



**Tecnológico  
de Monterrey**

## **Evidencia 2. Revisión 2**

Luis Fernando Valderrábano García | **A01644530**

José Pedro Gastélum Beltrán | **A00227608**

Mario Feng Wu | **A01644768**

Octavio Sebastián Hernández Galindo | **A01638993**

Angel Gabriel Camacho Pérez | **A01743075**

Jonathan Roman Velasco | **A01644522**

**TC2008B.302**

**Fecha de entrega:**  
18 de noviembre del 2025

# Índice

<b>1. Diagramas de Clase de Agentes.....</b>	<b>2</b>
1.1. Entidad Padre.....	2
1.2. Manager.....	4
1.3. Explorer.....	5
1.4. Recolector.....	6
1.5. Purgator.....	7
<b>2. Diagramas de Interacción (AIP).....</b>	<b>8</b>
<b>3. Diagramas de Actividad o de Estado.....</b>	<b>10</b>
¿Cómo visualizan el mundo virtual?.....	13
¿Qué elementos virtuales (edificios, objetos, etc. ) son clave para su propuesta?.....	14
¿Cómo visualizan los modelos de los agentes, y de otros objetos relevantes?.....	14
Planta.....	14
Entry.....	15
TrashZone.....	15
SafeZone.....	16
(Se cambiara de color).....	16
Manager.....	17
Explorer.....	18
Recolector.....	18
Purgator.....	19
(se cambiara de color).....	19
Entity.....	20
Agregar esquemáticos o bocetos de dibujo para apoyar estas descripciones.....	20
<b>4. Referencias.....</b>	<b>20</b>

## 1. Diagramas de Clase de Agentes

### 1.1. Entidad Padre

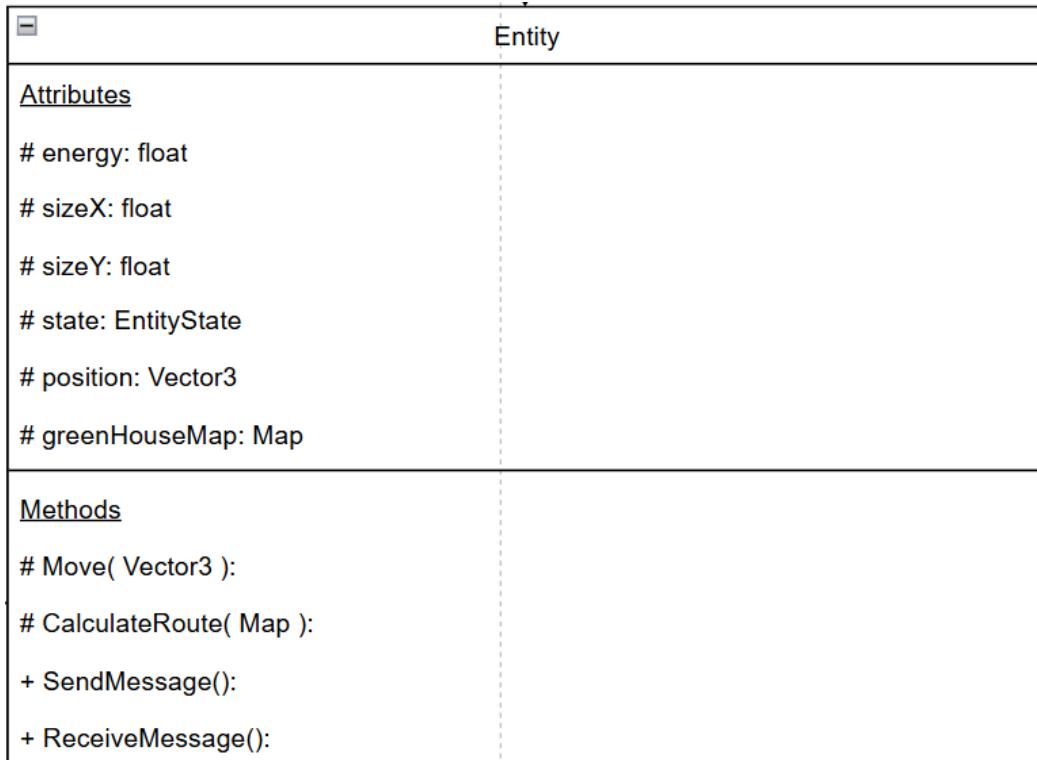


Figura 1. Entidad Padre. Creado en [Draw.io](#) <sup>[1]</sup>

En la Figura 1, diseñamos un diagrama de clase padre, cuyos atributos son heredados a todos los agentes que planteamos en nuestro diseño del reto. Estos son atributos que modelan la estructura básica de los agentes con la información requerida para desempeñarse en conjunto en la simulación del invernadero. Cada tipo de agente también utiliza todos los métodos nombrados, a excepción del agente Manager que únicamente utiliza los métodos de mensajes.

## 1.2. Manager

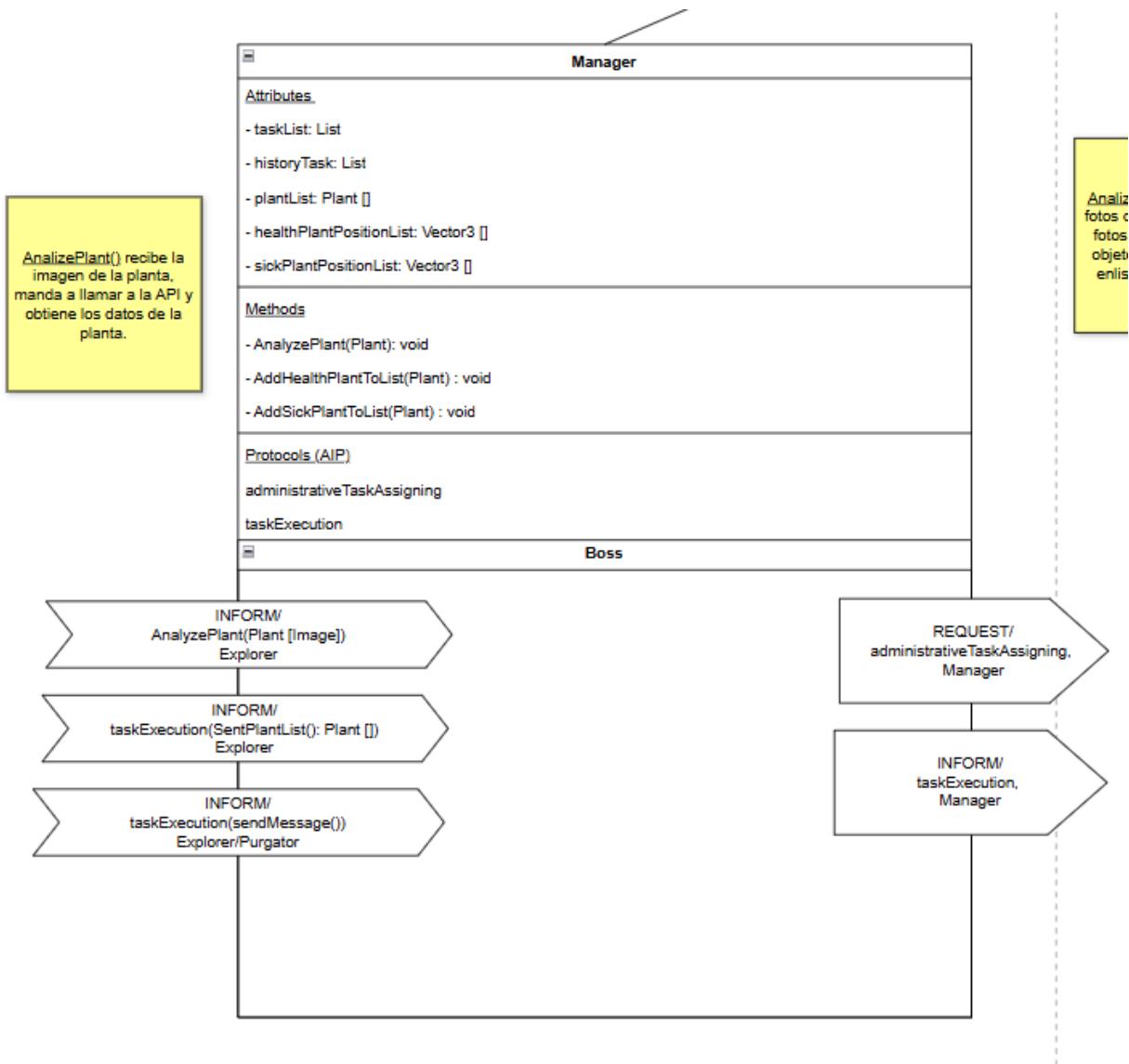


Figura 2. Agente Manager. Creado en [Draw.io](#) <sup>[1]</sup>

En la Figura 2, creamos el diagrama de clase del agente Manager, la cual tiene como principal función analizar las fotografías de las plantas para determinar en qué estados se encuentran(enferma, lista para cosechar, inmadura) y posteriormente asignar a un agente recolector o purgador a una respectiva planta.

### 1.3. Explorer

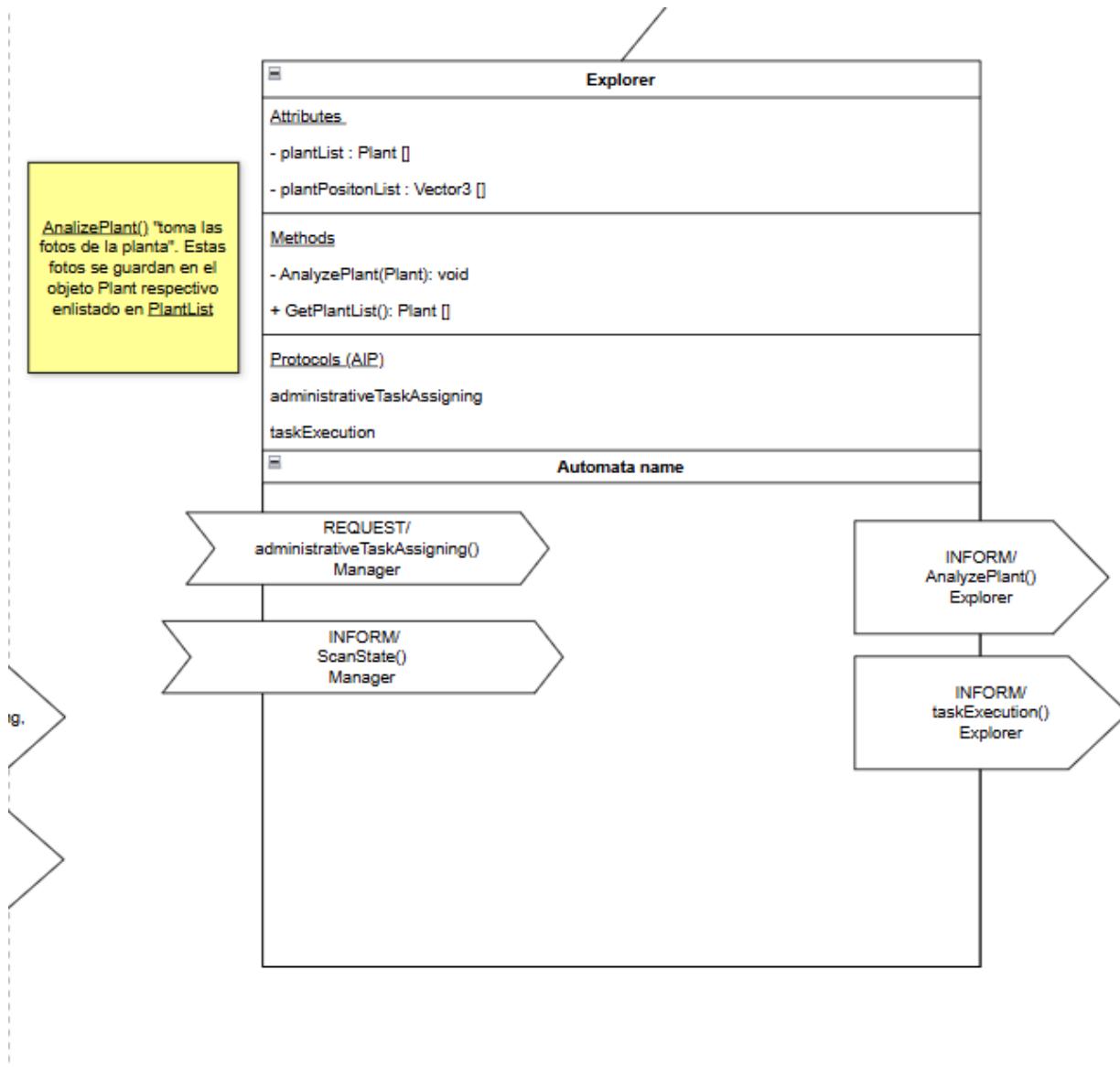


Figura 3. Agente Explorer. Creado en [Draw.io](#) <sup>[1]</sup>

En la Figura 3, creamos el diagrama de clase del agente Explorer, la cual tiene como principal función de patrullar todo el invernadero tomando fotos de las plantas y enviarlas al manager para que las analice.

#### 1.4. Recolector

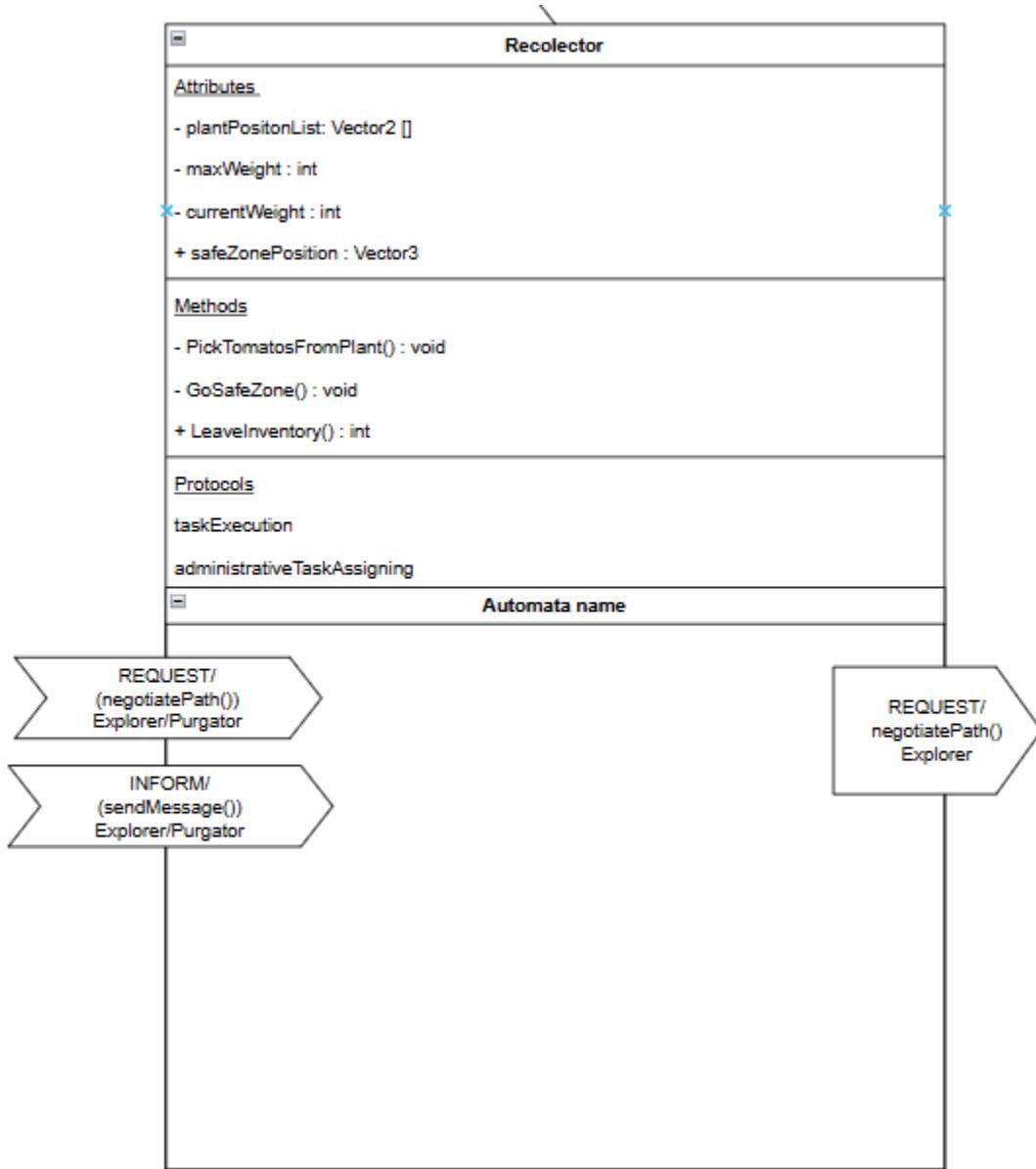


Figura 4. Agente Explorer. Creado en [Draw.io](#) <sup>[1]</sup>

En la Figura 4, creamos el diagrama de clase del agente Recolector, la cual tiene como principal función recolectar las plantas que están listas para cosechar y dejarlas en la zona asignada .

## 1.5. Purgator

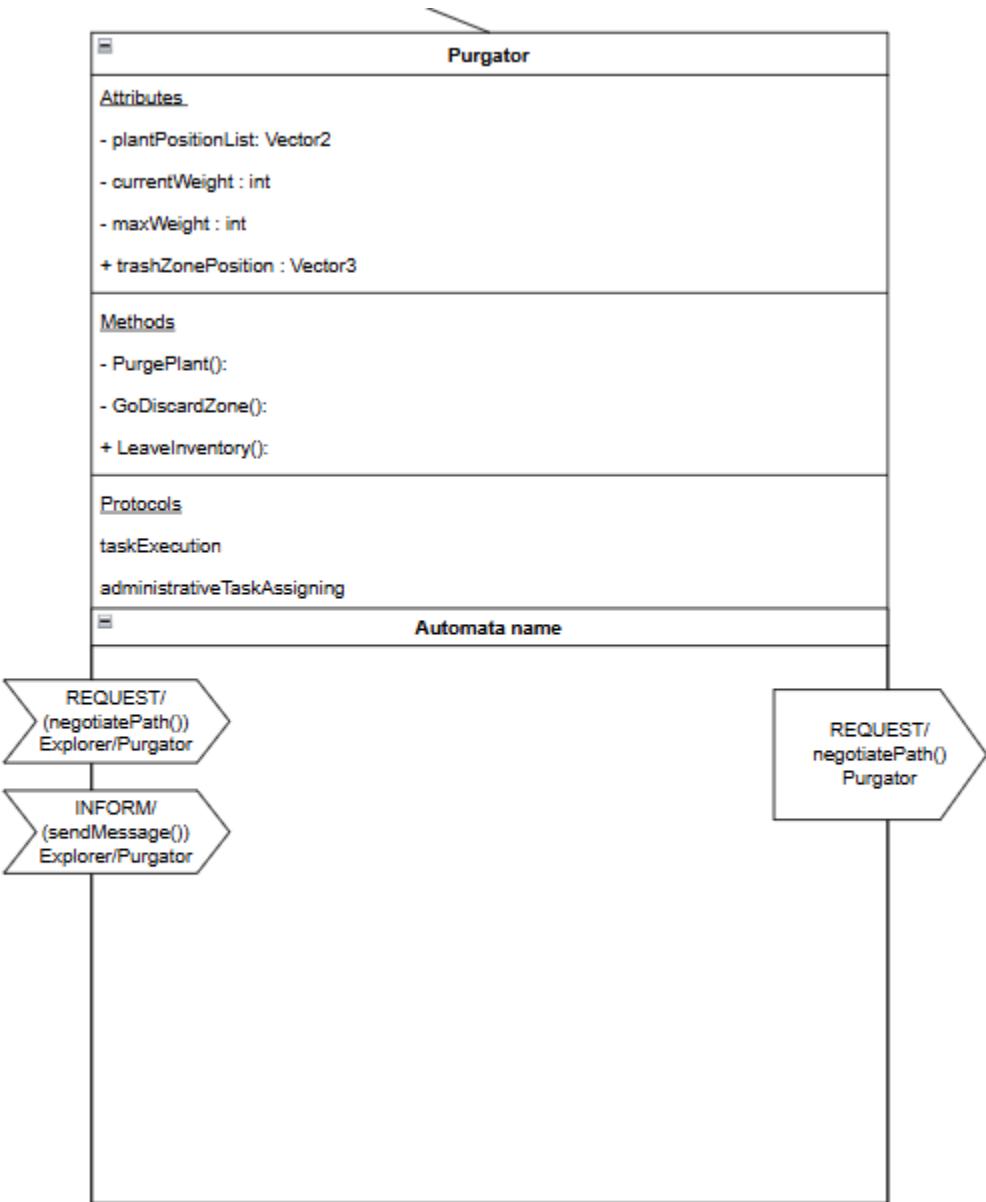


Figura 5. Agente Explorer. Creado en [Draw.io](#) <sup>11</sup>

En la Figura 5, creamos el diagrama de clase del agente Purgator, la cual tiene como principal función eliminar las plantas enfermas.

## 2. Diagramas de Interacción (AIP)

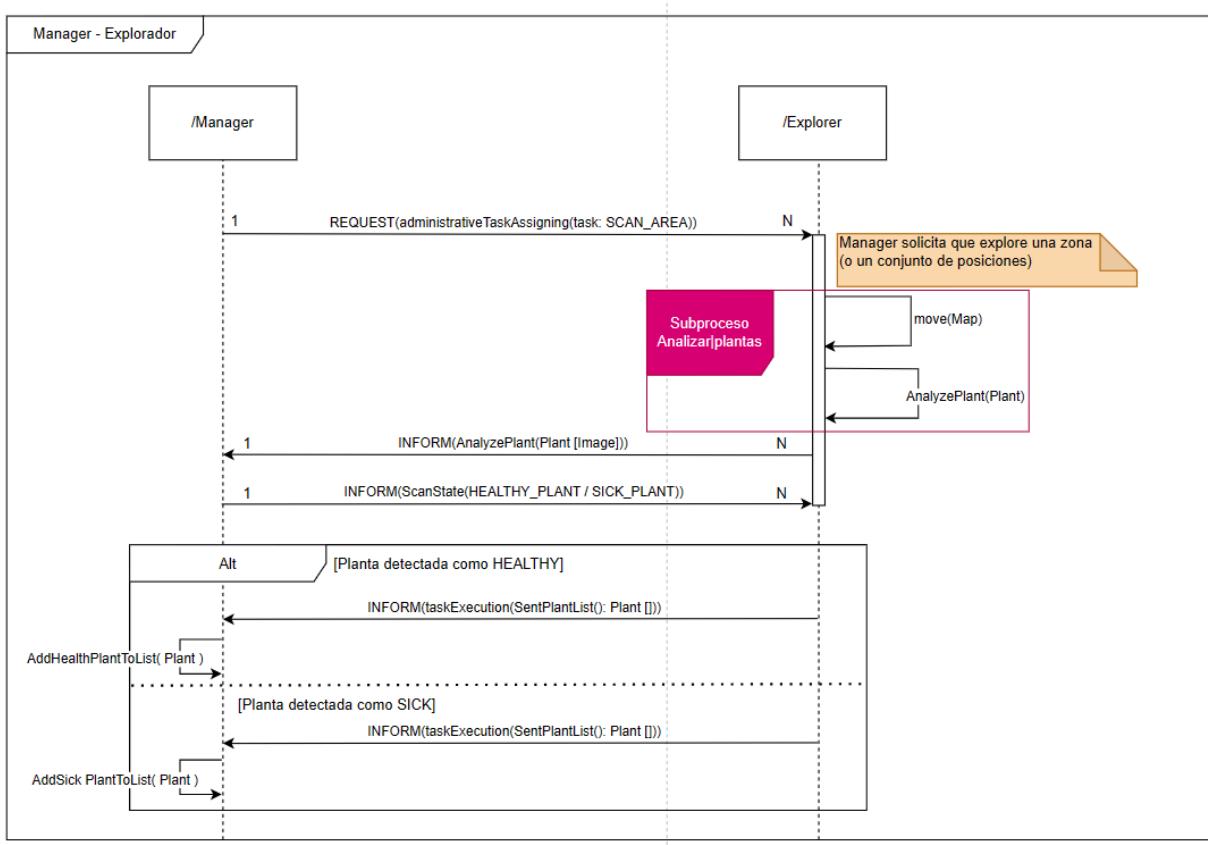


Figura 6. AIP\_AsignarTarea . Creado en [Draw.io](#) <sup>[1]</sup>

En la Figura 6 se presenta el diagrama AIP del proceso mediante el cual el manager designa al explorador una zona específica para escanear. A continuación, el explorador se desplaza por el mapa y lleva a cabo el análisis de las plantas como un subproceso. Finalmente, devuelve al manager un listado que distingue entre plantas sanas y plantas enfermas.

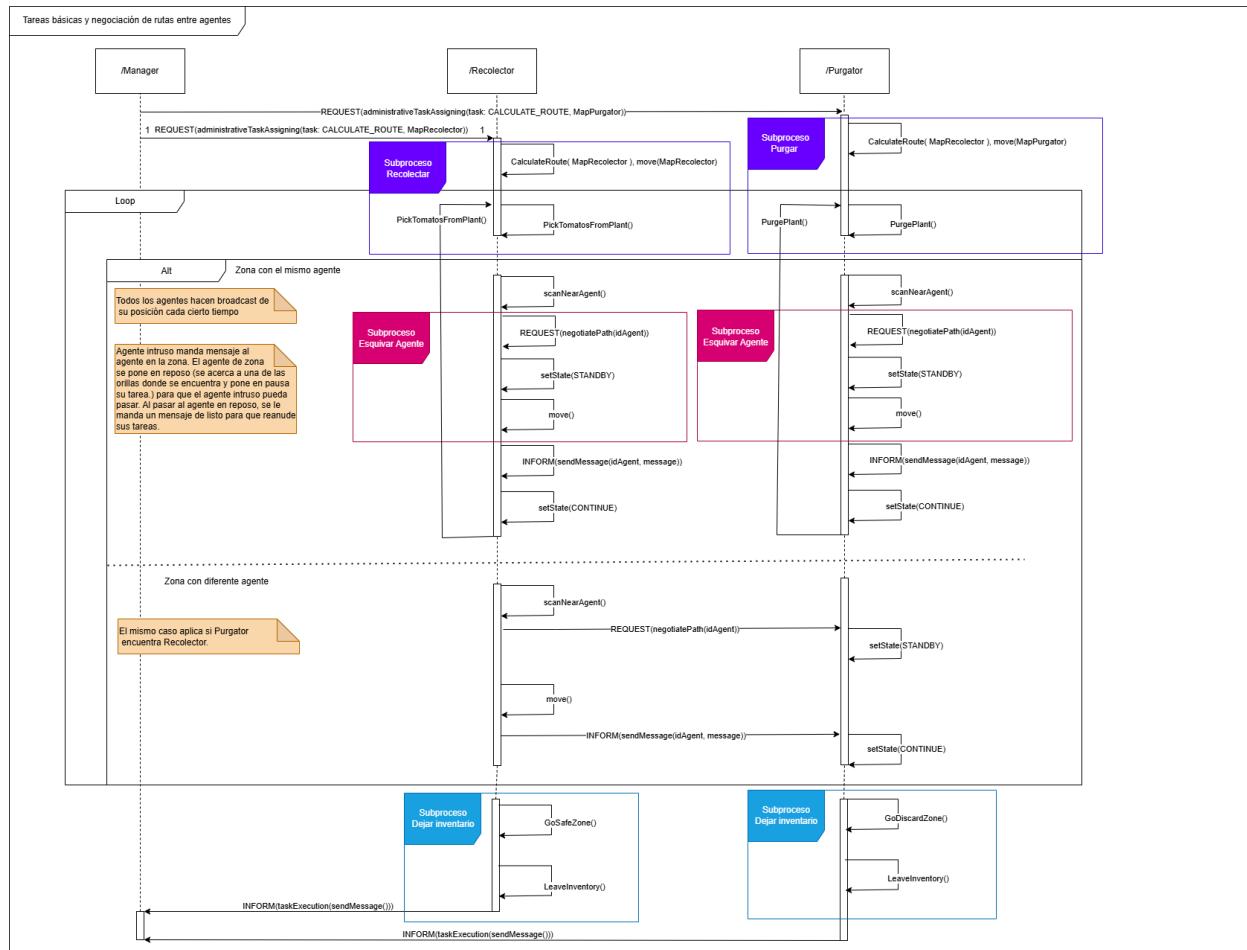


Figura 7. AIP\_subproceso\_tareas técnicas. Creado en [Draw.io](#) [1]

En la Figura 7, se muestra el diagrama AIP del proceso cuando se hace un REQUEST de parte del manager hacia los recolectores y purgadores para realizar su actividad normal y un INFORM de parte de los mismos hacia el manager para informar que la tarea solicitada sea ha realizado con éxito; recolectar / purgar plantas o tomates. En este mismo diagrama se representan los diagramas de subprocessos que invocan al recibir el intercambio de mensajes. El subprocesso de recolectar, purgar, el subprocesso esquivar agente y dejar inventario.

### 3. Diagramas de Actividad o de Estado

Recolector

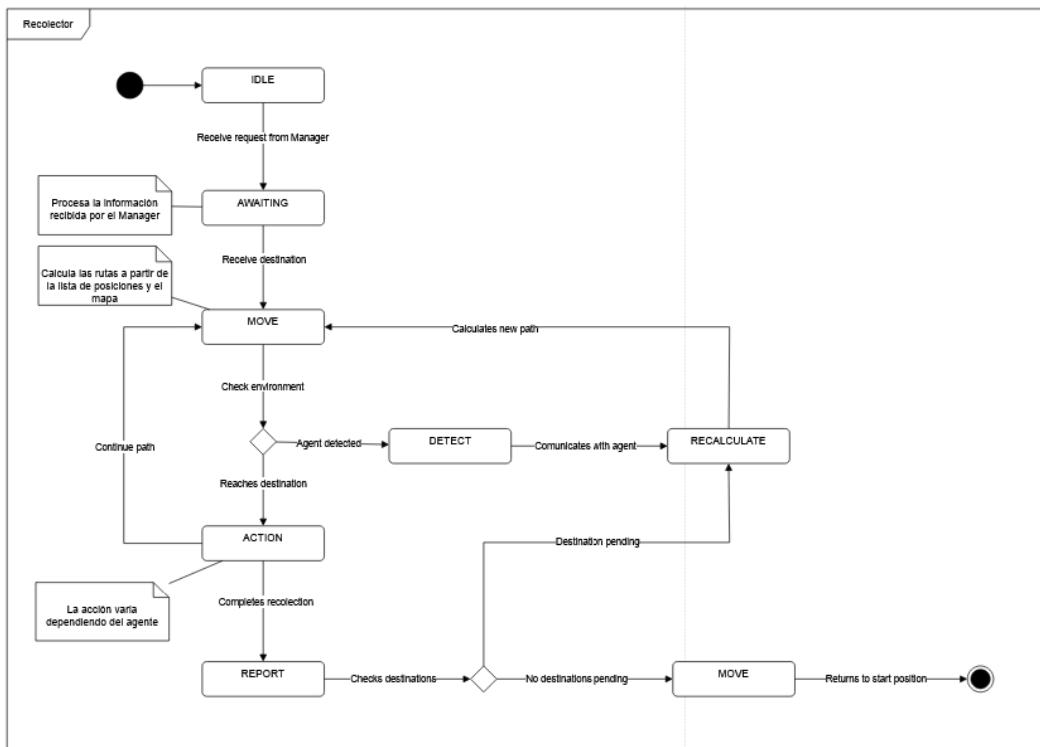


Figura 8. Diagrama de Actividad Recolector. Creado en [Draw.io](#) <sup>[1]</sup>

En la Figura 8, se muestra el diagrama de actividad del agente Recolector. Inicia su actividad esperando recibir información del manager. Tras recibir los destinos, calcula las rutas necesarias y se mueve mientras revisa su entorno detectando agentes y recalculando su ruta cuando sea necesario. Al llegar a su destino realiza la acción correspondiente y una vez termina, vuelve a su posición original.

## Purgator

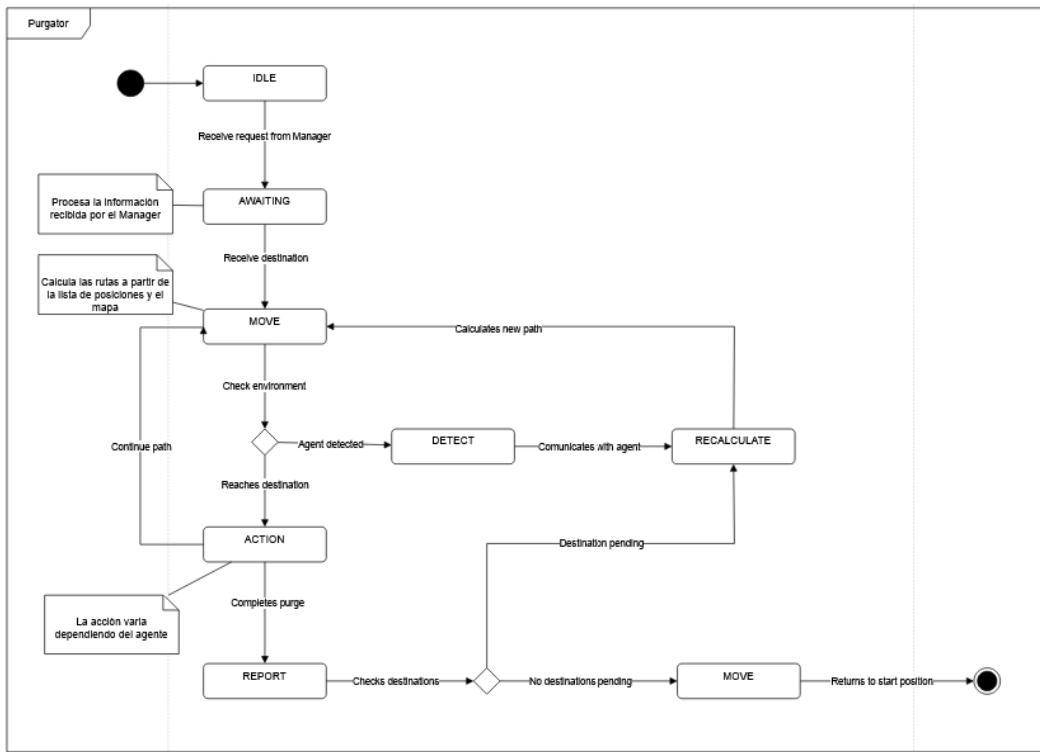


Figura 9. Diagrama de Actividad Purgator. Creado en [Draw.io](#) [1]

En la Figura 9, se muestra el funcionamiento del agente purgador. Comienza esperando a que le llegue información del Manager. Tras recibir sus objetivos, calcula las rutas usando el mapa y se mueve observando constantemente su entorno reaccionando a agentes. Al llegar a su destino ejecuta la acción asignada y continúa hasta terminar con todos los destinos.

## Explorador

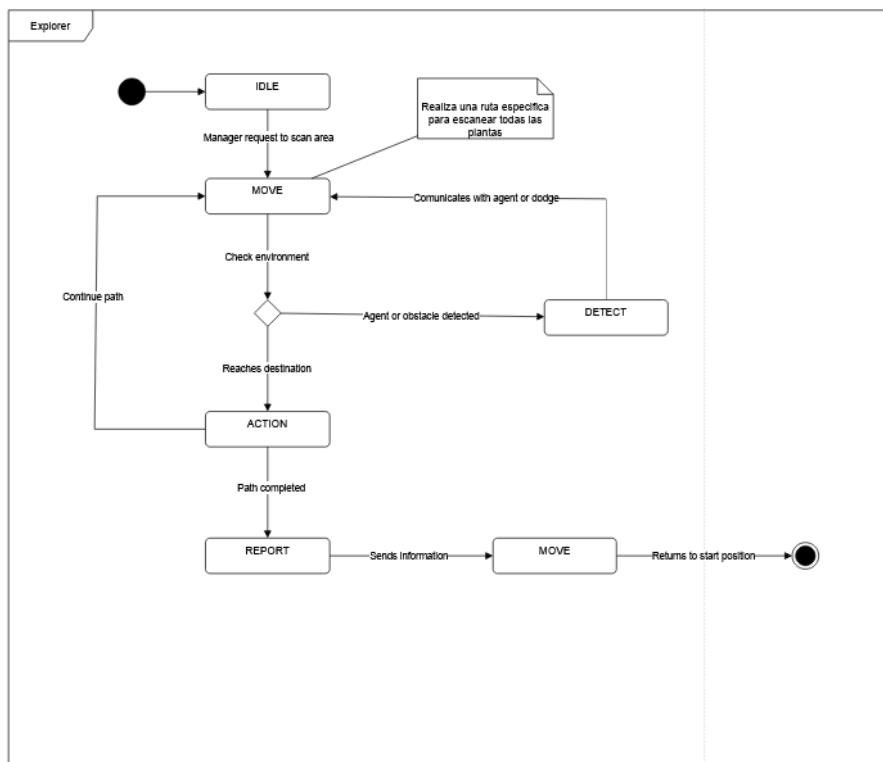


Figura 10. Diagrama de Actividad Explorador. Creado en [Draw.io](#) 

En la Figura 10, se muestra el diagrama de actividad del explorador. Su ciclo inicia en un estado inactivo hasta recibir la solicitud del Manager para escanear un área. Después sigue una ruta predeterminada, verificando su entorno constantemente y reaccionando a otros agentes u obstáculos. Finalmente envía un reporte al Manager y regresa a su posición original.

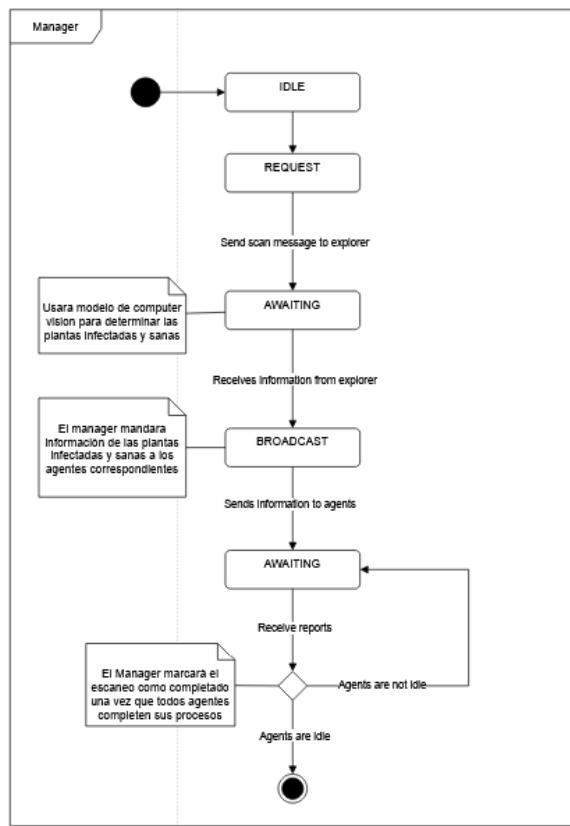


Figura 11. Diagrama de Actividad Manager. Creado en [Draw.io](#) 

En la Figura 11, se muestra el diagrama de actividad del manager, quien coordina el proceso de escaneo y análisis de plantas. Envía una solicitud al explorador y espera la información, después analiza la información para detectar plantas sanas e infectadas y distribuye los datos a los agentes correspondientes. El ciclo termina cuando todos los agentes terminan sus tareas.

### ¿Cómo visualizan el mundo virtual?

Una versión simplificada de la realidad, nuestro entorno es un invernadero de 500x500 que contiene plantas de tomates y que tiene agentes trabajando para mantenerlo. Las plantas de tomates están simplificadas a 3 partes en su estructura, y se toman en cuenta atributos como peso de tomates o estado de salud para poder simularlas. En esta simulación, la manera en la que los agentes interactúan con el entorno y entre sí es vital para lograr que este escenario virtual se asemeje más a la realidad.

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

Se tiene contemplado planta, explorador, recolector y purgador, zona de entrega de tomates y zonas de desecho de plantas enfermas,

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

Nuestros principales agentes que se desempeñan en el inventario son:

### **Planta**

Una Planta es un objeto que representa una planta con una ubicación en el espacio, una imagen asociada y varios valores que indican su peso y cuántas partes de ella están dañadas.

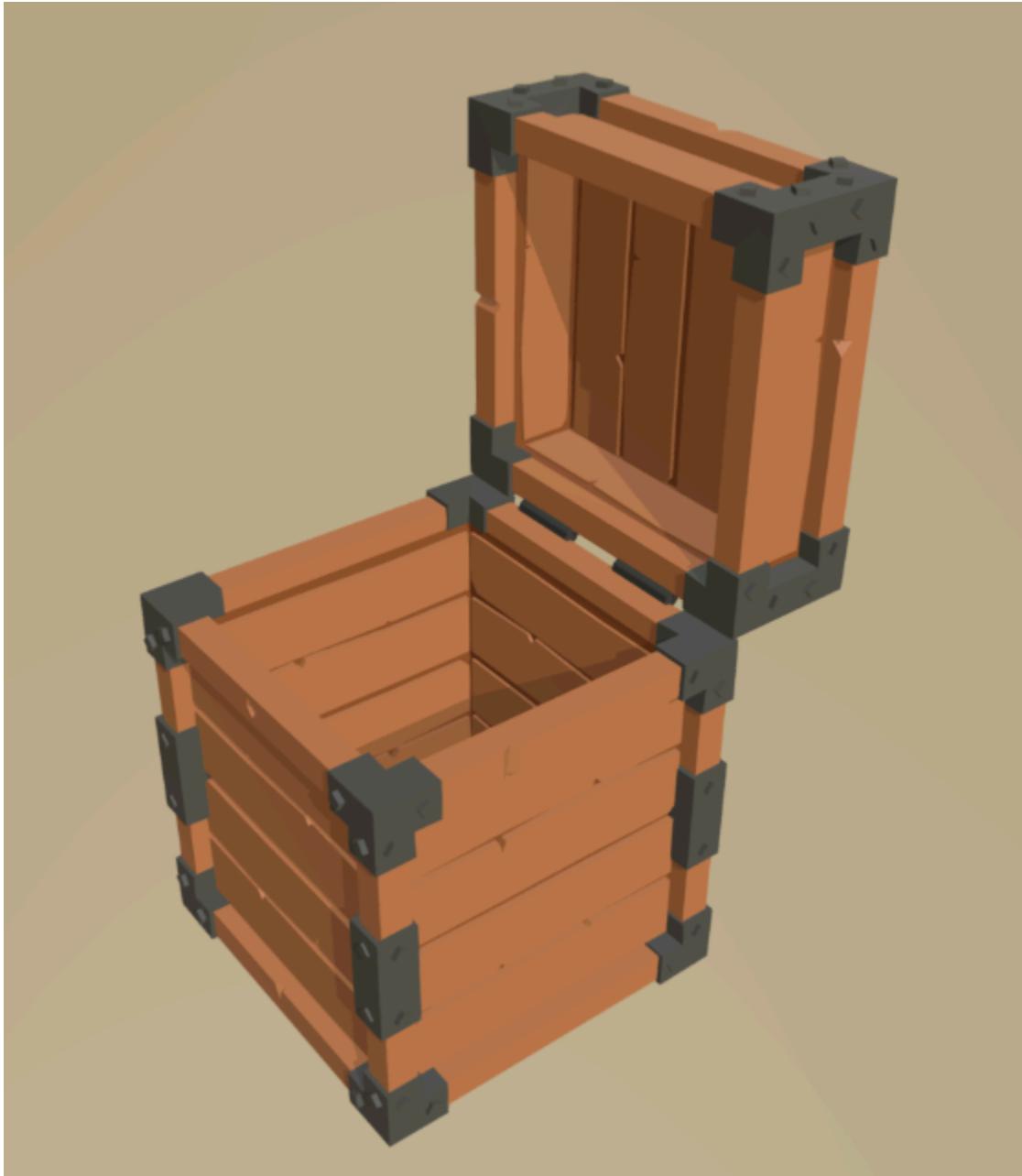


## Entry

Zona de entrada definida simplemente por su tamaño, es decir, su largo y ancho.

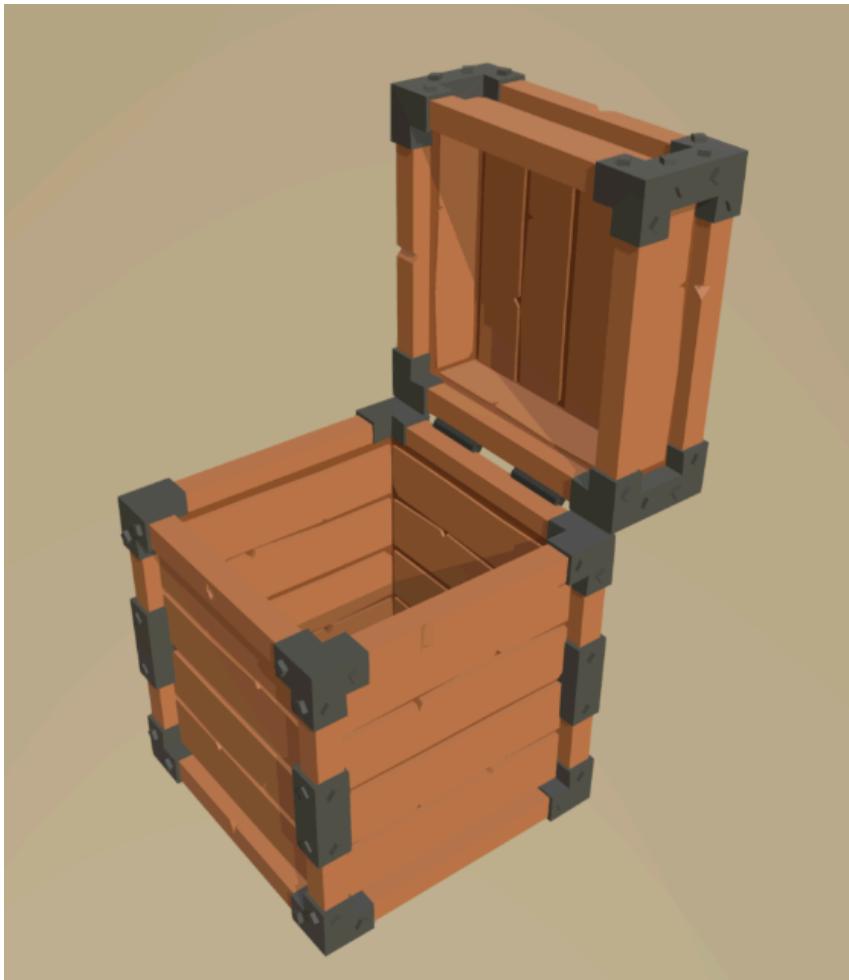
## TrashZone

Área delimitada por dimensiones de largo y ancho, que registra cuántas plantas han sido enviadas allí y está conectada a una zona de entrada.



## SafeZone

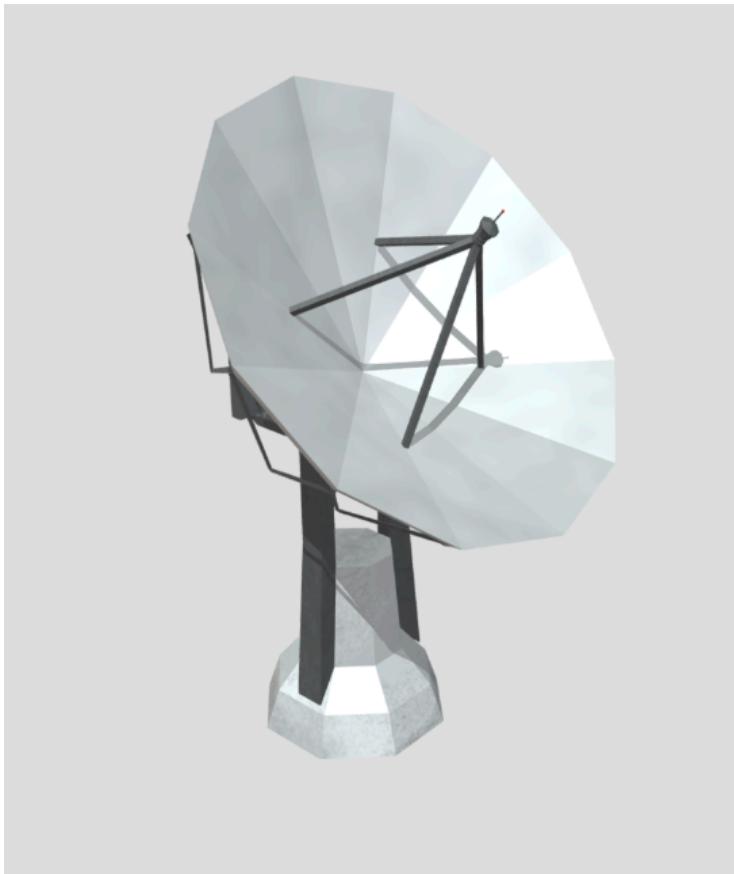
Zona segura con cierto tamaño, un punto de acceso y un registro del peso total que acumula lo que se almacena dentro.



(Se cambiara de color)

## Manager

Objeto que guarda listas de tareas, registros de tareas pasadas, un conjunto de plantas y posiciones clasificadas según si están sanas o enfermas.



### Explorer

Objeto que almacena una lista de plantas que ya fueron analizadas y otra lista con las posiciones espaciales de las plantas que debe revisar, ambas ordenadas para llevar registro de dónde está cada una.



### Recolector

Objeto que lleva una lista de posiciones de plantas sanas que debe recoger, junto con un peso máximo que puede cargar y un peso actual que va aumentando conforme recolecta; además conoce la posición exacta donde se encuentra la SafeZone.



### Purgator

Objeto similar al recolector pero enfocado en plantas enfermas: guarda sus posiciones, tiene un límite máximo de peso para desechos, un peso actual acumulado y sabe dónde está ubicada la TrashZone.



(se cambiara de color)

## Entity

Objeto general con energía, tamaño definido por largo y ancho, un estado actual, una posición espacial y acceso al mapa del invernadero que representa el entorno donde se mueve.

Cabe mencionar que todos los modelos seleccionados se sacaron de [3]

**Agregar esquemáticos o bocetos de dibujo para apoyar estas descripciones.**



Figura 12. Boceto de simulación. Creado en Gemini [2]

## 4. Referencias

- [1] “draw.io” n.d., <https://www.drawio.com/>.
- [2] “gemini” n.d., <https://gemini.google.com/> .
- [3] Sketchfab. “Sketchfab Website.” *Sketchfab*. Consultado el 24 de agosto de 2025. <https://sketchfab.com/>.