



Tecnológico de Monterrey

Juan Andrés Castellanos Huerta **A01644650**

Emiliano Duran Fuentes **A01638902**

Wendy del Carmen Martinez Macias **A01645818**

José Alfredo Vergara **A01645391**

Ivana Banderas Elliot **A01638966**

Materia

Modelación de sistemas multiagentes con gráficas computacionales

Profesor

Iván Axel Dounce Nava

Fecha de entrega:

18 de noviembre del 2025

Sistema Multiagente:

Diagrama de Clases Estándar: En este el agente principal que es el buscador sigue la ruta para encontrar enfermedades después las guarda usando GuardarTomate y su posición con PosInfección también para saber su ruta óptima toma en cuenta la posición donde esté inicia (PosInicial) y donde debe acabar (PosFinal) para así tener la mejor Ruta (Ruta), también guarda el id del agente Cortador. Después está el segundo agente que es el Agente Cortador donde se le pasa las posiciones de la infección gracias al Agente Buscador, tiene un metodo para cortar los tomates infectados (CortarTomate), este al igual que el otro tiene su posición final e inicial y la ruta óptima la obtiene gracias a la ruta que siguió el Agente Buscador y donde se encuentra los tomates infectados.

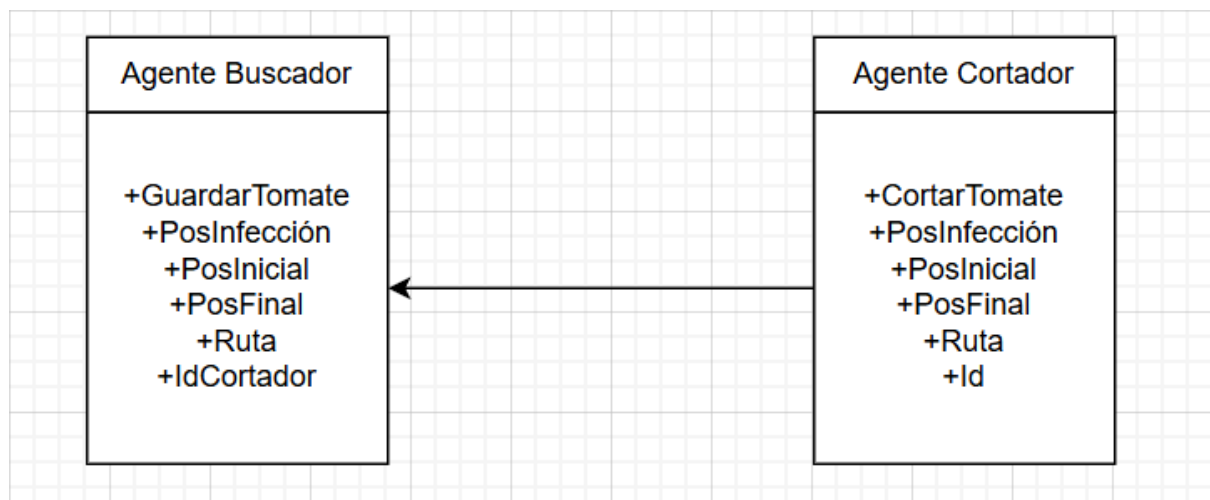


Diagrama de Clase de Agente: En este diagrama se están representando los dos agentes y cómo es que van a interactuar el uno con el otro. Lo que tendrán en común el uno con el otro serán la PosFinal, PosInicial, Ruta, y la PosInfeccion, las cuales serán utilizadas para saber dónde está qué está la infección. En el caso del agente buscador tendrá las acciones de recorrer toda la cosecha con RecorrerArea, enviara donde es que está la infección si es que encuentra una usando GurdarTomate que es como guardará la información de donde esta ubicado. Específicamente su utilidad va a ser recorrer toda la cosecha, guardar donde es que

están ubicados los tomates con plaga, y decirle al cortador donde tiene que hacer el trabajo con EnviarInfo.

Por otro lado, se tiene el agente cortador, que usará el mismo StateDescription pero con propósitos distintos. En el caso de este agente, sus acciones serán RecorrerRuta(), con lo cual solo recorre la ruta donde estén los tomates infectados, o sea, a diferencia del otro agente solo se detendrá donde se le indico. RecibirInfo() para saber donde es que están los tomates, guarda la información en la acción de GuardarToamte(), e irá a cortarlo. Su utilidad será recorrer esta ruta y cortar los tomates infectados.

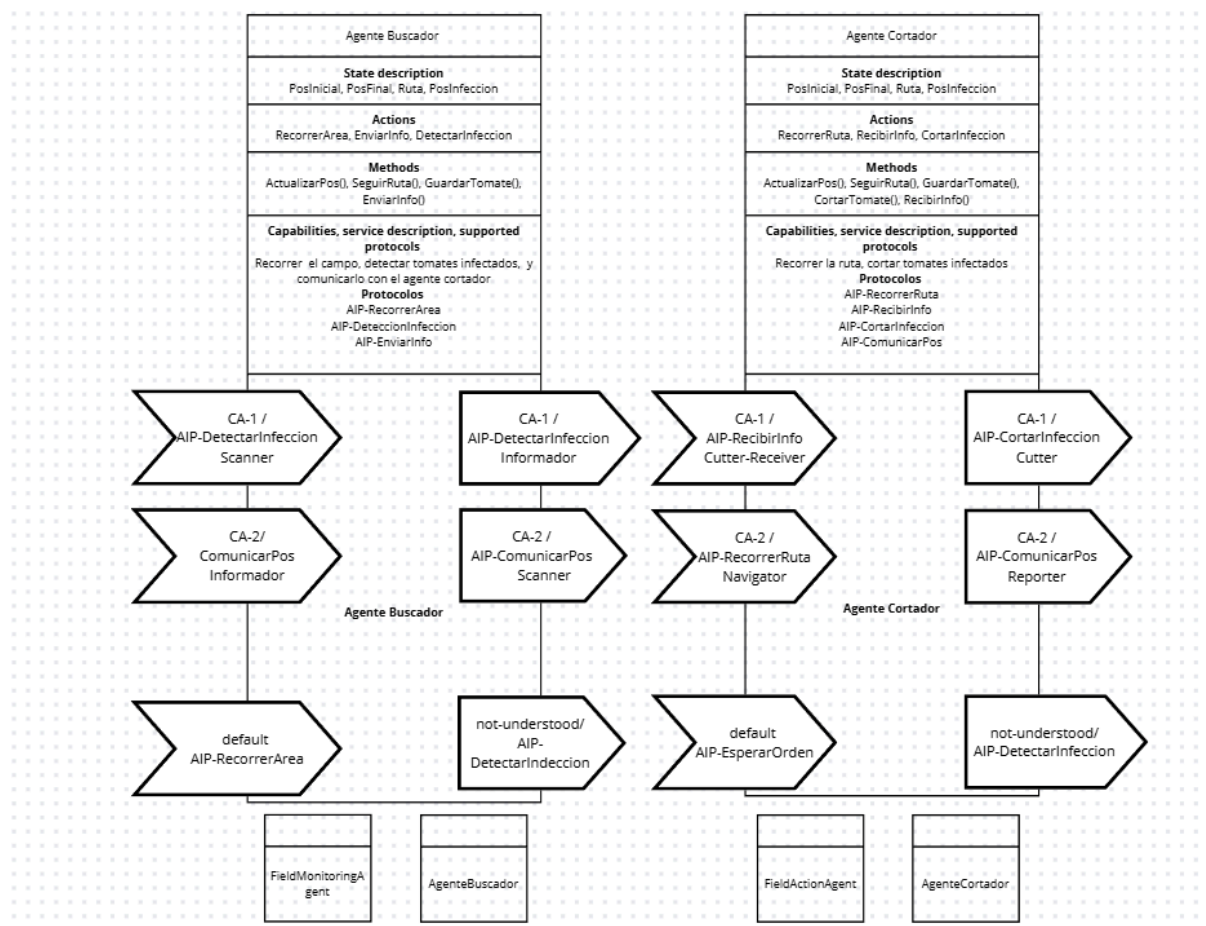


Diagrama de Interacción (AIP): En este diagrama podemos ver el flujo planeado para lo que viene siendo la única interacción entre los agentes, que es el intercambio de la ruta por

donde deberá pasar el agente cortador junto con la información de la ubicación de las plantas infectadas.

Primero el Agente Buscador comienza a recorrer su ruta y mientras hace esto va detectando las plantas que estén infectadas con la plaga hasta terminar de recorrer toda el área que está cubierta por la ruta. Una vez que este proceso termina manda la ubicación de todas las plantas infectadas al Agente Cortador y comienza a buscar en la siguiente área (si es que aun la hay). Después de eso el Agente Cortador guarda la ruta que le envió el Agente Buscador y le envía una notificación de recibido, posteriormente recorre la ruta mientras corta las plantas que recibió como infectadas hasta terminar de recorrer toda la ruta y finalmente le notifica al Agente Buscador que terminó con el cortado y está listo para recibir más información.

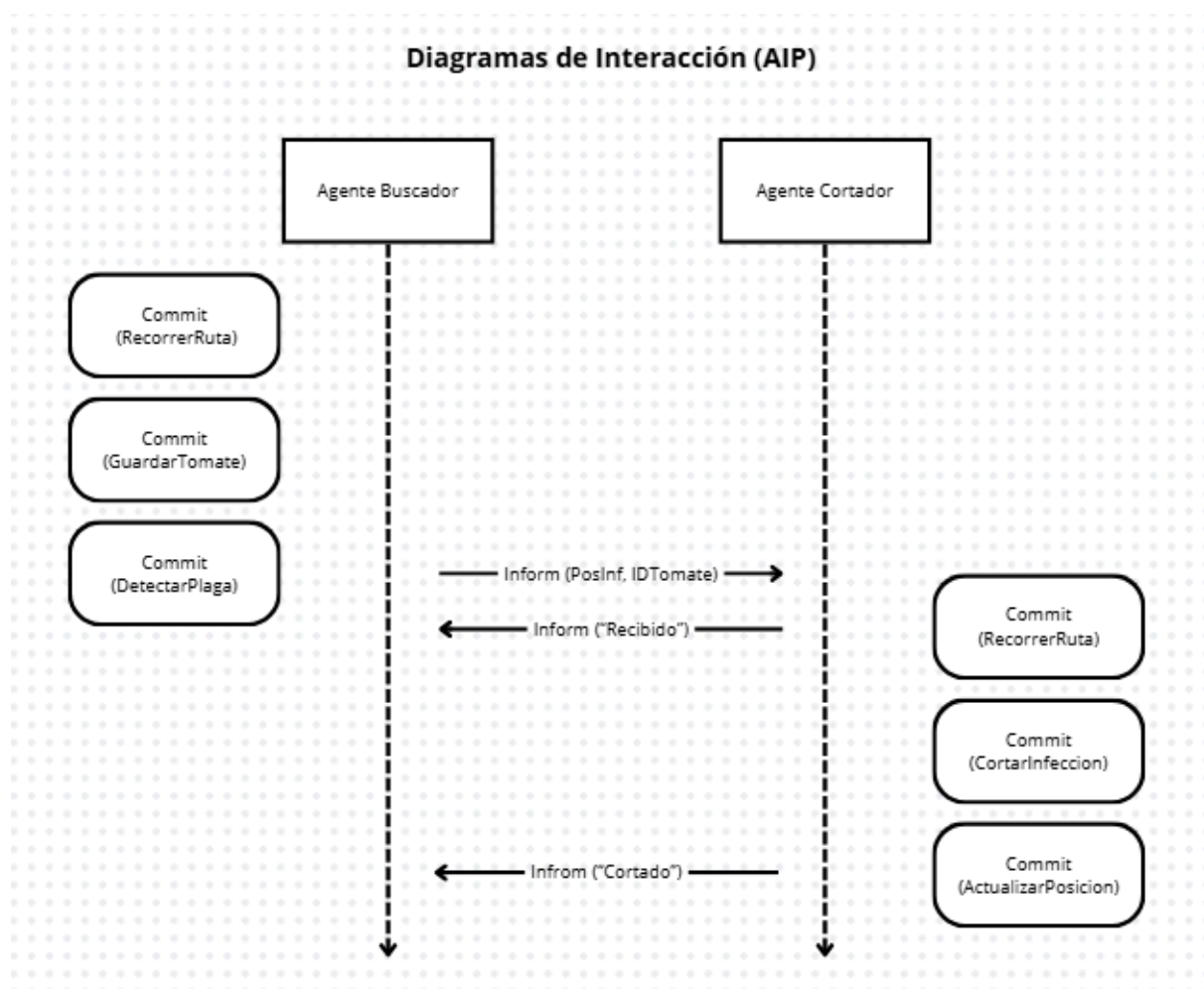
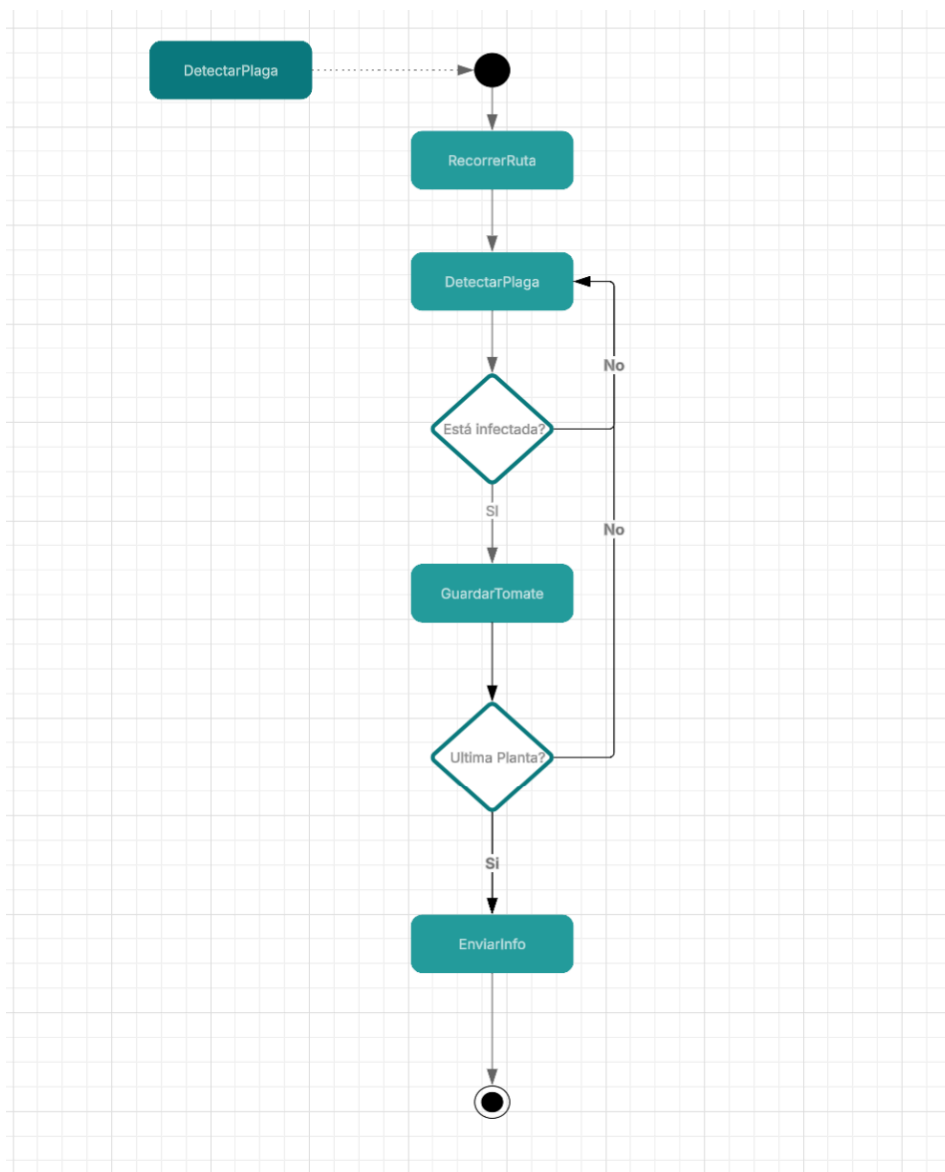


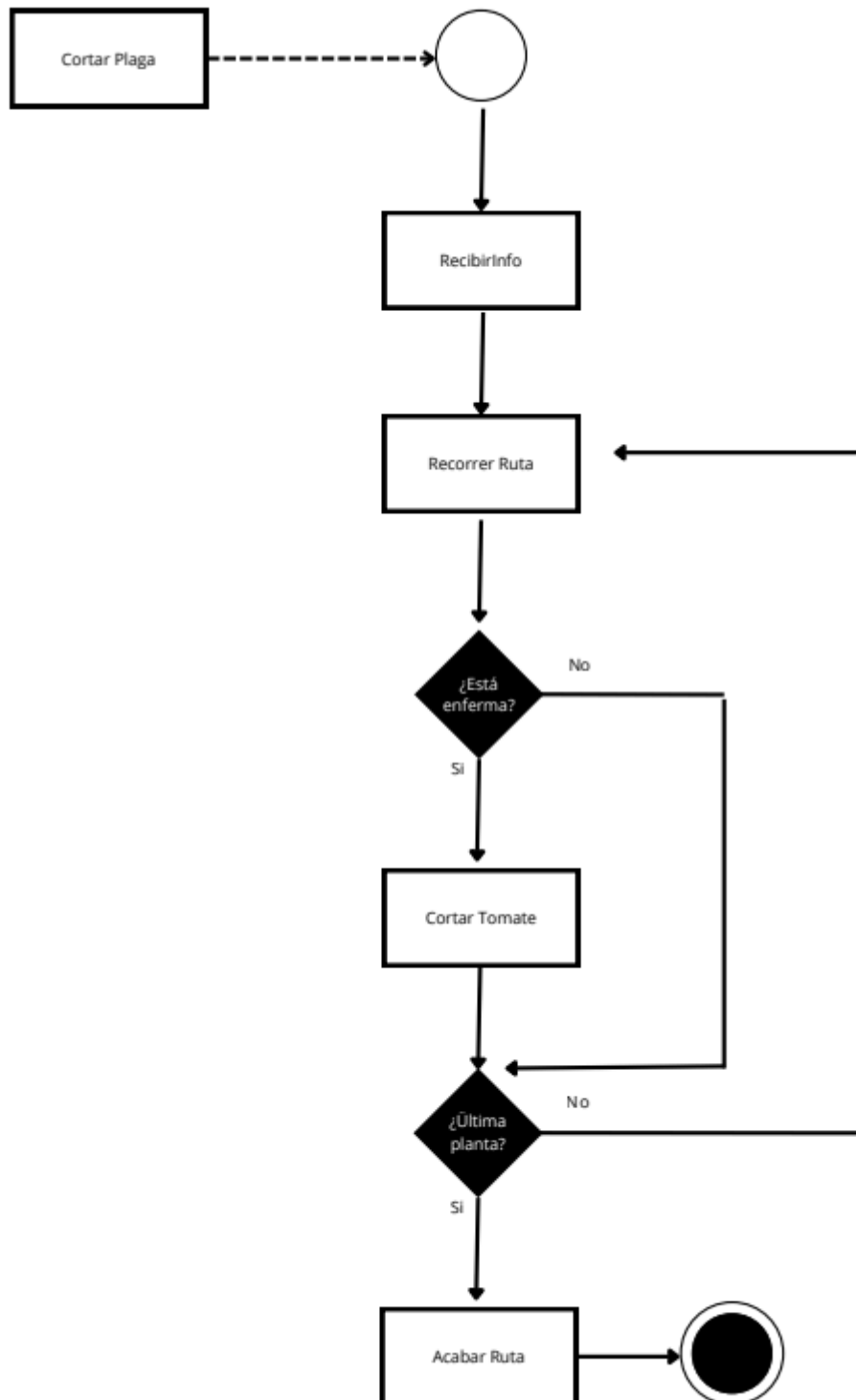
Diagrama de Actividad para el Agente Buscador:



El diagrama representa el comportamiento del Agente Buscador, cuya tarea es recorrer el campo para detectar posibles plantas infectadas. Primero, inicia su recorrido siguiendo la ruta predefinida (RecorrerRuta). En cada planta realiza la acción de detectar plaga (DetectarPlaga) y, si identifica que la planta está infectada, registra la información en (GuardarTomate). En caso de que la planta no esté infectada, el agente continúa avanzando y retoma el proceso de detección con la siguiente planta. Después de revisar cada planta, evalúa si ha llegado al final de la ruta. Si aún faltan plantas, continúa el proceso pero si ya llegó a la última planta,

entonces envía la información recopilada sobre las posiciones infectadas al Agente Cortador mediante (EnviarInfo).

Diagrama de Actividad para el Agente Cortador



En este diagrama representa las actividades del agente cortador, que se inicializará una vez que el agente buscador termine su flujo de actividades, es por eso que este modelo se inicia una vez que la información sea recibida (siendo la posición de las plantas infectadas y la indicación de iniciar con la ruta desde el agente detector hacia el cortador), después y al igual que el anterior diagrama, se sigue una ruta ya preestablecida (que forma parte de nuestras asunciones el conocer la ruta del plano para evitar obstáculos), revisa en la lista de posiciones infectadas y no infectadas si la planta que está revisando la tiene que cortar o no (es decir depende del estado de esa posición en concreto), si está infectada, la corta y revisa si fue la última planta, si lo fue se detiene y espera indicaciones nuevas para el siguiente área, caso contrario, si la planta no está infectada no la corta y solo revisa si esa fue la última planta, si no lo fue, regresa a recorrer la ruta (avanzar) y si sí lo fue, se detiene (acaba la ruta) y espera indicaciones del agente detector para el siguiente área.

Gráficos computacionales:

¿Cómo visualizan el mundo virtual?

Un invernadero con plantas de tomate tanto del lado derecho como del lado izquierdo en medio hay un camino estrecho por donde puede cabe solo una persona y en el caso del reto un solo robot a la vez, tal vez habría varias de estas filas donde se encuentran las plantas de tomate, también sería bueno implementar alguno que otro obstáculo por simular posibles cambios en la ruta.

¿Qué elementos virtuales (edificios, objetos, etc.) son clave para su propuesta?

Los elementos más importantes serían las diferentes líneas donde se encuentran las plantas de tomate (sembrados), los robots para simular cómo es la ruta y como es que estos pudieran reaccionar al momento de enfrentarse a un obstáculo en la ruta que causen que la ruta pueda cambiar. Puntos de inicio y puntos de fin según la ruta que estos tienen planeado hacer.

¿Cómo visualizan los modelos de los agentes, y de otros objetos relevantes?

Principalmente visualizamos a los agente como unos robots que cambian según su función el Agente Buscador sería un robot con dos cámaras para poder ver ambos lados para las plantas de tomate y que así pueda detectar las plantas infectadas, en el caso del agente cortador sería un robot al igual que el otro, sin embargo, en lugar de tener las cámaras de detección este tendría una herramienta para cortar las plantas.

En el caso de las plantas de tomate habría muchas en ambos lados de distintos colores para simular las plantas infectadas y que puedan ser detectadas.

Una Textura para el suelo.

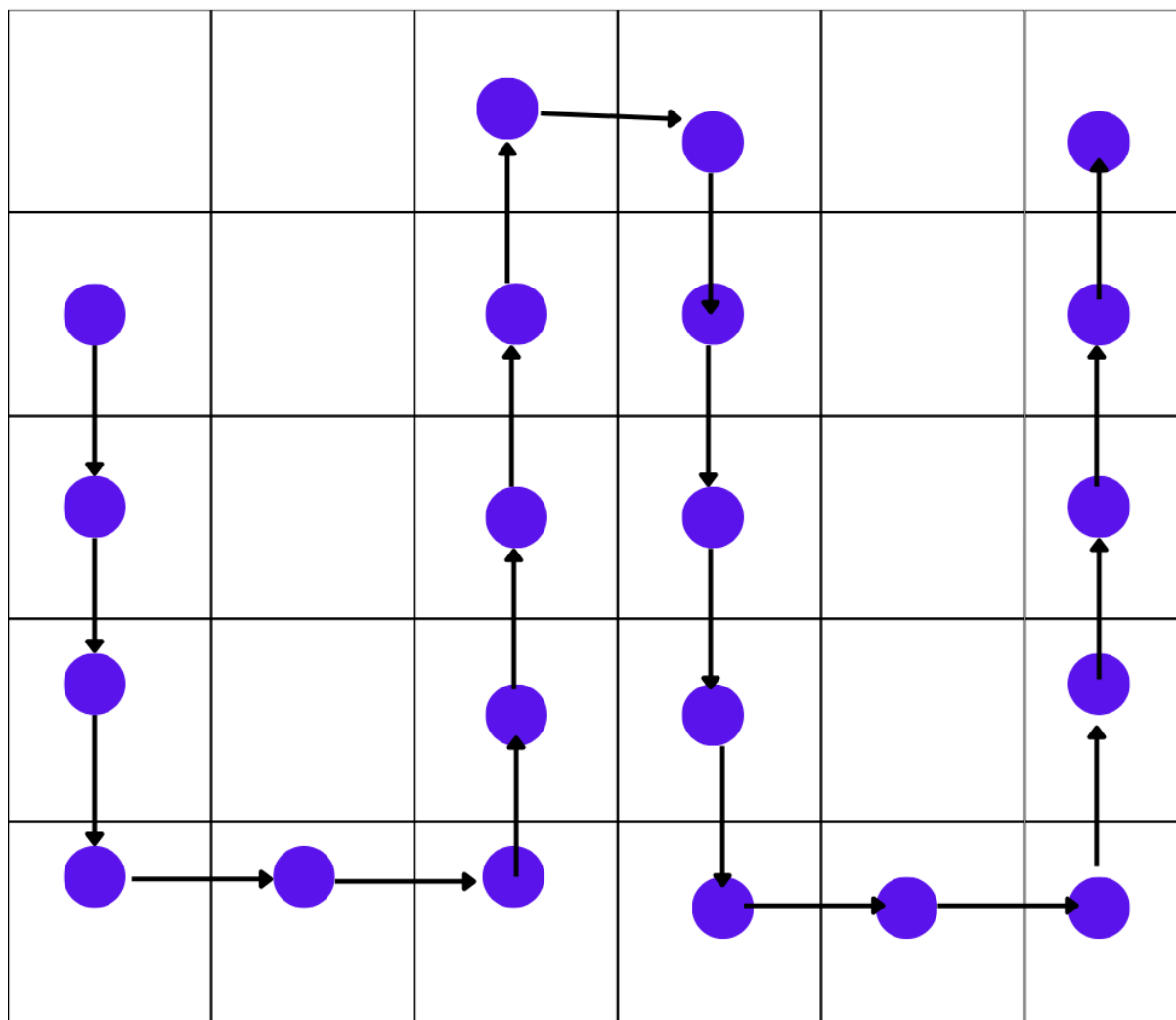
Por último algunas colisiones para simular algún tipo de obstáculo que se pueda interponer en el camino del robot y que pueda buscar nuevas rutas para lograr su tarea de la manera más rápida posible.

Agregar esquemáticos o bocetos de dibujo para apoyar estas descripciones.

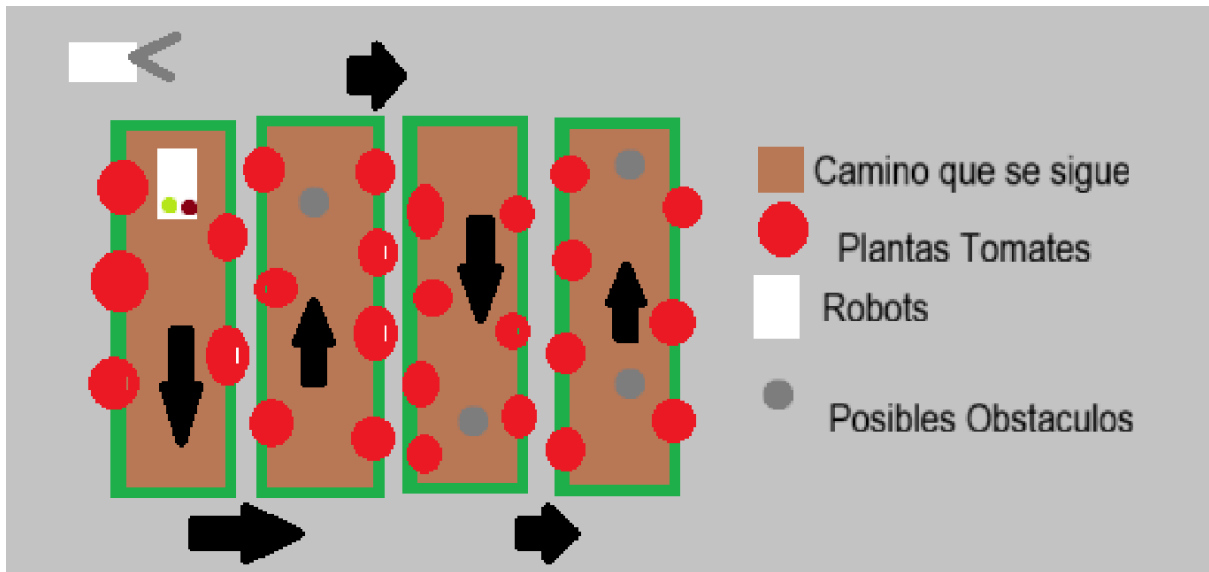
Asunciones:

Estamos asumiendo que conocemos previamente el terreno, o el plano, por lo cual nos permitirá trazar o dividir el plano en rejas (grid) y aprovechar esta propiedad como una ventaja, la forma en la que trazaremos la ruta será a través de nodos (o puntos en el espacio) y sus respectivas uniones, es decir grafos dirigidos al cual nuestros agentes ya tendrán acceso a ella y conocerán la ruta, solo hará falta seguirla y no detectarla en tiempo real como tal.

Ejemplo (el ejemplo por el momento no incluye el plano para que se vea mejor la rejilla):



Lo cual nos permite conocer el flujo del plano, esta representación se implementara dependiendo del plano, en este caso, la representación se vería así, sabido esto, y viéndolo a un ejemplo mas grafico, el plano con su respectiva planeación de rutas se vería así (sin rejilla)



Ejemplo de plano aplicado con todo y rejilla (primer paso ya explicado):

