



Tecnológico de Monterrey

Campus Guadalajara

Evidencia 2. Revisión 1.

Clay Rodolfo Gutiérrez Herrera - A01220835

Luis Alfredo Carmona Martínez - A01707658

Luis Fernando Díaz Hernández - A01639435

Emilio Hernandez Flores - A01637284

Dylan Pereyra López - A01540618

Alejandro Gutiérrez Zamudio - A00227610

Iván Axel Dounce Nava

Javier Félix Rendón

Jesús Israel Hernández Hernández

Eduardo Morales Vargas

Modelación de Sistemas Multiagentes con gráficas computacionales - Grupo 302

10 de Noviembre del 2025

Conformación del equipo de trabajo	5
1. Clay Gutiérrez	5
Fortalezas	5
Áreas de oportunidad	5
Expectativas	6
2. Alfredo Carmona	6
Fortalezas	6
Áreas de oportunidad	7
Expectativas	7
3. Luis Díaz	7
Fortalezas	7
Áreas de oportunidad	8
Expectativas	8
4. Emilio Hernández	8
Fortalezas	8
Áreas de oportunidad	9
Expectativas	9
5. Dylan Pereyra	9
Fortalezas	9
Áreas de oportunidad	10
Expectativas	10
6. Alejandro Gutiérrez	10
Fortalezas	10
Áreas de oportunidad	10
Expectativas	11
Expectativas grupales	11
Compromisos grupales	11
Herramientas colaborativas:	13
Github	13
Discord (Invitación para los profesores a unirse)	13
Descripción del reto	13
Propuesta general de solución	14
Flujo completo del sistema	16
0. Modelo del invernadero (base común)	16
1. Planificación inicial (Patólogo)	17
2. Misión del ScoutBot	17
3. Descarga en la Base (ScoutBot → Patólogo)	19
4. Análisis Etapa 1 (Patólogo – screening masivo)	20
5. Activación del PickBot Enfermero (Etapa 2 – verificación focalizada)	21
6. Análisis Etapa 2 (Patólogo – confirmación de diagnóstico)	21

7. (Opcional) Activación del PickBot de sanos	22
8. Ciclo	22
Agentes	23
1. Agente Patólogo Digital (PathologistAgent)	23
1.1. Identificación del agente y relaciones	23
1.1.1. Responsabilidades clave:	23
1.1.2. Relaciones con otros agentes y elementos:	24
1.2. Tipo de arquitectura del Patólogo	27
1.3. Componentes arquitectónicos del Patólogo	28
1.3.1. Capa Reactiva (componente del agente híbrido)	28
1.3.2. Capa Deliberativa (BDI) del Patólogo	34
1.3.3. Integración Híbrida	40
2. Agente ScoutBot (Agente Explorador)	41
2.1. Identificación del agente y relaciones	41
2.1.1. Responsabilidades clave	41
2.1.2. Relaciones con otros agentes y elementos	43
2.2. Tipo de arquitectura del ScoutBot	44
2.3. Componentes arquitectónicos del ScoutBot	45
2.3.1. Capa de Percepción	45
2.3.2. Capa de Control Reactivo	45
2.3.3. Capa de Acción	47
2.3.4. Características clave del diseño reactivo	47
3. PickBot Enfermero (Agente Recolector Especializado)	48
3.1. Identificación del agente y relaciones	48
3.1.1. Responsabilidades clave	48
3.1.2. Relaciones con otros agentes y elementos	49
3.2. Tipo de arquitectura del PickBot Enfermero	51
3.3. Componentes arquitectónicos del PickBot Enfermero	52
3.3.1. Capa de Percepción	52
3.3.2. Capa de Control Reactivo	53
3.3.3. Capa de Acción	55
3.3.4. Características clave del diseño reactivo	55
4. Agente PickBot de Sanos (Agente Recolector de Producción Segura)	56
4.1. Identificación del agente y relaciones	56
4.1.1. Responsabilidades clave	56
4.1.2. Relaciones con otros agentes y elementos	57
4.2. Tipo de arquitectura del PickBot de Sanos	59
4.3. Componentes arquitectónicos del PickBot de Sanos	60
4.3.1. Capa de Percepción	60
4.3.2. Capa de Control Reactivo	60

4.3.3. Capa de Acción	62
4.3.4. Características clave del diseño reactivo	63
Plan de trabajo	63
1. Organización general del plan	63
2. Plan de trabajo por semanas	64
Semana 1 – Definición y base colaborativa	64
Objetivos principales:	64
Actividades:	64
Responsables (sugerido):	65
Intervalo de esfuerzo estimado (por persona)	65
Semana 2 – Modelo del invernadero y agentes básicos (Primera revisión)	65
Objetivos principales:	65
Actividades planeadas para la primera revisión:	65
Responsables y esfuerzo estimado:	66
Resultado esperado al final de Semana 2 (primera revisión):	66
Semana 3 – Integración del Patólogo y flujo ScoutBot → Patólogo	67
Objetivos principales:	67
Actividades:	67
Responsables:	68
Actividades pendientes al cierre de Semana 3 (esperado):	68
Semana 4 – Screening, PickBots y pipeline completo	68
Objetivos principales:	68
Actividades:	68
Responsables:	69
Actividades pendientes al cierre de Semana 4 (esperado):	70
Semana 5 – Refinamiento, validación y presentación final	70
Objetivos principales:	70
Actividades:	70
Responsables:	71
Aprendizaje adquirido	72
1. Comprensión de sistemas multiagentes aplicados a un problema real:	72
2. Diferenciación entre Digital Shadow, Digital Twin y Digital Environment:	72
3. Diseño de arquitecturas de agentes:	72
4. Importancia de centralizar la inteligencia pesada:	73
5. Trabajo colaborativo estructurado:	73
6. Enfoque iterativo e incremental:	73

Conformación del equipo de trabajo

1. Clay Gutiérrez

Fortalezas

- Capacidad para diseñar arquitecturas de sistemas complejos.
- Pensamiento estructurado y orientación a la documentación clara y consistente.
- Experiencia previa usando herramientas modernas (GitHub, control de versiones, herramientas colaborativas).
- Facilidad para conectar el problema técnico con el contexto real (operación en campo, restricciones de hardware, comunicación).
- Compromiso alto con la calidad de las entregas y seguimiento de planes de trabajo.

Áreas de oportunidad

- Tendencia a abarcar demasiado alcance en paralelo; requiere acotar entregables para asegurar versiones funcionales tempranas.
- Necesidad de delegar de forma más explícita para equilibrar carga de trabajo dentro del equipo.
- Mantener tiempos realistas entre detalle técnico y requerimientos académicos del bloque.

Expectativas

- Consolidar conocimientos en sistemas multiagentes aplicados a problemas reales (detección temprana de enfermedades en agricultura).
- Profundizar en el diseño de arquitecturas híbridas (reactivo + deliberativo/BDI).
- Mejorar la integración entre simulación (Unity), control centralizado y modelos de decisión basados en datos.

2. Alfredo Carmona

Fortalezas

- Habilidad para la comunicación efectiva y la escucha activa, facilitando la colaboración, el intercambio de ideas y la construcción de soluciones en consenso.
- Pensamiento estructurado y alta capacidad de organización, implementando una planificación detallada antes de la ejecución para minimizar imprevistos y optimizar el flujo de trabajo.
- Enfoque pragmático orientado a resultados, priorizando la funcionalidad inicial de las soluciones para después refinarlas y optimizarlas de forma iterativa.
- Capacidad para integrar activamente la retroalimentación y las diversas perspectivas del equipo, fomentando un ambiente colaborativo para construir soluciones consensuadas.

Áreas de oportunidad

- Optimizar el proceso de síntesis de ideas para estructurar y priorizar planes de acción con mayor agilidad.
- Implementar técnicas de priorización y gestión de carga de trabajo para mitigar la confusión y reducir la incidencia del trabajo bajo presión.
- Desarrollar metodologías para la estimación de tiempos y alcance, buscando un balance más preciso entre la ambición del proyecto y los plazos reales.

Expectativas

- Adquirir un dominio práctico del entorno de Unity, con un enfoque específico en la manipulación de cámaras y la implementación de mecánicas de movimiento.
- Explorar y analizar aplicaciones prácticas y casos de uso de los sistemas de agentes en contextos del mundo real.
- Desarrollar una comprensión conceptual sólida de los agentes inteligentes, incluyendo sus modelos de interacción y convivencia.

3. Luis Díaz

Fortalezas

- Habilidad para expresar ideas técnicas de forma clara, ordenada y comprensible, facilitando la colaboración y la transferencia de conocimiento.
- Compromiso continuo con la mejora de mis conocimientos.

- Participación activa en proyectos colaborativos, asumiendo roles de coordinación y fomentando un ambiente de trabajo digno.
- Priorizar la funcionalidad antes de la optimización y aplicar metodologías sistemáticas en el desarrollo de soluciones.

Áreas de oportunidad

- Fortalecer la administración de tiempos y prioridades para equilibrar la profundidad técnica con la eficiencia en la entrega de resultados.
- Reforzar la confianza en el propio criterio profesional al momento de tomar decisiones.
- Saber delegar responsabilidades.

Expectativas

- Aprender los fundamentos del manejo de las cámaras y demás elementos técnicos de Unity.
- Entender e integrar correctamente los conceptos relacionados a los multiagentes y saber aplicarlos en situaciones de la vida real.
- Identificar la mejor arquitectura posible para el diseño al que se quiere llegar.

4. Emilio Hernández

Fortalezas

- Experiencia en el uso de herramientas colaborativas para el trabajo de diseño de software.

- Experiencia plena en el uso de entornos de desarrollo y simulación 3D como Unity y el uso de scripts y gameobjects para crear entornos creíbles.
- Experiencia anterior con el manejo de topográfica, texturizado, renderizado, UV mapping y animación en entornos de simulación 3D como Blender.
- Pensamiento estructurado y lógico para la resolución de problemas computacionales.

Áreas de oportunidad

- Fortalecer técnicas de manejo de tiempo y división de labor.
- Mejorar hacia una actitud mucho más independiente y proactiva

Expectativas

- Profundizar mi percepción del uso de agentes para la resolución de problemas de alta complejidad / alto nivel de mantenimiento.
- Convertirme proficiente en la automatización de metodologías de resolución autosuficientes.
- Realizar trabajos profundos enfocadas a las estructuras de resolución usando sistemas de multiagentes.

5. Dylan Pereyra

Fortalezas

- Habilidad de poder escuchar a las personas, no solamente escucharlas si no puedo comprenderlas y entender sus emociones
- Pensamiento crítico, para analizar la información objetivamente, evaluar argumentos y resolver problemas de manera lógica

- Tengo experiencia en Unity así como en blender, unity puedo comprender las máquinas de estado y puedo modelar ciertos objetos en blender para usarlos en Unity
- Tengo un poco de experiencia liderando

Áreas de oportunidad

- Trabajo, pero trabajo mucho mejor bajo presión
- Experiencia nula con sistemas multiagentes, pero capaz de entenderlos en un periodo corto de tiempo

Expectativas

- Refinar mis conocimientos en Unity ya que no he trabajado en Unity desde hace tiempo
- Poder comprender a profundidad los conceptos de sistemas multiagentes e implementarlos en el proyecto que se realizará
- Profundizar en las diferentes arquitecturas que se nos presentarán

6. Alejandro Gutiérrez

Fortalezas

- Me gusta el orientado a objetos
- Entiendo lógica de otros rápido
- Experiencia previa de Unity

Áreas de oportunidad

- Mi planeación de tiempo no es muy buena

- No soy muy bueno siendo líder

Expectativas

- Que el reto sea entretenido de desarrollar
- Lograr un buen trabajo en equipo entre todos

Expectativas grupales

- Desarrollar una propuesta formal sólida, coherente y técnicamente defendible para el reto de detección temprana de infección mediante un sistema multiagente.
- Lograr un prototipo funcional en Unity que represente:
 - El invernadero como grafo de segmentos.
 - El comportamiento básico de ScoutBots, PickBot Enfermero y Patólogo Digital, opcional PickBot de sanos.
- Mantener una documentación clara, consistente y alineada con los lineamientos del bloque.
- Repartir responsabilidades de manera equitativa, aprovechando las fortalezas de cada integrante.

Compromisos grupales

El equipo asume los siguientes compromisos:

- Compromiso de comunicación:

- Definir y utilizar de forma constante un canal oficial (Discord y repositorio en GitHub).
 - Reportar avances, bloqueos y necesidades de apoyo de forma oportuna.
- Compromiso de organización:
 - Mantener un plan de trabajo actualizado con actividades, responsables y fechas.
 - Respetar acuerdos internos y tiempos límite fijados por el equipo y por el curso.
- Compromiso técnico:
 - Asegurar que todas las decisiones de diseño (agentes, arquitecturas, flujos) se documenten.
 - Mantener la consistencia entre modelo conceptual, simulación en Unity y documentación escrita.
- Compromiso de calidad:
 - Revisar en equipo los entregables antes de su envío.
 - Cuidar coherencia gráfica (tipografía, colores, estilo, diagramas) en todos los documentos.
- Compromiso de aprendizaje:

- Aprovechar el proyecto como espacio para experimentar con ideas de sistemas multiagentes, no solo para cumplir.
- Compartir conocimiento entre integrantes para reducir brechas y fortalecer al equipo como unidad.

Herramientas colaborativas:

Github

<https://github.com/Clay2605293/MASTomatoes>

Discord (Invitación para los profesores a unirse)

<https://discord.gg/pvtdnZsYBV>

Descripción del reto

En la producción de tomate en México, la presencia del Tomato brown rugose fruit virus (ToBRFV) representa una amenaza crítica para la productividad y la seguridad alimentaria. De acuerdo a la información recibida en clase, actualmente, la detección de posibles infecciones depende de inspecciones visuales realizadas por trabajadores en campo. Cuando un recolector identifica síntomas sospechosos, informa al capataz, quien a su vez escala el caso a un especialista o patólogo vegetal. Este proceso es lento, sujeto a error humano y altamente reactivo: cuando se confirma la infección, con frecuencia ya es necesario retirar plantas o secciones completas del invernadero, generando pérdidas económicas significativas.

El reto consiste en diseñar una solución que permita acelerar y sistematizar la detección temprana de posibles infecciones, reduciendo la dependencia exclusiva de la inspección manual y del juicio tardío de expertos humanos. La solución debe ser capaz de:

- Recorrer el invernadero de forma estructurada.
- Capturar información relevante del estado de las plantas y frutos.
- Detectar posibles síntomas de enfermedad con mayor cobertura y trazabilidad.
- Separar los flujos de manejo de frutos sanos, sospechosos y potencialmente infectados.
- Disminuir el tiempo entre la observación inicial y la toma de decisiones informadas.

Todo esto se abordará en un entorno simulado, utilizando un invernadero modelado digitalmente para experimentar con estrategias multiagente sin requerir hardware físico ni datos reales en esta etapa.

Propuesta general de solución

La propuesta se basa en el desarrollo de un sistema multiagente dentro de un ambiente digital, donde un invernadero simulado en Unity se representa como un grafo de segmentos de cultivo. Sobre este entorno se coordinan distintos tipos de agentes con responsabilidades claras y complementarias.

El núcleo del sistema es el Patólogo Digital (PathologistAgent), un agente híbrido que concentra toda la inteligencia del sistema. Este agente:

- Conoce el modelo completo del invernadero.
- Planifica y asigna misiones de inspección a agentes móviles.
- Procesa las imágenes recibidas.
- Evalúa el estado sanitario de plantas y frutos.
- Coordina la recolección diferenciada de producción sana y muestras sospechosas.
- Genera alertas cuando identifica plantas potencialmente infectadas.

Para la exploración se utilizan ScoutBots, agentes móviles reactivos que ejecutan misiones simples definidas por el Patólogo Digital. Cada ScoutBot recorre segmentos específicos del invernadero, captura imágenes de plantas y frutos según parámetros establecidos, y al finalizar regresa a la base central para descargar la información. Estos agentes no realizan diagnóstico: funcionan como sensores móviles de bajo costo computacional.

La propuesta incluye además dos tipos de agentes recolectores especializados:

- PickBot Enfermero: agente encargado de acudir a plantas con frutos sospechosos, recolectar muestras y llevarlas a la zona de análisis del Patólogo Digital para una evaluación más detallada.

- PickBot de Sanos (opcional): agente dedicado exclusivamente a la recolección de producción en zonas confirmadas como seguras, manteniendo separado el flujo de frutos sanos del manejo de muestras sospechosas.

Con esta arquitectura, el sistema simulado permite:

- Probar estrategias de planificación, cobertura y coordinación multiagente.
- Centralizar el uso de modelos de visión y diagnóstico en un único agente inteligente.
- Representar de forma clara el ciclo completo: inspección → sospecha → verificación → acción.
- Sentar bases conceptuales que, en una etapa futura, podrían integrarse con sensores reales, robots físicos o esquemas de Digital Shadow / Digital Twin.

Esta propuesta formal define el marco conceptual sobre el cual el equipo construirá la simulación, la lógica de los agentes y el plan de trabajo del bloque.

Flujo completo del sistema

0. Modelo del invernadero (base común)

- El invernadero se representa como un grafo de segmentos:
 - Nodos = segmentos de fila / puntos clave (F1_A, F1_B, F2_A, etc.) + Base central.
 - Aristas = pasillos transitables entre esos puntos.

- Cada segmento tiene asociada una lista de plantas, y cada planta puede tener varios frutos.

Este grafo lo conoce solo el Patólogo; los ScoutBots sólo reciben sus misiones ya procesadas.

1. Planificación inicial (Patólogo)

1. El Patólogo:

- Detecta cuántos ScoutBots están disponibles.
- Divide los segmentos del invernadero en subconjuntos balanceados (zonas).
- Asigna a cada ScoutBot una lista ordenada de segmentos a visitar.

2. Define también:

- Parámetros de captura (cuántas imágenes por planta / por segmento).
- Reglas básicas de misión (qué hacer si hay bloqueo, cuándo regresar).

Comunicación:

El Patólogo envía a cada ScoutBot su misión inicial cuando están en la Base central.

2. Misión del ScoutBot

Cada ScoutBot ejecuta una misión simple, casi totalmente reactiva:

1. Sale de la Base → va al primer segmento asignado.
2. En cada segmento:

- Recorre la fila siguiendo un patrón fijo (por ejemplo lineal).
- Para cada planta relevante:
 - Toma 1–3 fotos:
 - Vista general de planta.
 - Vista donde se observen frutos.
 - (Opcional) otra vista si el layout lo requiere.
 - Guarda:
 - Imágenes en memoria local.
 - Metadata:
 - ID de segmento
 - ID/planta
 - Timestamp.

3. Si detecta un obstáculo físico (maquinaria, personas, etc.) que impide seguir:

- Aplica una regla simple:
 - Marca ese segmento como “no inspeccionado”.
 - Pasa al siguiente segmento de su lista.
- No recalcula rutas globales complejas, solo:
 - “No pude aquí, sigo con mi lista”.

4. Cuando termina su lista (más los reintentos posibles):

- Regresa a la Base central.

Durante la misión normal no manda fotos una por una por la red; solo acumula.

3. Descarga en la Base (ScoutBot → Patólogo)

Al llegar a la Base:

1. El ScoutBot se acopla (dock).

2. Transfiere en lote:

- Todas las imágenes capturadas.
- Bitácora de qué segmentos fueron:
 - Inspeccionados con éxito.
 - Saltados por bloqueo.

3. El Patólogo:

- Integra estos datos en su modelo del invernadero.
- Sabe:
 - Qué fue revisado.
 - Qué sigue pendiente.
 - Dónde hay datos para analizar.

4. Análisis Etapa 1 (Patólogo – screening masivo)

Con las imágenes descargadas:

1. El Patólogo corre su módulo de visión computarizada/diagnóstico (simulado):

- Detecta frutos en cada imagen.
- Para cada fruto identificado:
 - Clasifica como:
 - SANO
 - SOSPECHOSO
 - CRÍTICO si se ve grave.

2. Actualiza el mapa de estado:

- Plantas con solo frutos sanos → zona segura.
- Plantas con frutos sospechosos y críticos → zona en observación.

3. Genera una lista de objetivos sospechosos:

- Coordenadas aproximadas
- Nivel de sospecha
- Prioridad.

Si detecta segmentos no inspeccionados (por obstáculos o fallos):

- Puede planear una nueva misión con ScoutBots en una iteración posterior (esto ya es decisión del Patólogo que definiremos en su BDI).

5. Activación del PickBot Enfermero (Etapa 2 – verificación focalizada)

Con la lista de frutos sospechosos:

1. El Patólogo ordena al PickBot Enfermero:
 - Visitar esas plantas específicas (usando el grafo para generar rutas eficientes).
2. El PickBot Enfermero:
 - Llega a la planta sospechosa.
 - Simula:
 - Recolección del fruto sospechoso
 - Lleva el fruto a la zona de análisis del Patólogo (en la Base).

6. Análisis Etapa 2 (Patólogo – confirmación de diagnóstico)

El Patólogo realiza una segunda pasada, con una cámara que toma imágenes ultra detalladas:

1. Con el fruto físico:
 - Aplica reglas/modelo más estricto.
2. Decide:
 - Enfermo confirmado:

- Marca la planta (y potencialmente su entorno) como infectada.
- Genera alerta a humano (capataz/agricultor).
- Descartado / baja sospecha:
 - Planta queda como sana o en observación.

3. Actualiza su mapa global:

- Esto ya es parte central de sus creencias.

7. (Opcional) Activación del PickBot de sanos

Con la información consolidada:

1. El Patólogo puede:

- Mandar el PickBot de sanos a cosechar zonas confirmadas como seguras.

2. Esto refuerza:

- Separación de flujos: un robot no contamina al otro.
- Uso eficiente de la información generada por el análisis central.

8. Ciclo

- El proceso se repite en ciclos:
 - Nuevas misiones de ScoutBots para nuevas fechas/lotes.
 - Actualización de creencias.

- Siempre con la lógica:
 - ScoutBots = movimiento + captura.
 - Patólogo = análisis, decisión, coordinación.
 - PickBots = ejecución física específica según orden del Patólogo.

Agentes

1. Agente Patólogo Digital (PathologistAgent)

1.1. Identificación del agente y relaciones

Nombre del agente: Patólogo Digital (PathologistAgent)

Tipo: Agente central de decisión, coordinación y diagnóstico.

Rol en el sistema: Es el cerebro global del invernadero simulado.

1.1.1. Responsabilidades clave:

1. Conocer el modelo digital del invernadero:
 - Representado como grafo (nodos = segmentos de filas con plantas asociadas; aristas = pasillos/conexiones).
2. Detectar y gestionar recursos:
 - Saber cuántos ScoutBots están disponibles y su estado.
 - Saber si existen PickBots (sanos) y PickBot Enfermero.
3. Asignar trabajo:
 - Dividir el invernadero en filas y asignarlas a cada ScoutBot.

4. Integrar información:

- Recibir reportes de cobertura (qué filas fueron inspeccionadas).
- Recibir reportes de detección: plantas/frutos sanos, sospechosos, críticos.

5. Tomar decisiones sobre acción:

- Decidir cuándo activar PickBots sanos para recolectar producción segura.
- Decidir cuándo activar el PickBot Enfermero para recolectar muestras sospechosas.
- Evaluar las muestras sospechosas (diagnóstico simulado).
- Marcar plantas como:
 - Sana confirmada
 - Sospechosa en observación
 - Enferma confirmada (disparando alerta a humano).

1.1.2. Relaciones con otros agentes y elementos:

● Con ScoutBots (Agentes Exploradores)

- Envía:
 - Misiones iniciales:
 - Listas de segmentos (nodos del grafo) a visitar.

- Parámetros de captura (número de fotos, etc.).
 - Reglas básicas (qué hacer si hay bloqueo, cuándo regresar).
- Recibe (al regreso a Base):
 - Lote de imágenes capturadas.
 - Bitácora de segmentos inspeccionados y no inspeccionados.
- Relación conceptual:
 - El Patólogo usa a los ScoutBots como sensores móviles del entorno. Los ScoutBots no interpretan; solo capturan y reportan.
- **Con PickBot Enfermero (agente recolector especializado)**
 - Envía:
 - Lista de plantas/frutos sospechosos a verificar.
 - Rutas óptimas (en términos de segmentos o puntos objetivo).
 - Recibe:
 - Confirmación de recolección de cada fruto sospechoso.
 - Disponibilidad actual del PickBot.
 - Relación conceptual:

- El Patólogo decide qué frutos ameritan inspección especializada y usa al PickBot Enfermero como herramienta para traer muestras físicas al módulo de análisis detallado.

- **Con PickBot de sanos (agente opcional)**

- Envía:
 - Tareas de cosecha en zonas confirmadas como seguras.
- Recibe:
 - Confirmación de tareas completadas.
- Relación conceptual:
 - Separa la cadena “segura” de la “sospechosa/enferma”, reduciendo riesgo de contaminación cruzada.

- **Con el Humano (operador/agricultor)**

- Envía:
 - Alertas cuando se confirma infección.
 - Reportes de zonas seguras/no seguras.
- Recibe (opcional como centro de configuración):
 - Parámetros de política (umbrales, prioridades).
 - Confirmaciones de acciones reales (fuera de la simulación).
- Relación conceptual:

- No se modela como agente; es un actor externo.

- **Con el medio de comunicación**

- El canal no es un agente, es un elemento del sistema.
- El Patólogo define protocolos de mensajes, los transmite y los recibe.

1.2. Tipo de arquitectura del Patólogo

El Patólogo Digital se define como un agente híbrido.

Justificación:

- Tiene una capa reactiva:
 - Responde de forma inmediata a eventos:
 - Llegada de un ScoutBot con datos.
 - Llegada de muestras del PickBot Enfermero.
 - Detección de inconsistencias (segmentos no inspeccionados).
 - Aplica reglas simples de actualización de estado.
- Tiene una capa deliberativa (BDI):
 - Mantiene un modelo interno del invernadero.
 - Formula metas (detectar temprano, minimizar riesgo, maximizar rendimiento sano).
 - Selecciona y mantiene planes (intenciones) para:
 - Cobertura del invernadero.

- Gestión de sospechosos.
- Uso de recursos (ScoutBots/PickBots).
- Generación de alertas.
- Combina ambas:
 - La capa reactiva detecta eventos.
 - La capa deliberativa actualiza creencias y ajusta intenciones según esos eventos.

1.3. Componentes arquitectónicos del Patólogo

1.3.1. Capa Reactiva (componente del agente híbrido)

La capa reactiva se encarga de respuestas inmediatas, basadas en reglas simples:

Capa de Percepción

La capa de percepción del Patólogo Digital se encarga de recibir y estructurar los eventos provenientes del entorno y de los demás agentes. No toma decisiones estratégicas; su función es transformar entradas en información utilizable por las capas superiores.

Fuentes de percepción:

- ScoutBots
 - SCOUT_MISSION_COMPLETED(scout_id, datos_misión)
 - Incluye:

- Lista de segmentos inspeccionados.
 - Lista de segmentos no inspeccionados.
 - Referencias a imágenes capturadas por segmento/planta.
- PickBot Enfermero
 - SAMPLE_DELIVERED(fruto_id, planta_id, segmento_id)
 - Indica que una muestra sospechosa ha sido entregada en la Base.
- PickBot de sanos (opcional)
 - HARVEST_COMPLETED(zona_id)
 - Indica finalización de recolección en una zona segura.
- Estado de recursos
 - PICKBOT_AVAILABLE(tipo)
 - SCOUT_AVAILABLE(scout_id)
- Alertas internas
 - ANOMALY_IN_DATA
 - Disparada cuando se detectan inconsistencias en la información recibida (segmentos sin datos, resultados contradictorios, etc.).

Funciones principales de la capa de percepción:

- Validar la estructura de los mensajes recibidos.
- Asociar cada evento a:
 - Segmentos del grafo.
 - Plantas y frutos correspondientes.
 - Agente emisor.
- Entregar estos eventos de forma normalizada a la Capa de Control Reactivo para su procesamiento inmediato.
- Registrar trazas mínimas para depuración y trazabilidad.

Capa de Control Reactivo

La capa de control reactivo implementa las reglas de estímulo–respuesta del Patólogo Digital. Opera sobre los eventos recibidos por la capa de percepción y ejecuta acciones inmediatas sobre el estado interno, sin realizar planificación compleja.

Reglas principales:

- Regla 1: Finalización de misión de ScoutBot
 - Evento: SCOUT_MISSION_COMPLETED(scout_id, datos_misión)
 - Acciones:
 - Marcar al scout_id como disponible.
 - Registrar segmentos marcados como inspeccionados.

- Registrar segmentos no inspeccionados como PENDIENTE_REVISIÓN o NO_INSPECCIONADO.
 - Activar el disparador para el Análisis Etapa 1 (screening masivo) sobre las imágenes asociadas.
- Regla 2: Actualización de estado de segmento
 - Evento: SEGMENT_STATUS_UPDATE(segmento_id, estado)
 - Acciones:
 - Actualizar en el modelo interno el estado del segmento:
 - INSPECCIONADO, NO_INSPECCIONADO, BLOQUEADO, PENDIENTE_REVISIÓN.
- Regla 3: Entrega de muestras sospechosas
 - Evento: SAMPLE_DELIVERED(fruto_id, planta_id, segmento_id)
 - Acciones:
 - Registrar que el fruto sospechoso está disponible para análisis detallado.
 - Activar el disparador para el Análisis Etapa 2 asociado a esa muestra.
 - Mantener al PickBot Enfermero en estado “en evaluación” o “disponible”, según la configuración.
- Regla 4: Disponibilidad de PickBots

- Evento: PICKBOT_AVAILABLE(tipo)
- Acciones:
 - Actualizar la disponibilidad del PickBot correspondiente.
 - Dejar preparado este estado para que la capa deliberativa pueda asignar nuevas tareas si existen intenciones pendientes.
- Regla 5: Detección de anomalías en datos
 - Evento: ANOMALY_IN_DATA
 - Acciones:
 - Marcar segmentos o zonas afectadas como PENDIENTE_REVISIÓN.
 - Registrar la necesidad de una futura misión de reinspección.

Características:

- No selecciona estrategias globales; solo mantiene el estado inmediato coherente.
- Actúa como puente entre la percepción y la lógica BDI:
 - Actualiza creencias básicas.
 - Dispara procesos deliberativos cuando corresponde (sin decidir su contenido).

Capa de Acción

La capa de acción del Patólogo Digital agrupa las operaciones concretas que el agente ejecuta como resultado directo de sus reglas reactivas. Estas acciones se reflejan sobre su estado interno y sobre los demás agentes del sistema.

Acciones principales:

- UPDATE_SEGMENT_STATUS(segmento_id, estado)
 - Actualiza el estado de inspección del segmento en el modelo interno.
- REGISTER_SCOUT_AVAILABILITY(scout_id)
 - Marca al ScoutBot como disponible para futuras misiones.
- STORE_MISSION_DATA(scout_id, datos_misión)
 - Asocia imágenes y bitácoras recibidas con segmentos, plantas y frutos.
- TRIGGER_STAGE1_ANALYSIS(lote_imágenes)
 - Inicia el proceso de Análisis Etapa 1 sobre las imágenes descargadas.
 - Esta acción conecta la capa reactiva con la capa deliberativa, que definirá cómo interpretar los resultados.
- REGISTER_SAMPLE(fruto_id, planta_id, segmento_id)
 - Registra la llegada de una muestra sospechosa para análisis detallado.
- TRIGGER_STAGE2_ANALYSIS(fruto_id, planta_id)
 - Inicia el proceso de Análisis Etapa 2 sobre una muestra específica.

- Prepara la información para que la capa deliberativa actualice diagnósticos y tome decisiones.
- UPDATE_PICKBOT_STATUS(tipo, estado)
 - Actualiza el estado de los PickBots (disponible, en misión, en espera).
- FLAG_REVIEW_ZONE(zona_id)
 - Marca una zona como candidata para reinspección en próximas misiones, cuando se detectan inconsistencias.

Características:

- Todas las acciones son atómicas y de bajo costo computacional.
- No implican por sí mismas decisiones estratégicas; habilitan información y condiciones para que la capa deliberativa (BDI) formule y ajuste intenciones.
- Garantizan que el estado interno del Patólogo esté siempre sincronizado con los eventos del entorno y de los demás agentes.

1.3.2. Capa Deliberativa (BDI) del Patólogo

En la parte superior del agente híbrido, el Patólogo funciona como un agente BDI (Beliefs–Desires–Intentions).

Creencias (Beliefs)

El Patólogo mantiene un modelo interno que incluye:

1. Modelo del invernadero:
 - Grafo $G(V, E)$:

- V: Base central, segmentos de filas.
- E: pasillos transitables.
- Para cada segmento:
 - Lista de plantas.
 - Para cada planta:
 - IDs de frutos asociados.

2. Estado de inspección:

- Por segmento:
 - NO_INSPECCIONADO, INSPECCIONADO, BLOQUEADO, PENDIENTE_REVISIÓN.
- Por planta/fruto (screening):
 - DESCONOCIDO, SANO, SOSPECHOSO, CRÍTICO.

3. Recursos:

- ScoutBots:
 - Disponibles / en misión / fallando.
 - Última zona asignada.
- PickBot Enfermero:
 - Disponibilidad, posición (a nivel de segmento).
- PickBot de sanos (opcional):

- Disponibilidad, carga de trabajo.

4. Resultados de análisis:

- Etapa 1 (screening masivo):
 - Lista de frutos sospechosos por planta/segmento.
- Etapa 2 (confirmación):
 - Lista de frutos analizados en detalle.
 - Plantas marcadas como:
 - ENFERMO_CONFIRMADO
 - EN_OBSERVACIÓN
 - DESCARTADO.

5. Políticas/umbrales:

- Probabilidad mínima para marcar como sospechoso.
- Probabilidad mínima para marcar como enfermo confirmado.
- Reglas para decidir cuándo generar alerta a humano.
- Reglas para replanear inspecciones en segmentos bloqueados/no inspeccionados.

Deseos (Desires)

Objetivos de alto nivel que guían el comportamiento del Patólogo:

1. D1: Garantizar cobertura completa o suficiente del invernadero.

- Minimizar segmentos sin inspeccionar.
- 2. D2: Detectar tempranamente posibles infecciones.
 - Identificar frutos/plants sospechosos lo antes posible.
- 3. D3: Reducir falsos negativos críticos.
 - Evitar dejar pasar una planta infectada como sana.
- 4. D4: Minimizar impacto productivo.
 - Solo activar medidas fuertes (alerta/aislamiento) cuando haya evidencia suficiente.
 - Aprovechar los datos para identificar zonas seguras para cosecha.
- 5. D5: Optimizar uso de recursos.
 - No desperdiciar ScoutBots ni PickBots.
 - Reducir recorridos innecesarios.

Intenciones (Intentions)

Las intenciones son planes concretos que el Patólogo adopta basándose en sus creencias y deseos.

Principales intenciones:

1. I1: Planificación de misiones de ScoutBots
 - Plan:
 - Dividir los segmentos del grafo en zonas balanceadas.

- Generar listas ordenadas de segmentos para cada ScoutBot.

- Asignar misiones desde la Base.

- Se mantiene mientras haya áreas por inspeccionar.

2. I2: Análisis Etapa 1 (screening masivo)

- Plan:

- Procesar todas las imágenes recibidas en lote.

- Detectar frutos y clasificarlos (SANO/SOSPECHOSO/CRÍTICO).

- Actualizar mapa de salud.

- Construir lista de objetivos sospechosos.

- Se activa tras cada descarga de ScoutBots.

3. I3: Gestión de sospechosos con PickBot Enfermero

- Plan:

- Tomar la lista de frutos sospechosos.

- Ordenarlos por prioridad (nivel de sospecha, cercanía, densidad).

- Asignar al PickBot Enfermero una ruta eficiente.

- Esperar muestras y lanzar Etapa 2.

- Iterativo y condicionado por disponibilidad del PickBot.

4. I4: Análisis Etapa 2 (confirmación)

- Plan:
 - Evaluar frutos traídos por el PickBot Enfermero con cámara ultra detallada.
 - Si se confirma infección:
 - Marcar planta y zona como infectadas.
 - Generar alerta al humano.
 - Si se descarta:
 - Actualizar estado a “sano” o “en observación”.
- Busca cumplir D2 y D3.

5. I5: Reinspección / Replanificación

- Plan:
 - Detectar segmentos NO_INSPECCIONADO o BLOQUEADO.
 - Preparar nuevas misiones para próximos ciclos.
- Mantiene la cobertura (D1).

6. I6: Activación del PickBot de sanos (opcional)

- Plan:
 - Identificar zonas marcadas como seguras.
 - Asignar rutas de cosecha a PickBot sanos.

- Alineado con D4 y D5.

Cada intención puede ser reevaluada si cambian las creencias:

- Si aparece un nuevo brote grave, la prioridad puede pasar a confirmación rápida y alertas.

1.3.3. Integración Híbrida

Resumiendo cómo se combinan componentes:

1. La capa reactiva:

- Escucha eventos (llegan datos, llegan muestras, cambios de disponibilidad).
- Actualiza creencias inmediatas.
- Dispara o ajusta planes.

2. La capa deliberativa BDI:

- Sobre esas creencias:
 - Decide qué intención activar o modificar:
 - ¿Nuevo ciclo de scouts?
 - ¿Envío del enfermero?
 - ¿Generar alerta?
- Ejerce control de alto nivel sobre ScoutBots y PickBots.

En resumen, el Patólogo queda definido como:

- Agente híbrido (reactivo + BDI).
- Centro de coordinación.
- Punto de integración de toda la información del invernadero.
- Responsable de las decisiones críticas del sistema.

2. Agente ScoutBot (Agente Explorador)

2.1. Identificación del agente y relaciones

Nombre del agente: ScoutBot (Agente Explorador)

Tipo: Agente móvil de exploración y captura de información.

Rol en el sistema: Actúa como sensor móvil del invernadero; recorre segmentos asignados, captura imágenes de plantas y frutos, y reporta la información al Patólogo Digital al finalizar su misión.

2.1.1. Responsabilidades clave

- Ejecutar misiones definidas por el Patólogo Digital:
 - Recibir una lista ordenada de segmentos (nodos del grafo) a inspeccionar.
 - Desplazarse físicamente (en la simulación) a través de esos segmentos.
- Capturar información visual:
 - En cada planta relevante dentro del segmento:

- Tomar de 1 a 3 imágenes:
 - Vista general de la planta.
 - Vista donde se observen frutos.
 - (Opcional) vista adicional según configuración definida por el Patólogo.
 - Asociar a cada imagen:
 - ID de segmento.
 - ID de planta.
 - Timestamp.
- Manejo local de obstáculos:
 - Si detecta un obstáculo físico que impide inspeccionar un segmento:
 - Marca dicho segmento como “no inspeccionado”.
 - Continúa con el siguiente segmento de su lista.
- Gestión de datos:
 - Almacenar temporalmente en memoria local:
 - Imágenes capturadas.
 - Bitácora de segmentos inspeccionados y no inspeccionados.
 - Regresar a la Base central al finalizar la misión.
 - Transferir en lote toda la información al Patólogo Digital.

2.1.2. Relaciones con otros agentes y elementos

- **Con el Patólogo Digital (PathologistAgent)**

- Recibe:

- Misiones iniciales:

- Lista de segmentos a recorrer.

- Parámetros de captura de imágenes.

- Reglas de comportamiento ante bloqueo.

- Puntos de retorno a Base.

- Envía (al regresar a Base):

- Conjunto de imágenes capturadas.

- Bitácora de misión:

- Segmentos inspeccionados con éxito.

- Segmentos marcados como no inspeccionados por bloqueo.

- Relación conceptual:

- El ScoutBot es un ejecutor reactivo de la estrategia definida por el Patólogo; no interpreta la información, solo la captura y la entrega.

- **Con PickBot Enfermero y PickBot de sanos**

- No existe coordinación directa.
- La interacción es indirecta, mediada por el Patólogo Digital, quien decide las acciones de los PickBots en función de los datos proporcionados por los ScoutBots.
- **Con el medio de comunicación y la Base central**
 - Utiliza la Base como punto de sincronización:
 - Inicio de misión.
 - Fin de misión y descarga de datos.

2.2. Tipo de arquitectura del ScoutBot

El ScoutBot se define como un agente reactivo.

Justificación:

- No mantiene un modelo interno complejo del invernadero.
- No realiza diagnóstico ni planificación global.
- Responde a estímulos y reglas simples:
 - Seguir la lista de segmentos recibida.
 - Capturar imágenes según parámetros fijados.
 - Saltar segmentos bloqueados.
 - Regresar a Base al completar su misión.

- Toda la “inteligencia pesada” (análisis, priorización, diagnóstico, replanificación global) reside en el Patólogo Digital.

2.3. Componentes arquitectónicos del ScoutBot

2.3.1. Capa de Percepción

- Sensores simulados (a nivel conceptual):
 - Posición dentro del invernadero (conocer el segmento actual).
 - Detección de presencia de planta en la posición indicada.
 - Detección de obstáculo en el camino (maquinaria, personas, bloqueo temporal).
 - Confirmación de llegada a Base central.
- Entrada de configuración:
 - Misión recibida desde el Patólogo:
 - Lista ordenada de segmentos.
 - Parámetros de captura (número de fotos por planta).

2.3.2. Capa de Control Reactivo

Basada en reglas simples de estímulo-respuesta:

- Regla 1: Inicio de misión
 - Si se recibe una misión y el ScoutBot está en Base:
 - Cargar lista de segmentos.

- Cambiar estado a “en misión”.
 - Ir al primer segmento asignado.
- Regla 2: Llegada a segmento asignado
 - Si se llega a un segmento:
 - Recorrer la fila con un patrón predefinido.
 - Para cada planta marcada como relevante:
 - Capturar las imágenes configuradas.
 - Guardar imagen + metadata en memoria local.
- Regla 3: Obstáculo detectado en segmento
 - Si durante el recorrido de un segmento se detecta un obstáculo que impide la inspección adecuada:
 - Marcar el segmento como “no inspeccionado”.
 - Registrar el evento en la bitácora interna.
 - Continuar con el siguiente segmento de la lista.
 - No recalcular rutas globales; sólo seguir el orden definido.
- Regla 4: Fin de lista de segmentos
 - Si no hay más segmentos pendientes en la misión:
 - Cambiar estado a “regreso a Base”.
 - Navegar desde la posición actual hasta la Base central.

- Regla 5: Llegada a Base
 - Si llega a la Base y está en estado “regreso a Base”:
 - Iniciar transferencia de datos al Patólogo:
 - Imágenes capturadas.
 - Bitácora de segmentos inspeccionados/no inspeccionados.
 - Cambiar estado a “disponible”.
 - Esperar nueva misión.

2.3.3. Capa de Acción

- Acciones principales:
 - MOVE_TO(segmento): desplazarse al segmento indicado.
 - SCAN_PLANT(planta_id): capturar imágenes según parámetros.
 - MARK_UNINSPECTED(segmento): marcar un segmento como no inspeccionado por bloqueo.
 - RETURN_TO_BASE(): regresar a la Base central.
 - UPLOAD_DATA(): enviar al Patólogo las imágenes y la bitácora.

2.3.4. Características clave del diseño reactivo

1. No realiza inferencias sobre salud de plantas o frutos.
2. No gestiona prioridades globales ni redistribuye segmentos.

3. No coordina misiones con otros ScoutBots directamente.
4. Su comportamiento completo se puede describir como:
 - a. “Seguir misión → Capturar → Registrar → Regresar → Descargar”.

En conjunto, los ScoutBots forman una capa distribuida de percepción simple, controlada y explotada inteligentemente por el agente Patólogo Digital.

3. PickBot Enfermero (Agente Recolector Especializado)

3.1. Identificación del agente y relaciones

Nombre del agente: PickBot Enfermero

Tipo: Agente móvil recolector especializado para muestras sospechosas.

Rol en el sistema: Ejecutar misiones de recolección focalizada de frutos sospechosos identificados por el Patólogo Digital, trasladándolos a la zona de análisis para su evaluación detallada.

3.1.1. Responsabilidades clave

- Ejecutar órdenes emitidas por el Patólogo Digital:
 - Recibir una lista de objetivos sospechosos:
 - Plantas y frutos específicos.
 - Ubicaciones expresadas en términos de segmentos del grafo.
 - Desplazarse hasta dichos objetivos siguiendo rutas definidas o derivadas del modelo del invernadero.

- Recolección especializada:
 - En cada planta objetivo:
 - Localizar el fruto sospechoso indicado.
 - Simular la recolección del fruto sospechoso.
 - Asegurar su transporte controlado hacia la Base central para evitar contaminación de otras zonas.
- Entrega y confirmación:
 - Depositar los frutos sospechosos en la zona de análisis del Patólogo Digital.
 - Reportar la lista de objetivos atendidos y aquellos que no pudieron ser recolectados (por bloqueo u otra causa).
- Gestión básica de estado:
 - Indicar al Patólogo:
 - Inicio de misión.
 - Finalización de misión.
 - Disponibilidad para nuevas tareas.

3.1.2. Relaciones con otros agentes y elementos

- **Con el Patólogo Digital (PathologistAgent)**
 - Recibe:

- Misiones de recolección:
 - Lista de frutos sospechosos (planta, segmento, identificador).
 - Orden sugerido o ruta (o puntos a visitar sobre el grafo).
- Instrucciones operativas específicas (por ejemplo: prioridad de ciertos casos críticos).

○ Envía:

- Confirmaciones de recolección:
 - Fruto_id / planta_id recolectado.
 - Segmentos atendidos.
- Notificaciones de imposibilidad de acceso:
 - Bloqueos o problemas encontrados.
- Señales de “misión completada” y “disponible”.

○ Relación conceptual:

- El PickBot Enfermero es un ejecutor especializado y reactivo de las decisiones del Patólogo, enfocado en la etapa de verificación y diagnóstico de alta precisión.

● **Con ScoutBots**

- No existe comunicación directa.

- La relación es indirecta:
 - Los ScoutBots generan los datos de sospecha.
 - El Patólogo utiliza esos datos para asignar misiones al PickBot Enfermero.
- **Con PickBot de sanos**
 - No existe coordinación directa.
 - Ambos actúan bajo las órdenes del Patólogo para mantener separadas las cadenas de manejo de frutos sanos y sospechosos.
- **Con la Base central**
 - Utiliza la Base como:
 - Punto de partida y regreso.
 - Punto de entrega de muestras sospechosas para análisis detallado.
 - La comunicación y transferencia de información se realiza a través de los mecanismos definidos por el Patólogo y el sistema.

3.2. Tipo de arquitectura del PickBot Enfermero

El PickBot Enfermero se define como un agente reactivo.

Justificación:

- No mantiene un modelo completo del invernadero.

- No realiza diagnóstico ni toma decisiones estratégicas.
- Su comportamiento se basa en seguir misiones, reaccionar a objetivos e informar resultados.
- Aplica reglas simples:
 - Ir al objetivo indicado.
 - Recolectar fruto sospechoso.
 - Regresar a Base y entregar.
 - Reportar disponibilidad o incidencias.

3.3. Componentes arquitectónicos del PickBot Enfermero

3.3.1. Capa de Percepción

- Sensores simulados (conceptuales):
 - Posición actual dentro del invernadero.
 - Detección de llegada a un segmento objetivo.
 - Detección de la planta objetivo dentro del segmento.
 - Confirmación de interacción con el fruto sospechoso (recolección).
 - Detección de obstáculos que impidan el acceso a un objetivo.
 - Confirmación de llegada a la Base central.
- Entrada de configuración:
 - Lista de objetivos sospechosos enviada por el Patólogo:

- Segmento, planta, fruto_id.

- Parámetros de manipulación segura definidos por el sistema.

3.3.2. Capa de Control Reactivo

Conjunto de reglas de estímulo–respuesta que definen su comportamiento:

- Regla 1: Inicio de misión
 - Si el PickBot Enfermero está en Base y recibe una lista de objetivos:
 - Cargar lista de objetivos.
 - Cambiar estado a “en misión”.
 - Ir al primer objetivo de la lista.
- Regla 2: Llegada a segmento objetivo
 - Si se alcanza el segmento asociado a un objetivo:
 - Localizar la planta indicada.
 - Posicionarse junto al fruto sospechoso.
- Regla 3: Recolección de fruto sospechoso
 - Si el fruto sospechoso es accesible:
 - Simular PICK_FRUIT(fruto_id).
 - Marcar objetivo como “recolectado”.
 - Almacenar el fruto en el contenedor seguro del PickBot.

- Si el acceso está bloqueado:
 - Marcar objetivo como “no recolectado por bloqueo”.
 - Registrar el incidente en la bitácora interna.
- Regla 4: Siguiendo objetivo
 - Si existen objetivos pendientes:
 - Seleccionar el siguiente objetivo de la lista.
 - Desplazarse hacia su segmento correspondiente.
 - Si no hay más objetivos pendientes:
 - Cambiar estado a “regreso a Base”.
 - Ejecutar RETURN_TO_BASE().
- Regla 5: Llegada a Base
 - Si llega a la Base con estado “regreso a Base”:
 - Ejecutar DELIVER_SAMPLES():
 - Entregar todos los frutos sospechosos recolectados al área de análisis del Patólogo.
 - Enviar reporte:
 - Lista de objetivos recolectados.
 - Lista de objetivos no atendidos y motivos.
 - Cambiar estado a “disponible”.

- Esperar nuevas instrucciones.

3.3.3. Capa de Acción

Acciones principales del PickBot Enfermero:

- MOVE_TO(segmento): desplazarse hacia el segmento objetivo.
- APPROACH_PLANT(planta_id): posicionarse junto a la planta indicada.
- PICK_FRUIT(fruto_id): simular la recolección del fruto sospechoso.
- MARK_UNREACHABLE(objetivo_id): registrar que el objetivo no pudo ser atendido.
- RETURN_TO_BASE(): regresar a la Base central al finalizar la misión.
- DELIVER_SAMPLES(): transferir las muestras sospechosas al módulo de análisis del Patólogo.
- REPORT_STATUS(): informar al Patólogo el resultado de la misión y la disponibilidad actual.

3.3.4. Características clave del diseño reactivo

- Operación basada en misiones definidas externamente por el Patólogo Digital.
- Lógica centrada en cumplir objetivos específicos sin modificar la estrategia global.
- No ejecuta análisis de salud, ni reorganiza prioridades de manera autónoma.

- Su función principal es cerrar el ciclo de verificación: transportar información física (frutos sospechosos) desde el campo hasta el punto de diagnóstico detallado.

Con este diseño, el PickBot Enfermero complementa el trabajo de los ScoutBots y del Patólogo Digital, asegurando una segunda etapa de revisión precisa y controlada para la detección del Tomato brown rugose fruit virus dentro del entorno simulado.

4. Agente PickBot de Sanos (Agente Recolector de Producción Segura)

4.1. Identificación del agente y relaciones

Nombre del agente: PickBot de Sanos

Tipo: Agente móvil recolector de frutos/planta en zonas confirmadas como seguras.

Rol en el sistema: Ejecutar misiones de cosecha en áreas clasificadas como no infectadas por el Patólogo Digital, manteniendo separado el flujo de producción sana del flujo de muestras sospechosas o infectadas.

4.1.1. Responsabilidades clave

- Ejecutar órdenes emitidas por el Patólogo Digital:
 - Recibir listas de segmentos, plantas o zonas marcadas como seguras.
 - Desplazarse a dichas zonas siguiendo rutas basadas en el grafo del invernadero.
- Recolección de producción segura:

- En cada planta o zona segura asignada:
 - Simular la recolección de frutos o la cosecha correspondiente.
 - Garantizar que solo se manipule producción previamente confirmada como sana.
- Entrega y confirmación:
 - Transportar la producción sana al punto designado (Base central/área de acopio).
 - Reportar al Patólogo Digital:
 - Zonas atendidas.
 - Cantidad o estado de la cosecha (a nivel conceptual).
- Gestión básica de estado:
 - Indicar inicio de misión, finalización y disponibilidad para nuevas tareas.

4.1.2. Relaciones con otros agentes y elementos

- **Con el Patólogo Digital (PathologistAgent)**
 - Recibe:
 - Misiones de cosecha:
 - Listas de segmentos o plantas marcadas como seguras.
 - Orden o prioridades sugeridas.

- Instrucciones operativas:
 - Ventanas de tiempo, restricciones, etc. (si se modelan).

- Envía:

- Confirmaciones de recolección en zonas seguras.
- Notificación de zonas inaccesibles por bloqueo u otra causa.
- Indicaciones de finalización de misión y estado de disponibilidad.

- Relación conceptual:

- El PickBot de Sanos es un ejecutor directo de la estrategia de aprovechamiento productivo definida por el Patólogo, sin intervenir en diagnóstico ni toma de decisiones.

- **Con PickBot Enfermero**

- No existe comunicación directa.
- La separación física y funcional entre ambos agentes reduce la probabilidad de contaminación cruzada:
 - PickBot de Sanos opera solo en áreas seguras.
 - PickBot Enfermero manipula únicamente frutos sospechosos.

- **Con ScoutBots**

- Relación indirecta:

- Las decisiones sobre qué zonas son seguras se basan en la información capturada por los ScoutBots y procesada por el Patólogo.

- **Con la Base central**

- Utiliza la Base como:
 - Punto de inicio de misión.
 - Punto de entrega de producción sana.
 - Punto de sincronización con el Patólogo Digital.

4.2. Tipo de arquitectura del PickBot de Sanos

El PickBot de Sanos se define como un agente reactivo.

Justificación:

- No mantiene un modelo completo del invernadero.
- No realiza análisis de estado sanitario.
- Su comportamiento está guiado por:
 - Misiones concretas emitidas por el Patólogo.
 - Reglas simples de navegación y recolección.
- La lógica estratégica (qué cosechar, cuándo, en qué orden global) reside completamente en el Patólogo Digital.

4.3. Componentes arquitectónicos del PickBot de Sanos

4.3.1. Capa de Percepción

- Sensores simulados:
 - Posición actual dentro del invernadero.
 - Detección de llegada a un segmento o planta objetivo.
 - Confirmación de recolección de frutos/planta.
 - Detección de obstáculos que impidan el acceso a una zona segura.
 - Confirmación de llegada a la Base o punto de acopio.
- Entrada de configuración:
 - Lista de zonas/segmentos/planta seguras enviada por el Patólogo.
 - Parámetros de operación:
 - Forma de recolección.
 - Cantidad máxima por misión (si se modela).

4.3.2. Capa de Control Reactivo

Reglas de estímulo–respuesta que definen su comportamiento:

- Regla 1: Inicio de misión
 - Si el PickBot de Sanos está en Base y recibe una lista de zonas seguras:
 - Cargar la lista de objetivos.

- Cambiar estado a “en misión”.
 - Ir al primer segmento o planta de la lista.
- Regla 2: Llegada a zona segura
 - Si se alcanza un segmento o planta marcada como segura:
 - Ejecutar HARVEST_SAFE():
 - Simular la recolección de frutos/planta definidos como sanos.
 - Registrar la zona como atendida.
- Regla 3: Obstáculo en zona segura
 - Si la zona segura no es accesible por bloqueo:
 - Marcar objetivo como “no atendido por bloqueo”.
 - Registrar el incidente en la bitácora.
 - Continuar con el siguiente objetivo.
- Regla 4: Siguiente objetivo
 - Si existen objetivos pendientes en la misión:
 - Seleccionar el siguiente objetivo de la lista.
 - Desplazarse al segmento/planta correspondiente.
 - Si no hay más objetivos:
 - Cambiar estado a “regreso a Base”.

- Ejecutar RETURN_TO_BASE().
- Regla 5: Llegada a Base
 - Si llega a la Base con estado “regreso a Base”:
 - Ejecutar DELIVER_HARVEST():
 - Entregar la producción sana recogida.
 - Enviar reporte al Patólogo Digital:
 - Zonas atendidas.
 - Zonas no atendidas y motivos.
 - Cambiar estado a “disponible”.

4.3.3. Capa de Acción

Acciones principales del PickBot de Sanos:

- MOVE_TO(segmento): desplazarse al segmento objetivo seguro.
- APPROACH_PLANT(planta_id): posicionarse junto a la planta seleccionada.
- HARVEST_SAFE(): simular recolección de frutos/planta en zona segura.
- MARK_UNREACHABLE(objetivo_id): registrar zona segura no atendida por bloqueo.
- RETURN_TO_BASE(): regresar al punto de acopio/Base.
- DELIVER_HARVEST(): entregar la producción sana recolectada.
- REPORT_STATUS(): informar estado de misión y disponibilidad.

4.3.4. Características clave del diseño reactivo

- Opera exclusivamente sobre información validada por el Patólogo Digital.
- No modifica rutas globales ni redefine qué es seguro; solo ejecuta.
- Ayuda a materializar el objetivo de maximizar el aprovechamiento de producción sana, manteniendo una separación clara entre:
 - Flujos de cosecha segura.
 - Flujos de diagnóstico y manejo de sospechosos.

Con este diseño, el PickBot de Sanos complementa al Patólogo y al PickBot Enfermero, cerrando el ciclo operativo de detección, verificación y aprovechamiento seguro dentro del invernadero simulado.

Plan de trabajo

1. Organización general del plan

Roles:

- **Integrante 1** – Clay Gutiérrez: Coordinación general, diseño de arquitectura multiagente, documentación.
- **Integrante 2** - Emilio Hernandez: Líder técnico Unity (escena 3D, navegación básica).
- **Integrante 3** - Alfredo Carmona: Lógica de agentes en C# (ScoutBots).
- **Integrante 4** - Alejandro Gutiérrez: Lógica de agentes en C# (PickBots).

- **Integrante 5** - Dylan Pereyra: Soporte en Patólogo Digital (BDI, flujos, documentación técnica).
- **Integrante 6** - Luis Díaz: Pruebas, métricas, pulido visual y consistencia gráfica.

2. Plan de trabajo por semanas

Semana 1 – Definición y base colaborativa

Objetivos principales:

- Formalizar la comprensión del reto y del contexto (ToBRFV, necesidad de detección temprana).
- Definir agentes, roles y arquitecturas (Patólogo híbrido, ScoutBots, PickBots).
- Establecer herramientas colaborativas.

Actividades:

1. Definición final de la propuesta conceptual:
 - Reto, agentes, relaciones, arquitecturas.
2. Redacción del documento base:
 - Secciones: descripción del reto, agentes, arquitecturas, flujo general.
3. Creación de herramientas colaborativas:
 - Repositorio en GitHub para documentación y código.
 - Canal oficial de comunicación (Discord).

4. Definir roles internos y responsables por componente (Unity, agentes, docs).

Responsables (sugerido):

- Integrante 1: coordinación del documento y arquitectura multiagente.
- Integrante 2: configuración inicial de proyecto Unity.
- Integrante 6: creación y organización del repositorio + lineamientos de commits.
- Todo el equipo: acordar canal de comunicación y reglas de trabajo.

Intervalo de esfuerzo estimado (por persona)

4–6 horas en la semana.

Semana 2 – Modelo del invernadero y agentes básicos (Primera revisión)

Objetivos principales:

- Contar con la base visual y lógica mínima del sistema.
- Tener lista la propuesta formal documentada para la primera revisión.

Actividades planeadas para la primera revisión:

1. Modelo del invernadero como grafo de segmentos:
 - Definir nodos (segmentos + Base) y aristas (pasillos).
 - Representación conceptual en el documento + boceto en Unity.
2. Prototipo en Unity:
 - Escena con invernadero simplificado.

- Un ScoutBot con movimiento básico siguiendo segmentos predefinidos.

3. Definición formal de agentes:

- Patólogo Digital (híbrido).
- ScoutBots, PickBot Enfermero, PickBot de sanos (reactivos).
- Componentes arquitectónicos documentados (como ya se definió).

4. Actualización del plan de trabajo en el documento.

Responsables y esfuerzo estimado:

- Modelo de grafo + documentación:
 - Integrante 1 + Integrante 5 – 3–4 horas.
- Escena base Unity + movimiento simple ScoutBot:
 - Integrante 2 + Integrante 3 – 5–7 horas.
- Revisión y pulido de documento para primera revisión:
 - Integrante 1 + Integrante 6 – 3–4 horas.

Resultado esperado al final de Semana 2 (primera revisión):

- Documento formal coherente y completo en:
 - Descripción del reto.
 - Agentes y arquitecturas.
- Prototipo inicial:

- Invernadero básico.
- ScoutBot recorriendo segmentos asignados.

Semana 3 – Integración del Patólogo y flujo ScoutBot → Patólogo

Objetivos principales:

- Implementar el ciclo de misión del ScoutBot.
- Integrar el Patólogo como coordinador lógico (sin toda la complejidad aún).

Actividades:

1. Implementar lógica de misión del ScoutBot:
 - Recibir lista de segmentos.
 - Recorrer, simular captura de imágenes, registrar bitácora.
 - Regresar a Base.
2. Implementar módulo básico del Patólogo:
 - Asignar misiones a ScoutBots.
 - Recibir datos al regreso (simulados).
 - Actualizar estado de segmentos (inspeccionado / no inspeccionado).
3. Documentar el flujo completo:
 - Desde planificación inicial hasta descarga de datos.

Responsables:

- ScoutBot misión completa:
 - Integrante 3 – 5–7 horas.
- Patólogo (coordinación básica + recepción de datos):
 - Integrante 1 + Integrante 5 – 5–7 horas.
- Pruebas y documentación:
 - Integrante 6 – 3–4 horas.

Actividades pendientes al cierre de Semana 3 (esperado):

- Implementar screening masivo (clasificación SANO/SOSPECHOSO/CRÍTICO simulada).
- Implementar PickBot Enfermero y PickBot de sanos.

Semana 4 – Screening, PickBots y pipeline completo**Objetivos principales:**

- Completar el flujo técnico de detección en dos etapas.
- Integrar PickBot Enfermero y PickBot de sanos en la simulación.

Actividades:

1. Análisis Etapa 1 en el Patólogo:
 - Simular procesamiento de “imágenes”:
 - Asignar estados SANO/SOSPECHOSO/CRÍTICO.

- Generar lista de objetivos sospechosos.
2. Implementar PickBot Enfermero:
- Recibir lista de objetivos.
 - Recorrer segmentos, “recolectar” frutos sospechosos.
 - Regresar a Base y notificar.
3. Implementar PickBot de sanos (opcional pero recomendado):
- Recibir zonas seguras.
 - Simular cosecha en esas zonas.
4. Integrar Etapa 2:
- Patólogo recibe muestras del PickBot Enfermero.
 - Simula análisis detallado.
 - Marca plantas como enfermas confirmadas o descartadas.
 - Genera alertas.

Responsables:

- Análisis Etapa 1 + Etapa 2 en Patólogo:
 - Integrante 1 + Integrante 5 – 5–7 horas.
- Implementación PickBot Enfermero:
 - Integrante 4 – 5–7 horas.

- Implementación PickBot de sanos:
 - Integrante 2 + Integrante 4 – 4–6 horas.
- Integración final y pruebas:
 - Integrante 6 + todo el equipo – 4–6 horas.

Actividades pendientes al cierre de Semana 4 (esperado):

- Refinar comportamiento.
- Mejorar visualización, métricas y presentación final.

Semana 5 – Refinamiento, validación y presentación final

Objetivos principales:

- Pulir simulación, documentación y coherencia gráfica.
- Preparar la presentación final del reto.

Actividades:

1. Refinamiento técnico:

- Ajustar comportamientos de agentes.
- Revisar consistencia entre arquitectura teórica y comportamiento en Unity.

2. Validación del flujo:

- Verificar:
 - Misiones de ScoutBots.

- Screening del Patólogo.
- Acciones del PickBot Enfermero y PickBot de sanos.

3. Documentación final:

- Actualizar documento con:
 - Plan de trabajo final.
 - Aprendizaje adquirido.
 - Capturas/diagramas del sistema.

4. Preparación de presentación:

- Diapositivas.
- Guión de explicación del flujo y decisiones de diseño.

Responsables:

- Refinamiento técnico:
 - Integrante 2, 3, 4 – 6–8 horas.
- Documentación final:
 - Integrante 1 + Integrante 5 + Integrante 6 – 5–7 horas.
- Presentación:
 - Todo el equipo.

Aprendizaje adquirido

Al corte de esta propuesta formal, el equipo identifica los siguientes aprendizajes:

1. Comprensión de sistemas multiagentes aplicados a un problema real:

- Entender cómo descomponer el reto agrícola (detección temprana de infección) en agentes con roles claros:
 - Sensores móviles (ScoutBots),
 - Recolectores especializados (PickBots),
 - Centro inteligente de decisión (Patólogo Digital).

2. Diferenciación entre Digital Shadow, Digital Twin y Digital Environment:

- Reconocer por qué, en esta etapa, un Digital Environment (invernadero simulado) es la elección adecuada:
 - No se cuenta con datos reales ni integración física bidireccional.
 - Se requiere un entorno controlado para experimentar con estrategias de agentes.

3. Diseño de arquitecturas de agentes:

- Identificar y modelar:
 - Agentes reactivos (ScoutBots y PickBots).
 - Un agente híbrido con capa reactiva + BDI (Patólogo Digital).

- Entender la importancia de separar:

- Percepción,
- Decisión,
- Ejecución.

4. Importancia de centralizar la inteligencia pesada:

- Valorar el enfoque de “robots tontos + cerebro central”:
 - Dispositivos simples en campo.
 - Procesamiento intensivo concentrado en el Patólogo.
 - Facilidad para escalar, actualizar modelos y controlar calidad de decisiones.

5. Trabajo colaborativo estructurado:

- Reconocer la necesidad de:
 - Un repositorio central (GitHub) con versión controlada.
 - Un canal de comunicación estable y activo (Discord).
 - Un plan de trabajo con responsables y fechas claras.

6. Enfoque iterativo e incremental:

- Comprender que el sistema se construye en capas:
 - Primero el modelo conceptual sólido.

- Luego la simulación básica.
- Luego la integración de agentes y lógica de decisiones.
- Evitar sobrecargar la primera versión con complejidad innecesaria.

Este bloque de trabajo se utilizará como base; en las siguientes entregas el equipo actualizará el plan y el apartado de aprendizaje adquirido conforme avance la implementación y las pruebas del sistema.