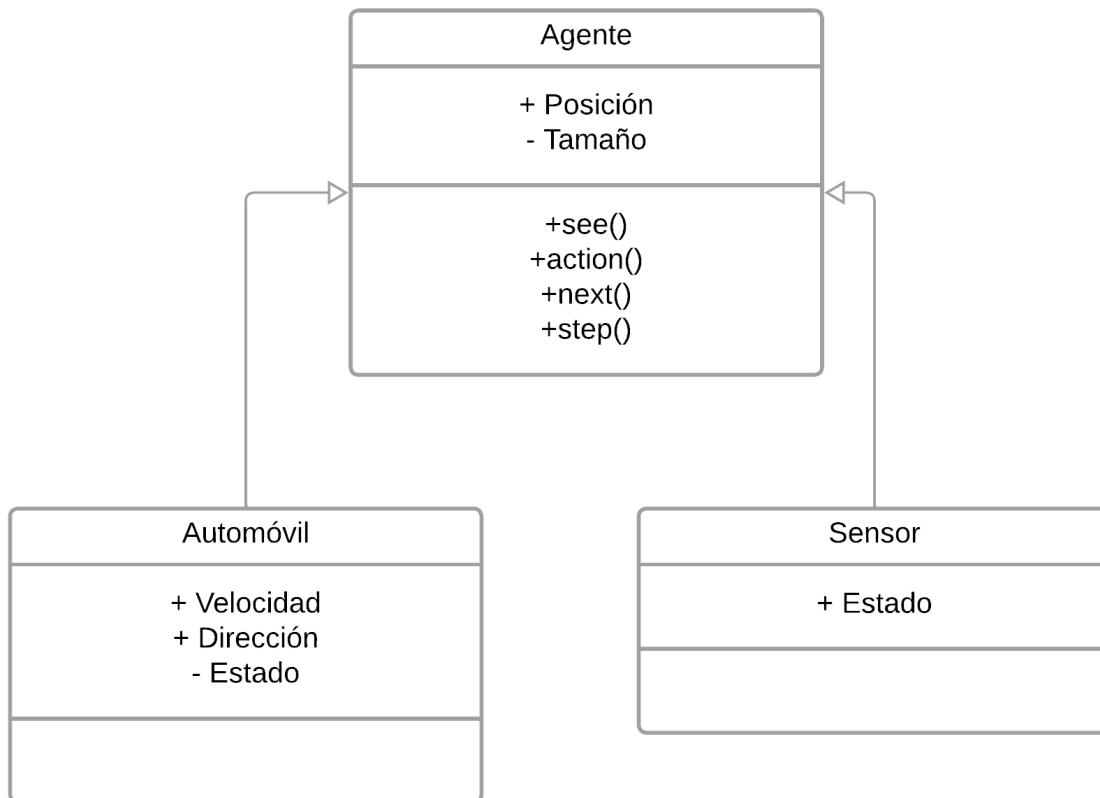
Gestionar y asignar de manera eficiente los espacios de estacionamiento en una determinada área urbana tiene como objetivo principal evitar la circulación innecesaria de vehículos en busca de lugares para estacionarse. Este enfoque busca mitigar la congestión vehicular y mejorar la fluidez del tráfico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

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



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

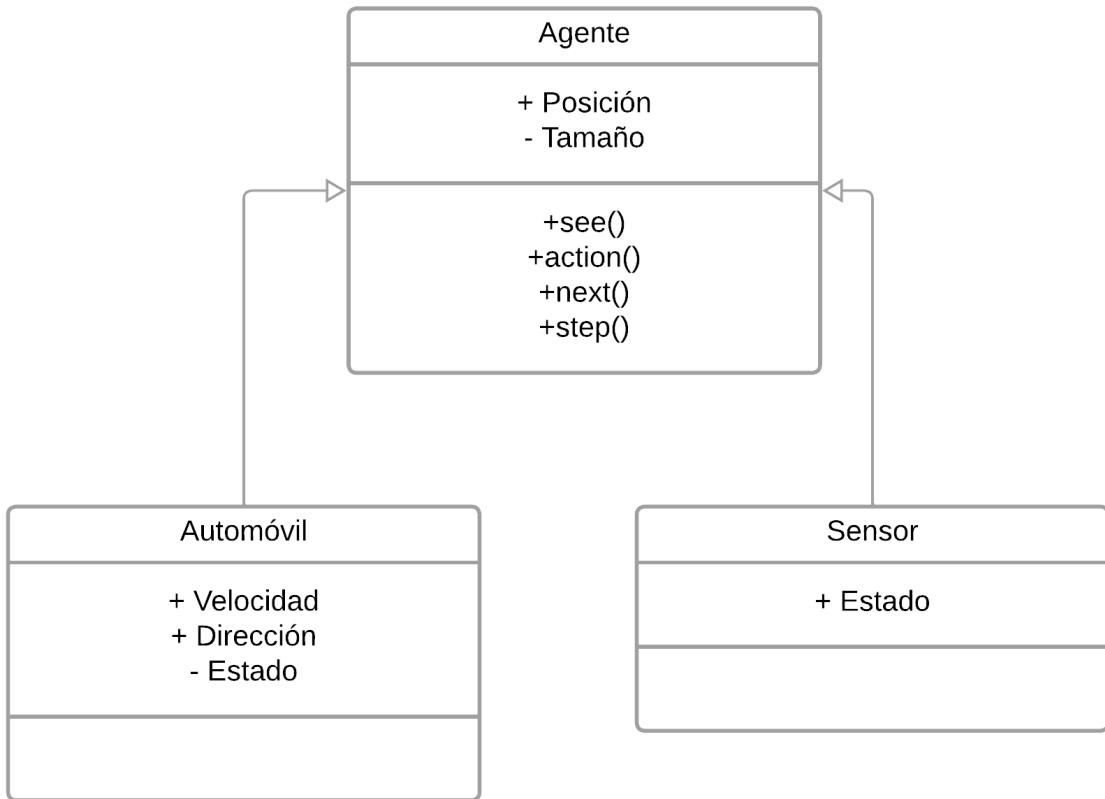
mejorar la fluidez del tráfico al asegurar una distribución ordenada y controlada de los espacios de estacionamiento disponibles. Este proceso contribuye a optimizar la utilización de recursos urbanos y promover una movilidad más eficiente al minimizar la búsqueda de estacionamiento, reduciendo así el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

Objetivo	Fecha de inicio de trabajo	Fecha de conclusión	Horas dedicadas por dia
<i>Comunicación entre agentes</i>	Domingo 4 de Marzo, 2024	Jueves 7 de Marzo, 2024	1-2 hora de trabajo al día
<i>Toma de decisiones multiagente</i>	Viernes 8 de Marzo, 2024	Lunes 11 de Marzo, 2024	3-4 de hora de trabajo
<i>Integración del reto</i>	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 (Reactiva)
- 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:

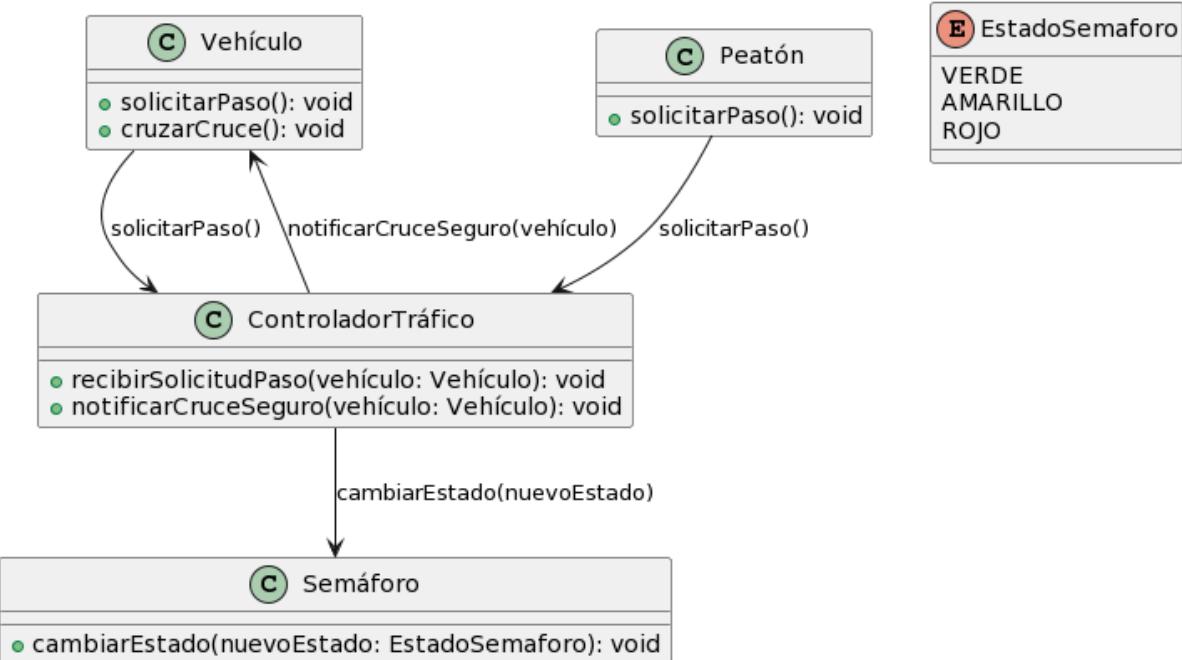
- Tipo de Agente: Deductivo
- Funcionalidad: Toma decisiones basadas en reglas y conocimiento previo.
- Interacción: Recibe solicitudes de paso de vehículos, coordina con los semáforos y comunica a los vehículos cuando pueden cruzar.

→ 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 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 Vehiculo {
    +solicitarPaso(): void
    +cruzarCruce(): void
}

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

class ControladorTráfico {
    +recibirSolicitudPaso(vehiculo: Vehiculo): void
    +notificarCruceSeguro(vehiculo: Vehiculo): void
}

class Peaton {
    +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>



Tecnológico
de Monterrey

*Modelación de sistemas multiagentes con
gráficas computacionales*

M5. Revisión de avance 3

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

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:

→ TerrainAgent

Tipo de agente: práctico

Funcionalidad: representa el terreno o las áreas sin ocupar por vehículos o semáforos.

Interacción: No participa en interacciones directas con otros agentes y sirve como marcador de áreas vacías en la simulación.

→ TrafficLightAgent

Tipo de agente: reactivo

Funcionalidad: representa los semáforos del cruce controlando el flujo de vehículos.

Interacción: El agente trafficLightAgent cambia su estado de luz (verde, amarillo, rojo) según un conjunto de reglas definidas. Los vehículos observan el estado de los semáforos para decidir si avanzan o se detienen.

→ CarAgent

Tipo de agente: reactivo

Funcionalidad: representa los vehículos que circulan por el cruce.

Interacción: los agentes carAgent se mueven siguiendo reglas basadas en las luces de los semáforos y la presencia de otros vehículos en las cercanías. Pueden cambiar de dirección en ciertas condiciones, como cuando llegan a un borde del área de simulación o cuando deciden girar.

Código de los agentes implementados (link al Google Collaboratory):

- <https://colab.research.google.com/drive/1QIURnntuAOU6xJ50YhqH9d0rSnRuIlOx?usp=sharing>

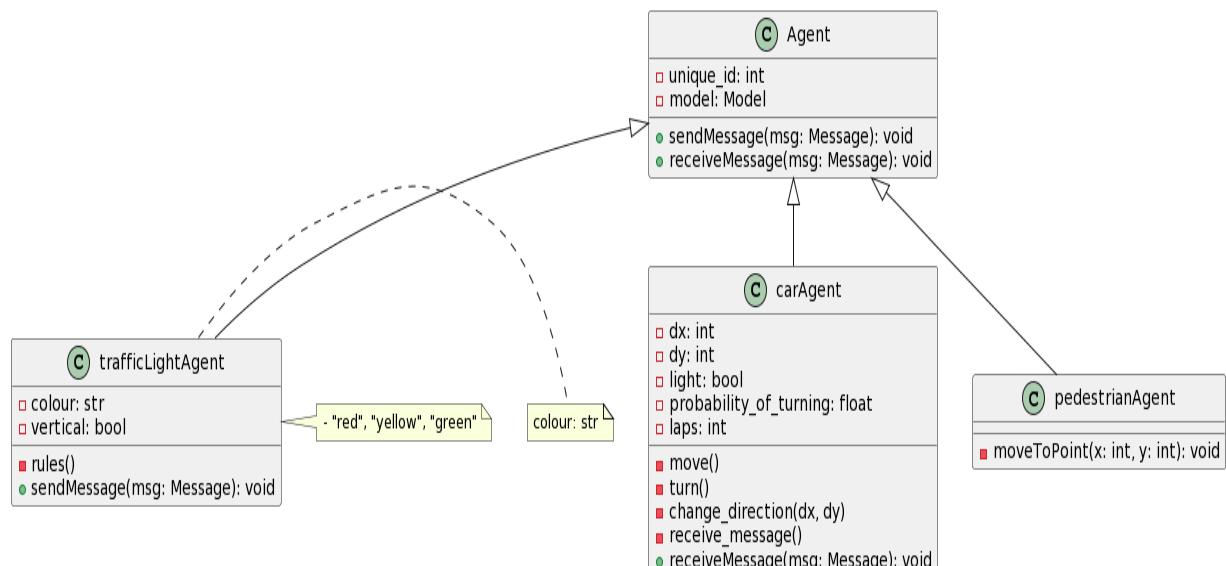
Código de la implementación grafica (link al repositorio de Github con el proyecto de Unity 3D):

- https://github.com/HayabusaLA/TC2008B.570_SP_MovilidadUrbana.git

Link al repositorio de Github:

- <https://github.com/itschavaa/TC2008B>

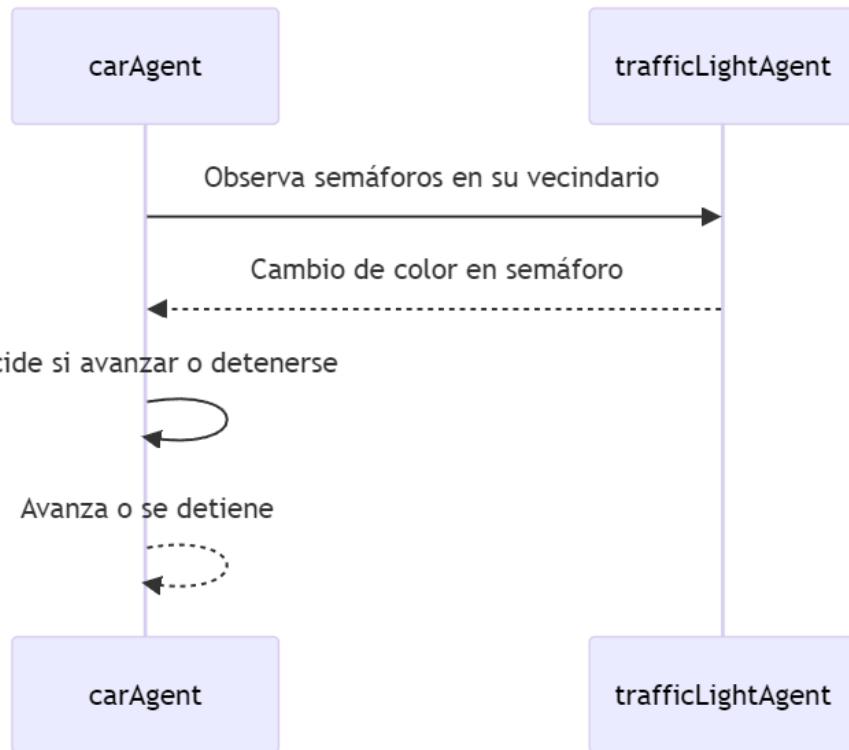
Diagrama de clases para agentes actualizado



Descripción del diagrama

En este diagrama de clases de agentes tanto el agente `trafficLightAgent` como el agente `carAgent` pueden enviar mensajes del tipo inform. `trafficLightAgent` tiene un atributo `colour` para representar el color actual del semáforo y un método `rules()` para definir las reglas de cambio de color. `carAgent` representa los vehículos en la simulación, con atributos como dirección de movimiento (`dx`, `dy`), estado del semáforo (`light`), probabilidad de girar (`probability_of_turning`), y métodos para moverse y cambiar de dirección y `pedestrianAgent` representa a los peatones, con un método para moverse a un punto específico en el entorno.

Diagrama de secuencia para agentes o protocolos de interacción



Descripción del diagrama

Este diagrama visualiza la interacción entre los vehículos y los semáforos, mostrando cómo los agentes se comunican para tomar decisiones en el sistema de control de tráfico.

Protocolos de interacción (Votación)

Descripción general

- Semáforos como candidatos: En el protocolo de Votación, los agentes votan por un candidato específico. En este caso, los vehículos (carAgent) "votan" por el estado de los semáforos (trafficLightAgent), considerándolos como los candidatos a seguir.
- Decisión basada en mayoría: En el protocolo de Votación, se toma una decisión basada en la mayoría de votos. En el código, los vehículos deciden su movimiento basándose en la mayoría de semáforos en verde en su vecindario.
- Comunicación indirecta: En el protocolo de Votación, los agentes no necesitan comunicarse directamente entre sí, sino que observan el resultado de las votaciones para tomar decisiones. En el código, los vehículos observan el estado de los semáforos para decidir si avanzan o no.

Pasos detallados del protocolo

1. Inicio del proceso:

Cada vehículo (carAgent) observa los semáforos (trafficLightAgent) en su vecindario para determinar el número de semáforos en verde.

2. Votación:

Cada vehículo "vota" por avanzar si la mayoría de los semáforos en su vecindario están en verde.

3. Recuento de votos:

Los vehículos cuentan el número de semáforos en verde en su vecindario para determinar si deben avanzar o detenerse.

4. Decisión:

Si la mayoría de los semáforos están en verde, los vehículos avanzan; de lo contrario, se detienen.

5. Iteración:

El proceso se repite en cada paso de la simulación, con los vehículos observando y votando según el estado actual de los semáforos en su vecindario.

6. Fin del proceso:

Todos los vehículos en el cruce han tomado una decisión sobre si avanzar o detenerse, basándose en el estado de los semáforos en su vecindario

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.

Objetivo	Fecha de inicio de trabajo	Fecha de conclusión	Horas dedicadas por día	Porcentaje de avance	Tiempo en el que se completo	Persona encargada
Comunicación entre agentes	Sábado 9 de Marzo, 2024	Martes 12 de Marzo, 2024	2-3 hora de trabajo al día	90% completado, aún se necesitan más pruebas de	3 días exactos para completar esta implementación. No hubo diferencias en el	Salvador

				fiabilidad.	tiempo estimado.	
Toma de decisiones multiagente	Domingo 10 de Marzo, 2024	Miércoles 13 de Marzo, 2024	3-4 de hora de trabajo	95% Los agentes trabajan conforme a lo establecido, se requieren breves pruebas y validaciones .	En 4 días para completar esta implementación, existieron diversos errores y correcciones a realizar para comprobar que el programa funcionara como se tenía planeado.	Yudith
Modelación en Unity	Viernes 8 de marzo, 2024.	Miércoles 13 de marzo, 2024	3 horas de trabajo	100% el trabajo se encuentra completo, únicamente se agregan detalles visuales que no afecten en el desarrollo del trabajo principal.	2 días para completar el escenario básico donde se montó la simulación. Los días extras se están ocupando para embellecer el paisaje más que para agregarle funcionalidad..	Luis
Integración del reto	Martes 12 de Marzo, 2024	Jueves 14 de Marzo, 2024	1-2 horas de trabajo al día	80% Ambas partes del back-end como el front-end se encuentran en etapas finales, sólo se necesita lograr una comunicación fiable y pruebas de funcionamiento correcto.	En proceso de integrar todos los elementos.	Todos los integrantes del equipo

Aprendizaje logrado:

Como equipo, esta experiencia de trabajo colaborativo nos ha enseñado valiosas lecciones que sin duda fortalecen nuestro proceso de desarrollo. Durante la implementación del plan de trabajo que abarcó desde el sábado 9 de marzo hasta el jueves 14 de Marzo del 2024, cada uno de nosotros se enfrentó a desafíos y logros significativos en nuestras respectivas áreas.

Salvador, a cargo de la comunicación entre agentes, aprendió la importancia de la minuciosidad en las pruebas de fiabilidad. Aunque alcanzamos el 90% de avance en el tiempo esperado, comprendimos que la paciencia y la exhaustividad son claves para asegurar un funcionamiento óptimo.

Yudith, encargada de la toma de decisiones multiagente, experimentó la necesidad de flexibilidad y adaptación. Aunque alcanzamos un sólido 95% de progreso, las correcciones y pruebas adicionales mostraron la importancia de estar preparados para ajustes inesperados en el proceso.

Luis, liderando la modelación en Unity, nos enseñó sobre la importancia del equilibrio entre funcionalidad y estética. Al haber completado el trabajo al 100%, reconoció que dedicar tiempo extra para embellecer el entorno no debe comprometer la eficiencia en el desarrollo principal.

Finalmente, en la integración del reto, todos como equipo entendimos la vital importancia de la comunicación y la colaboración. Aunque estamos en un 80% de avance, comprendimos que la unión de las partes back-end y front-end requiere un esfuerzo conjunto para garantizar la fiabilidad y el correcto funcionamiento del programa.

Como cierre, hemos aprendido que la planificación cuidadosa, la adaptabilidad ante imprevistos, el equilibrio entre funcionalidad y estética, y la comunicación efectiva son pilares fundamentales para alcanzar nuestros objetivos de manera exitosa. Esta experiencia nos deja con lecciones sólidas que seguiremos aplicando en futuros proyectos, consolidando así nuestro crecimiento como equipo cohesionado y efectivo.

PROPUESTA DE SOLUCION

MOVILIDAD URBANA

SALVADOR RODRÍGUEZ
PAREDES A01704562

LUIS ADRIAN ABARCA GOMEZ
A01798043

YUDITH KORINA HERNÁNDEZ
PALACIOS A00834231

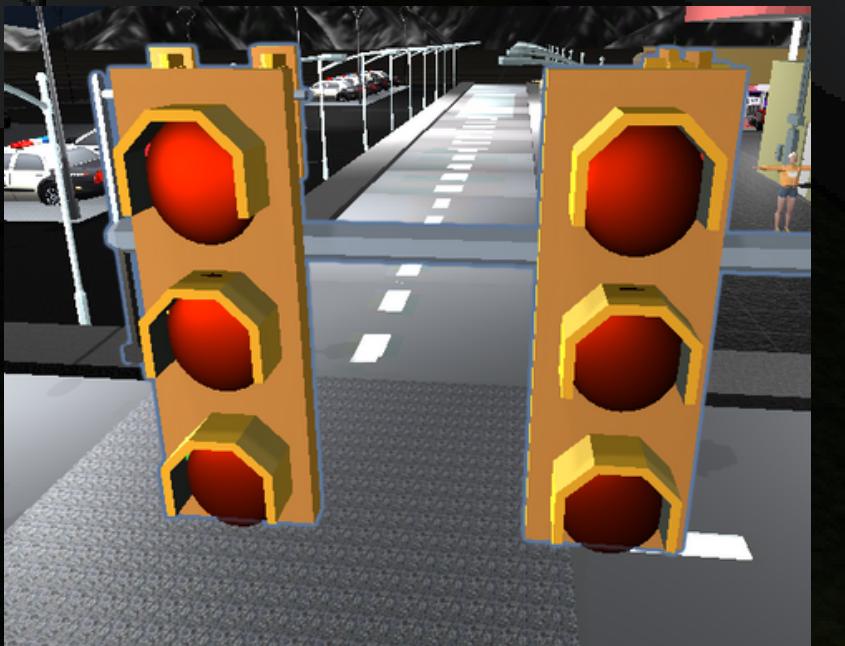
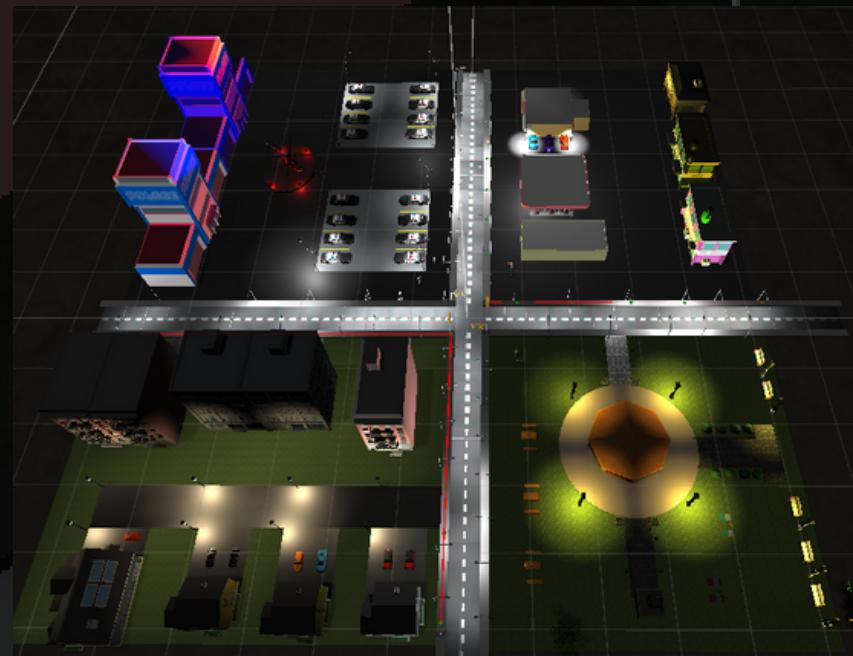
Problematica escogida

Coordinacion de tiempos entre semáforos, así, logrando reducir la congestión de un cruce. O, quizás, indicar en qué momento un vehículo va a cruzar una intersección y que de esta forma, el semáforo puede determinar el momento y duración de la luz verde.

Método de resolución

El modelo representa una intersección de dos calles de doble sentido en una cuadricula discreta. Los vehículos deben esperar la señal del semáforo para avanzar y tienen una probabilidad de girar a la derecha o izquierda, dependiendo del carril. Los agentes son:

1. Terreno: Delimita el espacio de circulación de los vehículos.
2. Semáforo: Regula el color de la luz para controlar el flujo vehicular.
3. Vehículo: Sujeto a la luz del semáforo y puede girar.



Agentes involucrados

Autos

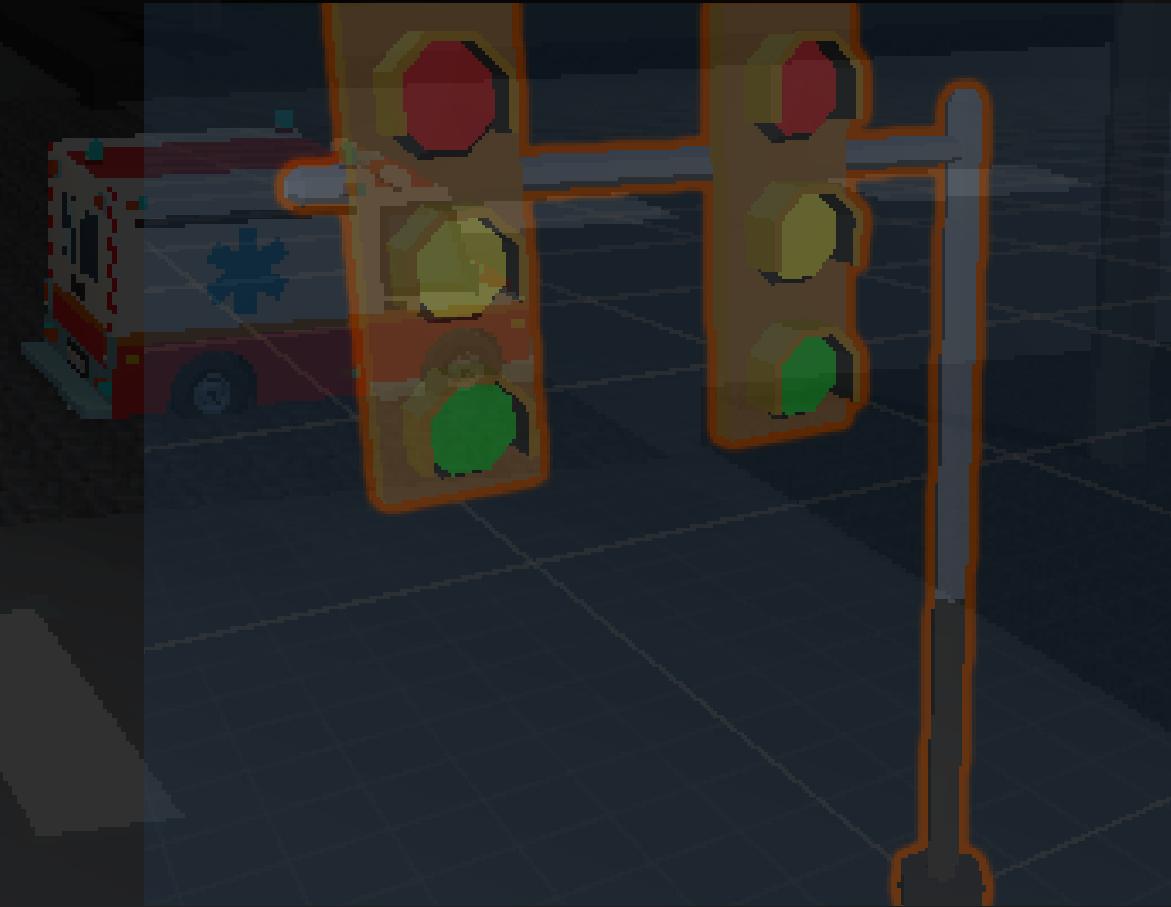
- La orientación del agente se fija al crearlo y puede cambiarse con el método `change_direction()`.
- En la intersección, el agente puede cambiar su orientación al carril con una probabilidad si no hay vecinos de terreno o semáforo.
- Reglas de acción:
 - Avanza si la casilla adelante está desocupada y no hay semáforos con luz roja en la misma orientación.
 - Avanza si la casilla adelante está libre y hay un semáforo en verde en su vecindad.
 - No avanza si la casilla adelante está bloqueada por otro agente.

Semaforo

- Si no hay ningún vehículo con la misma orientación (vertical u horizontal) cerca del semáforo, su color cambia a amarillo.
- Si hay un vehículo con la misma orientación cerca del semáforo y ningún semáforo en un radio de 3 casillas en la misma orientación está en verde, su color cambia a verde.
- Si hay un vehículo con la misma orientación cerca del semáforo y al menos uno de los semáforos en un radio de 3 casillas en la misma orientación está en verde, su color cambia a rojo.

Comunicación y razonamiento

Ambos agentes son de tipo deductivo, ya que siguen un conjunto de reglas predeterminadas y utilizan su conocimiento previo, estos presentan una comunicación de tipo broadcast, donde el semáforo manda un mensaje de que está en verde a los coches para que puedan cambiar su estado a cruce.





Modelación de sistemas multiagentes y gráficas computacionales

Aprendizajes de reto

Alumna: Yudith Korina Hernández Palacios A00834231

¿Por qué seleccionaron el modelo multiagentes utilizado?

El modelo multiagentes se seleccionó porque este permitía simular de manera más realista la interacción entre los diferentes elementos en el sistema de tráfico urbano, como semáforos, vehículos y peatones, por lo que este enfoque parece ideal para estudiar y mejorar la coordinación de los semáforos y reducir la congestión vehicular.

¿Cuáles fueron las variables que se tomaron al momento de tomar la decisión?

Las variables que se consideraron fueron la complejidad del sistema de tráfico urbano, la capacidad de representar la interacción entre diferentes agentes de forma realista, así como la flexibilidad para implementar estrategias de coordinación entre semáforos y movilidad de los vehículos y peatones.

¿Cuál es la interacción de esas variables con respecto al resultado de la simulación?

La interacción entre estas variables permitió obtener una simulación más precisa y realista del tráfico urbano, lo que facilitó la implementación de estrategias de coordinación entre semáforos y la movilidad de los vehículos y peatones, lo que contribuyó a reducir la congestión vehicular.

¿Por qué seleccionaron el diseño gráfico presentado?

El entorno en Unity fue el ideal por su capacidad para representar de manera clara y visualmente atractiva la simulación del tráfico urbano. Se buscó un diseño que fuera intuitivo y fácil de entender para el usuario, lo que facilita la visualización de la coordinación de los semáforos y la movilidad de los vehículos en el cruce simulado.

¿Cuáles son las ventajas que encuentras en la solución final presentada?

Considero que la solución final posee varias ventajas, como la capacidad de simular de manera realista la interacción entre semáforos y vehículos así como la posibilidad de implementar estrategias de coordinación entre semáforos para reducir la congestión vehicular, y la visualización clara y atractiva de la simulación a través del entorno en Unity.

¿Cuáles son las desventajas que existen en la solución presentada?

Una posible desventaja de la solución presentada podría ser la complejidad en la implementación y ajuste de los parámetros de la simulación, así como la necesidad de realizar pruebas exhaustivas para asegurar su correcto funcionamiento.

¿Qué modificaciones podrías hacer para reducir o eliminar las desventajas mencionadas?

Pienso que para reducir la complejidad en la implementación y ajuste de los parámetros, se podrían desarrollar herramientas de configuración más intuitivas y fáciles de usar. Además, observo que se podrían realizar pruebas para agilizar el proceso de verificación del funcionamiento de la simulación.

Una reflexión sobre tu proceso de aprendizaje.

A lo largo de este proceso fui capaz de aprender la importancia de la planificación cuidadosa, la flexibilidad para adaptarnos a cambios inesperados, y la comunicación efectiva en equipo. Asimismo, mis expectativas se cumplieron en su totalidad con las que tenía al inicio del bloque y considero que las herramientas que nos han impartido en esta materia han sido verdaderamente útiles para mi formación profesional como ITC.