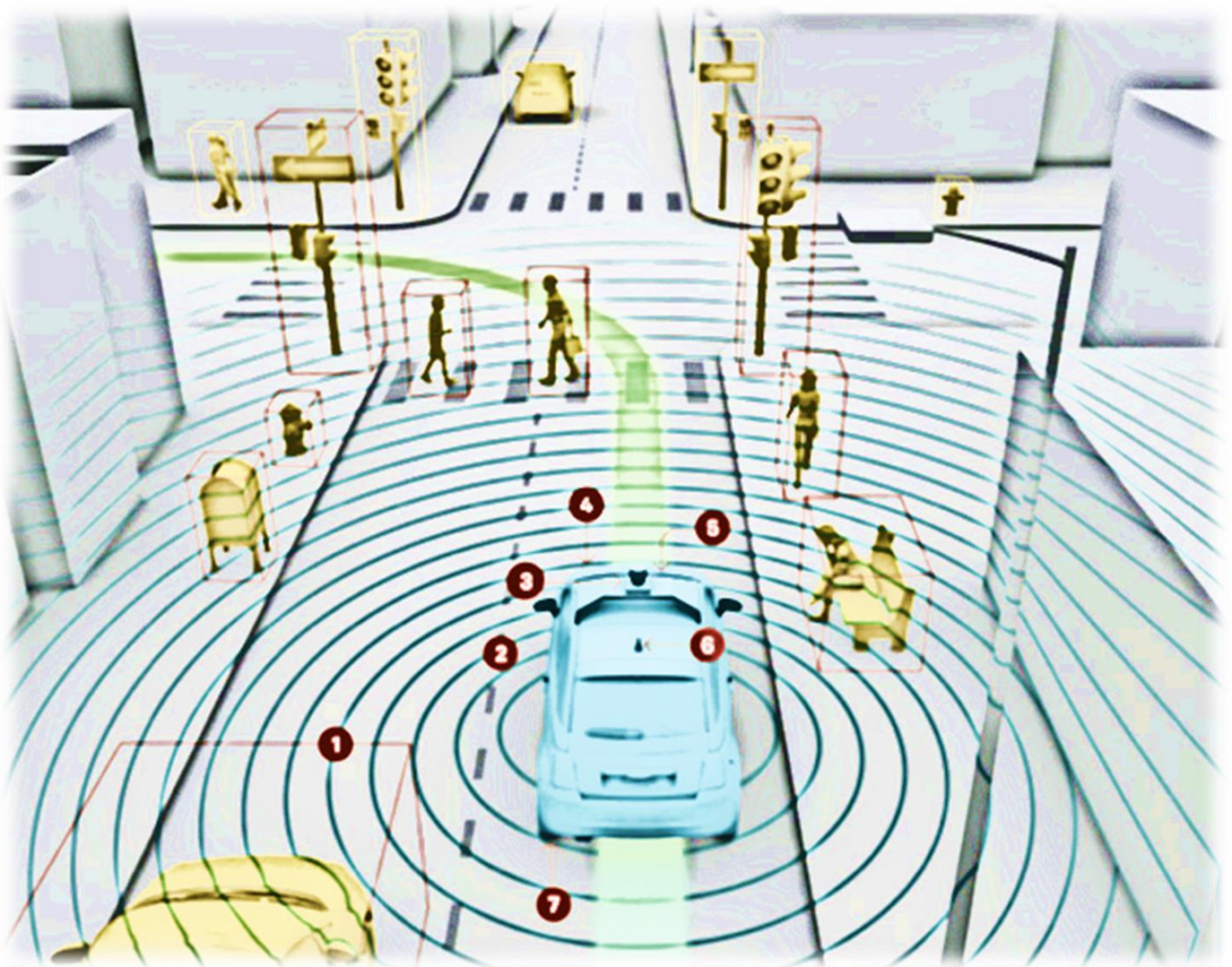


---

# PRÁCTICA 3

---

Desarrollo Basado en Agentes



AARÓN RODRÍGUEZ BUENO  
BRYAN MORENO PICAMÁN  
HUGO MALDONADO CÓZAR  
JOSE DAVID TORRES DE LAS MORENAS

## Descripción del Sistema.

Para el desarrollo de esta práctica y desde un inicio se ha planteado un sistema compuesto principalmente de un controlador y varios agentes "motrices" que hacen uso del controlador para temas como sincronización o resolución de conflictos, de esta forma se busca que la funcionalidad y movimiento se separasen del control.

Después de las primeras reuniones el sistema por el que el grupo se decidió y que posteriormente implementamos consta de varios agentes que se van a describir a continuación:

- **Agent Controller.**- Encargado de controlar y dirigir el sistema, despierta a los agentes y consulta la información necesaria, también se encarga de resolución de conflictos y elecciones críticas.
- **Agent Car.**- Se trata de cada uno de los agentes que pueden moverse por el sistema, cada uno funciona de forma distinta dependiendo de sus capacidades, las cuales no conoce hasta que el sistema se pone en marcha y las recibe del servidor.

Estos son los principales agentes del sistema y con los cuales se trabajará, existen otras consideraciones a tener en cuenta sobre ellos pero que serán explicadas más adelante.

Con esta organización se comenzó con el diseño del sistema, a partir de este y siguiendo los procedimientos típicos en el desarrollo de software se crearon los diagramas que nos ayudarían durante el desarrollo y la implementación. A continuación se pueden ver estos diagramas:

[Diagrama de Clases](#)

[Diagrama de Secuencia](#)

[Diagramas de Actividad](#)

Como se puede observar en los diagramas, además del uso de solo dos tipos de agentes, disponemos de muchas otras clases que ayudan a compartimentar y separar mejor la funcionalidad del sistema, así pues disponemos de las siguientes clases:

- **Agent.**- Clase de la que heredan todos los agentes y que tiene características básicas como envío o recepción de mensajes.
- **TypeAgent (Fly & NotFly).**- Esta conjunción de clase abstracta y subclases se hace para diferenciar que agentes pueden volar y que agentes no, esto influye en su capacidad de actuación en el mapa y también en la forma de calcular el camino hacia los objetivos, su función principal es esta última y por ello es importante separar bien que capacidades tiene.

Por ultimo aunque no menos importante tenemos implementados otros componentes del sistema, que si bien son importantes no forman parte del mismo y se mantienen fuera de los diagramas pero se comentan brevemente a continuación:

- **MyDrawPanel.**- Ventana grafica que muestra cómo se mueven los agentes en el sistema a tiempo real, muy útil para ver paso a paso los movimientos de los agentes.
- **Map&Node.**- Librería externa que nos ayuda en el cálculo de caminos hasta el goal, básicamente contiene la implementación del algoritmo A\*, es usado por las clases Fly y NotFly.

## Descripción de Heurísticas.

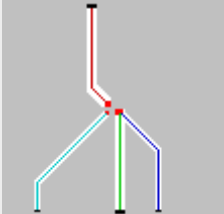
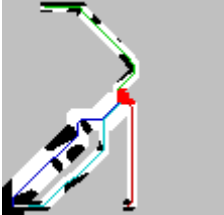
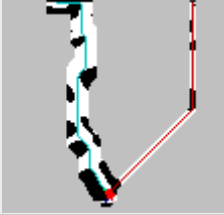

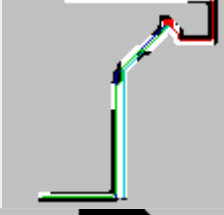
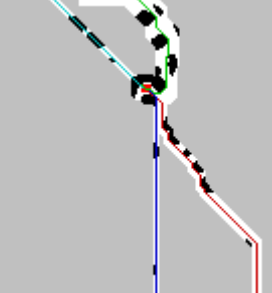
Desde el primer momento y basándonos en la experiencia de la práctica anterior, decidimos que el sistema debería ser lo más óptimo posible es decir debía conseguir tener los máximos agentes posibles en el objetivo en cada una de las iteraciones, para esto se pensaron una serie de características y funcionalidades que debíamos implementar:


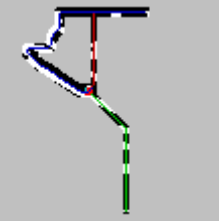
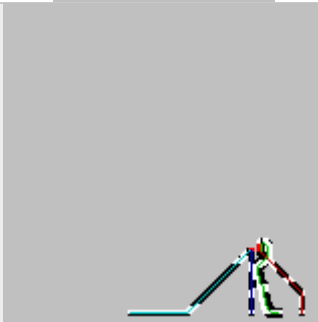
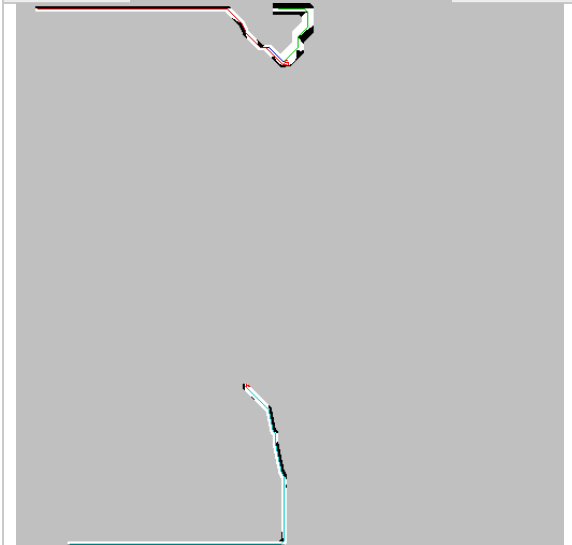
- **Mapeado y memoria.**- Nuestro sistema debía poder recorrer el mapa completo y almacenar la información entre iteraciones, de forma que se pudieran aprovechar para encontrar las soluciones óptimas para los problemas. Aprovechando las capacidades de un agente en concreto (volador que atraviesa paredes), decidimos realizar mapeos completos del entorno y almacenarlo de forma que pudiera ser utilizado a posteriori.
- **Goal Optimo.**- Tener un mapa y la posición del objetivo no era suficiente, aun necesitábamos una manera de movernos al goal en el menor número de pasos posibles, llegados a este paso en el que disponíamos de información completa sobre el entorno, capacidades y elementos del sistema, la solución parecía bastante clara un algoritmo A\* que calculara la mejor ruta posible para cada agente hacia el goal. Para eso optamos por una librería externa que ya tenía la funcionalidad implementada y la adaptamos ligeramente a nuestro problema.
- **Conservación de la energía.**- Los agentes debían consumir la energía mínima para llegar a su objetivo, como es lógico si un agente desperdicia energía puede hacer que otros agentes no lleguen a los objetivos, esto apareció de forma casi consecutiva al cálculo del camino optimo, ya que sabiendo el camino optimo y cuanto fuel consumimos, podíamos calcular fácilmente hasta que momento el agente debería ser capaz de hacer refuel. De esta forma solo se desperdicia cierta cantidad de energía en casos específicos en los que el agente se encuentra de inicio a una distancia hasta el goal que le supone un gasto inferior al fuel con el que se inicializa en el sistema.
- **Funcionamiento sin cortes.**- En última instancia y para hacer más fácil el uso de nuestra implementación personal al problema, había casos en los que una exploración completa del mapa y su uso posterior no era posible en 1 o 2 ejecuciones solo, es decir un mapeado podía conllevar más de 1 ejecución, así pues se ha dispuesto de un sistema de codificación y almacenaje de información que nos permite restaurar sesiones de exploración parciales y completarlas sin ningún tipo de interacción por parte el usuario. Con esto, el sistema calcula cuando tiene el mapa completo y reinicia las comunicaciones con el servidor las veces que sea necesario para completarlo, en caso de tener información completa también pasa de explorar a resolver el sistema sin ningún tipo de interacción, elige que agentes pueden llegar, calcula los path hasta sus objetivos (todos los goal son distintos para que los agentes no puedan chocar), así pues nuestra implementación funciona de forma independiente a los posibles mapas que se encuentre.

Estas son las características básicas de nuestro sistema, las cuales se han mantenido e implementado en el mismo y mejoran su funcionalidad y optimalidad, ofreciendo siempre en cada iteración (si disponemos del mapa completo) la cantidad máxima de vehículos en el goal en el menor número de pasos posible.

## Resultados y Valoración.

A continuación podemos ver los resultados de la ejecución de nuestro sistema en cada uno de los mapas propuestos, como apunte en estas ejecuciones los mapas ya estaban almacenados y se conocían por el sistema, los mapas testeados eran los que tenían un fuel limitado:

RESULTADOS			
			Mapa1 - CONV-c3oi6o9d Agentes: <ul style="list-style-type: none"><li>- 2 Voladores 3x3</li><li>- 2 No Voladores 5x5</li><li>- 281 Unidades de fuel usadas</li></ul>
			Mapa2 - CONV-9k38x80g Agentes: <ul style="list-style-type: none"><li>- 1 No volador 11x11</li><li>- 3 No Voladores 5x5</li><li>- 417 Unidades de fuel usadas</li></ul>
			Mapa3 - CONV-hhoigr5j Agentes: <ul style="list-style-type: none"><li>- 1 Volador 3x3</li><li>- 2 No Voladores 11x11</li><li>- 1 No Volador 5x5</li><li>- 1011 Unidades de fuel usadas</li></ul>
			Mapa4 - CONV-jo2cs32k Agentes: <ul style="list-style-type: none"><li>- 2 Voladores 3x3</li><li>- 2 No Voladores 5x5</li><li>- 331 Unidades de fuel usadas</li></ul>
			Mapa5 - CONV-640ebw81 Agentes: <ul style="list-style-type: none"><li>- 2 Voladores 3x3</li><li>- 2 No Voladores 5x5</li><li>- 691 Unidades de fuel usadas</li></ul>
			Mapa6 - CONV-hz0zd00g Agentes: <ul style="list-style-type: none"><li>- 2 Voladores 3x3</li><li>- 1 No Volador 5x5</li><li>- 1 No Volador 11x11</li><li>- 736 Unidades de fuel usadas</li></ul>

	<p>Mapa7 - CONV-n0d7ci11</p> <p>Agentes:</p> <ul style="list-style-type: none"><li>- 2 Voladores 3x3</li><li>- 2 No Voladores 5x5</li><li>- 274 Unidades de fuel usadas</li></ul>
	<p>Mapa8 - CONV-4bwdsbqa</p> <p>Agentes:</p> <ul style="list-style-type: none"><li>- 2 Voladores 3x3</li><li>- 2 No Voladores 5x5</li><li>- 365 Unidades de fuel usadas</li></ul>
	<p>Mapa9 - CONV-998t2wfr</p> <p>Agentes:</p> <ul style="list-style-type: none"><li>- 3 Voladores 3x3</li><li>- 1 No Volador 5x5</li><li>- 284 Unidades de fuel usadas</li></ul>
	<p>Mapa10 - CONV-1g15zh0p</p> <p>Agentes:</p> <ul style="list-style-type: none"><li>- 3 No Voladores 5x5</li><li>- 1 No Volador 11x11</li><li>- 1014 Unidades de fuel usadas</li></ul>

[Link a carpeta con trazas elegidas](#)