



Tecnológico de Monterrey

**Modelación de sistemas multiagentes con gráficas computacionales (Gpo
570)**

Evidencia 2. Revisión 1 y 2

David Isaí Díaz Avilés (A01666448)

Edgar Martínez Retes (A01706825)

Elisheba Hannai Trejo Leyva (A01736967)

Ian García González (A01706892)

Óscar Orlando De la Paz Ibarra (A01644484)

Pablo Javier Arreola Velasco (A01747824)

23 de Febrero de 2026

Conformación del equipo

David Isaí Díaz Avilés

Fortalezas

- Capacidad para comprender cómo la autonomía y la interacción social influyen en sistemas complejos.
- Pensamiento sistémico para analizar tanto el comportamiento individual como el colectivo.
- Habilidad para traducir problemas reales en modelos multiagente estructurados.
- Experiencia en el diseño e implementación de sistemas distribuidos dentro de entornos de simulación.

Áreas de oportunidad

- Mejorar la organización y planificación del trabajo en equipo.
- Fortalecer el cumplimiento puntual de fechas de entrega.
- Profundizar en la documentación formal del diseño (diagramas, arquitecturas y protocolos).
- Optimizar la validación del sistema mediante pruebas y métricas de desempeño.

Expectativas del bloque

- Aplicar los conceptos teóricos en un proyecto funcional dentro de un entorno de simulación o videojuego.
- Implementar y comparar distintas arquitecturas de agentes (reactiva, deliberativa e híbrida).
- Diseñar mecanismos de coordinación y comunicación entre agentes.
- Analizar el comportamiento emergente generado por la interacción colectiva.

Edgar Martínez Retes

Fortalezas

- Compromiso, responsabilidad, amabilidad, disponibilidad, experiencia.

Áreas de oportunidad

- Paciencia, habilidad para delegar, falta de documentación.

Expectativas del bloque

- Poder ver más a fondo el comportamiento de los agentes más que la parte de modelación, ya que al final soy estudiante de ITC y lo que más me interesa es el diseño de estas funciones e interacciones.

Elisheba Hannai Trejo Leyva

Fortalezas

- Capacidad de análisis y resolución de problemas en entornos tecnológicos.
- Perseverancia para comprender temas complejos y aprender nuevas herramientas.
- Creatividad al diseñar soluciones y estrategias en proyectos de programación
- Pensamiento lógico y estructurado para implementar agentes y sistemas
- Responsabilidad y compromiso al trabajar en equipo.
- Buena organización en proyectos y entregas académicas.
- Disposición para aprender de errores y mejorar continuamente.

Áreas de oportunidad

- Fortalecer la confianza inicial al enfrentar herramientas o entornos nuevos.
- Mejorar la seguridad al tomar decisiones técnicas sin dudar tanto al principio.
- Optimizar la gestión del tiempo en proyectos largos o complejos,
- Practicar más la implementación desde cero para ganar fluidez en desarrollo.
- Confiar más en mis conocimientos y habilidades
- Continuar reforzando conceptos avanzados de programación e inteligencia artificial.

Expectativas del bloque

En esta materia espero fortalecer mis habilidades en el diseño e implementación de agentes inteligentes, así como comprender con mayor profundidad cómo funcionan los sistemas multiagente en entornos reales y simulados. Me interesa desarrollar un pensamiento más estratégico al momento de programar, combinando análisis, lógica y creatividad para resolver problemas complejos de manera autónoma.

Como compromiso personal, me propongo participar activamente en clase, dedicar tiempo constante a la práctica y no limitarme únicamente a cumplir con lo mínimo requerido. También me comprometo a enfrentar los retos técnicos con mayor seguridad, confiar más en mis capacidades y buscar soluciones incluso cuando al inicio parezcan difíciles. Asimismo, asumiré la responsabilidad de trabajar de manera organizada y colaborativa, aportando ideas y apoyando a mi equipo en cada etapa del proyecto.

Por último, mi expectativa es cerrar el curso no solo con conocimientos técnicos adquiridos, sino también con una mayor confianza en mi capacidad de desarrollar soluciones inteligentes y bien fundamentadas.

Pablo Javier Arreola Velasco

Fortalezas

- Responsable con el cumplimiento de plazos y fechas de entrega.
- Persona curiosa con interés constante por aprender cosas nuevas.
- Experiencia en desarrollo con Unity y en técnicas de inteligencia artificial clásica.
- Enfoque orientado al progreso y a la mejora continua del trabajo.

Áreas de oportunidad

- Poca paciencia con procesos largos y cuyos resultados no son inmediatos.
- Tendencia a enfocarse demasiado en detalles debido al perfeccionismo.
- Preferencia por estructuras de trabajo organizadas, por lo que puede resultar desafiante trabajar en equipos con poca definición de roles o responsabilidades.

Expectativas del bloque

- Desarrollar mayor paciencia en proyectos que requieren largos periodos de desarrollo y cuyos resultados no son proporcionales al esfuerzo.
- Adquirir experiencia trabajando con datos reales y desarrollando soluciones aplicadas a problemas del mundo real.

Ian García González

Fortalezas

- Facilidad para comunicar ideas técnicas de forma clara y accesible al equipo.
- Experiencia en programación orientada a objetos y diseño de patrones de software.
- Capacidad de adaptación rápida a nuevas tecnologías y frameworks.
- Habilidad para identificar y depurar errores de forma eficiente en sistemas complejos.

Áreas de Oportunidad

- Mejorar la participación activa en discusiones grupales y revisiones de código. Mejorar la gestión de prioridades cuando se tienen múltiples entregas simultáneas.
- Fortalecer habilidades de documentación técnica, incluyendo diagramas UML y especificaciones formales.
- Mejorar la capacidad de síntesis al presentar resultados y avances frente al equipo.
- Desarrollar mayor disciplina en la escritura de pruebas unitarias y de integración.

Expectativas del bloque

En este bloque espero comprender a profundidad cómo los agentes inteligentes interactúan y se coordinan dentro de sistemas multiagente, particularmente en escenarios de simulación. Me interesa explorar cómo las decisiones individuales de cada agente generan comportamientos emergentes a nivel colectivo, y cómo esto puede aplicarse a problemas reales. Me comprometo a mantener una comunicación constante con el equipo, cumplir con las entregas en tiempo y forma, y documentar adecuadamente cada fase del proyecto para facilitar la colaboración y la evaluación del sistema desarrollado.

Óscar Orlando De la Paz Ibarra

Fortalezas

- Capacidad para integrar conocimientos de desarrollo de software con modelado de sistemas inteligentes y arquitecturas distribuidas.
- Experiencia práctica en la implementación de soluciones tecnológicas reales, lo que facilita trasladar conceptos teóricos a aplicaciones funcionales.
- Pensamiento estructurado para diseñar sistemas complejos considerando tanto la lógica interna de los agentes como su interacción dentro del entorno.
- Habilidad para identificar oportunidades de optimización y mejora continua en procesos técnicos.
- Enfoque orientado a resultados, con visión tanto técnica como estratégica en proyectos de desarrollo.

Áreas de oportunidad

- Profundizar en la formalización académica del diseño de agentes (diagramas, especificaciones y protocolos formales).
- Delegar con mayor claridad cuando el proyecto involucra múltiples responsabilidades técnicas.
- Optimizar la gestión del tiempo cuando se combinan actividades académicas con proyectos externos.
- Dedicar más espacio a la experimentación controlada y análisis sistemático de resultados antes de pasar a implementación final.

Expectativas del bloque

En este bloque espero consolidar una comprensión más sólida de las arquitecturas de agentes, especialmente en la integración de modelos deliberativos, reactivos e híbridos dentro de entornos de simulación. Me interesa fortalecer la capacidad de diseñar sistemas multiagente no solo desde la perspectiva técnica, sino también desde el análisis del comportamiento emergente y la toma de decisiones distribuidas.

Busco desarrollar un enfoque más formal en la documentación y validación de sistemas, aplicando métricas y pruebas que permitan evaluar objetivamente el desempeño de los agentes. Asimismo, me propongo contribuir activamente al equipo, aportar soluciones estructuradas y asumir un rol de liderazgo técnico cuando sea necesario, manteniendo siempre una actitud colaborativa.

Mi meta al finalizar el curso es no solo contar con un proyecto funcional, sino haber fortalecido mi capacidad para diseñar, analizar e implementar sistemas inteligentes complejos con una base teórica bien fundamentada y aplicable a contextos reales.

Metas colectivas

- Desarrollar un proyecto funcional de sistemas multiagente dentro de un entorno de simulación o videojuego, integrando distintas arquitecturas (reactiva, deliberativa e híbrida) y comparando su desempeño.

- Diseñar e implementar mecanismos efectivos de comunicación, coordinación y toma de decisiones entre agentes, analizando el comportamiento emergente generado por su interacción colectiva.
- Profundizar en el entendimiento práctico del diseño de agentes inteligentes, priorizando la implementación, experimentación y análisis de resultados más allá de la modelación teórica.
- Aplicar herramientas de desarrollo y técnicas de inteligencia artificial en un contexto cercano a problemas reales, fortaleciendo la conexión entre teoría y práctica.
- Consolidar un pensamiento estratégico y sistémico que permita abordar problemas complejos de manera estructurada, colaborativa y orientada a resultados.

Acuerdos de trabajo

- Establecer una organización clara de roles, tareas y tiempos de entrega, fomentando la responsabilidad individual y el cumplimiento puntual de los acuerdos.
- Mantener una comunicación constante, abierta y respetuosa, promoviendo la colaboración, la retroalimentación constructiva y la delegación equilibrada de responsabilidades.
- Documentar adecuadamente el diseño del sistema (diagramas, arquitecturas, protocolos y métricas), asegurando claridad y calidad técnica en el proyecto.
- Implementar pruebas, validaciones y métricas de desempeño que permitan evaluar objetivamente el funcionamiento de los agentes y del sistema en conjunto.
- Asumir una actitud de mejora continua, enfrentando los retos técnicos con perseverancia, paciencia y disposición para aprender de los errores.
- Participar activamente en cada etapa del proyecto, aportando ideas, creatividad y compromiso para alcanzar un resultado sólido tanto a nivel técnico como
- colaborativo.
- Utilizar como herramientas de trabajo colaborativo WhatsApp como canal de comunicación rápida y seguimiento continuo, Zoom para reuniones de planeación y revisión de avances, y un [repositorio compartido en GitHub](#) para el control de versiones, organización del código y documentación colaborativa del proyecto.

Descripción del reto a desarrollar

El reto consiste en desarrollar una simulación de una flota de robotaxis autónomos que operan dentro de una ciudad digital. En el entorno aparecen solicitudes de viaje generadas dinámicamente por pasajeros que desean trasladarse de un punto de origen a un destino específico. El sistema debe asignar cada solicitud al taxi más conveniente y permitir que el vehículo planifique y ejecute su recorrido de forma autónoma, optimizando el tiempo de espera del pasajero y la eficiencia del trayecto. El objetivo principal es minimizar el tiempo promedio de espera y viaje, asegurando conducción segura, cumplimiento de señales de tránsito y capacidad de adaptación ante cambios simples del entorno (como congestión o taxi ocupado). La simulación representa un modelo simplificado de los sistemas utilizados en plataformas reales de movilidad autónoma.

Identificación de agentes involucrados

En la simulación intervienen tres tipos principales de agentes: el Despachador de Flota, los Taxis Autónomos y los Pasajeros.

Agente Despachador de Flota

El Agente Despachador de Flota es el encargado de la coordinación global del sistema. Su función es recibir las solicitudes de viaje generadas por los pasajeros, consultar el estado y ubicación de los taxis disponibles y asignar el vehículo más conveniente de acuerdo con criterios básicos como cercanía y disponibilidad. Este agente mantiene una visión general del sistema y actúa como entidad central de decisión.

Agente Taxi Autónomo

Los Agentes Taxi Autónomo son los vehículos encargados de ejecutar físicamente los trayectos. Cada taxi debe desplazarse hacia el punto de recogida, transportar al pasajero hasta su destino y actualizar constantemente su estado dentro del sistema (disponible, asignado, en viaje). Asimismo, deben interactuar con el entorno urbano respetando señales básicas de tránsito y evitando colisiones.

Agente Pasajero

El Agente Pasajero representa la entidad que solicita el servicio de transporte. Su función principal es generar una solicitud de viaje especificando origen y destino, esperar la asignación de un taxi y cambiar su estado cuando es recogido y posteriormente cuando el viaje finaliza.

Las relaciones entre los agentes se estructuran de la siguiente manera: el Pasajero envía su solicitud al Despachador; el Despachador asigna un Taxi; y el Taxi ejecuta el servicio e informa su estado al sistema.

Arquitectura Multiagente

El diseño del sistema incorpora distintos tipos de arquitectura con el propósito de representar niveles diferenciados de inteligencia y toma de decisiones. El Agente Despachador de Flota utiliza una arquitectura deliberativa basada en el modelo BDI (Creencias, Deseos e Intenciones), ya que requiere evaluar información global antes de tomar decisiones estratégicas de asignación. El Agente Taxi Autónomo implementa una arquitectura híbrida, combinando un componente reactivo para la seguridad inmediata con un componente deliberativo para la planificación y ejecución de la ruta. El Agente Pasajero se modela mediante una arquitectura reactiva simple, dado que su comportamiento se limita a responder a eventos del entorno sin necesidad de planificación compleja.

Componentes Arquitectónicos de Cada Agente

Agente Despachador (Arquitectura Deliberativa BDI)

En el caso del Despachador, sus creencias incluyen la ubicación y estado de cada taxi, las solicitudes activas y las distancias estimadas entre vehículos y pasajeros. Sus deseos se orientan a minimizar el tiempo de espera promedio, maximizar el número de viajes completados y mantener una distribución eficiente de la flota. Sus intenciones consisten en ejecutar el algoritmo de asignación del taxi más cercano disponible y mantener actualizado el estado general del sistema.

Agente Taxi Autónomo (Arquitectura Híbrida)

El Taxi Autónomo integra dos niveles de comportamiento. En su componente reactivo, cuenta con capas encargadas de la seguridad inmediata, como frenar ante obstáculos, respetar semáforos y mantener distancia con otros vehículos. Este nivel opera con prioridad sobre cualquier otro proceso.

En su componente deliberativo, el taxi mantiene creencias relacionadas con su ubicación actual, la ruta asignada y el estado del viaje. Sus deseos incluyen completar el trayecto de manera eficiente y segura. Sus intenciones se traducen en dirigirse al punto de recogida,

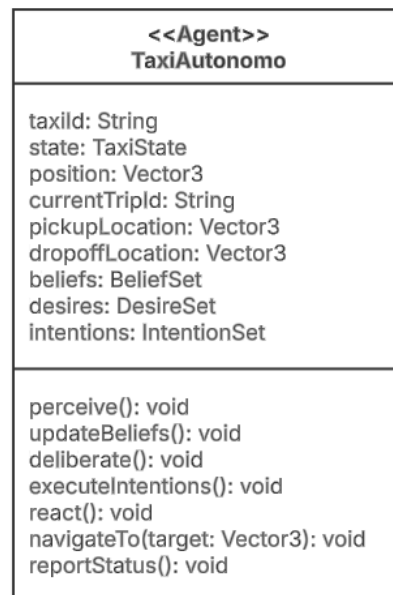
transportar al pasajero hacia el destino final y recalcular la ruta en caso de detectar un bloqueo simple o desviación necesaria.

Agente Pasajero (Arquitectura Reactiva)

El Pasajero opera bajo un esquema reactivo básico. Genera una solicitud de viaje, espera la asignación correspondiente y cambia su estado cuando el taxi llega al punto de recogida. Posteriormente, finaliza el viaje al llegar al destino.

Diagramas AUML

TaxiAutonomo:



El diseño del agente integra una arquitectura híbrida, combinando un componente deliberativo basado en el modelo BDI (Creencias, Deseos e Intenciones) para la planificación y toma de decisiones, junto con un componente reactivo orientado a la seguridad inmediata y al cumplimiento de reglas básicas de tránsito. Los atributos del agente modelan su estado interno, incluyendo información de posición, viaje actual y estructuras mentales. Las operaciones definidas representan sus capacidades principales, tales como percepción del entorno, actualización de creencias, deliberación, ejecución de intenciones, reacción ante eventos inmediatos y comunicación con el sistema.

DespachadorFlota:

<<Agent>> DespachadorFlota
activeRequests: List<TripRequest> fleetStatus: Map<String, TaxiStatus> beliefs: BeliefSet desires: DesireSet intentions: IntentionSet assignmentPolicy: AssignmentPolicy
receiveRequest(r: TripRequest): void updateTaxiStatus(s: TaxiStatus): void selectBestTaxi(r: TripRequest): String assignTrip(taxiId: String, r: TripRequest): void handleIncident(taxiId: String): void closeTrip(tripId: String): void

El agente DespachadorFlota representa la entidad coordinadora del sistema multiagente. Su responsabilidad principal es gestionar las solicitudes de viaje, mantener información actualizada sobre el estado de la flota y tomar decisiones de asignación utilizando una política definida. Su arquitectura es deliberativa basada en el modelo BDI, ya que requiere evaluar información global del sistema antes de seleccionar el taxi más conveniente. Los atributos modelan su conocimiento del entorno y de los recursos disponibles, mientras que las operaciones reflejan sus capacidades de recepción, evaluación, asignación y supervisión del servicio.

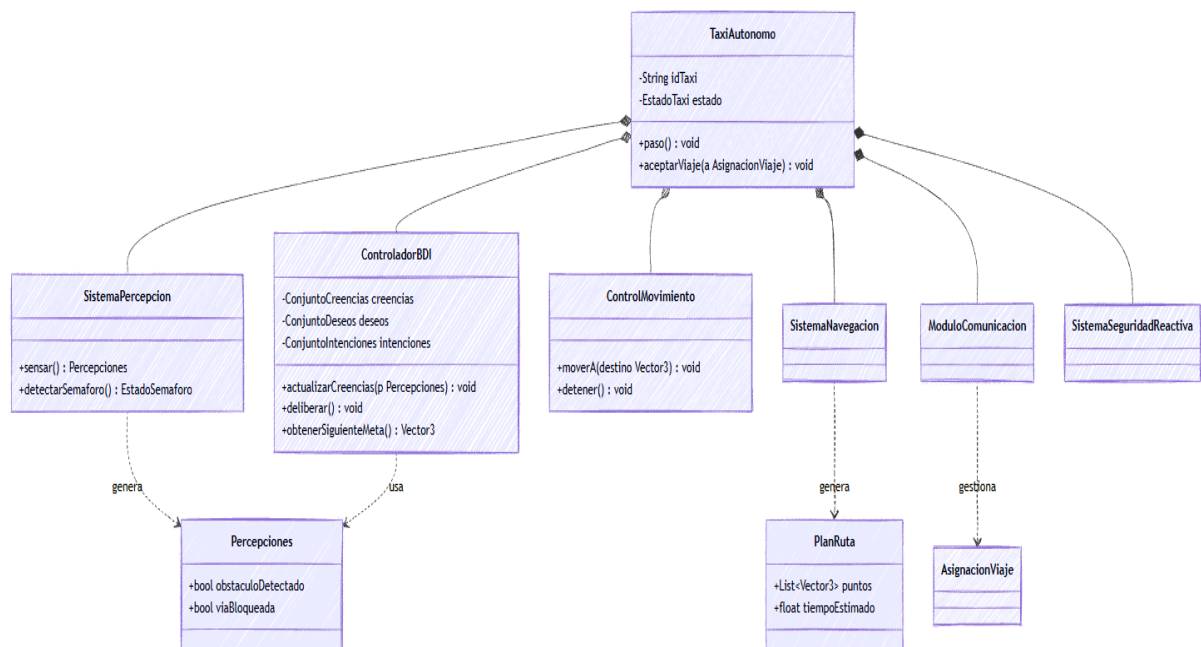
Pasajero:

<<Agent>> Pasajero
passengerId: String state: PassengerState origin: Vector3 destination: Vector3 currentRequestId: String waitTime: float maxWaitTime: float
createRequest(origin: Vector3, destination: Vector3): TripRequest sendRequest(r: TripRequest): void receiveAssignment(taxiId: String, etaPickup: float): void updateWait(dt: float): void cancelRequest(): void boardTaxi(): void confirmArrival(): void

El agente Pasajero representa la entidad que demanda el servicio de transporte dentro de la simulación. Su comportamiento es principalmente reactivo, ya que se enfoca en generar una solicitud de viaje, esperar la asignación de un taxi y responder a eventos del sistema como la llegada del vehículo, el inicio del viaje y la confirmación de arribo. Los atributos describen su estado, el origen y destino del trayecto, así como variables de control de espera para modelar condiciones como cancelación por tiempo excedido. Las operaciones reflejan sus capacidades esenciales de solicitud, seguimiento y cierre del servicio.

Diagramas de los subsistemas de los agentes

TaxiAutonomo



El presente diagrama describe la descomposición interna del agente TaxiAutonomo en subsistemas funcionales que permiten implementar su arquitectura híbrida. El agente se compone de módulos especializados que separan las responsabilidades de percepción, deliberación, navegación, control de movimiento, comunicación y seguridad reactiva, lo que favorece una estructura modular y escalable.

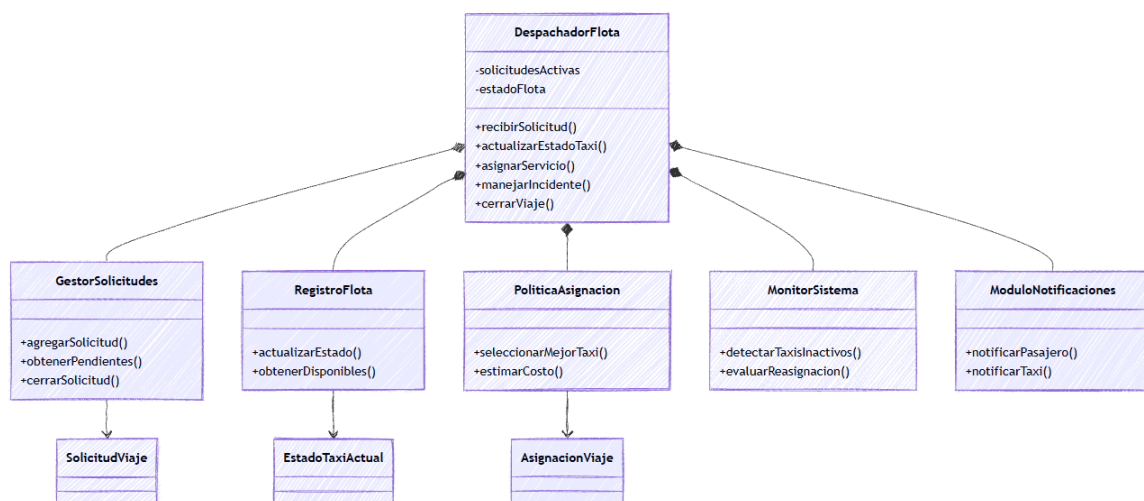
El SistemaPercepcion se encarga de obtener información del entorno y generar objetos de tipo Percepciones, los cuales contienen datos relevantes como detección de obstáculos o vías

bloqueadas. Esta información es utilizada por el ControladorBDI para actualizar sus creencias, deliberar sobre el siguiente objetivo y determinar la meta inmediata del agente. Paralelamente, el SistemaSeguridadReactiva permite responder de forma inmediata ante situaciones críticas, priorizando la seguridad sobre cualquier otro proceso deliberativo.

El SistemaNavegacion genera instancias de PlanRuta a partir de los puntos de origen y destino, permitiendo estructurar el desplazamiento del vehículo. El ControlMovimiento ejecuta físicamente el desplazamiento dentro del entorno virtual. Finalmente, el ModuloComunicacion gestiona la interacción con el sistema global, enviando el estado del taxi y recibiendo asignaciones de viaje representadas por AsignacionViaje.

Esta organización modular refleja una arquitectura híbrida coherente, donde los componentes reactivos y deliberativos coexisten y se coordinan dentro del agente, manteniendo una separación clara de responsabilidades y facilitando la mantenibilidad del sistema.

DespachadorFlota



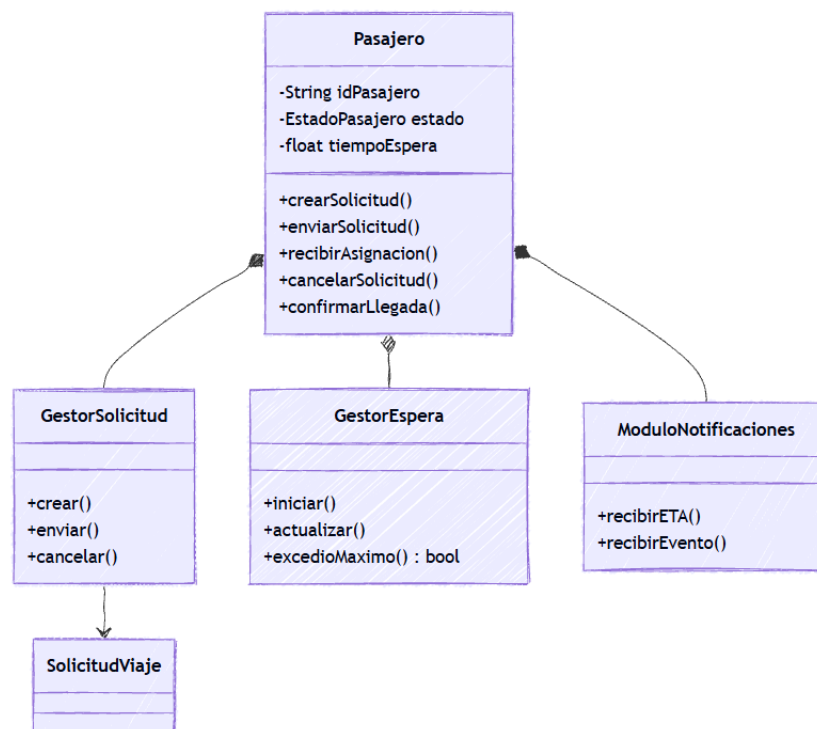
Su función principal es administrar solicitudes de viaje, mantener información actualizada de la flota de taxis y ejecutar decisiones de asignación que permitan operar el servicio de manera eficiente.

El GestorSolicitudes se encarga de registrar y controlar el ciclo de vida de las solicitudes, desde su recepción hasta su cierre. El RegistroFlota mantiene el estado actualizado de cada taxi a partir de reportes periódicos, permitiendo identificar vehículos disponibles y sus

condiciones operativas. La *PoliticaAsignacion* encapsula la lógica de decisión para seleccionar el taxi más conveniente para una solicitud, utilizando criterios como cercanía o tiempo estimado de llegada, y generando la *AsignacionViaje* correspondiente.

Adicionalmente, el *MonitorSistema* permite supervisar la operación para detectar taxis inactivos o incidentes que requieran reevaluación del servicio, mientras que el *ModuloNotificaciones* concentra la comunicación de eventos relevantes hacia taxis y pasajeros. En conjunto, esta descomposición modular separa responsabilidades críticas del despachador, facilitando su implementación, mantenibilidad y adaptación a cambios en la estrategia de asignación.

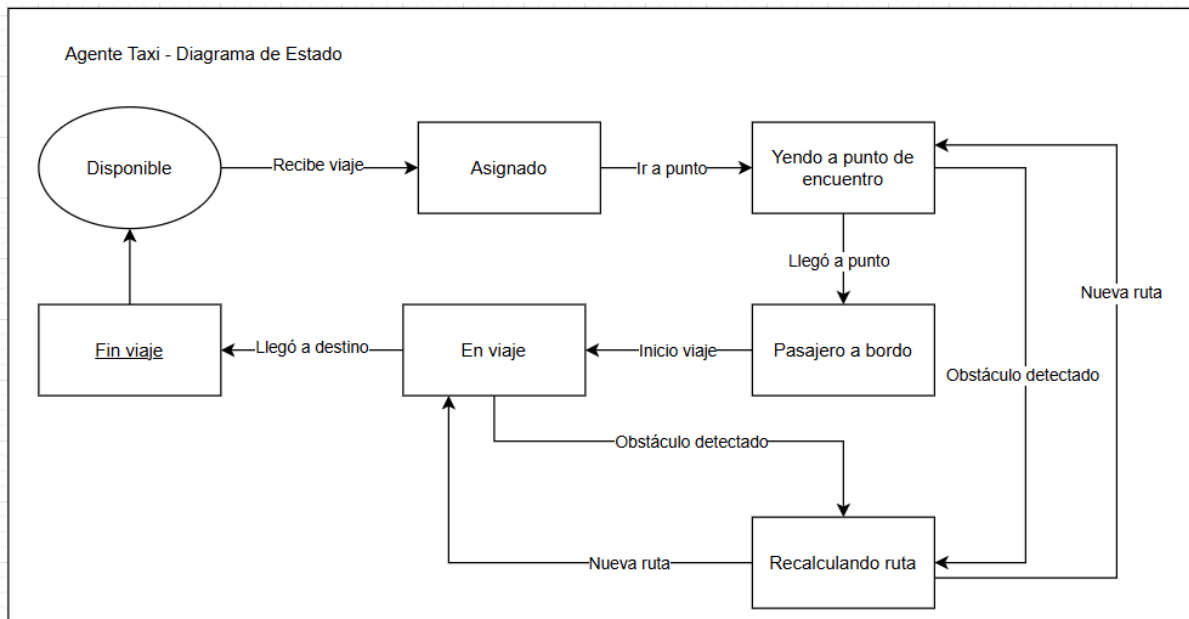
Pasajero



La funcionalidad se divide en un *GestorSolicitud* para crear, enviar o cancelar solicitudes de viaje, un *GestorEspera* para controlar el tiempo de espera y determinar si se excede un umbral máximo, y un *ModuloNotificaciones* para recibir eventos relevantes del sistema, como asignación de taxi o estimaciones de llegada. Esta separación modular permite mantener el diseño del pasajero simple y coherente, facilitando su implementación y pruebas dentro del sistema multiagente.

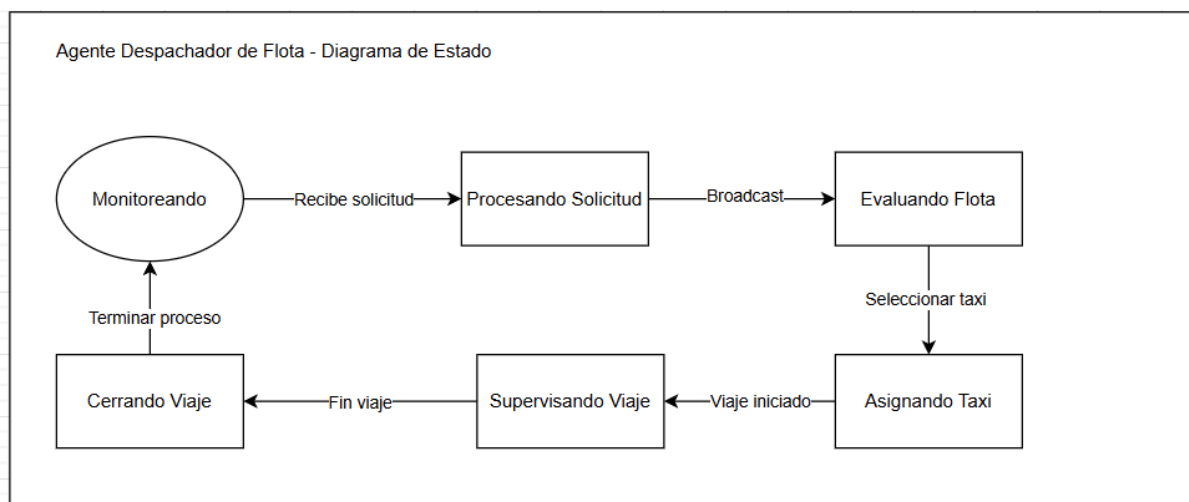
Diagramas de estado

Taxi Autónomo



Este diagrama modela el ciclo de vida del agente TaxiAutonomo a través de sus estados operativos principales. El taxi comienza en estado Disponible y, al recibir una asignación, transita hacia Asignado y posteriormente hacia Yendo a punto de encuentro. Una vez que el pasajero aborda, el agente entra en estado En viaje. Si durante la navegación se detecta un obstáculo, el taxi transita a Recalculando ruta y retoma el flujo al resolverlo. El viaje concluye al llegar al destino, regresando al estado Disponible.

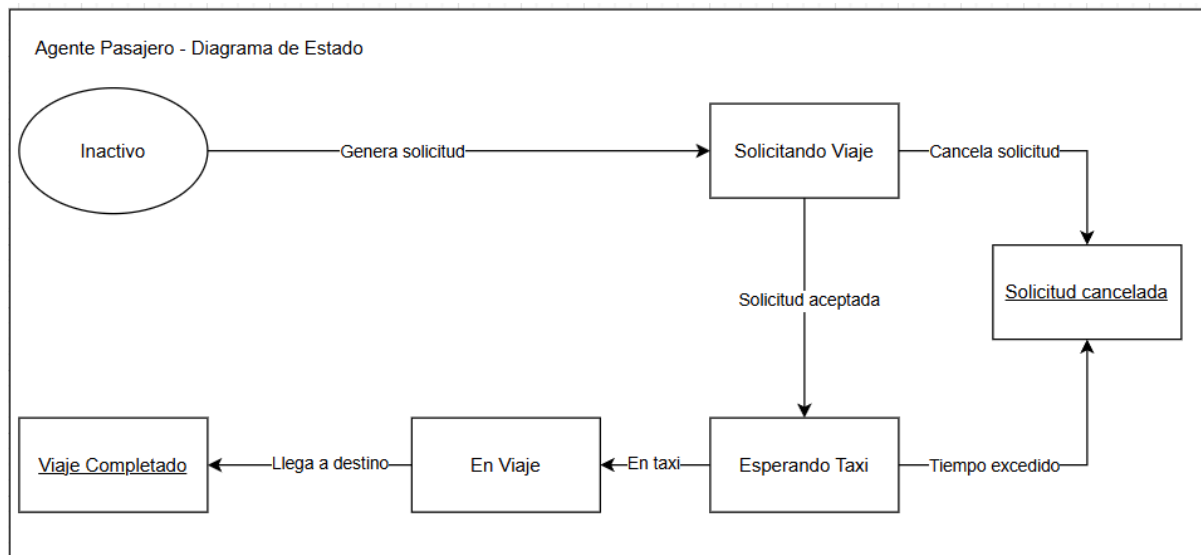
Despachador de Flota



Este diagrama representa el comportamiento del agente DespachadorFlota como un ciclo continuo de monitoreo y gestión. Parte del estado Monitoreando y, al recibir una solicitud,

avanza hacia Procesando Solicitud. Mediante un broadcast consulta el estado de la flota en Evaluando Flota, selecciona el mejor taxi y transita a Asignando Taxi. Una vez iniciado el viaje, supervisa su progreso en Supervisando Viaje hasta que concluye, cerrándolo y regresando al estado inicial de monitoreo.

Pasajero

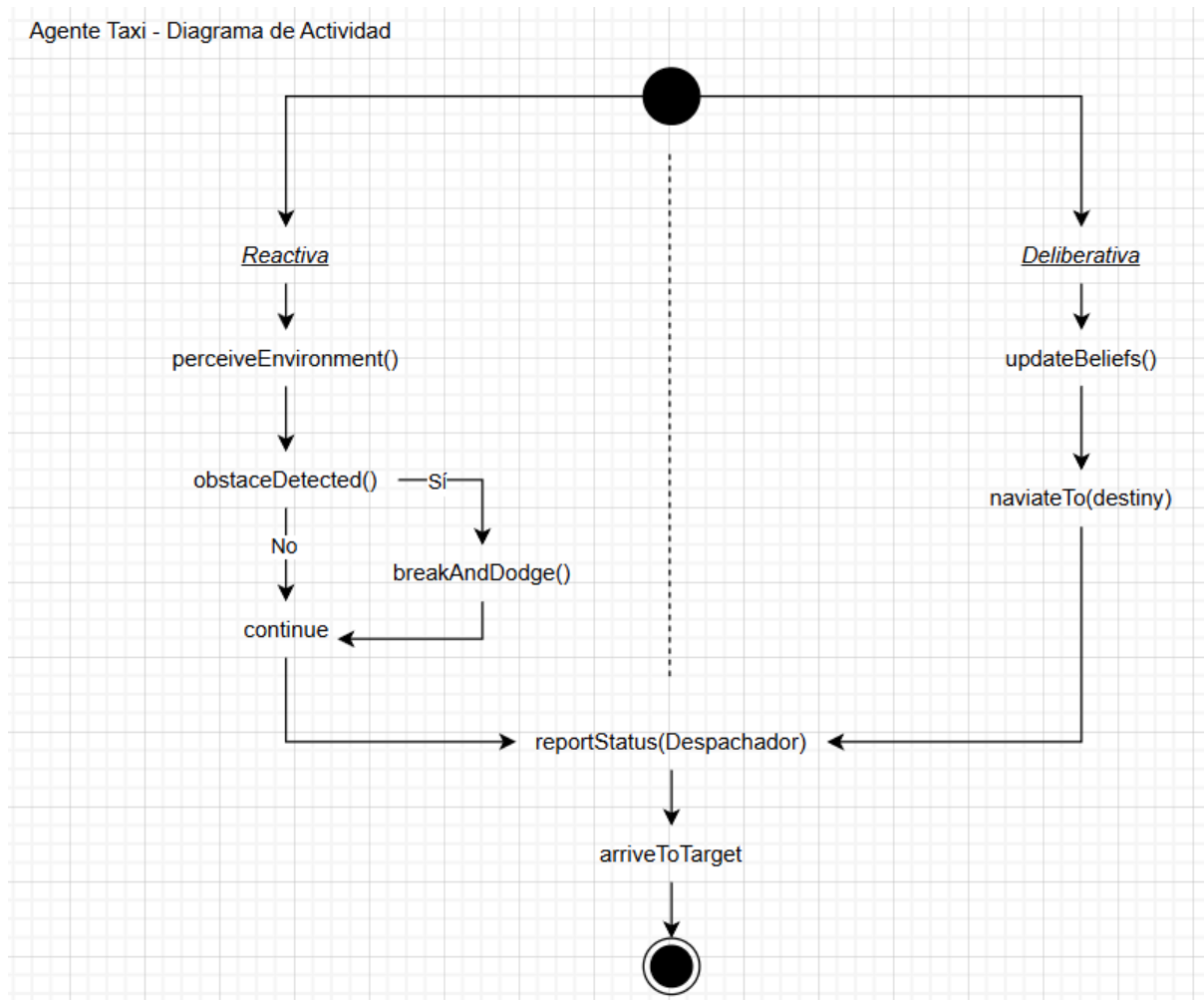


Este diagrama describe los estados por los que atraviesa el agente Pasajero desde que genera una solicitud hasta que completa su viaje. Parte del estado Inactivo, genera una solicitud y entra en Solicitando Viaje. Si la solicitud es aceptada, pasa a Esperando Taxi; si el tiempo de espera se excede, transita a Solicitud Cancelada. Al llegar el taxi, el pasajero entra en estado En Viaje, finalizando en Viaje Completado al arribar al destino.

Diagramas de actividad

Taxi Autónomo

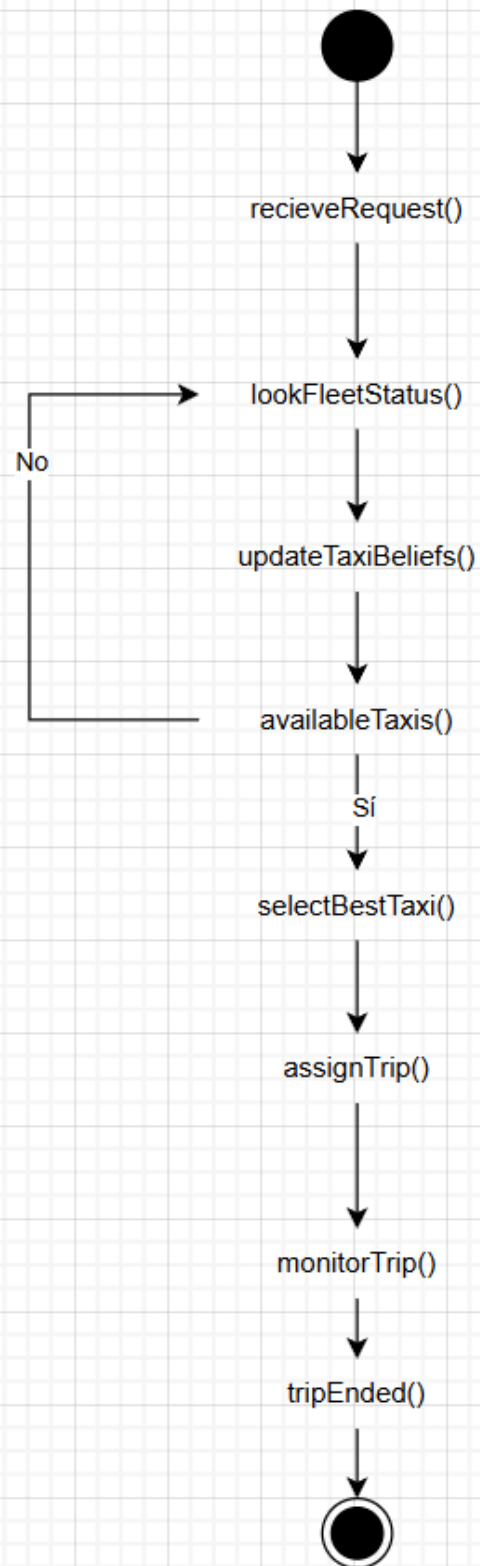
Agente Taxi - Diagrama de Actividad



Este diagrama ilustra el flujo de ejecución del agente TaxiAutonomo mediante dos carriles paralelos que representan su arquitectura híbrida. El carril reactivo ejecuta la percepción del entorno y responde inmediatamente ante obstáculos con `breakAndDodge()`. Simultáneamente, el carril deliberativo actualiza las creencias del agente y ejecuta la navegación hacia el destino. Ambos flujos convergen al reportar el estado al Despachador, finalizando cuando el taxi llega al objetivo.

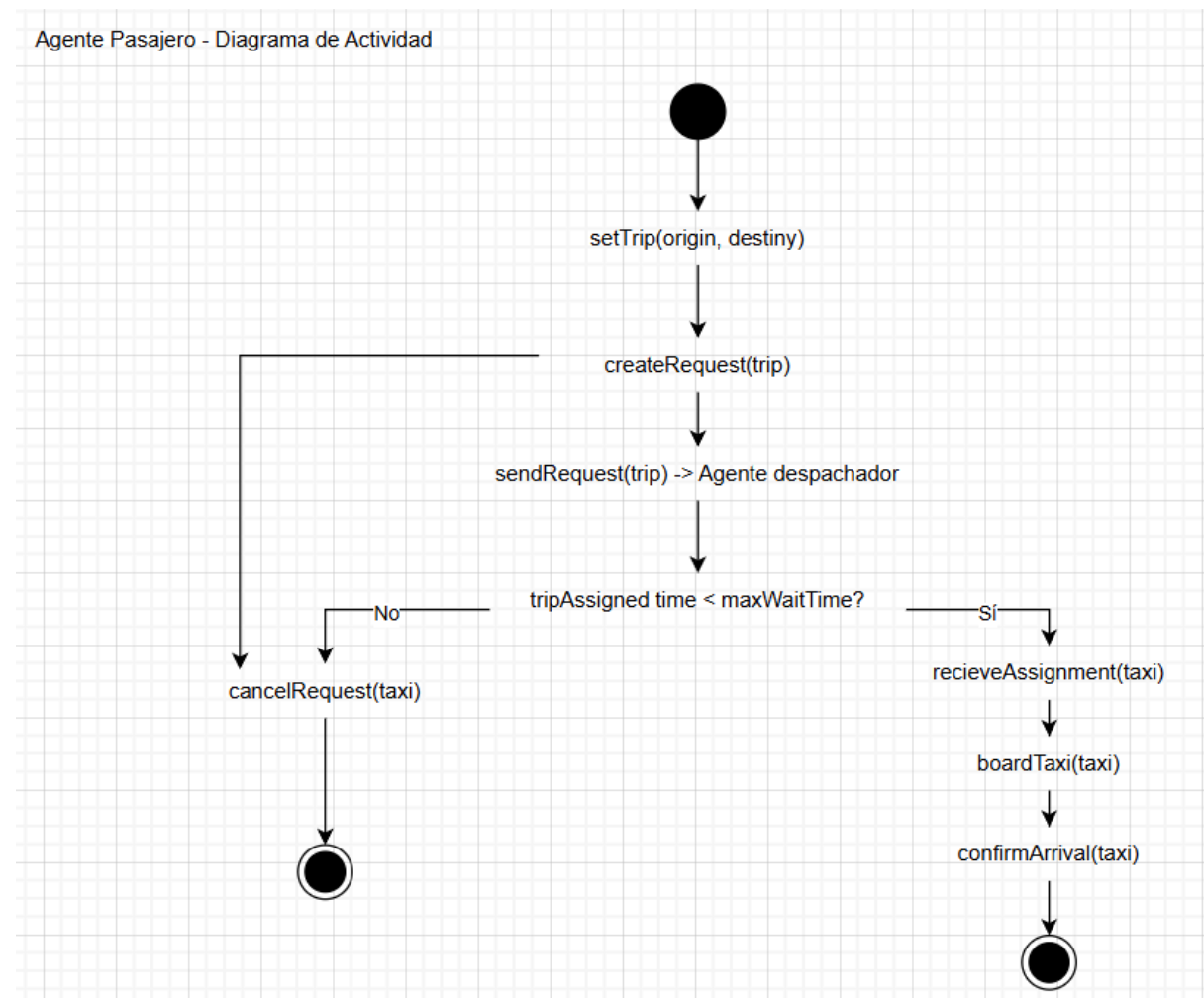
Despachador de Flota

Agente Despachador - Diagrama de Actividad



Este diagrama describe el proceso de asignación que ejecuta el agente DespachadorFlota ante cada solicitud recibida. Tras recibir la solicitud con `receiveRequest()`, consulta y actualiza el estado de la flota. Si no hay taxis disponibles, el proceso espera y vuelve a consultar. Una vez identificado un taxi, ejecuta `selectBestTaxi()` y `assignTrip()`, procediendo a monitorear el viaje activo hasta que `tripEnded()` señala su conclusión.

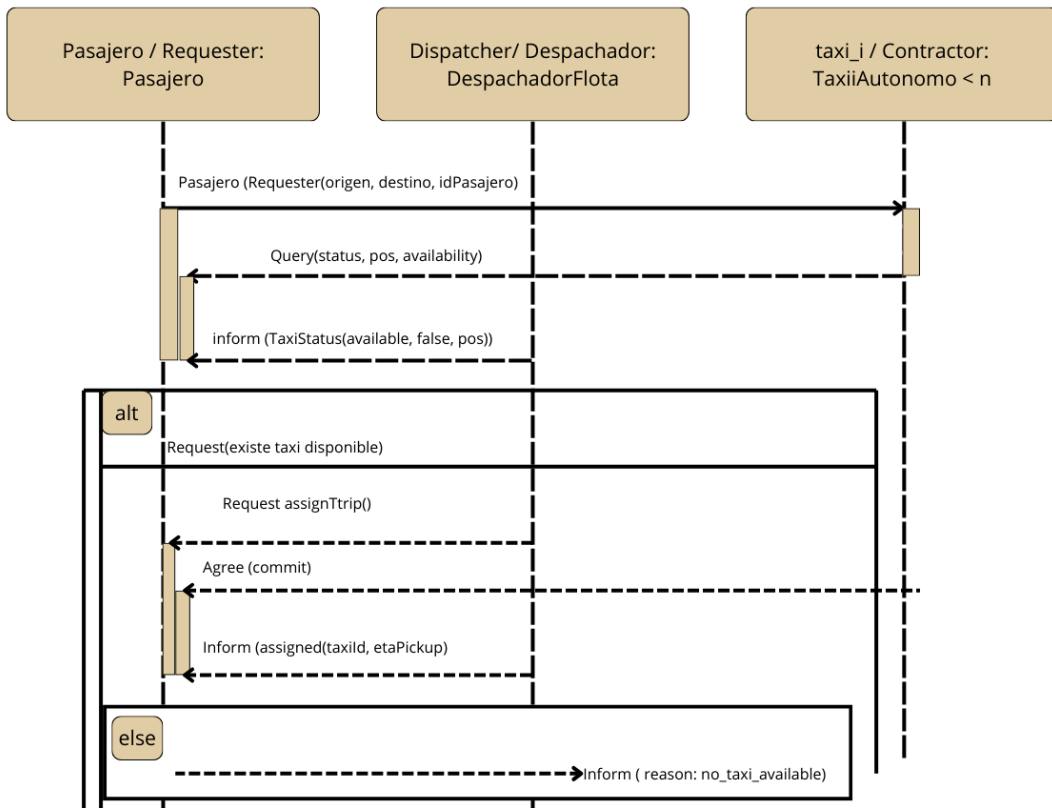
Pasajero



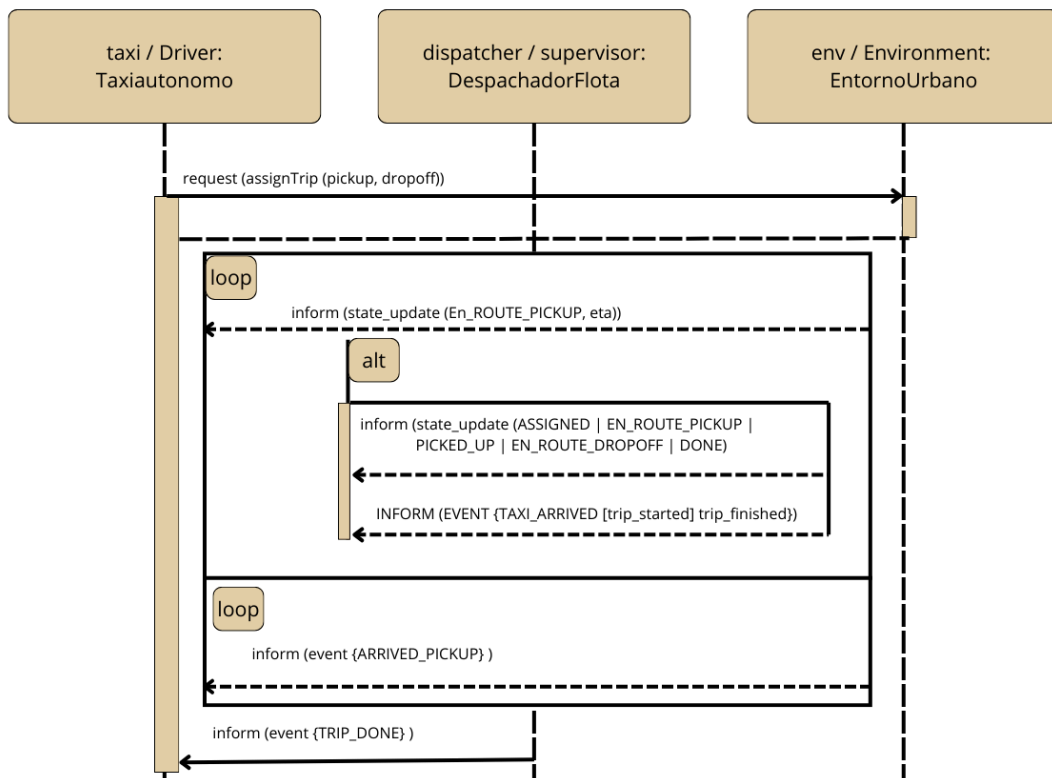
Este diagrama modela el flujo de acciones del agente Pasajero desde la creación de su solicitud hasta la confirmación de arribo. Comienza definiendo origen y destino con `setTrip()`, crea y envía la solicitud al Despachador. Luego evalúa si la asignación llega dentro del tiempo máximo permitido: si no, ejecuta `cancelRequest()` y termina; si sí, recibe la asignación, aborda el taxi con `boardTaxi()` y confirma la llegada al destino con `confirmArrival()`.

Diagramas AUML

AUML Protocolo: Asignación de viaje (Despacho)



AUML Protocolo Ejecución segura del viaje (taxi híbrido)



Arquitectura gráfica

La propuesta gráfica se basaría, principalmente, en principios de estética visual, portabilidad, rendimiento y esencialismo. Bajo este enfoque se privilegia el realismo lógico de la simulación por sobre el realismo gráfico, permitiéndonos distribuir la mayor cantidad de recursos computacionales hacia el modelado del comportamiento de los agentes sin comprometer la calidad gráfica. Visualizamos así, un mundo virtual donde inequívocamente quede plasmada como pregunta guiadora “¿Qué información visual es estrictamente necesaria para interpretar el estado y la dinámica de los agentes?”.

Los modelos de los agentes se visualizarán mediante geometría low poly, priorizando siluetas claras y formas reconocibles por encima del detalle superficial y su iluminación será mayormente precalculada (baked), evitando sistemas dinámicos complejos que no aportan información relevante al análisis del comportamiento. Los detalles secundarios o ornamentales, como señalética urbana, acabados de los vehículos o follajería, se resolverán mediante assets 2D y texturas planas, evitando geometría adicional innecesaria.

Los elementos virtuales clave dentro de la propuesta serán únicamente aquellos que influyan directamente en la dinámica del sistema y en la toma de decisiones de los agentes. En este sentido, la infraestructura vial constituye el componente central del entorno: calles claramente delimitadas, carriles definidos, intersecciones y cruces peatonales estructuran el espacio de movilidad y condicionan el comportamiento de los taxis autónomos.

Asimismo, se incorporarán elementos normativos como semáforos, señales básicas de tránsito y topes, ya que introducen restricciones y variaciones en la ejecución de las rutas, permitiendo observar cómo los agentes reaccionan ante reglas explícitas del entorno. Estos componentes cumplen una función reguladora, afectando directamente la planificación y el comportamiento reactivo.

Los edificios y volúmenes urbanos se representarán de forma simplificada, funcionando principalmente como delimitadores espaciales que estructuran la ciudad digital sin añadir complejidad innecesaria. De igual manera, cualquier otro objeto 3D presente en el entorno deberá justificar su existencia en términos funcionales, ya sea como elemento de regulación, restricción o referencia espacial.

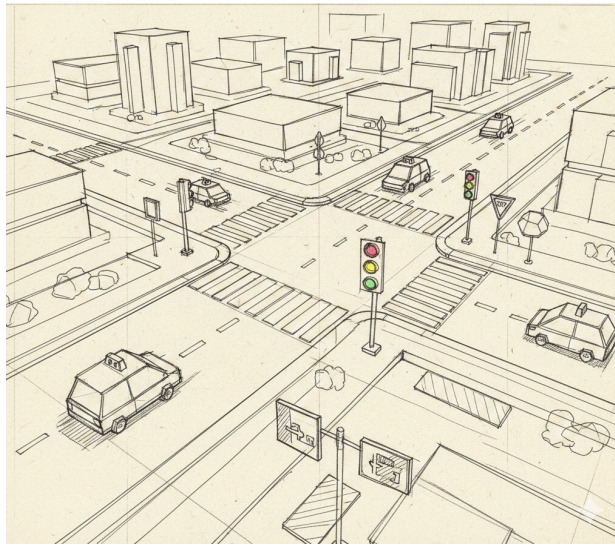


Figura 1: Sketch con Gemini de una intersección urbana low poly sin texturas.

Plan de trabajo

- Actividades de la Primera y Segunda Revisión**

Actividad	Responsable(s)	Fecha de Entrega	Esfuerzo Estimado	Estado
Definición de arquitectura y tipos de agentes	David Díaz / Óscar De la Paz	18-Feb	6 hrs	Completado
Elaboración de Diagramas de Clase y AUML	Ian García / David Díaz	20-Feb	8 hrs	Completado
Diseño de Diagramas de Estado y Actividad	Elisheba Trejo / Pablo Arreola	21-Feb	10 hrs	Completado

Definición de Propuesta Gráfica (Low Poly)	Edgar Martínez / Pablo Arreola	22-Feb	4 hrs	Completado
Documentación de fortalezas y acuerdos	Equipo completo	23-Feb	2 hrs	Completado

- **Actividades Pendientes**

Actividad	Responsable(s)	Fecha Inicio	Fecha Fin	Esfuerzo Estimado
Setup de Escenario: Importación de assets low-poly y configuración de la malla vial en Unity.	Pablo Arreola / David Díaz	24-Feb	27-Feb	10 hrs
Core de Inteligencia (BDI): Programación del ciclo de deliberación del Despachador y Taxis.	Edgar Martínez/ Óscar De la Paz	26-Feb	04-Mar	18 hrs
Capa Reactiva y Sensores: Implementación de Raycasts para frenado y detección de semáforos.	Ian García / Elisheba Trejo	01-Mar	05-Mar	12 hrs

Interfaz de Usuario (UI): Dashboard para visualizar tiempos de espera y estados de la flota.	Edgar Martínez / Pablo Arreola	05-Mar	08-Mar	8 hrs
Validación y Métricas: Pruebas de simulación para medir el impacto de la congestión.	Elisheba Trejo / Ian García	09-Mar	11-Mar	8 hrs
Entrega Final: Pulido de documentación, video de demostración y reporte de resultados.	Equipo Completo	11-Mar	12-Mar	6 hrs

Aprendizaje adquirido

- Comprendimos con mayor claridad las diferencias entre arquitecturas reactivas, deliberativas (BDI) e híbridas, y cómo seleccionar la más adecuada según el rol y nivel de complejidad de cada agente.
- Fortalecimos nuestra capacidad para traducir un problema general en una estructura formal de agentes con responsabilidades y relaciones bien definidas.
- Aprendimos más sobre nuestras fortalezas individuales como equipo, identificando quién se orienta más al diseño, quién a la implementación, quién a la organización y quién al análisis, lo que nos permitió distribuir mejor las responsabilidades.