



UNIVERSIDAD  
DE LA REPÚBLICA  
URUGUAY



FACULTAD DE  
INGENIERÍA

# Inspección de cultivos con drones para la detección de malezas

Informe de Proyecto de Grado presentado por

Matias Dolgay, Fabricio Trinidad y Romina Sosa

en cumplimiento parcial de los requerimientos para la graduación de la carrera de Ingeniería en  
Computación de Facultad de Ingeniería de la Universidad de la República

Supervisor

Facundo Benavides

Montevideo, 5 de septiembre de 2025



Inspección de cultivos con drones para la detección de malezas por Matias Dolgay, Fabricio Trinidad y Romina Sosa tiene licencia [CC Atribución 4.0](#).

# Agradecimientos

Quisiéramos expresar el más profundo agradecimiento a nuestro tutor de tesis, Facundo Benavides, quien fue nuestro pilar fundamental durante todo el desarrollo de este proyecto.

# Resumen

El presente proyecto aborda el desarrollo de un sistema autónomo para la inspección de cultivos utilizando vehículos aéreos no tripulados (drones) y procesamiento de imágenes. El objetivo principal es detectar automáticamente las hileras de cultivo y generar trayectorias eficientes para realizar una cobertura fotográfica completa del terreno. El sistema fue diseñado para operar tanto en simulación como en condiciones reales, permitiendo validar su comportamiento y precisión.

El dron se controla mediante la plataforma PX4 (un sistema de control de vuelo de código abierto), en conjunto con ROS2 (Robot Operating System 2, un framework de desarrollo para robots), que facilita la integración de nodos encargados de distintos aspectos del sistema. Estos nodos controlan el dron en modo manual o autónomo, planifican misiones a partir de puntos GPS y toman imágenes durante el vuelo. Posteriormente, las imágenes capturadas son procesadas mediante una herramienta externa llamada OpenDroneMap (ODM), que permite generar ortomosaicos y mapas de alturas georreferenciados del terreno. Sobre este mapa de alturas se aplican técnicas de procesamiento digital de imágenes utilizando Python y OpenCV, incluyendo umbralización, operaciones morfológicas, detección de contornos y análisis estructural. Para representar el patrón de siembra, se emplea un diagrama de Voronoi generalizado que se genera a partir de los contornos de las hileras detectadas. A través de este modelo, se construye un camino de navegación para que el dron pueda realizar un vuelo que recorra las hileras de cultivo de forma ordenada y precisa. La planificación se adapta automáticamente a la disposición del campo y a la distancia entre hileras.

La solución fue validada en un entorno simulado, permitiendo iterar sobre diferentes configuraciones, rutas y estrategias sin riesgos reales. La evaluación del sistema demostró su capacidad para generar rutas adaptadas al terreno y realizar vuelos autónomos exitosos, con imágenes alineadas espacialmente que permiten una inspección visual precisa del cultivo. Además, la solución implementada sirve como base para el desarrollo de un sistema que detecte malezas de forma visual en las imágenes generadas por vuelos autónomos de drones.

**Palabras clave:** Drones, inspección agrícola, vuelo autónomo, planificación de trayectorias, ROS2, PX4, procesamiento de imágenes, Voronoi

# Índice general

<b>1. Introducción</b>	<b>1</b>
<b>2. Revisión de antecedentes</b>	<b>2</b>
2.1. Elección de artículos	2
2.1.1. Criterios de selección	2
2.1.2. Artículos seleccionados	3
2.1.3. Artículos complementarios	3
2.1.4. Respuestas a las preguntas de investigación	4
2.2. Malezas	4
2.2.1. Malezas en Uruguay	4
2.2.2. Control de malezas	5
2.3. Drones	5
2.3.1. Estrategias de navegación	5
2.3.2. Casos de uso de UAV en la industria y la agricultura	6
2.3.3. Uso de drones en la agricultura de precisión	6
<b>3. Desarrollo de la solución</b>	<b>9</b>
3.0.1. Conceptos básicos y asunciones de trabajo	9
3.1. Inspección de cultivos	10
3.1.1. Definición del área de interés	10
3.1.2. Generación del plan de reconocimiento	13
3.1.3. Vuelo de reconocimiento	17
3.1.4. Procesamiento de imágenes y generación del mapa	19
3.1.5. Detección de hileras	21
3.1.6. Vuelo entre hileras	27
3.2. Arquitectura del sistema	30
3.2.1. Diagramas de despliegue	30
3.2.2. Diagrama de casos de uso	32
<b>4. Experimentación</b>	<b>33</b>
4.1. Vuelo de reconocimiento	34
4.1.1. Altitud	34
4.1.2. Recorrido	35
4.1.3. Tiempo de ejecución	35
4.2. Generación del ortomosaico	36
4.3. Generación del recorrido entre hileras	37
4.3.1. Cultivos horizontales	37
4.3.2. Cultivos oblicuos	38
4.4. Vuelo entre hileras	38
4.5. Ejecuciones con viento	39
<b>5. Conclusiones y Trabajo Futuro</b>	<b>42</b>
5.1. Contribuciones	42
5.2. Conclusiones	42
5.3. Dificultades encontradas	43
5.4. Trabajos a futuro	43
A.1. ROS 2	46

# Capítulo 1

## Introducción

En el marco de la agricultura moderna, la inspección eficiente de cultivos es esencial para garantizar una producción sostenible, rentable y de alta calidad. Contar con información precisa sobre el estado del terreno, el desarrollo de las plantas, su distribución y posibles anomalías permite a los agricultores tomar decisiones oportunas y fundamentadas. Sin embargo, en grandes extensiones de terreno, esta tarea representa un gran desafío, ya que los métodos tradicionales, basados en recorridos manuales, requieren una gran inversión de tiempo, esfuerzo y recursos, además de estar sujetos a errores humanos (Abdulsalam et al., 2023).

El principal problema son las malezas, plantas no deseadas que crecen entre los cultivos y compiten por recursos esenciales como agua, luz solar y nutrientes. Esta competencia tiene un impacto significativo tanto en la productividad agrícola como en la calidad del producto final. El control de malezas presenta múltiples dificultades operativas. Requiere una mano de obra intensiva, ya que se necesitan realizar monitoreos manuales frecuentes.

En el contexto de un proyecto de investigación más amplio orientado a la detección automática de malezas, el presente trabajo busca desarrollar un sistema de planificación y de vuelos autónomos que permita la captura de imágenes aéreas de los cultivos.

Esto es realizado mediante el uso de vehículos aéreos no tripulados (UAVs), comúnmente conocidos como drones. Estas aeronaves presentan ventajas significativas frente a otros métodos de monitoreo, al ofrecer una solución accesible y eficaz para la captura de imágenes aéreas. Además, permiten el acceso a zonas de difícil o peligroso acceso para los agricultores.

El objetivo principal de este trabajo es desarrollar un sistema automatizado de planificación de trayectorias que genere la infraestructura necesaria para facilitar la posterior implementación de algoritmos de detección automática de malezas. Específicamente, se busca crear un sistema que planifique trayectorias de vuelo óptimas entre las hileras de cultivos frutales, donde se logre capturar imágenes del suelo con la calidad suficiente para detectar malezas.

Para lograr este objetivo, diseñamos un sistema que se basa en varias etapas. La primera etapa consiste en un recorrido inicial realizado manualmente por un operador, el cual se utiliza como referencia para generar una ruta que permita una primera inspección de reconocimiento totalmente autónoma. Esta ruta asegura una cobertura completa del terreno libre de colisiones ni solapamientos, permitiendo así la construcción de un mapa detallado del cultivo.

La siguiente etapa utiliza el mapa generado en la etapa previa para identificar y detectar las hileras de árboles mediante técnicas de procesamiento de imágenes, lo que permite construir trayectorias de vuelo optimizadas entre las hileras en base a la hipótesis de que es mejor volar entre hileras para capturar imágenes de las malezas.

Para evaluar el sistema propuesto se utilizó un simulador de robótica que permite crear entornos virtuales realistas para probar y validar las trayectorias de vuelo generadas en situaciones controladas. Constituyendo el paso previo necesario antes de la implementación en plataformas reales.

Este informe se estructura en cinco capítulos. El Capítulo 1, correspondiente a esta sección, establece los objetivos, plantea la problemática y expone el alcance del trabajo, así como los resultados alcanzados. El Capítulo 2 desarrolla una revisión de antecedentes, abordando artículos relevantes y explorando el uso de drones en la agricultura con énfasis en las estrategias de navegación. El Capítulo 3 detalla el desarrollo técnico del sistema, incluyendo la navegación del dron, la creación y el procesamiento del mapa del cultivo. El Capítulo 4 describe la experimentación realizada para validar el sistema, mientras que, finalmente, el Capítulo 5 concluye con el trabajo señalado y las posibles líneas de trabajo futuro.

# Capítulo 2

## Revisión de antecedentes

A continuación se incluye una breve descripción de las metodologías empleadas en la revisión de estudios previos, con el fin de realizar un análisis más efectivo y eficiente de los artículos relacionados. También se analizan los avances tecnológicos más recientes y los diferentes enfoques de navegación utilizados en la agricultura para el monitoreo con drones, de cultivos a gran escala. Finalmente, se introducen algunos conceptos fundamentales que permiten comprender en mayor profundidad el desarrollo del proyecto.

### 2.1. Elección de artículos

Los artículos seleccionados para esta investigación fueron obtenidos a través del portal Timbó, utilizando la base de datos Scopus y una variante de la metodología de Revisión Sistemática, denominada Revisión Rápida (Pizard et al., 2019), con algunas modificaciones realizadas por el equipo. A diferencia de una revisión sistemática completa, que sigue un proceso exhaustivo, la revisión rápida permite alcanzar el objetivo sin cumplir con todos los pasos convencionales. Este enfoque implica definir previamente una serie de criterios y características para filtrar rápidamente los artículos menos relevantes sin necesidad de una lectura exhaustiva, rescatando las principales ideas que puedan contribuir al desarrollo del proyecto.

Durante el proceso de la revisión, se formularon las siguientes preguntas de investigación que guiaron el análisis:

- RQ1: ¿Qué vehículos aéreos autónomos existen y cómo funcionan?
- RQ2: ¿Existe algún proyecto open source para la inspección de cultivos con ROS 2?
- RQ3: ¿Existe algún proyecto open source para detectar malezas con ROS 2?

En el proceso de búsqueda, se establecieron términos clave, sinónimos y la cadena de búsqueda a utilizar en los motores de búsqueda. Este protocolo aseguró que todos los miembros del equipo aplicaran criterios consistentes durante la recopilación de artículos.

#### 2.1.1. Criterios de selección

La selección de estudios se basó en criterios de inclusión y exclusión, evaluando principalmente las introducciones de los artículos. Para resolver los conflictos, cada miembro leyó el artículo correspondiente y se decidió en conjunto su inclusión o exclusión.

Los criterios de inclusión establecieron que los artículos debían ser publicados a partir de 2016 inclusive, contener información sobre drones o cultivos, incluir datos sobre software de manejo de drones y procesamiento de imágenes y constituir estudios primarios o secundarios. Por otro lado, los criterios de exclusión descartaron aquellos artículos sin texto completo disponible, no disponibles en español o inglés o que no sean de libre acceso.

De la búsqueda estructurada en Scopus se obtuvieron 20 artículos potencialmente relevantes. Luego, se llevó a cabo una revisión manual en la que cada integrante del equipo leyó la introducción de los artículos seleccionados. Tras discutir y evaluar su pertinencia en relación con el tema de estudio, se decidió conservar 9 de ellos para su análisis detallado. Estos artículos finales sirvieron como base para profundizar en el estudio del uso de drones en la agricultura.

### 2.1.2. Artículos seleccionados

Los estudios seleccionados abordan tres áreas principales: detección de malezas, navegación autónoma y aplicaciones agrícolas con UAVs. Entre los trabajos más relevantes se encuentra Abdulsalam et al., 2023, que presenta un enfoque monocular basado en visión para UAVs, utilizando una arquitectura denominada fused-YOLO para detectar múltiples tipos de malezas y estimar su posición relativa. Este sistema está diseñado para que se capturen imágenes desde drones que vuelan en trayectoria elíptica, proporcionando detección precisa y posicionamiento automático.

Bouguettaya et al., 2023 contribuyen con una revisión exhaustiva centrada en técnicas de aprendizaje profundo aplicadas a la detección de enfermedades de plantas usando imágenes aéreas desde UAVs. Analiza algoritmos y su potencial para diagnóstico temprano en agricultura de precisión.

La revisión realizada por Gokool et al., 2024 explora el potencial de la teledetección con UAV en pequeñas explotaciones agrícolas, centrada en el manejo integrado de malezas. Destaca cómo drones equipados con sensores remotos permiten monitorear y mapear la presencia de malezas, facilitando decisiones de manejo más precisas y sostenibles en contextos de agricultura de pequeña escala.

Pranaswi et al., 2024 compara el uso de UAVs equipados contra pulverizadores de mochila en el manejo de malezas para cultivos de trigo, demostrando que la tecnología UAV puede mejorar significativamente el control de malezas y la productividad. Meesaragandla et al., 2024 presentan una revisión integral que examina el uso de UAVs para pulverización de herbicidas y detección de malezas. Indicando una eficiencia significativamente mayor en cobertura, precisión y reducción del impacto ambiental frente a métodos tradicionales.

Complementando los sistemas aéreos, Ortataş et al., 2023 presenta una plataforma autónoma basada en vehículos terrestres no tripulados para cartografía y pulverización en agricultura de precisión. Esta investigación destaca sus capacidades para mapeo automatizado y aplicación precisa de insumos, indicando su relevancia como complemento o alternativa a sistemas aéreos.

El trabajo de Krestenitis et al., 2024 se enfoca en UAVs con sensores activos y planificación de trayectoria adaptativa aplicada a agricultura de precisión. Ofrece métodos para minimizar errores en el monitoreo de cultivos, mejorando la cobertura y calidad del mapeo mediante segmentación semántica.

El estudio presentado en C. Kim et al., 2023 aborda un método económico y eficiente para estimar desplazamientos de sensores en plataformas UAV mediante tie-points entre imágenes adyacentes, evitando la necesidad de estaciones base de control en tierra. Esto mejora la precisión y fluidez de las operaciones fotogramétricas en agricultura de precisión.

Castelar Wembers et al., 2024 presenta un prototipo automatizado para la inspección de palas de aerogeneradores utilizando UAVs equipados con LiDAR. Desarrollado sobre la plataforma DJI M300, aporta un sistema robusto y autónomo para la detección de daños, con alta precisión y eficiencia en contextos exigentes como la energía eólica offshore.

### 2.1.3. Artículos complementarios

Además de los trabajos seleccionados, se revisaron estudios recientes sobre inteligencia artificial aplicada a la clasificación de cultivos, sistemas de navegación autónoma para drones agrícolas y nuevas tendencias en sensores de bajo costo. Si bien estos estudios no fueron elegidos porque, o bien no trataban de lo buscado, tenían pequeños fragmentos que eran útiles para nuestros estudios, como definiciones de malezas o cómo se utilizan en las sociedades.

El informe de Dellepiane et al., 2023 contextualiza la problemática de las malezas como plantas indeseadas que afectan el bienestar y las actividades humanas, describiendo su relación histórica con los sistemas de producción agrícola y ganadera, describe y analiza distintas estrategias de manejo. El documento de Ministerio de Ganadería, 2025 describe cómo las malezas reducen la productividad agrícola y forrajera en Uruguay. Examina propuestas de manejo nacional para contrarrestar esta problemática, enfocándose en los impactos ambientales, económicos y estratégicos en políticas agrarias.

El documento de International Civil Aviation Organization (ICAO), 2011 establece el marco regulatorio para la integración segura de sistemas de aeronaves no tripuladas, incluyendo definiciones operativas y requisitos funcionales.

Istiak et al., 2023 ofrecen una revisión sistemática sobre la adopción de imágenes UAV en la gestión agrícola, destacando la creciente implementación de imágenes aéreas en agricultura de precisión y los beneficios del monitoreo y la planificación. Y. Wang et al., 2023 propone un algoritmo de planificación de ruta para inspecciones de puentes mediante UAVs, considerando condiciones

meteorológicas (viento) que afectan la seguridad de las operaciones. Su objetivo es optimizar trayectorias para maximizar la eficiencia y proteger los componentes costosos del UAV.

El estudio de D. Kim y Moon, 2024 presenta un nuevo modelo integrado de planificación estratégica y operativa para drones, combinando la ubicación óptima de instalaciones y la programación de rutas. Por otra parte, Raptis et al., 2023 presenta CoFly, una plataforma nidykar de código abierto para agricultura de precisión con UAVs. Utiliza tecnologías de IA para análisis automatizado de datos agrícolas.

Por último, el estudio de Ajayi et al., 2023 evalúa el rendimiento del modelo YOLOv5 frente a redes CNN tradicionales para clasificar cultivos y malezas utilizando imágenes aéreas tomadas por UAVs. Se prueba sobre distintos cultivos (espinaca, caña, pimiento), mostrando precisión elevada y viabilidad para implementaciones en tiempo real.

#### 2.1.4. Respuestas a las preguntas de investigación

Con la investigación realizada, se obtuvieron las siguientes respuestas a las preguntas de investigación relacionadas a los artículos elegidos:

Con respecto a RQ1, los estudios analizados revelan diversas configuraciones de UAVs. Krestenitis et al., 2024: Presenta UAVs con sensores activos y planificación de vuelo adaptativa, detallando tanto su configuración de hardware como su integración con algoritmos de segmentación para agricultura de precisión. El trabajo de C. Kim et al., 2023 aborda el problema técnico de estimar desplazamientos de sensores en UAVs, aportando información sobre calibración y configuración de plataformas aéreas, mientras que Castelar Wembers et al., 2024 describe UAVs equipados con LiDAR para inspección automatizada de estructuras (aerogeneradores), demostrando capacidades técnicas aplicables a la agricultura de precisión.

En relación a RQ2, encontramos dos trabajos que presentan enfoques compatibles con plataformas open source como ROS 2: Krestenitis et al., 2024 por su diseño modular para adquisición de imágenes y análisis en tiempo real y C. Kim et al., 2023 debido a que su metodología de estimación de offsets es aplicable en entornos open source para calibración de UAVs en inspección de cultivos.

En cuanto a RQ3, Abdulsalam et al., 2023 propone un enfoque basado en redes neuronales profundas integradas en UAVs para la detección y localización de malezas, que puede adaptarse a sistemas ROS 2. Pranaswi et al., 2024 estudian la combinación de UAVs y pulverizadores manuales, optimizando el manejo de malezas con enfoques transferibles a plataformas open source. Las revisiones de Gokool et al., 2024 y Meesaragandla et al., 2024 ofrecen técnicas de teledetección y pulverización que podrían implementarse en UAVs controlados por ROS 2.

## 2.2. Malezas

Las malezas son plantas que crecen en lugares no deseados y que interfieren con el desarrollo de los cultivos. Estas plantas compiten directamente con las especies de interés por recursos esenciales como agua, nutrientes y espacio, lo que puede afectar negativamente el crecimiento de los cultivos. Además, las malezas pueden contaminar el producto final y, en algunos casos, ser hospederas de plagas y enfermedades que afectan la producción agrícola y ganadera. Por estas razones, la presencia de malezas puede generar consecuencias adversas significativas para la productividad y la calidad en los sistemas productivos. (Dellepiane et al., 2023).

### 2.2.1. Malezas en Uruguay

En Uruguay, el campo natural representa aproximadamente el 86 % del área destinada al pastoreo ganadero (unas 11 millones de hectáreas), lo que lo convierte en la principal fuente de alimento para la producción de carne. Sin embargo, la presencia de malezas en estos campos constituye un problema significativo: no solo reducen la productividad de las especies forrajeras de interés, sino que también disminuyen la superficie útil para el pastoreo y amenazan la biodiversidad de las especies nativas. Aunque se han realizado investigaciones importantes sobre la biología y el control de estas especies (como la campaña nacional contra la “Margarita de Piria” de Investigación Agropecuaria (INIA), 2020), la falta de una estrategia sostenida y coordinada ha contribuido a que las malezas sigan siendo un problema vigente a nivel nacional. (Ministerio de Ganadería, Agricultura y Pesca (MGAP), 2023)

### 2.2.2. Control de malezas

El control de malezas en el campo natural y en sistemas agrícolas representa un desafío importante debido a los elevados costos económicos, el uso intensivo de recursos y la gran demanda de mano de obra especializada. Las estrategias actuales requieren aplicaciones frecuentes de herbicidas, monitoreos constantes y manejo manual que pueden resultar ineficientes y poco sostenibles a largo plazo. Por esta razón, el desarrollo de sistemas automatizados para la detección y manejo de malezas cobra relevancia, ya que podrían optimizar los recursos, reducir costos y minimizar el impacto ambiental, facilitando un control más preciso y eficiente de estas plantas no deseadas. (Dellepiane et al., 2023)

## 2.3. Drones

Un dron o UAV (Unmanned Aerial Vehicle, por sus siglas en inglés) es un vehículo aéreo no tripulado. Consiste en un vehículo aéreo que es controlado de forma remota o autónoma y que no requiere de un piloto a bordo. De acuerdo con la Organización de la Aviación Civil Internacional, un dron se define como un “conjunto de elementos configurables que consisten en: un avión remotamente pilotado, su estación de piloto remoto asociada, los enlaces de comandos y control requeridos y cualquier otro elemento del sistema que se requiera en cualquier punto durante la operación de vuelo”. (International Civil Aviation Organization (ICAO), 2011)

El dron puede variar en tamaño, desde pequeños vehículos de uso recreativo hasta grandes aeronaves utilizadas para aplicaciones comerciales e industriales. Su capacidad para ser controlado remotamente o programado para operar de forma autónoma hace que los drones sean versátiles y adecuados para una amplia gama de aplicaciones, tales como monitoreo ambiental, inspección de infraestructuras, fotografía aérea, agricultura de precisión, y actividades de búsqueda y rescate, entre otros. (Istiak et al., 2023)

### 2.3.1. Estrategias de navegación

Existen diversos métodos de navegación que se ajustan a los requerimientos específicos de cada misión. En el contexto de este proyecto, uno de los aspectos clave es la elección del tipo de navegación que empleará el dron durante la fase inicial de reconocimiento del cultivo. De forma general, pueden identificarse dos enfoques principales de navegación para drones:

#### Navegación adaptativa

Este tipo de navegación se basa en la adaptación dinámica del dron a los cambios en su entorno durante el recorrido del vuelo. Para esto, mientras el dron explora el cultivo, va procesando el camino recorrido y generando el próximo punto a recorrer de manera simultánea. Utilizando una variedad de sensores (como cámaras, LIDAR, o sensores de proximidad), el dron puede detectar obstáculos, cambios en el terreno o variaciones en las condiciones del cultivo, y modificar su ruta en consecuencia. (Krestenitis et al., 2024)

La principal ventaja de este tipo de recorrido es que el usuario queda liberado de las responsabilidades asociadas al control directo del vuelo, ya que el dron posee un nivel de autonomía suficiente para completar el vuelo de forma independiente. Esto permite que incluso usuarios con poca o ninguna experiencia en el manejo de UAVs puedan operar el sistema sin dificultades significativas. Sin embargo, esta autonomía conlleva la necesidad de una mayor capacidad de procesamiento, debido a la ejecución simultánea de diversas tareas, como la planificación de la ruta en tiempo real, la detección de obstáculos y el análisis de datos en tiempo real. Como consecuencia, se requiere un dron de mayor tamaño y capacidad, lo que, si bien mejora la eficiencia, puede presentar ciertos inconvenientes, como una menor agilidad en el vuelo y un aumento de los costos operativos debido a su mayor complejidad y tamaño.

#### Navegación predefinida

Tal como su nombre lo indica, este tipo de navegación se basa en una ruta de vuelo predefinida con anterioridad, la cual el dron tiene que seguir sin modificaciones en tiempo real. Se utiliza especialmente cuando las condiciones del terreno a recorrer son estables y la ruta que debe seguir el dron está claramente definida con anterioridad. En este tipo de navegación, el dron generalmente se guía a través de sistemas de posicionamiento global (GPS), siguiendo las coordenadas preprogramadas. (Ortataş et al., 2023)

A diferencia del método anterior, en este caso el usuario debe participar de manera más activa en la interacción con el dron y el cultivo. Esto se debe a que el usuario tiene que proporcionar datos que permitan crear el mapa de puntos a recorrer, o en su defecto, realizar un recorrido manual del cultivo para generar dicho mapa. Sin embargo, este enfoque tiene la ventaja de que reduce la carga de procesamiento a bordo, ya que la imagen del cultivo puede procesarse posteriormente sin conexión, evitando la necesidad de realizar simultáneamente el procesamiento de imágenes y la generación de puntos en tiempo real. Esta simplificación en el procesamiento contribuye a que el dron consuma menos energía y sea de menor tamaño, lo que lo hace más ágil y más eficiente en términos operativos, ya que puede realizar tareas prolongadas sin sobrecargar sus recursos.

### 2.3.2. Casos de uso de UAV en la industria y la agricultura

Los UAV ofrecen ventajas decisivas en múltiples sectores gracias a su versatilidad, eficiencia y seguridad. A continuación se exponen los principales casos de uso:

**Inspección de infraestructuras:** La extensa red de puentes en China exige inspecciones periódicas que, por métodos tradicionales, implican altos costos, riesgos para el personal y complicaciones logísticas. En *Task-Driven Path Planning for Unmanned Aerial Vehicle-Based Bridge Inspection in Wind Fields* se propone un algoritmo de planificación de ruta que incorpora las condiciones de viento para garantizar una cobertura óptima y segura (Y. Wang et al., 2023)

**Mantenimiento de turbinas eólicas:** El acceso y la revisión de las palas de aerogeneradores son costosos y peligrosos. El uso de LiDAR montado en UAV permite generar modelos 3D de alta resolución y detectar defectos sin necesidad de personal en altura. El artículo “LiDAR-based automated UAV inspection of wind turbine rotor blades” detalla un flujo de trabajo automatizado que reduce tiempo y costo (Castelar Wembers et al., 2024)

**Optimización de rutas y localización:** Determinar los puntos de lanzamiento y las trayectorias óptimas para múltiples drones es un problema NP-hard. El estudio “Scheduling-Location Problem with Drones” plantea un modelo de optimización que minimiza tanto el tiempo de vuelo como los costos operativos, mejorando la eficiencia global de la flota (D. Kim y Moon, 2024)

**Agricultura de precisión:** En el sector agropecuario, los drones permiten la identificación y eliminación selectiva de malezas, así como la aplicación localizada de herbicidas. La revisión “Herbicide spraying and weed identification using drone technology in modern farms: A comprehensive review” muestra cómo estas técnicas aumentan la eficiencia, reducen el uso de químicos y mejoran la sostenibilidad de los cultivos (Meesaragandla et al., 2024)

**Detección de malezas mediante fused-YOLO y estimación de posición:** En este estudio se presenta un sistema basado en visión monocular para la detección automática de múltiples tipos de malezas y la estimación de su posición relativa, diseñado para integración en UAVs en aplicaciones de agricultura de precisión. Sobre imágenes obtenidas por una cámara monocular montada en un dron que sigue una trayectoria elíptica, se emplea una arquitectura profunda denominada *fused-YOLO* para clasificar y localizar las malezas. A continuación, un filtro de Kalman no lineal (UKF) refina la estimación de la posición exacta de cada planta, permitiendo aplicar tratamientos localizados con alta precisión y reducir el uso indiscriminado de herbicidas. Este enfoque optimiza tanto la eficiencia operativa como la sostenibilidad ambiental (Abdulsalam et al., 2023)

### 2.3.3. Uso de drones en la agricultura de precisión

Los UAVs juegan un papel fundamental en la **agricultura de precisión**, proporcionando una variedad de soluciones para el monitoreo de cultivos y la gestión agrícola. Estos drones se utilizan para diversas tareas, desde la **detección de maleza** hasta la **evaluación del crecimiento de los cultivos**. Los estudios realizados en el ámbito de los UAVs para la agricultura permiten optimizar las operaciones agrícolas y mejorar la eficiencia en la toma de decisiones.

#### Uso de drones para la detección de maleza y clasificación de cultivos

Varios estudios han demostrado que los UAVs, equipados con cámaras RGB o multispectrales, pueden ser utilizados para la **detección de maleza** y la **clasificación de cultivos**. Por ejemplo,

el artículo *"Performance evaluation of YOLO v5 model for automatic crop and weed classification on UAV images"* evalúa el modelo YOLO v5 para la clasificación automática de cultivos y maleza utilizando imágenes de UAVs (Ajayi et al., 2023). Esto permite realizar un monitoreo en tiempo real y aplicar tratamientos de manera localizada, optimizando el uso de insumos y reduciendo el impacto ambiental.

### Planes de vuelo autónomos y su optimización

La planificación de las rutas de vuelo es otro aspecto crucial en las operaciones con drones. En el artículo *"POTENTIAL-GUIDED UAV-FLIGHT PATH PLANNING FOR THE INSPECTION OF COMPLEX STRUCTURES"*, se presentan métodos de planificación de rutas de vuelo para UAVs, optimizando el vuelo en entornos complejos. Esta técnica también puede aplicarse en la agricultura, donde las rutas de vuelo de los UAVs pueden ser ajustadas para evitar obstáculos o áreas con mayor densidad de vegetación (Y. Wang et al., 2023).

### Monitoreo de cultivos y condiciones del terreno

El uso de UAVs para el monitoreo de cultivos también incluye la inspección de la salud de los cultivos mediante imágenes aéreas. Un ejemplo de esto es el artículo *"Automatic Monitoring of Maize Seedling Growth Using Unmanned Aerial Vehicle-Based RGB Imagery"* (Gao et al., 2023), que utiliza imágenes RGB de UAVs para monitorear el crecimiento de las plántulas de maíz. Esta tecnología permite realizar un seguimiento preciso del desarrollo de las plantas y aplicar intervenciones en tiempo real para mejorar el rendimiento de los cultivos.

### Plataformas abiertas para agricultura de precisión

El uso de plataformas abiertas para la gestión de UAVs en la agricultura está siendo cada vez más popular. Un ejemplo de esto es el artículo *"CoFly: An automated, AI-based open-source platform for UAV precision agriculture applications"* (Raptis et al., 2023), que presenta una plataforma automatizada y de código abierto para aplicaciones agrícolas de precisión utilizando UAVs. Estas plataformas ofrecen soluciones accesibles para los agricultores, permitiendo integrar **IA y aprendizaje automático** en las operaciones agrícolas.

### Consideraciones operativas para el uso de UAVs

Los UAVs en la agricultura no solo mejoran la eficiencia operativa, sino que también abordan problemas relacionados con el **costo y el tamaño** del dron, así como la **capacidad de procesamiento**. El uso de UAVs más pequeños y ágiles permite realizar vuelos más prolongados sin sobrecargar los recursos del dron. Sin embargo, este tipo de vuelos requiere de una mayor capacidad de procesamiento en tiempo real para la planificación de las rutas de vuelo y la detección de obstáculos, como se detalla en el artículo *"Autonomous Mapping and Spraying in Precision Agriculture using Unmanned Ground Vehicles"* (Ortataş et al., 2023).

### Conclusiones

Tras revisar la literatura relacionada con el tema, se observa que el uso de vuelos predefinidos es la opción elegida por la mayoría de los estudios y aplicaciones. La razón principal de esta preferencia radica en lograr una inspección más precisa del cultivo. En este enfoque, la prioridad está en dedicar la mayor parte del procesamiento a la identificación de las hileras del cultivo, en lugar de centrarse en la navegación del dron. Esto permite optimizar los recursos del sistema, asegurando que el dron se enfoque en la tarea más importante: monitorear y analizar con precisión el estado del cultivo, sin que el proceso de navegación consuma un exceso de recursos computacionales.

En este proyecto se implementa una estrategia de navegación predefinida con características adaptativas, combinando la simplicidad y eficiencia de las rutas planificadas con cierta capacidad de adaptación al entorno. En la etapa inicial, el dron es pilotado manualmente para realizar un vuelo de reconocimiento, durante el cual se capturan imágenes del cultivo con el objetivo de obtener información sobre la distribución de las hileras y las condiciones generales del terreno.

A partir de este primer vuelo, se genera un mapa del área, sobre el cual se realiza un procesamiento de imágenes para detectar las hileras de cultivo. Esta información permite planificar una nueva ruta optimizada, que el dron seguirá en vuelos posteriores de manera automática.

Si bien no se emplea una navegación adaptativa en tiempo real, la planificación de la trayectoria incorpora datos reales del entorno obtenidos previamente, lo que permite ajustar el recorrido de

manera eficiente. Esta aproximación permite maximizar el rendimiento del sistema, enfocando los recursos computacionales en el análisis del cultivo, sin comprometer la precisión del vuelo.

## Capítulo 3

# Desarrollo de la solución

En este capítulo se presenta el proceso de diseño e implementación de la solución propuesta para la inspección automatizada de cultivos mediante el uso de drones.

### 3.0.1. Conceptos básicos y asunciones de trabajo

En la figura 3.1, se presentan algunos conceptos básicos de manera gráfica que son esenciales para el entendimiento del proyecto.

- **Hilera:** sección del cultivo sembrada, la cual generalmente se encuentra sembrada en línea.
- **Perímetro:** línea que delimita la figura del cultivo, en el caso del ejemplo es un rectángulo.
- **Camino entre hileras:** espacio entre dos hileras de árboles dentro del perímetro del cultivo que permite el paso de personas o maquinaria.



Figura 3.1: Conceptos básicos de cultivos

El sistema desarrollado está pensado específicamente para cultivos frutales organizados en hileras paralelas entre sí, ya sean horizontales, verticales u oblicuas. Manteniendo espacios regulares entre ellas. Otra característica importante es que las copas de los árboles en una hilera son cercanas. Además, estos árboles deben tener una altura mayor a 1 metro. Esta información garantiza el correcto funcionamiento del sistema. Un ejemplo de un cultivo aceptable para la solución es el de la figura 3.1.

La implementación de la solución fue realizada sobre la plataforma ROS 2<sup>1</sup> en combinación con el simulador Gazebo, lo que facilitó la integración con modelos de drones, sensores y cultivos

<sup>1</sup>Más información en el anexo A.1

virtuales. En la simulación, se emplea como dron el modelo *x500* que se implementa con PX4 (PX4 Autopilot, 2024), este modelo integra los siguientes componentes que son relevantes para el proyecto:

- Cámara inferior: orientada hacia abajo para el análisis del terreno y como está distribuido el cultivo.
- Sistema de posicionamiento: GPS simulado y sensores inerciales que permiten mantener la estabilidad y conocer la ubicación precisa en todo momento.

El uso de simulación aporta ventajas como la posibilidad de ajustar parámetros ambientales y de modificar las características del dron sin necesidad de adquirir nuevo *hardware*. Además, permite iterar y validar el diseño del sistema antes de su implementación en campo real.

### 3.1. Inspección de cultivos

La automatización de la inspección se estructura como una secuencia de tareas a realizar. En esta sección se describe el conjunto de acciones que debe ejecutar el dron, desde la definición del área de interés hasta el recorrido final entre las hileras del cultivo, junto con la estrategia empleada para resolver cada etapa.

En la primera etapa del proceso, se lleva a cabo un vuelo manual mediante teleoperación para definir el área de interés del cultivo. Esta delimitación se realiza a través del trazado del perímetro de la zona a inspeccionar. A partir de los datos obtenidos durante esta fase inicial, se elabora un plan de vuelo de reconocimiento, donde se establece la ruta que seguirá el dron de manera autónoma con el propósito de recolectar información detallada del cultivo.

Durante este vuelo de reconocimiento, el dron captura imágenes de profundidad mediante cámaras *RGBD*<sup>2</sup>. Posteriormente, las imágenes son procesadas para generar un mapa de alturas del área sobrevolada. Este mapa es usado para la detección de hileras de árboles frutales presentes en el terreno, que normalmente se organizan en hileras, lo que permite ejecutar un segundo vuelo autónomo orientado a la navegación entre dichas hileras.

En el diagrama de la figura 3.2 se representa gráficamente el procedimiento detallado en los párrafos anteriores.

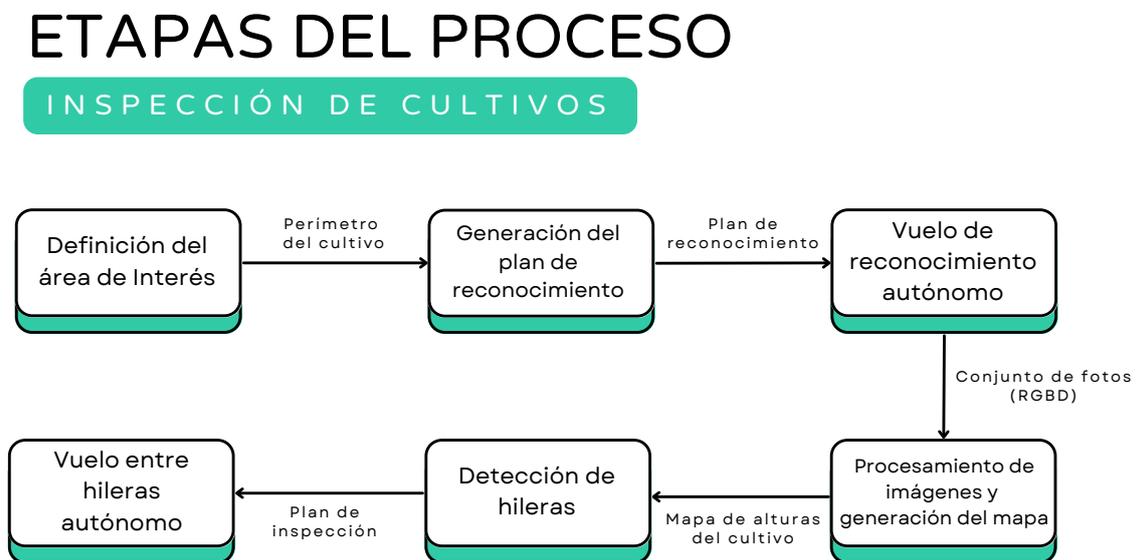


Figura 3.2: Etapas del proceso

#### 3.1.1. Definición del área de interés

El proceso inicia con un vuelo manual, llevado a cabo por un operador, durante el cual se registran las posiciones GPS del dron en formato de latitud y longitud. Estas coordenadas permiten

<sup>2</sup>Tipo de imagen que combina información de color (RGB) y profundidad (D). Cada píxel contiene los valores de color en los canales rojo, verde y azul y un valor que indica la distancia de ese punto respecto al sensor.

reconstruir la trayectoria seguida por el dron. Dicha trayectoria representa el contorno del área de cultivo que será objeto de inspección.

Como este conjunto de posiciones contiene un gran número de puntos (existiendo una gran redundancia), su presencia aumenta la complejidad computacional del análisis a realizar posteriormente. Para mitigar este problema, utilizamos el algoritmo de **Ramer-Douglas-Peucker** (Ramer, 1972), cuyo objetivo es reducir el número de puntos utilizados en la aproximación de una curva, manteniendo al mismo tiempo la forma del contorno. Este método fue seleccionado debido a su capacidad para simplificar los datos sin comprometer la fidelidad geométrica, una característica que lo diferencia de otras alternativas menos adecuadas como las que detallamos a continuación:

- **Clustering espacial:** Agrupa puntos cercanos, pero no conserva con precisión la forma del contorno.
- **Algoritmos de Envoltura Convexa:** Ignoran las concavidades del perímetro y no son fieles cuando el cultivo tiene formas complejas o irregulares.
- **Algoritmos de simplificación basados en cuadrícula:** Dependientes del tamaño fijo de las celdas, no mantienen la geometría exacta y sacrifican precisión en formas no uniformes.

En la figura 3.3 se presenta un ejemplo ilustrativo donde los algoritmos mencionados demuestran limitaciones significativas y no resultan ser la opción más adecuada. En contraste, el algoritmo **Ramer-Douglas-Peucker** es más flexible y fiel a la forma real del perímetro, ofreciendo una solución más adecuada para nuestro objetivo.

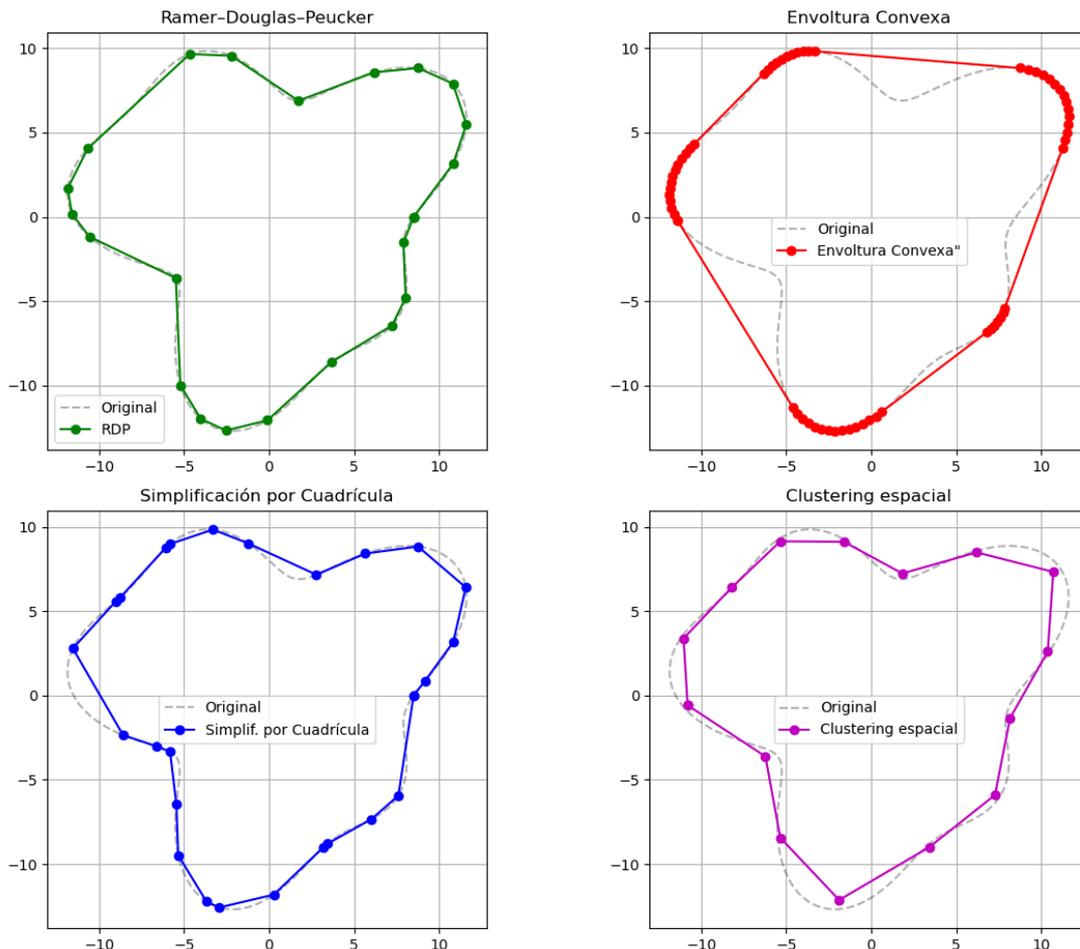


Figura 3.3: Comparación de algoritmos para la simplificación de contornos

En el algoritmo 1 se presenta un pseudocódigo de como se implementa esta simplificación. Sus parámetros son la lista de puntos a simplificar y un umbral de distancia  $\epsilon$ , que determina la tolerancia para considerar si un punto debe ser preservado en la aproximación.

---

**Algoritmo 1** Ramer Douglas Peucker

---

```
Funcion rdp(puntos, epsilon)
(* puntos contiene la lista de puntos,
epsilon es el valor del umbral a la distancia *)

primero <- puntos[1] (*Primer punto de la lista*)
ultimo <- puntos[-1] (*Ultimo punto de la lista*)

dmax = 0
indice = 0
(* Obtener punto con distancia maxima a la recta (primero, ultimo) *)
Para cada i en puntos hacer
  d <- distancia(puntos[i], recta(primero, ultimo))
  si d > dmax entonces
    indice <- i
    dmax <- d

(*Simplificar recursivamente*)
resultado <- {} (*Conjunto vacio*)
Si dmax > epsilon entonces
  (*Aplicar rdp para la primer mitad de la lista*)
  resultado1 = rdp(puntos[1..indice], epsilon)
  (*Aplicar rdp para la segunda mitad de la lista*)
  resultado2 = rdp(puntos[indice..ultimo], epsilon)
  resultado <- {resultado1, resultado2}
Sino
  resultado <- {primero, ultimo}

(*Devolver conjunto de puntos*)
Devolver resultado
```

---

Continuando con el perímetro, para  $\epsilon$  se utiliza por defecto un valor de  $1 \times 10^{-5}$  grados, que corresponde aproximadamente a 1,11 metros. Después de realizar esta simplificación, con los puntos resultantes, se construye un polígono para eliminar autointersecciones del perímetro (se despega y se aterriza dentro del cultivo) tal como se ve en la figura 3.4.

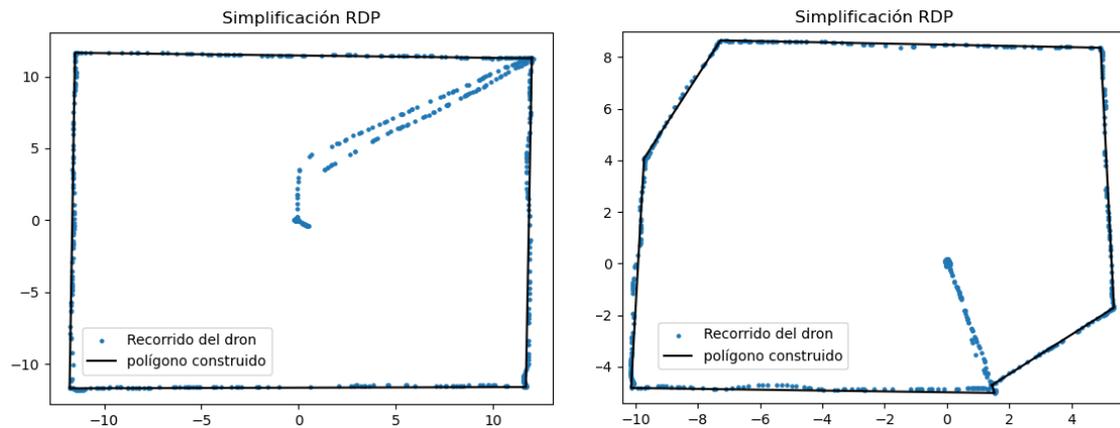


Figura 3.4: Simplificación del perímetro con Ramer Douglas Peucker

A continuación, se convierten los vértices del polígono resultante desde coordenadas geográficas en GPS a coordenadas en metros utilizando una transformación al marco de referencia local NED (North-East-Down), tomando como origen el punto de despegue del dron. En este marco de referencia, el eje  $x$  se alinea con el norte (N), el eje  $y$  con el este (E) y el eje  $z$  apunta hacia abajo (D).

Este cambio implica una transformación que aproxima una superficie terrestre como plana en torno a un punto de referencia, en este caso, el punto de despegue del dron. Esta aproximación

introduce cierto error debido a la curvatura de la Tierra, pero al ser un área de operación pequeña, se puede despreciar. Además, permite trabajar en metros, lo que simplifica los cálculos geométricos posteriores. Se presenta un esquema de la transformación realizada en la figura 3.5.

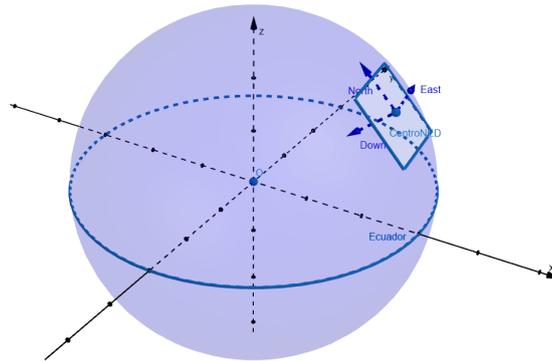


Figura 3.5: Transformación a NED

Durante el vuelo teleoperado, los datos de telemetría del dron se registran mediante una `bag`<sup>3</sup> de ROS2, la cual almacena la información en una base de datos `sqlite3`<sup>4</sup> en un archivo con formato `.db3`. Esta herramienta permite grabar y reproducir la información generada durante la operación del dron, incluyendo los mensajes de sensores y otros eventos relevantes.

Para procesar estos datos y generar una representación simplificada del perímetro del cultivo, se emplea un `nodo`<sup>5</sup> denominado `perimeter_generator`, encargado de analizar la trayectoria registrada y extraer los puntos clave que definen el área de interés. Este procedimiento se ilustra en la figura 3.6.

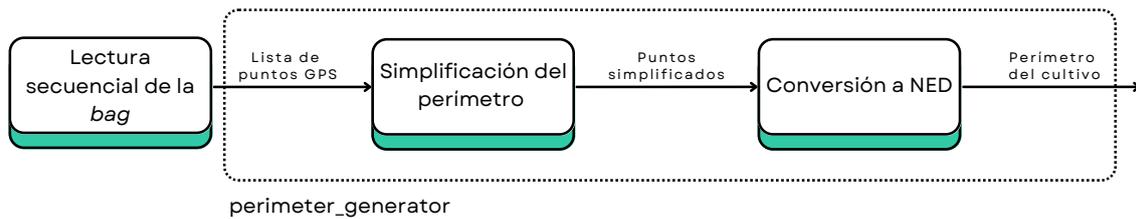


Figura 3.6: Esquema del nodo `perimeter_generator`

### 3.1.2. Generación del plan de reconocimiento

Una vez obtenido el perímetro de la parcela, se procede al siguiente paso: crear un recorrido autónomo que cubra todo el cultivo. Para ello, identificamos el lado más largo de la figura y trazamos líneas paralelas a este. Estas líneas se construyeron de manera que cubrieran la totalidad del área dentro del polígono delimitado, tal como se muestra en la figura 3.7.

La distancia entre las líneas fue determinada considerando múltiples factores operativos del dron y las características de la cámara. Tras evaluar diferentes parámetros, se estableció una altitud fija de 10 metros para el vuelo del dron y una velocidad máxima de  $2m/s$ , tomando fotografías a una frecuencia de 2 fotos por segundo. Aun así, para garantizar un solapamiento de al menos un 70% entre las imágenes, se analizó el campo de visión (FoV) de la cámara y se ajustó la distancia entre las hileras en función de la velocidad del dron.

<sup>3</sup>Herramienta que permite grabar y reproducir simulaciones de ROS2, almacenando datos de mensajes.

<sup>4</sup>Sistema de gestión de bases de datos relacionales.

<sup>5</sup>Unidad de ejecución de ROS2, es un programa independiente que realiza una tarea específica.

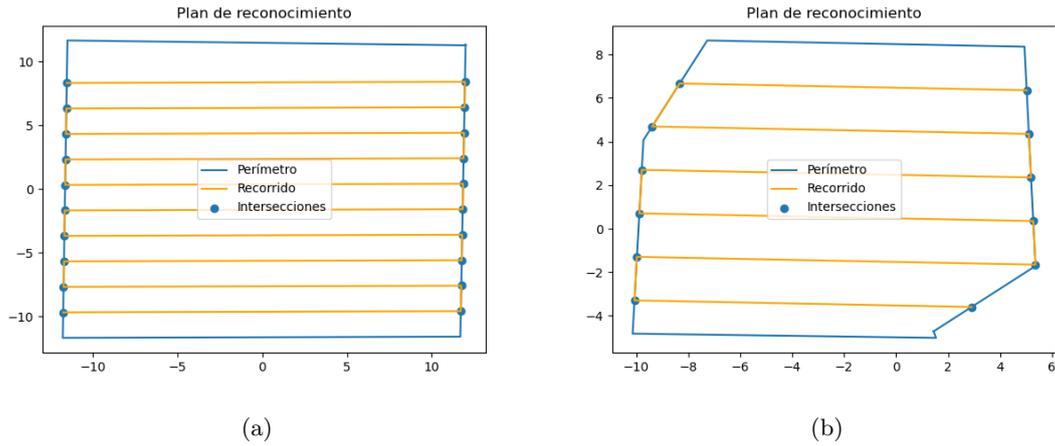


Figura 3.7: Ejemplos de trayectorias iniciales del dron

### Problemas identificados en el diseño inicial

El recorrido generado inicialmente lograba cubrir todo el cultivo. Sin embargo, identificamos un problema, los movimientos bruscos realizados por el dron al pasar de una línea paralela a otra afectaban significativamente la calidad de las imágenes capturadas.

Para abordar este problema, se incorporó un *gimbal* al modelo de dron utilizado.

### Gimbal

El *gimbal* es un dispositivo especializado que permite estabilizar la cámara al compensar automáticamente los movimientos y vibraciones del dron. Esto se logra mediante motores y sensores que ajustan la orientación de la cámara en los ejes  $x$  (*roll*),  $y$  (*pitch*) y  $z$  (*yaw*), garantizando que se mantenga fija en la dirección deseada incluso cuando el dron se mueve. (Soludrones, 2024)

En nuestra simulación y pruebas, optamos por utilizar el modelo *x500\_gimbal*, una variante del modelo *x500* que incluye un sistema de estabilización de cámara integrado. Esta versión está diseñada específicamente para aplicaciones donde la estabilización es fundamental, como la captura de imágenes georreferenciadas de alta calidad. (PX4 Autopilot, 2024)

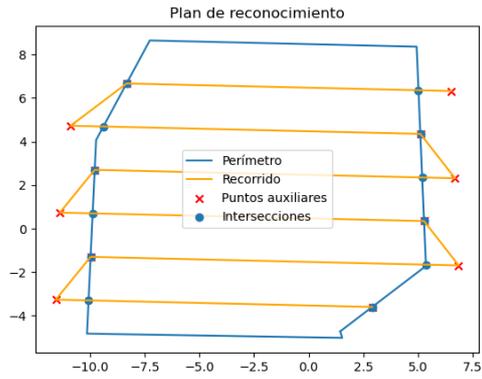
Para controlar este sistema, implementamos un nodo llamado *gimbal\_controller* que se encarga de mantener la cámara apuntando hacia abajo, contrarrestando los movimientos del dron.

Aunque la incorporación del *gimbal* mejoró la calidad de las imágenes, no resolvió completamente el problema. Al llegar a los puntos de intersección se seguía perdiendo calidad, ya que, el *gimbal* utilizado requería un tiempo para estabilizarse. Por lo que se concluyó que había que cambiar la estrategia de navegación.

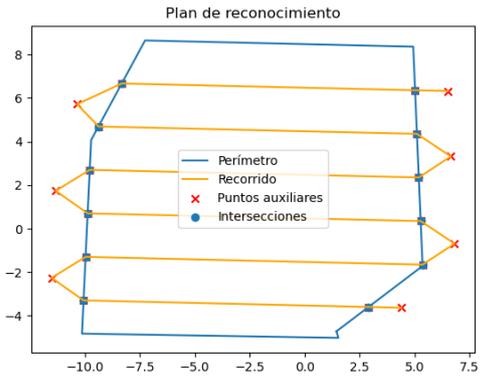
### Navegación

Durante el proyecto evaluamos distintas opciones para la navegación sobre las líneas paralelas. La figura 3.8 muestra dos de las alternativas consideradas inicialmente. Estas opciones se descartaron, porque al usar las intersecciones de las paralelas con el perímetro como puntos a recorrer, se perdía calidad en las imágenes cercanas a estos puntos.

En el caso del recorrido de 3.8a se perdía calidad al final de cada paralela y en 3.8b se perdía calidad en ambos extremos de las paralelas.



(a) Punto auxiliar al inicio de cada paralela



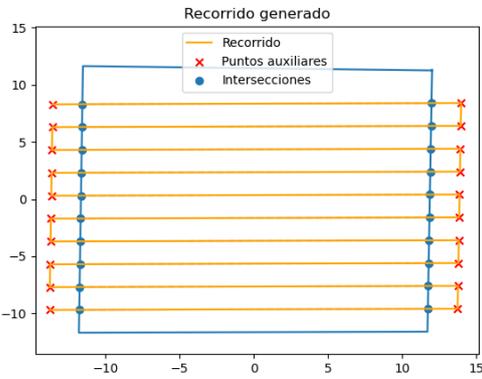
(b) Puntos auxiliares en la mitad de dos paralelas

Figura 3.8: Opciones sobre puntos auxiliares

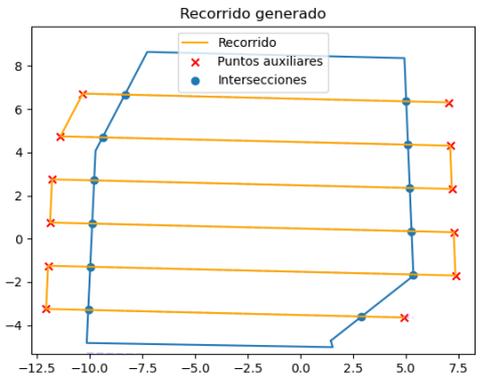
La estrategia implementada consiste en generar dos puntos auxiliares por cada paralela, evitando el uso de los puntos del perímetro como parte del recorrido. Estos puntos se colocaron a una distancia  $\delta$  (por defecto  $1,5m$ ) de las intersecciones entre las líneas y el perímetro de la parcela. Estos puntos desplazados se utilizan como referencia para que el dron realice los giros necesarios al final de cada línea, garantizando que el dron mantenga su inclinación al realizar la ruta y logrando su estabilización dentro del perímetro del cultivo.

Por último, se utiliza un nodo denominado *path\_maker*, que se encarga de obtener las líneas paralelas y generar el recorrido. Mediante el procedimiento previamente explicado, este nodo devuelve una lista de puntos que conforman la ruta para el primer vuelo de reconocimiento.

En la figura 3.9 se muestran ejemplos de los recorridos y en la figura 3.10 se presenta un esquema del flujo que se realiza para calcularlo:



(a)



(b)

Figura 3.9: Ejemplos de rutas de reconocimiento

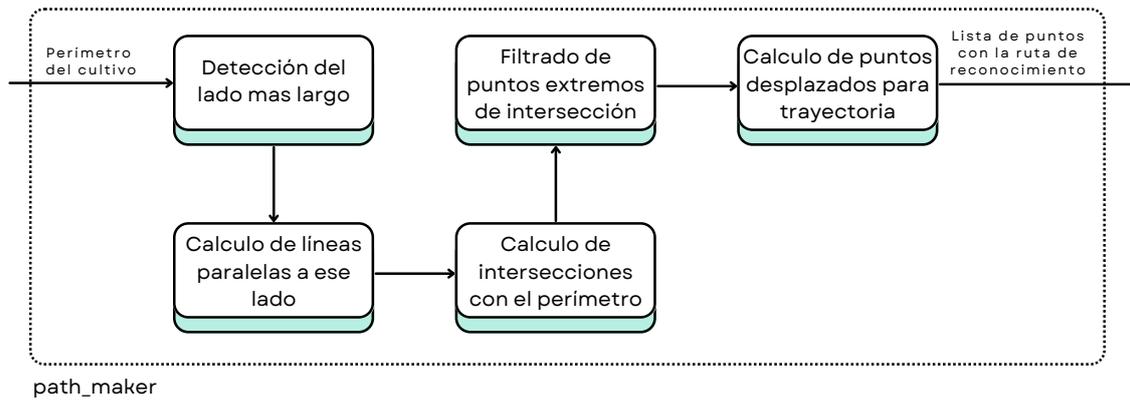


Figura 3.10: Esquema del nodo *path\_maker*

### Derivación geométrica de la fórmula del campo de visión

El siguiente análisis ilustra el fundamento geométrico de la fórmula para calcular el tamaño del área capturada por una cámara en función de su ángulo de visión (*Field of View*, FOV) y la distancia al plano observado.

El FOV define el ángulo  $\theta$  que abarca la escena visible desde la lente, formando un cono o pirámide. La mitad de ese ángulo se define como  $\text{FOV}/2$ , y se forma un triángulo rectángulo con un cateto adyacente como la distancia ( $D$ ) al objetivo, y el cateto opuesto como la mitad de la anchura ( $W/2$ ).

Por trigonometría en ese triángulo:

$$\tan\left(\frac{\theta}{2}\right) = \frac{W/2}{D} \Rightarrow W = 2 \cdot D \cdot \tan\left(\frac{\theta}{2}\right)$$

La idea clave es considerar la mitad del ángulo de visión total. Si el FOV angular (horizontal, vertical o diagonal) de la cámara es  $\theta$ , al trazar líneas rectas desde la lente hasta los extremos de la escena se forma un triángulo isósceles. La mitad de ese ángulo ( $\theta/2$ ) subtende un triángulo rectángulo con el eje central de la cámara.

En otras palabras, la fórmula se puede enunciar como:

*“Tamaño de la escena = 2 por la distancia por la tangente de la mitad del ángulo de visión.”*

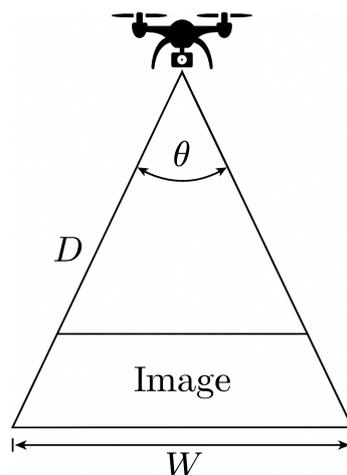


Figura 3.11: Diagrama de FOV

Esta derivación matemática proviene directamente de un triángulo rectángulo que representa la mitad del cono de visión, para encontrar la distancia. Si hay distorsión de lente o la superficie no es perpendicular, la relación es una buena aproximación pero no exacta, se discute en la tesis (Verdú, 2023).

## Descripción técnica de la cámara de profundidad OAK-D

La cámara estéreo de profundidad integrada en el dispositivo **OAK-D** utiliza dos sensores **OV9282** dispuestos en una configuración estéreo con una **línea base de 7.5 cm**, lo que permite una percepción precisa de la profundidad en tiempo real.

Cada sensor presenta los siguientes valores de **campo de visión angular**:

- Campo de visión horizontal (HFOV):  $72^\circ$
- Campo de visión vertical (VFOV):  $49^\circ$
- Campo de visión diagonal (DFOV):  $81^\circ$

Estos ángulos definen una pirámide de visión que, a una distancia conocida del plano observado (por ejemplo, el suelo), permite calcular el área cubierta por la cámara mediante relaciones trigonométricas. A una altura de **10 metros**, los tamaños del área visible proyectada son aproximadamente:

- Ancho horizontal: 14,53 m
- Alto vertical: 9,11 m
- Diagonal: 17,08 m

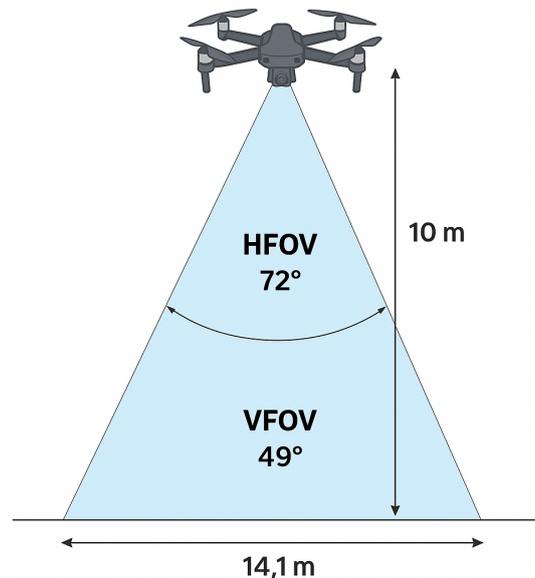


Figura 3.12: Diagrama de cálculo del HFOV y ejemplo para VFOV

Los valores anteriores se obtienen utilizando la siguiente expresión, derivada de la trigonometría básica aplicada a un triángulo rectángulo:

$$\text{Tamaño} = 2 \cdot \text{distancia} \cdot \tan\left(\frac{\text{FOV}}{2}\right)$$

Este análisis es fundamental para tareas de navegación, mapeo o reconocimiento en sistemas robóticos y vehículos aéreos no tripulados, ya que determina la cobertura por imagen y la resolución espacial alcanzable.

### 3.1.3. Vuelo de reconocimiento

En esta etapa comienza la etapa de vuelo del dron. Es importante tener en cuenta que el origen local del dron corresponde a la posición de despegue actual. Dado que esta posición puede no coincidir con la utilizada anteriormente como origen (descrita en la sección 3.1.1), es necesario hacer una traslación de los ejes al nuevo marco de referencia, como se muestra en la figura 3.13.

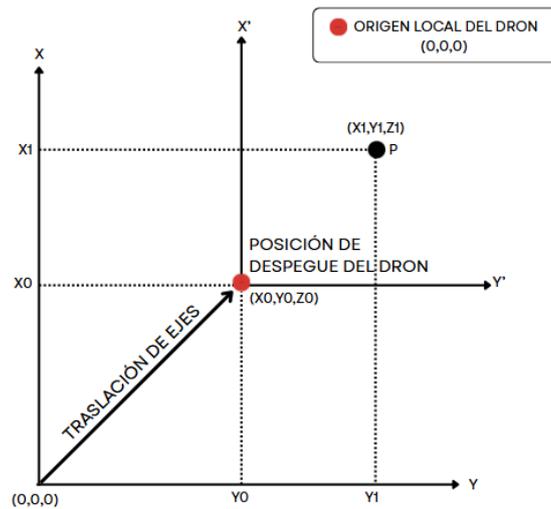


Figura 3.13: Traslación de los ejes de coordenadas

El recorrido del dron se define a través de los vértices más lejanos de cada línea trazada en la etapa anterior (puntos auxiliares). Para determinar si el dron ha alcanzado un punto específico en su recorrido, se utiliza un [tópico](#)<sup>6</sup> propio de PX4 llamado `fmu/out/vehicle_local_position` que publica información sobre la ubicación local del dron. Cuando el dron alcanza una posición  $p$  dentro del entorno  $\epsilon$  de un vértice  $v$ , es decir, se cumple  $|p - v| < \epsilon$ , el sistema registra su posición en una lista de puntos visitados; si  $v$  se encuentra dentro del perímetro calculado, se habilita la toma de imágenes. Este proceso se ilustra en el diagrama de flujo de la figura 3.14.

Las imágenes capturadas por la cámara del dron son de profundidad. Cada píxel de la imagen, además de la información de color RGB, tiene un valor entre 0 y 255 que representa la distancia al objeto más cercano. Sin embargo, estas imágenes no contienen metadatos GPS, por lo que se implementó un nodo denominado `image_saver`, encargado de procesar cada imagen capturada, adjuntándole la información de posicionamiento global (latitud, longitud y altitud) obtenida desde el [tópico](#) `/fmu/out/vehicle_global_position` de PX4.

Al finalizar el recorrido, luego de haber visitado todos los puntos previstos, el nodo procede a desactivar el modo `offboard`, desarmar el dron y publicar un mensaje final indicando la finalización exitosa de la misión. Esta secuencia marca el cierre automático de la operación tras concluir la captura de datos.

<sup>6</sup>Forma de comunicación entre nodos que permite que los nodos puedan intercambiar datos. Los nodos pueden publicar (enviar) información a un tema, y otros nodos pueden suscribirse.

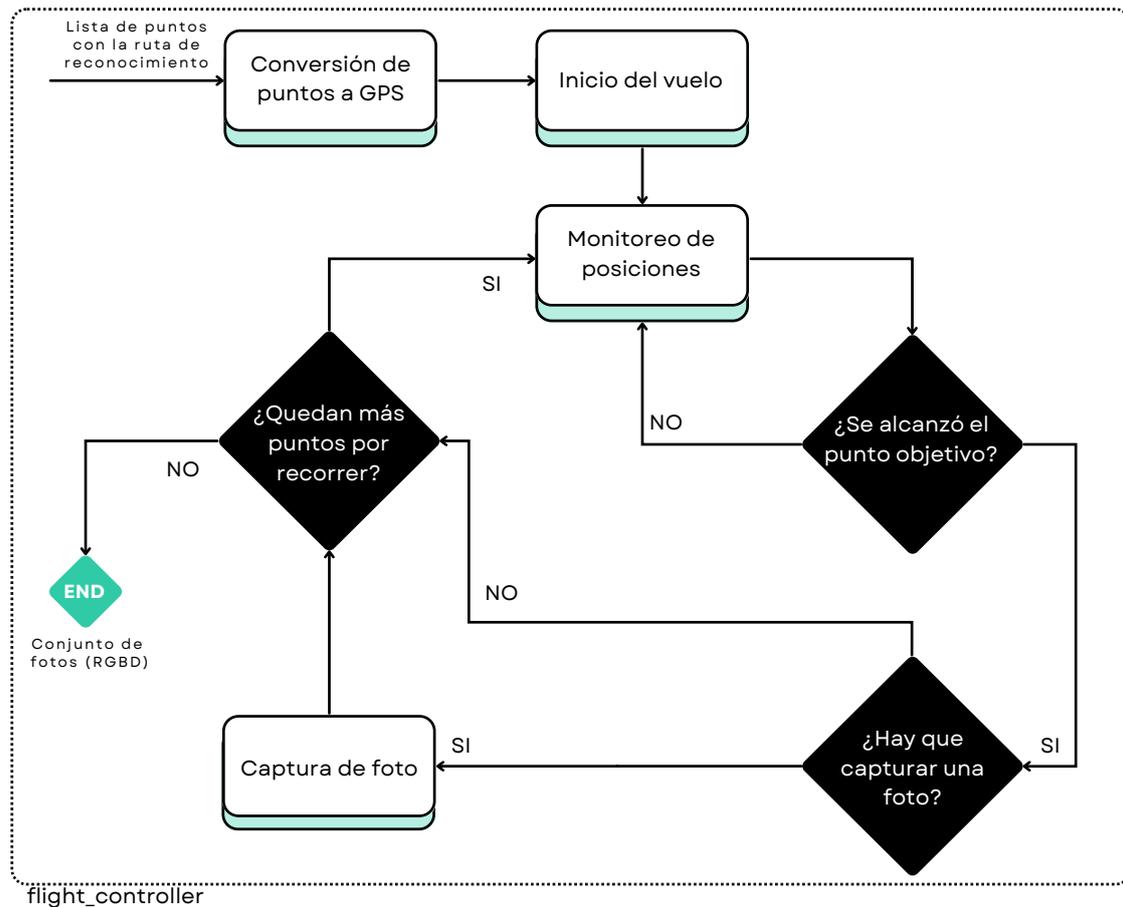


Figura 3.14: Esquema del nodo *flight\_controller*

### 3.1.4. Procesamiento de imágenes y generación del mapa

Tras realizar el recorrido inicial para el reconocimiento de la morfología del cultivo, se procede hacia la creación de un mapa de alturas (*ortomosaico*<sup>7</sup> de alturas) a partir de las fotografías capturadas por el dron, con el fin de poder identificar las hileras de los árboles frutales y realizar la inspección entre dichas hileras. La creación del ortomosaico conformó uno de los aspectos más complejos y desafiantes en el desarrollo del proyecto, requiriendo ensamblar múltiples imágenes tomadas desde el dron para generar un mapa georreferenciado. Este proceso involucró numerosos detalles técnicos que requirieron una extensa investigación, pruebas y ajustes para lograr un resultado satisfactorio.

#### Selección del software

El primer paso en este desafío fue identificar una herramienta libre y gratuita que permitiera procesar las imágenes capturadas durante el vuelo de reconocimiento del dron. Este paso resultó ser más difícil de lo esperado, ya que la mayoría de las herramientas disponibles son pagas, restringiendo nuestras opciones a un conjunto reducido de softwares entre los que se encontraban: QGIS, MicMacODM y ODM (OpenDroneMap).

- **QGIS:** Es un sistema de información geográfica (GIS) ampliamente utilizado para el análisis y la visualización de datos geoespaciales. Esta herramienta permite la creación de mapas utilizando diversas capas y formatos, por lo que inicialmente era una opción prometedora. Sin embargo, al intentar generar un mapa con las imágenes del dron, en lugar de ensamblar las fotos en un mosaico, el software las apilaba una sobre otra devolviendo una única imagen. Finalmente, descartamos QGIS debido a su incapacidad para producir un ortomosaico funcional con nuestros datos.

<sup>7</sup>Imagen detallada y georreferenciada de un lugar, creada mediante la unión de fotos de alta resolución tomadas desde drones u otras plataformas aéreas.

- **MicMac:** Es una herramienta desarrollada para generar ortomosaicos y modelos tridimensionales. Este software daba buenos resultados al trabajar con imágenes RGB, pero presentaba errores para procesar imágenes de profundidad como las que necesitábamos utilizar. Además, la instalación y configuración de MicMac resultaron ser especialmente complejas ya que generaban errores recurrentes. Después de invertir un tiempo considerable sin obtener resultados útiles, decidimos descartar esta herramienta y buscar alternativas.
- **ODM:** Es un software de código abierto, siendo uno de los más reconocidos en la comunidad de drones y procesamiento de imágenes aéreas. ODM tiene la ventaja de contar con una comunidad activa y extensa documentación, lo que inicialmente nos dio confianza en su potencial, por lo que había sido nuestra primer opción. Además cuenta con una interfaz gráfica llamada WebODM, que facilitó la comprensión del proceso de generación de mapas y permitió organizar y clasificar las distintas ejecuciones, tanto exitosas como fallidas.

Se priorizó el uso de herramientas ampliamente utilizadas en el ámbito de la fotogrametría y la agricultura de precisión, que cuentan con validación en la literatura científica. En este sentido, software como OpenDroneMap (ODM) ha sido recurrentemente empleado en investigaciones relacionadas con agricultura de precisión y monitoreo de cultivos, debido a su capacidad para generar ortomosaicos, modelos digitales de superficie y nubes de puntos a partir de imágenes obtenidas con UAVs. Además, ODM cuenta con una interfaz que permitía la facilidad de uso, cambiando parámetros de forma sencilla para adaptarse a nuestras necesidades y pruebas.

La clave para poder utilizar esta herramienta (ODM) fue basarnos en la hipótesis de que el software estaba optimizado para trabajar con imágenes de tres canales, como las tradicionales RGB. Es por esto que transformamos nuestras imágenes de profundidad con un único canal  $D$  a un formato RGB, replicando el valor de cada píxel del canal  $D$  en los tres canales  $R$ ,  $G$  y  $B$ . Logrando conservar la información original de profundidad en escala de grises. Este cambio permitió que ODM las procesara sin errores y generara un ortomosaico con las imágenes capturadas por el dron. Así, el conjunto de entradas consistió en un conjunto de imágenes RGBD georeferenciadas, y la salida fue un mapa de alturas, como se muestra en la figura 3.15.



Figura 3.15: Esquema de la generación del ortomosaico

El resultado final del ortomosaico cumple con los requisitos fundamentales del proyecto, especialmente en lo que respecta a la diferenciación entre el suelo y los cultivos. En la figura 3.16 se presentan ambas imágenes con el propósito de ilustrar la similitud entre ellas.

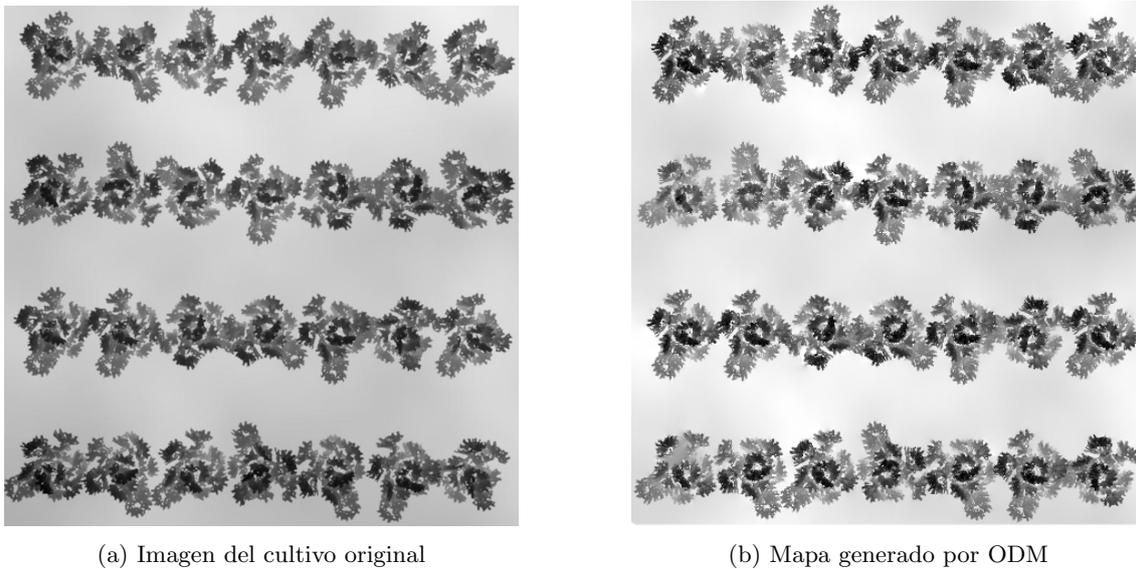


Figura 3.16: Comparación del cultivo con el mapa generado por ODM

La imagen generada proporciona la información necesaria para realizar la detección de hileras, y, combinada con el perímetro previamente obtenido del cultivo, permite generar el recorrido que debe seguir el dron, definiendo trayectorias paralelas a las hileras de plantas.

Una alternativa a esta solución adoptada es generar un ortomosaico en formato RGB y posteriormente añadir los datos de altura a cada píxel. Sin embargo, esta alternativa fue descartada ya que al incorporar los valores de altura se corre el riesgo de introducir desajustes, porque una posición en el espacio podría tener varios valores de altura debido a errores de registro o interpolación.

### 3.1.5. Detección de hileras

Con el nuevo mapa generado, se inicia el proceso de detección de las hileras del cultivo. Para lograr este objetivo, lo primero que se realiza es un filtrado de los puntos. Al ser un mapa de profundidades, podemos diferenciar las partes altas de la imagen de las partes más bajas. Las áreas altas corresponden a las copas de las plantaciones, representadas en color gris, mientras que las partes bajas corresponden al suelo, representadas en color blanco.

Para facilitar esta separación, se aplica una máscara binaria en la que los píxeles con una altura menor al umbral (establecido por defecto en 200) se asignan al valor 1 (blanco), mientras que los demás píxeles se asignan al valor 0 (negro). El umbral se utiliza para poder crear la máscara binaria, que era necesaria para utilizarla en los siguientes pasos de la identificación de las hileras, en las pruebas realizadas, un valor alto ocasionaba una distorsión en los contornos y no se identificaba correctamente las hileras, mientras que un valor bajo ocasionaba que se detectaran más contornos de los que realmente existían. Llegando al valor de 200, un valor que corresponde a una altura correcta en muchos cultivos probados, aun así, este es un parámetro que se puede cambiar y ajustarse al cultivo utilizado. Este proceso permite identificar de manera más precisa las áreas correspondientes a los árboles y al suelo, como se muestra en la figura 3.17a.

Luego se obtiene una lista de contornos a partir de la imagen binaria, donde cada contorno corresponde a un conjunto de puntos que define el límite de una región identificada (en nuestro caso, una hilera). Sin embargo, algunas de estas secciones no eran correctas debido a imperfecciones en la imagen o a la detección de objetos no deseados. Para filtrar estos casos, se estableció un umbral mínimo de área: solo se consideran aquellos contornos cuya superficie supera un cierto valor de píxeles. En nuestro caso, este número fue establecido a 500 por defecto. En la figura 3.17b se muestra un ejemplo de la detección de contornos. Se realizaron varias pruebas para decidir un valor específico que ayudara en diferentes cultivos, además de un estudio de la relación entre las hileras y su tamaño, si el valor era muy alto, los contornos de los árboles de distintas hileras se juntaban, generando un error en la detección del contorno. En cambio, al un valor muy bajo, no lograba unir adecuadamente las hileras.

Adicionalmente, para evitar la detección errónea de caminos entre hileras (que en ocasiones aparecen como espacios blancos entre los árboles) se incorporó una función de fusión de contornos. Esta función combina contornos que se encuentren a una distancia inferior al umbral determinado, ayudando a evitar errores producidos por el movimiento de los árboles o la segmentación fragmen-

tada de una misma hilera. En la figura 3.18a y 3.18b se puede observar el resultado de aplicar una máscara binaria a estas hileras fusionadas y luego detectar los contornos sobre esa máscara respectivamente.

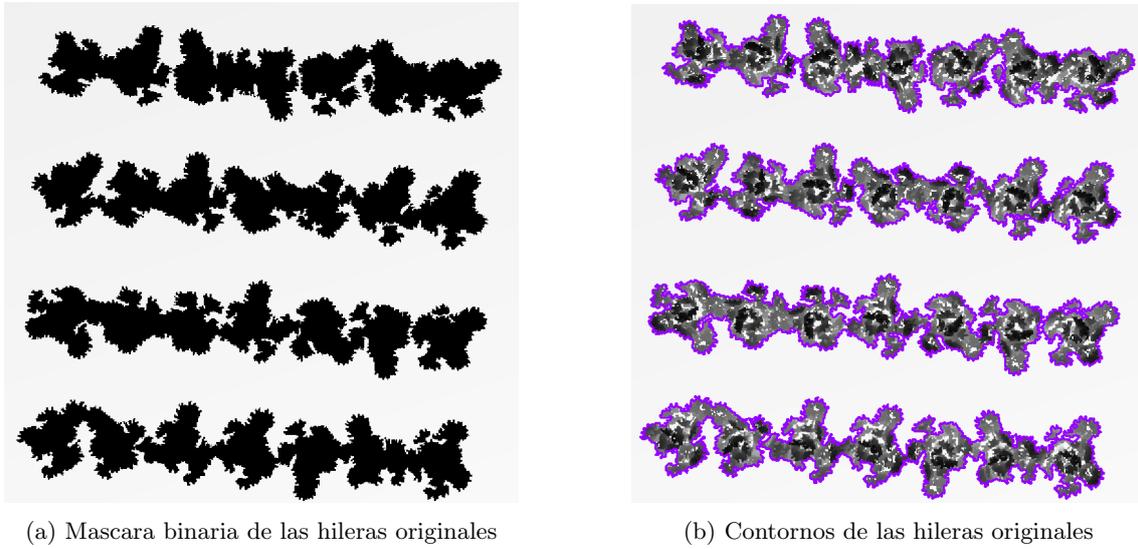


Figura 3.17: Detección de hileras originales

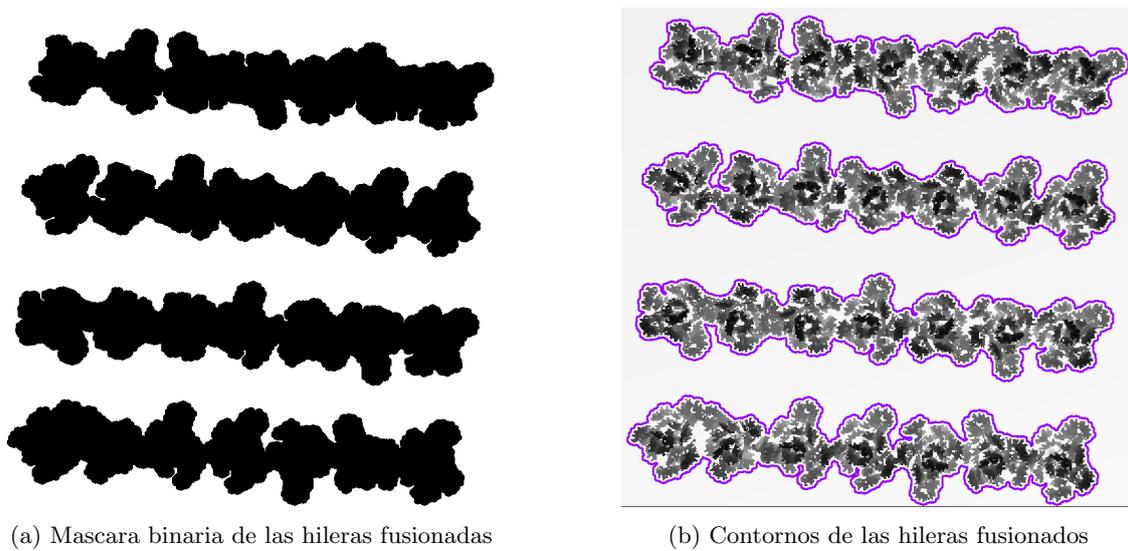


Figura 3.18: Detección de hileras fusionadas



Figura 3.19: Unificación de hileras con los contornos fusionados

En la Figura 3.19a se observa como, debido a las separaciones entre las copas de los árboles frutales, una hilera es fragmentada en tres secciones distintas. Como resultado, el algoritmo detecta tres objetos independientes en lugar de una única hilera continua. Sin embargo, tras aplicar el proceso de fusión de contornos, se puede apreciar en la Figura 3.19b que dichas discontinuidades han sido eliminadas. De esta forma, las secciones previamente separadas se unifican en un solo objeto, permitiendo identificar correctamente cada hilera y pudiendo diferenciar una hilera de las demás.

### Primeros enfoques para la detección del camino entre hileras

El primer enfoque adoptado para identificar los caminos ubicados entre las hileras del cultivo consistió en, una vez obtenidos los contornos de las hileras, generar un rectángulo que envolviera todos los puntos de cada contorno. La idea era que, a partir de estos rectángulos, se pudieran trazar líneas paralelas a sus lados principales y, de esta manera, definir los caminos por los cuales el dron debía navegar entre las hileras.

Como puede observarse en la imagen 3.20, logramos representar cada hilera mediante un rectángulo verde. Luego, para calcular la trayectoria entre dos hileras consecutivas, simplemente se determinaba el punto medio entre sus bordes enfrentados, obteniendo así una línea central que pasa entre ambas. Finalmente, nos quedábamos con tres puntos: los extremos y el punto medio de esta línea, que servirían como referencia para el trayecto del dron.

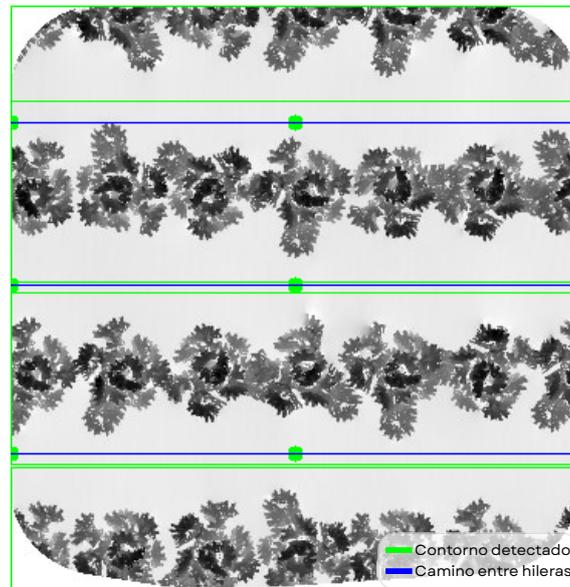


Figura 3.20: Detección de hileras mediante recuadros

Sin embargo, este método presentaba limitaciones cuando la imagen no estaba alineada en sentido horizontal o vertical, tal como muestra la figura 3.21. En esos casos, la detección de hileras a través de rectángulos no era fiable, ya que los rectángulos no se ajustaban adecuadamente a la orientación real de las hileras.

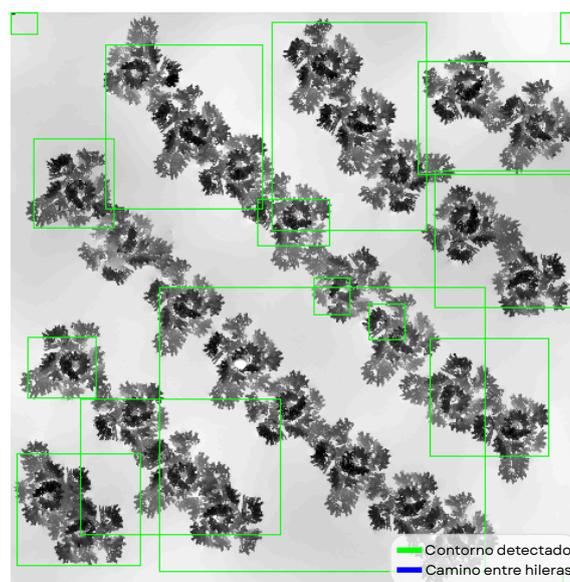
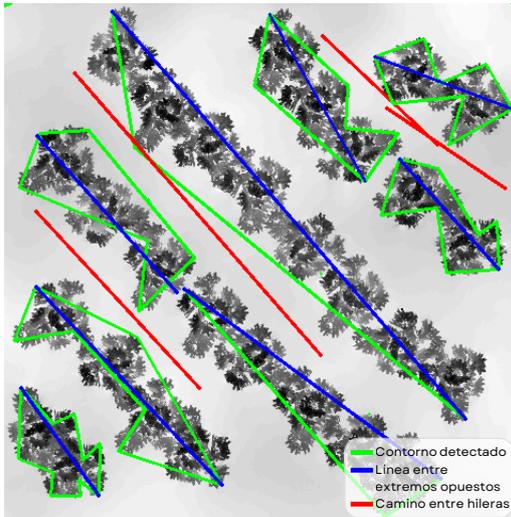


Figura 3.21: Detección de hileras con una imagen en diagonal

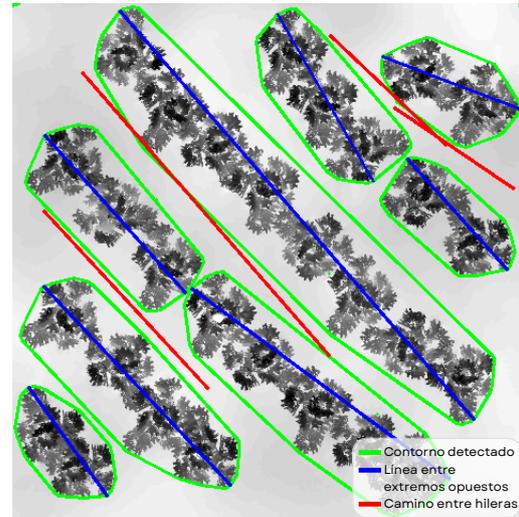
A partir de estas limitaciones, se probaron otros enfoques que involucraban distintas formas

de aproximar los contornos detectados, como el uso de cascos convexos, líneas extremas o líneas medias. Estos métodos exploran distintas formas de representar cada hilera y generar caminos entre ellas. A continuación, se presentan algunas de las pruebas realizadas:

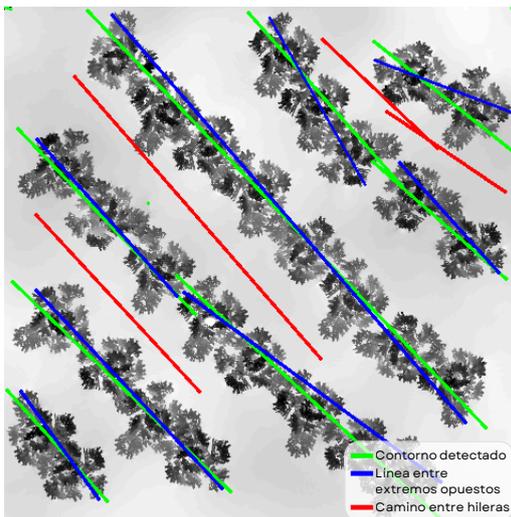
En la figura 3.22 se presentan los resultados obtenidos:



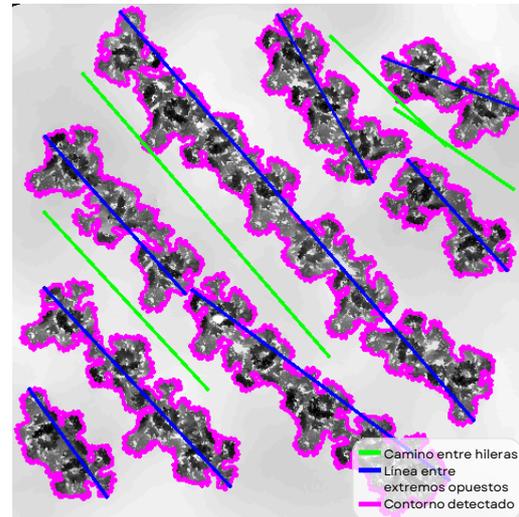
(a) Resultado con `approxContours`



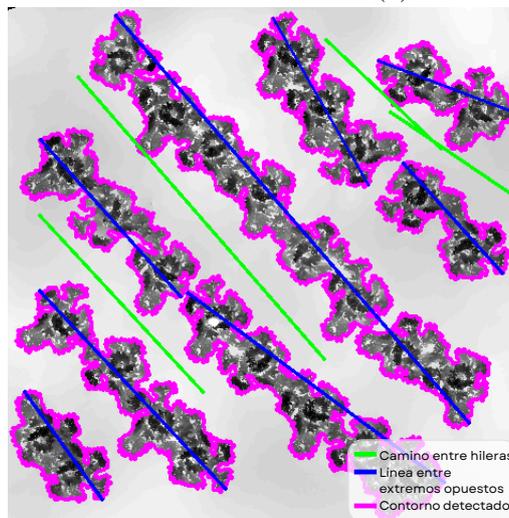
(b) Resultado con `convexHull`



(c) Resultado con `extremeLines`



(d) Resultado con `filteredContours`



(e) Resultado con `middleLines`

Figura 3.22: Resultados de los algoritmos

Otro de los enfoques explorados consistió en la generación de una máscara en la que se expandió el área correspondiente a las hileras mediante la superposición de una grilla regular. Este procedimiento tenía como objetivo estructurar el espacio de manera más uniforme, con la intención de facilitar la posterior determinación de los caminos de navegación entre hileras. Sin embargo, los resultados obtenidos no fueron satisfactorios. La figura 3.23 ilustra este caso aplicado a un caso con hileras dispuestas en orientación horizontal.

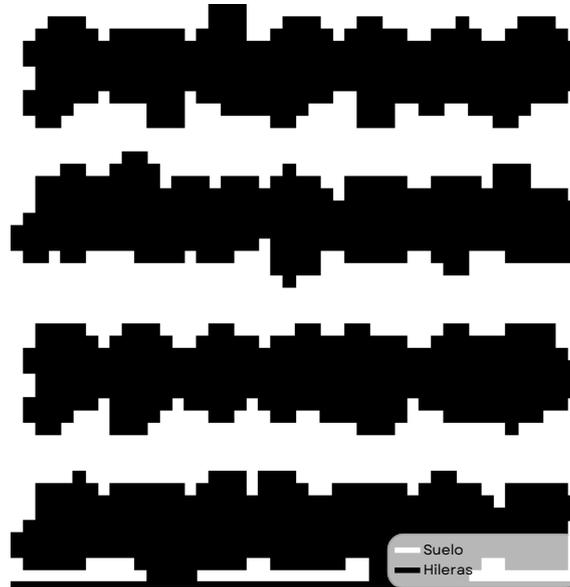


Figura 3.23: Resultado con `middleLines` en hileras horizontales

Si bien ninguno de estos métodos logró resolver el problema de forma robusta frente a variaciones en la orientación o forma de las hileras, permitieron entender mejor la estructura del problema y sentaron las bases para la siguiente etapa: el uso del Diagrama de Voronoi como herramienta para definir caminos de navegación.

### Voronoi generalizado

El diagrama de Voronoi consiste en dividir un espacio en regiones basadas en la proximidad a un conjunto dado de puntos, conocidos como “semillas” o puntos generadores. Cada región contiene todos los puntos que están más cerca de su semilla que de cualquier otra. Es decir, el diagrama asigna a cada semilla un área que abarca todos los puntos más cercanos a ella, creando un conjunto de celdas poligonales que cubren todo el espacio. (Analytics, 2022)

Sin embargo, este enfoque no se adaptaba a nuestras necesidades, ya que, al basarse en puntos de semilla, no podíamos representar adecuadamente nuestras hileras de árboles frutales. Sintetizar una hilera en un solo punto no era representativo de la realidad, ya que implica asumir una distribución uniforme que no refleja las características reales del cultivo, lo que podría llevar a una detección errónea, como se muestra en la figura 3.24b para el caso del cultivo en diagonal, donde las líneas generadas chocaban entre sí. Por esta razón, fue necesario utilizar la versión generalizada de Voronoi para trabajar directamente con polígonos que representaran de manera más precisa las áreas ocupadas por las hileras. Por ello, se optó por trabajar directamente con polígonos derivados de los contornos detectados para cada hilera, extraídos previamente a partir del mapa de alturas.

Como los contornos originales contenían una gran cantidad de puntos, se aplicó una técnica de muestreo por intervalos para reducir su cantidad sin perder representatividad. Esta reducción era necesaria para disminuir la complejidad computacional, ya que utilizar todos los puntos generaba un sobrecosto de procesamiento sin mejoras sustanciales en la calidad del resultado.

Para generalizar el Diagrama de Voronoi a este contexto, se introdujo una lista de etiquetas: a cada hilera se le asignó una etiqueta única que fue replicada en todos los puntos pertenecientes a ella. Así, las etiquetas permitieron identificar y clasificar a los puntos de acuerdo con su hilera correspondiente. Durante la construcción del diagrama, estas etiquetas evitaron que se generaran líneas entre puntos de la misma hilera, lo que resultaba esencial para delimitar correctamente los espacios de navegación entre hileras adyacentes.

Sin embargo, como medida adicional de robustez, se realiza un filtrado posterior de las aristas generadas. Este paso tiene como objetivo eliminar de forma defensiva cualquier línea que pueda

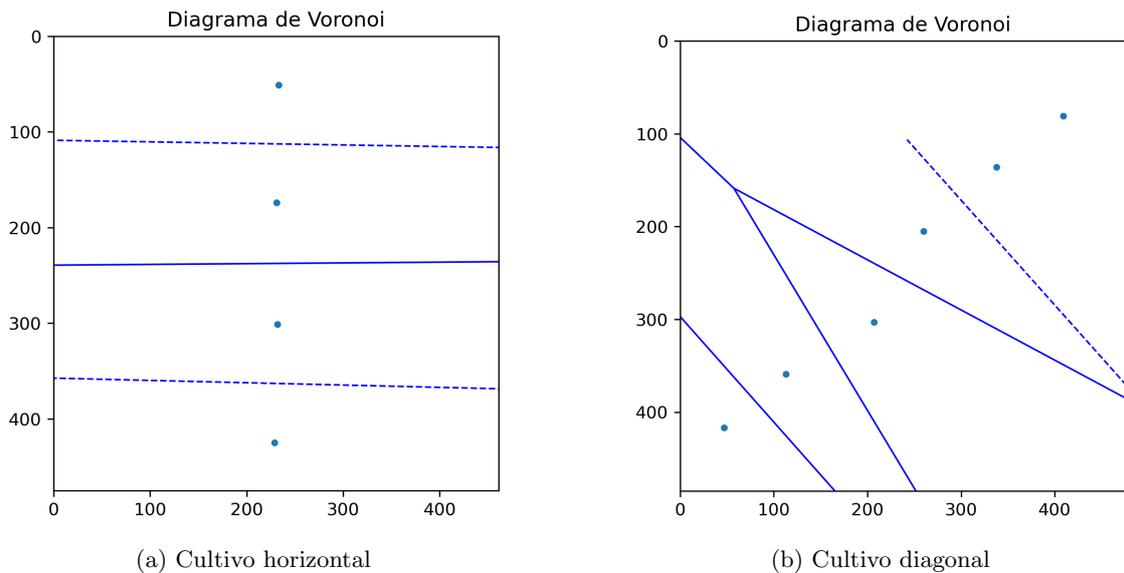


Figura 3.24: Detección de hileras usando Voronoi

haber sido trazada entre puntos con la misma etiqueta, ya sea por errores numéricos o imperfecciones en los datos. De este modo, se garantiza que las únicas líneas conservadas correspondan efectivamente a zonas navegables entre diferentes hileras.

Una vez construido el diagrama, y como medida adicional de robustez, se realiza un filtrado posterior de las aristas no deseadas entre puntos de una misma hilera, con el fin de asegurar que las líneas generadas no atraviesen las áreas ocupadas por la vegetación. De este modo, se conservan únicamente aquellas que delimitan las zonas navegables entre hileras adyacentes. Finalmente, se almacenaron los vértices de estas líneas del diagrama de Voronoi, los cuales fueron utilizados como base para planificar las rutas de navegación del dron.

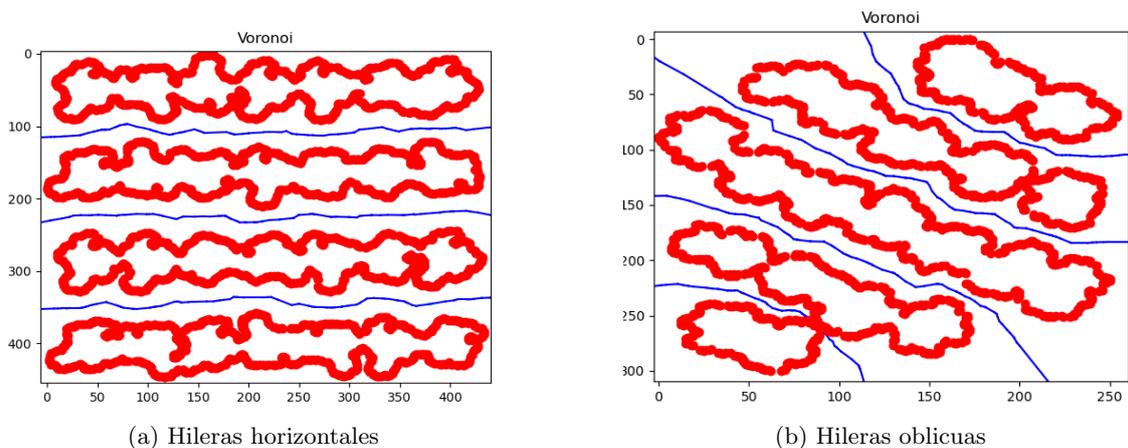


Figura 3.25: Detección de hileras usando Voronoi generalizado

En la figura 3.25 se pueden observar los contornos de las hileras representados en color rojo, y en azul, los caminos resultantes luego de aplicar el filtrado que elimina las líneas no relevantes. De esta forma, se obtienen únicamente las trayectorias que atraviesan las zonas intermedias entre hileras consecutivas, definiendo así los caminos de navegación para el dron.

El nodo `height_map_path` es el responsable de llevar a cabo todo el procedimiento para generar los caminos entre las hileras. En la figura 3.26 se muestran las etapas que lleva a cabo dicho nodo.

### Problemas encontrados con esta solución

Una limitación de la detección de hileras utilizando voronoi generalizado aparece cuando falta algún árbol en una hilera. En este caso, el algoritmo considera espacio como si fuera un camino entre dos hileras, haciendo que se genere un camino que no es óptimo para recorrer el cultivo. En la figura 3.27 observamos el resultado del algoritmo cuando al cultivo 3.25a le falta un árbol.

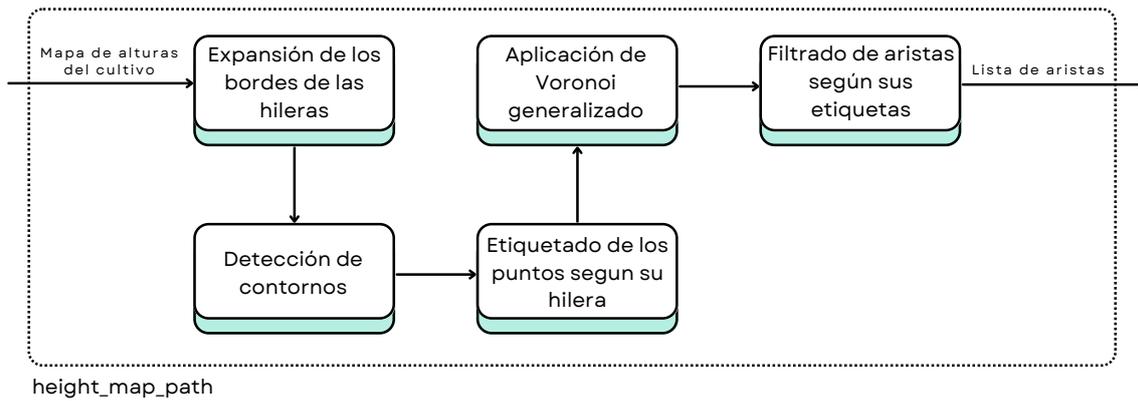


Figura 3.26: Diagrama del nodo height\_map\_path

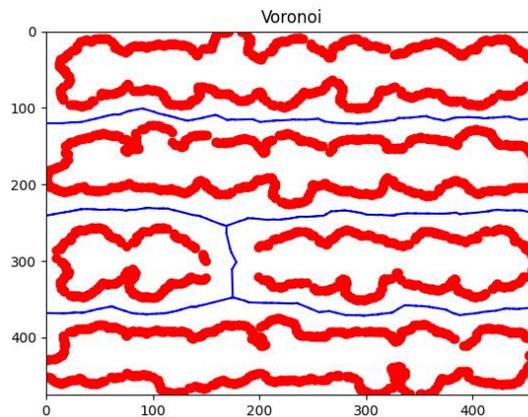


Figura 3.27: Detección de hileras usando Voronoi generalizado faltando un árbol

### 3.1.6. Vuelo entre hileras

Para este segundo vuelo, al igual que en el primero, es necesario indicarle al dron cuáles son las posiciones a las que debe dirigirse. Por ello, a partir del conjunto de vértices generados mediante el diagrama de Voronoi, se realiza una serie de transformaciones y análisis con el objetivo de obtener un camino navegable, expresado en coordenadas NED.

En primer lugar, se requiere ordenar los vértices de la manera más eficiente posible para minimizar la distancia total del recorrido, ya que a priori no se dispone de una heurística que indique cuál es el camino óptimo.

#### Ordenación de vértices

Para ordenar los vértices, se construye un rectángulo convexo que los contiene, utilizando valores mínimos y máximos en las coordenadas  $x$  e  $y$ . Para cada lado de este rectángulo, se toman como candidatos a vértice inicial aquellos puntos que estén a una distancia menor que  $\delta$  (por defecto en  $0,1m$ ) del borde correspondiente, y que, además, posean el valor mínimo y máximo en la coordenada paralela a dicho borde. En el algoritmo 2 se presenta un pseudocódigo que realiza estos cálculos y retorna los candidatos a vértice inicial.

Luego, para construir el camino, desde cada vértice candidato se construye un recorrido avanzando siempre hacia el vértice más cercano no visitado. Finalmente, se selecciona como orden óptimo el recorrido con menor distancia total, como se describe en el algoritmo 3.

---

**Algoritmo 2** Obtener candidatos

---

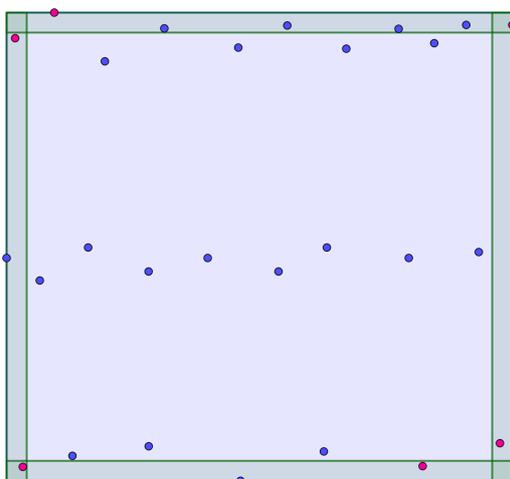
```
Funcion obtenerCandidatos(puntos, delta, min_x, max_x, min_y, max_y)
(* puntos contiene la lista de puntos,
delta es el valor de la distancia maxima *)
(* min_x contiene el minimo en x de la lista de puntos *)
(* min_y contiene el minimo en y de la lista de puntos *)
(* max_x contiene el maximo en x de la lista de puntos *)
(* max_y contiene el maximo en y de la lista de puntos *)

(* Definir 4 listas para los puntos cercanos a los minimos y maximos*)
lista [1..4] <- [[], [], [], []]
Para cada (x,y) en puntos hacer
  Si x - min_x <= delta entonces
    Agregar (x, y) a lista [1]
  Si max_x - x <= delta entonces
    Agregar (x, y) a lista [2]
  Si y - min_y <= delta entonces
    Agregar (x, y) a lista [3]
  Si max_y - y <= delta entonces
    Agregar (x, y) a lista [4]
vertices_candidatos <- {} (*conjunto vacio de vertices*)
Para i <- 1..2 hacer
  Agregar el minimo y maximo en y de lista [i] a vertices_candidatos
Para i en 3..4 hacer
  Agregar el minimo y maximo en x de lista [i] a vertices_candidatos
Devolver vertices_candidatos
```

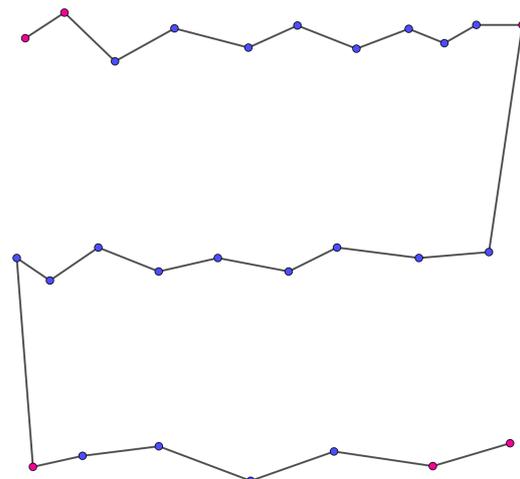
---

Este enfoque es viable porque la distancia entre los vértices dentro de las regiones de Voronoi que representan una hilera es significativamente menor que la distancia entre hileras. De no ser así, el algoritmo no funcionaría correctamente, ya que, al buscar el vértice más cercano, podría saltar a una hilera diferente. En otras palabras, la alta densidad de puntos generados en cada hilera garantiza que la distancia entre vértices consecutivos sea muy pequeña, permitiendo un orden correcto.

La figura 3.28a presenta un ejemplo del conjunto de vértices generados anteriormente por Voronoi dentro del cultivo, siendo los vértices en rojo aquellos posibles candidatos a ser el primer punto. Mientras que la figura 3.28b muestra el resultado luego de ordenar.



(a) Resultado del algoritmo 2



(b) Resultado del algoritmo 3

Figura 3.28: Vertices de Voronoi

Para detectar un salto de un camino a otro, se calcula el promedio de las distancias entre los vértices. Si la distancia entre dos vértices es mayor que el promedio multiplicado por un factor  $\alpha$  (por defecto 10) entonces se considera que hubo un salto.

---

**Algoritmo 3** Ordenar vértices

---

```
Funcion ordenarVertices(puntos, vertices_candidatos)
  (* Ordenar vertices *)
  distancia_minima <- infinito
  mejor_camino <- []
  Para i en vertices_candidatos hacer
    camino <- [i] (* lista con el indice i *)
    distancia <- 0
    no_visitados <- puntos - i
    Mientras quede un vertice sin visitar y
      distancia sea menor a distancia_minima hacer
        ultimo <- ultimo vertice del camino
        Agregar el vertice mas cercano a ultimo al camino
        Remove ultimo de no_visitados
        distancia <- distancia + distancia al vertice mas cercano

    Si distancia < distancia_minima entonces
      distancia_minima <- distancia
      mejor_camino <- camino

  Devolver mejor_camino
```

---

Posteriormente, se ajusta cada camino mediante una regresión polinómica de hasta quinto grado, con el objetivo de suavizar el recorrido y minimizar el error. El grado óptimo del polinomio se determina mediante validación cruzada, utilizando el error cuadrático medio como métrica de evaluación. A esta regresión se le aplica una simplificación RDP<sup>8</sup> con una tolerancia de 0,1m, para reducir la complejidad del recorrido tal como se muestra en la figura 3.29. Por último, los puntos resultantes se transforman al marco de referencia NED, mencionado anteriormente, para su integración con el sistema de navegación del dron. Técnica de corrección diferencial del GPS que permite obtener una geolocalización de alta precisión, con márgenes de error en el orden de los centímetros.

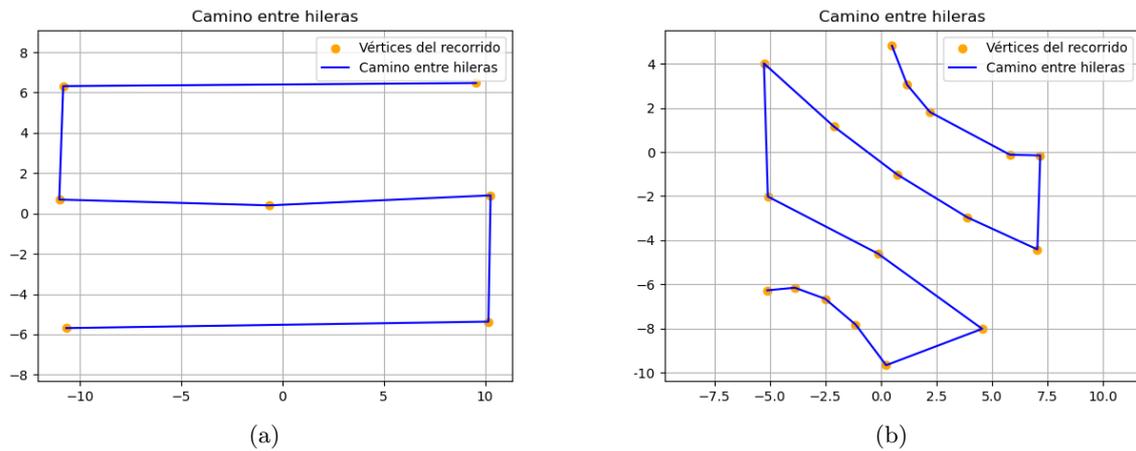


Figura 3.29: Camino entre hileras obtenido

### Inspección autónoma

Finalmente, el dron realiza el recorrido entre las hileras correspondientes, siguiendo la lista de puntos calculada previamente y tomando nuevas fotografías a lo largo del trayecto. Esta navegación se lleva a cabo bajo la hipótesis de que volar entre las hileras puede ofrecer una mejor capacidad de inspección y recolección de datos. Sin embargo, esta estrategia se implementa con fines experimentales, con el objetivo de evaluar su viabilidad y efectividad en comparación con el vuelo convencional.

---

<sup>8</sup>Ramer-Douglas-Pecker (Ramer, 1972)

## 3.2. Arquitectura del sistema

Para comprender la arquitectura general del sistema propuesto, es importante contextualizar brevemente las tecnologías empleadas. En este proyecto, el dron es controlado a través de PX4, una plataforma de código abierto ampliamente utilizada en sistemas de navegación aérea autónoma. Por otro lado, se utilizó ROS2 (Robot Operating System 2), un framework modular y distribuido diseñado para facilitar el desarrollo de sistemas robóticos complejos.

La comunicación entre ROS2 y PX4 se establece mediante un [middleware](#) llamado Micro XRCE-DDS, que permite el intercambio eficiente de mensajes entre el dron (cliente) y la estación de control (agente). Este mecanismo permite que los nodos desarrollados en ROS2 puedan recibir datos de navegación y sensores del dron, así como enviarle comandos de control de forma confiable y en tiempo real. (PX4 Autopilot, 2025)

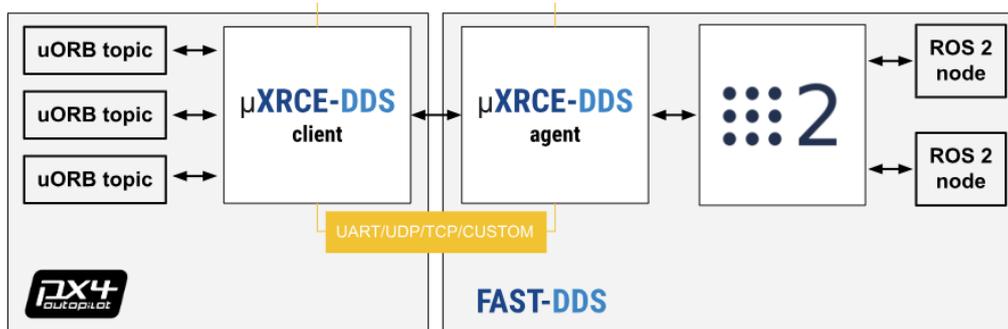


Figura 3.30: Diagrama de comunicación entre PX4 y ROS2, extraído de (PX4 Autopilot, 2025)

Sobre esta base, se diseñó una arquitectura de software compuesta por múltiples nodos ROS2, encargados de tareas específicas como la planificación de trayectorias, el control del vuelo, la toma de fotografías, y la ejecución de procesos externos. Uno de estos procesos es la reconstrucción del mapa de alturas, que se realiza mediante la herramienta OpenDroneMap (ODM), invocada desde un contenedor Docker. Esta integración permite automatizar por completo el flujo de trabajo, desde la captura inicial de imágenes hasta la generación de rutas óptimas entre hileras para el segundo vuelo.

En conjunto, nuestra arquitectura desarrollada combina tecnologías abiertas y robustas como ROS2, PX4 y Docker para resolver un problema real en el ámbito de la agricultura de precisión.

### 3.2.1. Diagramas de despliegue

Es importante mencionar que el entorno en el que probamos nuestro proyecto no es el mismo en el que esperamos que se utilice con un caso real. En nuestro caso utilizamos un entorno de simulación llamado Gazebo, el cual nos permitió realizar todas las pruebas necesarias sin ningún tipo de riesgo. Es por esto que, a continuación, presentamos dos diagramas de despliegue: uno correspondiente al entorno local de pruebas (figura 3.31) y otro correspondiente a un entorno real (figura 3.32).

Estos diagramas permiten visualizar la arquitectura física del sistema, mostrando cómo se distribuyen los nodos de ROS2 y cómo interactúan con componentes como PX4 y ODM. Además, permiten representar la interacción entre los nodos en los distintos entornos de ejecución, facilitando la comprensión de la infraestructura del sistema.

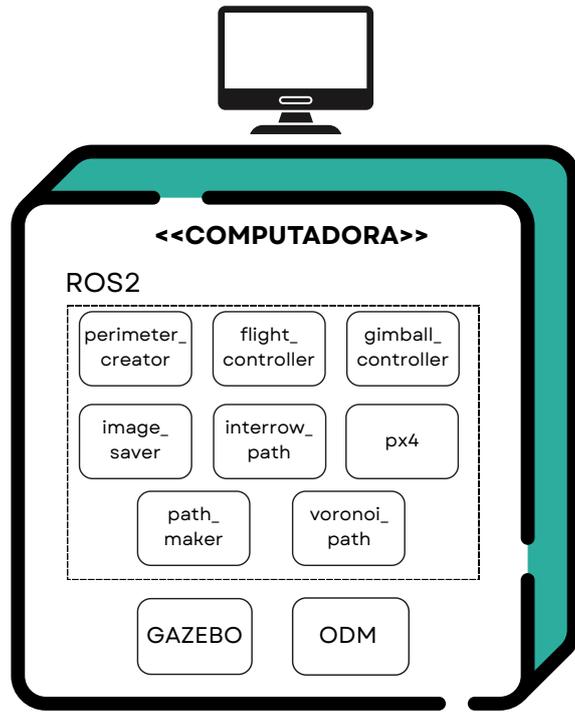


Figura 3.31: Diagrama de despliegue en un entorno simulado

En la figura 3.31, que muestra el diagrama de despliegue en un entorno simulado (en este caso, Gazebo), todos los componentes del sistema se encuentran integrados en una única arquitectura física: una computadora. Esto se debe a que tanto la simulación como el sistema de control se ejecutan localmente, es decir, dentro del mismo equipo. Por lo tanto, todos los módulos necesarios están instalados en esa máquina y se comunican internamente, sin necesidad de interacción con hardware externo o redes distribuidas.

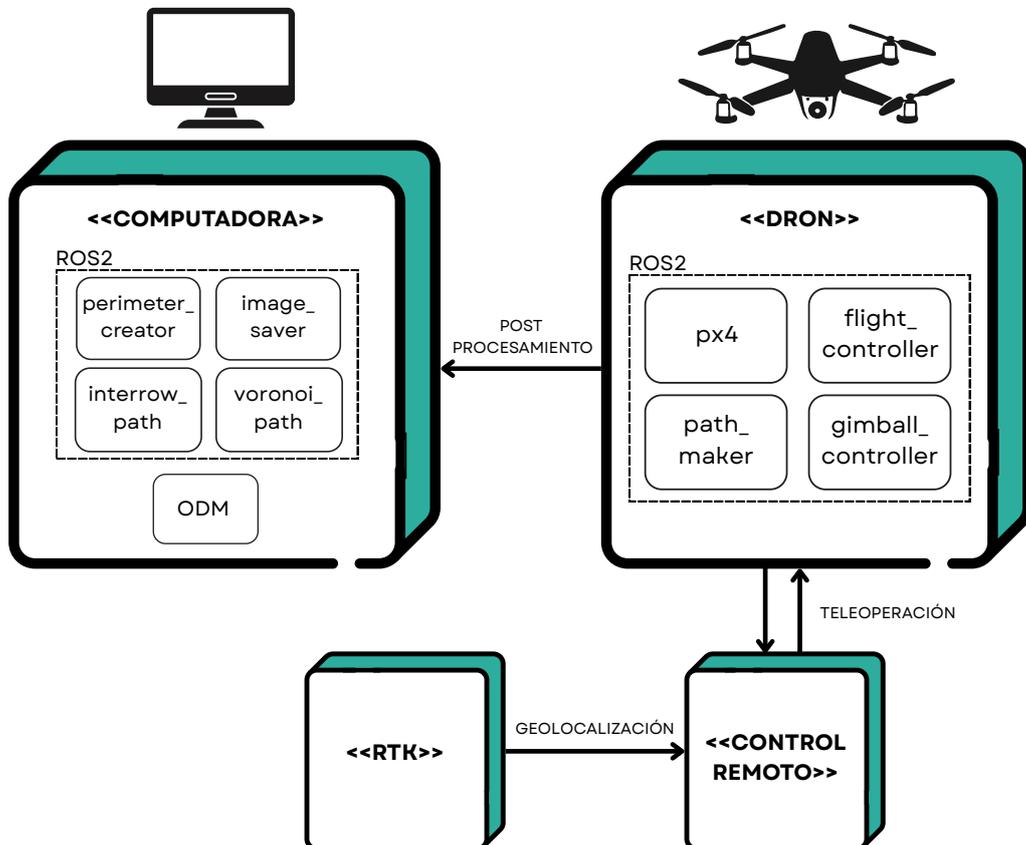


Figura 3.32: Diagrama de despliegue en un entorno real

En el caso del diagrama correspondiente a un entorno real representado por la figura 3.32, ya no tenemos una única arquitectura física ya que estamos trabajando con un dron real. En este caso, se utiliza una computadora para controlar el sistema, pero este solamente va a ejecutar los nodos de ROS2 encargados del procesamiento, mientras que los nodos encargados del vuelo se encuentran dentro del dron. Además, por otro lado, para realizar la teleoperación tenemos otro sistema físico aparte que es el control remoto que se comunica con el dron para poder ser teleoperado. Por último se necesita de un sistema RTK<sup>9</sup> para poder obtener la geolocalización del dron.

En el caso del diagrama correspondiente a un entorno real, representado en la figura 3.32, ya no se cuenta con una única arquitectura física, dado que el sistema se implementa sobre un dron real. En esta configuración, una computadora externa se encarga de ejecutar los nodos de ROS 2 responsables del procesamiento y la planificación, mientras que los nodos vinculados al control de vuelo se ejecutan directamente en la unidad de control del dron (por ejemplo, PX4).

Adicionalmente, se incorpora un tercer componente físico: el control remoto, que permite la teleoperación del dron mediante comunicación directa con su sistema de vuelo. Finalmente, se emplea un sistema de posicionamiento RTK para garantizar una geolocalización precisa del dron durante su operación en campo.

### 3.2.2. Diagrama de casos de uso

El diagrama de casos de uso es fundamental para comprender las interacciones entre los distintos actores y el sistema. En nuestra solución, consideramos especialmente relevante destacar el rol del teleoperador, dado que es el actor que interviene activamente en el proceso. En el diagrama 3.33 se puede observar cómo el teleoperador interactúa con el sistema, evidenciando las acciones y responsabilidades clave que desempeña dentro del flujo operativo.

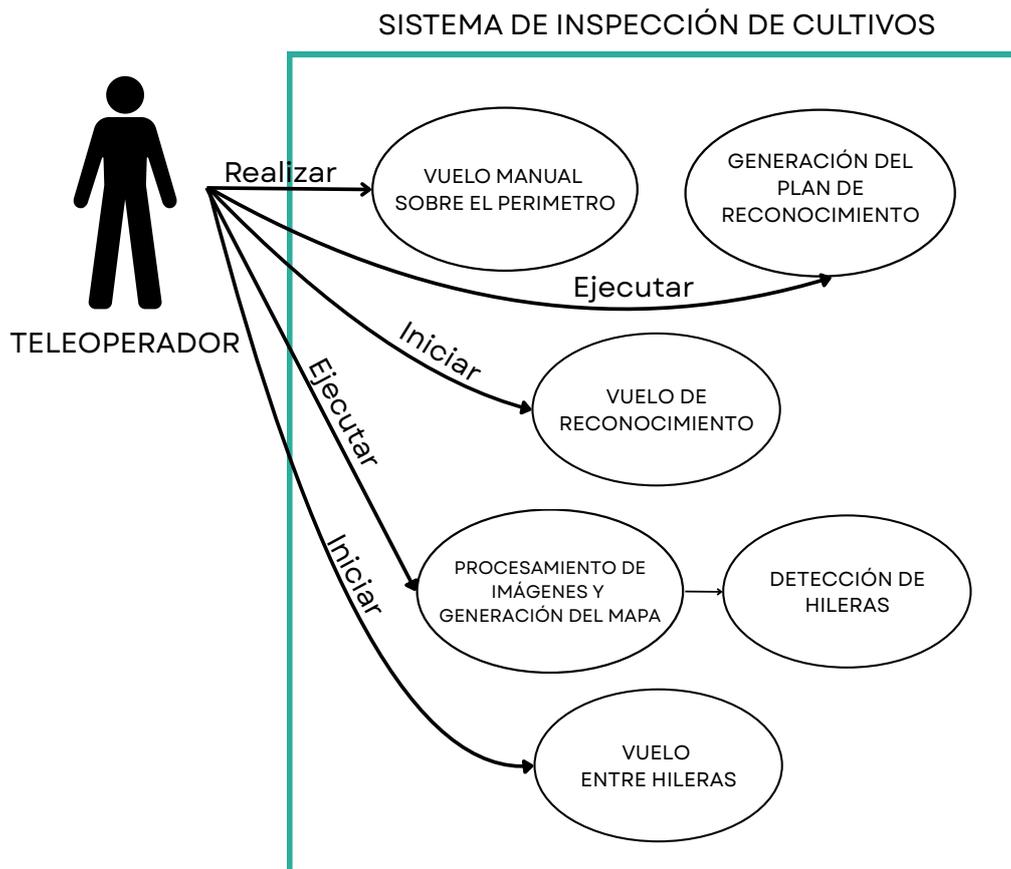


Figura 3.33: Diagrama de casos de uso

<sup>9</sup>Técnica de corrección diferencial del GPS que permite obtener una geolocalización de alta precisión, con márgenes de error en el orden de los centímetros.

## Capítulo 4

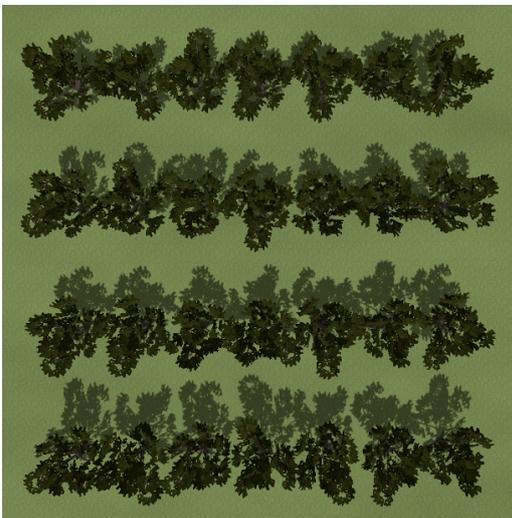
# Experimentación

Este apartado describe las pruebas realizadas para evaluar el desempeño del sistema en diferentes escenarios simulados. Los experimentos se organizan en categorías específicas, con diferentes objetivos.

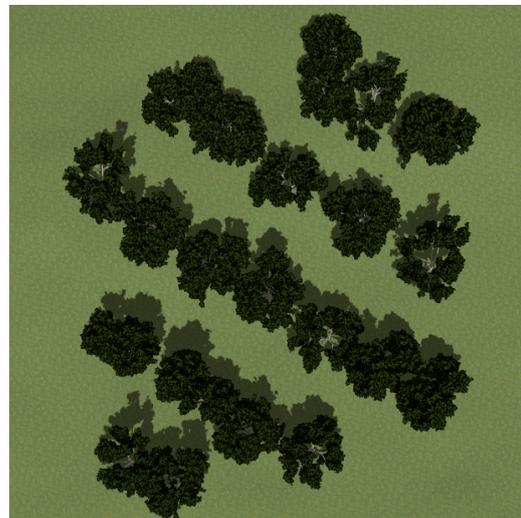
### Plataforma de pruebas

Para la simulación de los vuelos se utilizó Gazebo, un simulador de robótica avanzado que permite crear entornos virtuales realistas para probar y validar algoritmos sin necesidad de realizar experimentos en el mundo real. Esta herramienta permite simular sensores como cámaras, dispositivos Kinect, GPS, entre otros, lo que lo convierte en una opción ideal para el desarrollo y prueba de sistemas autónomos. En el contexto de este proyecto, Gazebo se empleó para recrear un entorno de cultivos y simular tanto el comportamiento del dron como el funcionamiento de sus sensores en distintos escenarios, permitiendo un control preciso sobre la trayectoria, la navegación y el procesamiento de imágenes.

Para probar la mayoría del sistema, se diseñaron dos mundos con algunos relieves, aunque predominantemente planos. Un mundo posee los árboles dispuestos de forma horizontal [4.1a](#) y el otro de forma oblicua [4.1b](#).



(a) Mundo con los árboles en forma horizontal



(b) Mundo con las árboles en forma oblicua

Figura 4.1: Mundos diseñados

## 4.1. Vuelo de reconocimiento

Con el fin de evaluar la consistencia del comportamiento del dron, se llevaron a cabo un total de 50 ejecuciones del vuelo de reconocimiento en cada mundo, registrando la altitud, el recorrido y el tiempo que tarda en realizarse cada ejecución.

### 4.1.1. Altitud

Durante cada vuelo de reconocimiento se registra la altura del dron, con el fin de comprobar si la solución mantiene una altura estable. En la figura 4.2 se presenta una gráfica correspondiente a cada cultivo, donde se observa como varía la altura durante cada instante del recorrido.

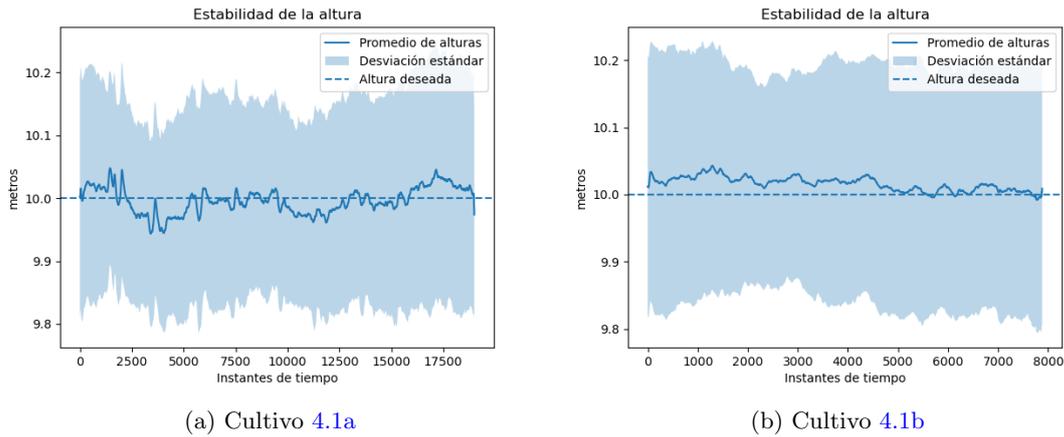


Figura 4.2: Promedio de alturas en cada cultivo

Como se observa en las gráficas, la altura se mantiene generalmente entre los 9,77 y 10,26 metros. Esta estabilidad en la altitud garantiza que las imágenes de profundidad capturadas tengan una escala y resolución similares, lo cual facilita la creación del ortomosaico.

Además, en cada instante, calculamos el error máximo cometido hacia abajo, bajo la hipótesis de que el error hacia arriba es menos importante, ya que, no habría ningún obstáculo. En la figura 4.3, podemos observar que el error hacia abajo máximo es de  $0,55m$  para ambos cultivos, concluyendo que el dron podría realizar el recorrido correctamente si no hay ningún obstáculo a menos de esa distancia.

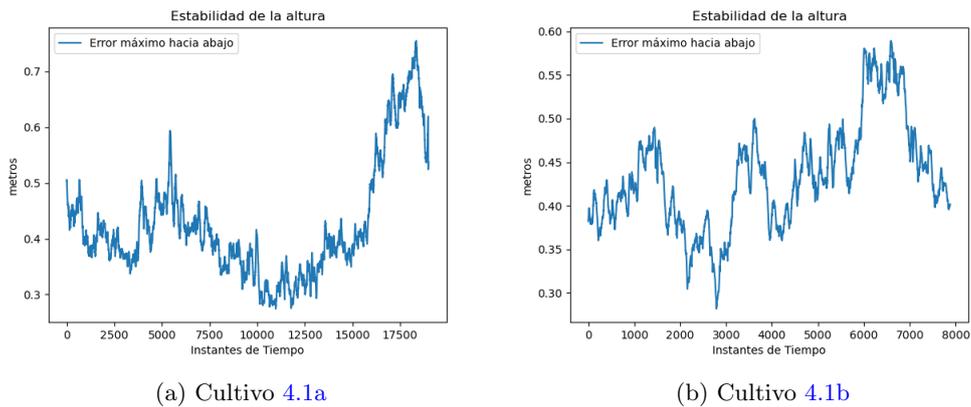


Figura 4.3: Máximo error hacia abajo

### 4.1.2. Recorrido

Para verificar que el recorrido se lleve a cabo correctamente, se comparan los recorridos realizados en cada ejecución con el recorrido calculado. En la figura 4.4 se muestran tanto las ejecuciones individuales como el recorrido planificado para cada cultivo.

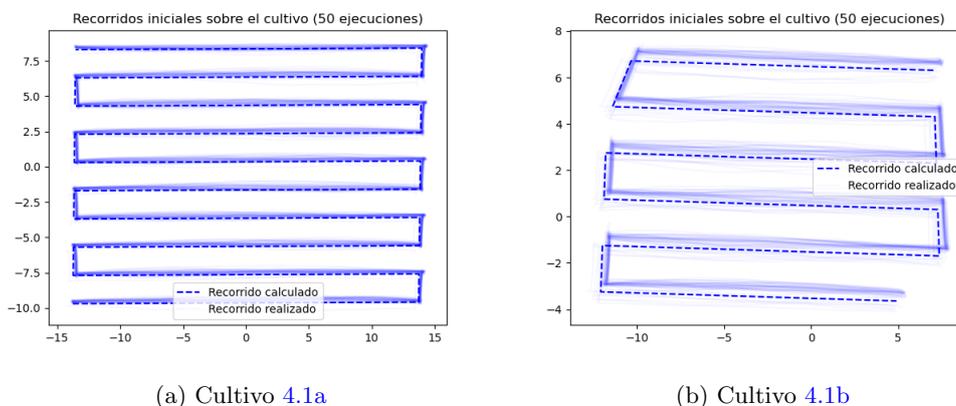


Figura 4.4: Recorrido realizado para cada cultivo

Aunque observemos que los recorridos realizados por el dron difieren levemente del planificado, la distancia entre cada pasada sobre el cultivo se mantiene, lo cual hace que la superposición de las imágenes de profundidad sea similar.

### 4.1.3. Tiempo de ejecución

En esta sección, se mide el tiempo requerido para ejecutar la misión completa, desde el primer vértice hasta el último, a una velocidad constante de  $2m/s$ . El tiempo promedio de ejecución para el cultivo 4.1a, que cubre una superficie de  $542,6m^2$ , fue de  $273,9s \pm 3,06s$ . Para el cultivo 4.1b, el tiempo promedio de ejecución fue de  $114,3s \pm 4,4s$  para completar el recorrido calculado, cubriendo una superficie de  $190,2m^2$ . A partir de estos datos, y considerando una distancia de  $2m$  entre cada paralela y un margen de  $1,5m$  para el gimbal, podemos calcular que el tiempo necesario para cubrir un metro cuadrado es de  $0,53s$ . Para evaluar la viabilidad de ejecutar un recorrido en un entorno real, tomamos como base un cultivo de aproximadamente  $5400m^2$  extraído de Google Maps<sup>1</sup>.

Con los cálculos realizados, lo que tardaría este recorrido en ejecutarse en un entorno real serían  $47,7 \pm 4$  minutos. Este recorrido sería viable para un dron como el *Matrice 350 RTK*<sup>2</sup> de la empresa *DJI*, que posee una autonomía teórica de 55 minutos (DJI, 2025).

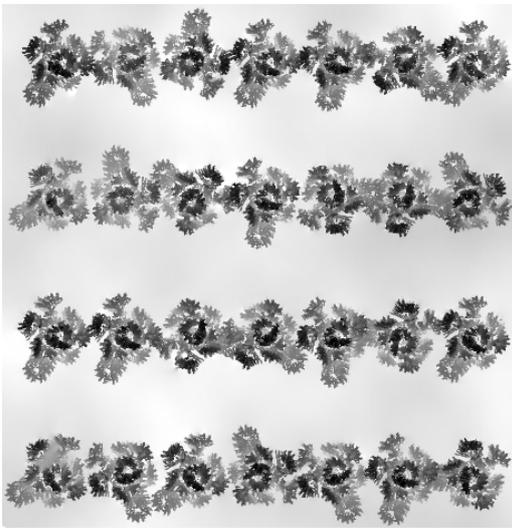
Para disminuir el tiempo de ejecución de la misión, y así asegurar que se pueda cubrir el área total del recorrido, se podría aumentar la velocidad del dron y la frecuencia en la que se captura una imagen de profundidad. Por ejemplo, si se incrementara la velocidad del dron al doble del valor actual ( $4m/s$ ), el tiempo de ejecución de la misión disminuirá a la mitad ( $23,35 \pm 2$  minutos). Sin embargo, debido a limitaciones en el sistema de simulación, no es posible ejecutar el sistema con estos parámetros. El tiempo requerido para la estabilización del gimbal impide que las imágenes se mantengan siempre orientadas hacia abajo, afectando así la calidad y precisión de la captura. Esto genera problemas en la generación del ortomosaico, ya que las imágenes mal orientadas pueden llevar a errores de superposición, afectando la alineación de las imágenes en el proceso de ensamblaje de ODM. Aun así, en un entorno real, esto podría llevarse a cabo sin mayores dificultades.

<sup>1</sup>Ubicación del cultivo:  $34^{\circ}40'04.0''S$   $56^{\circ}20'30.6''W$

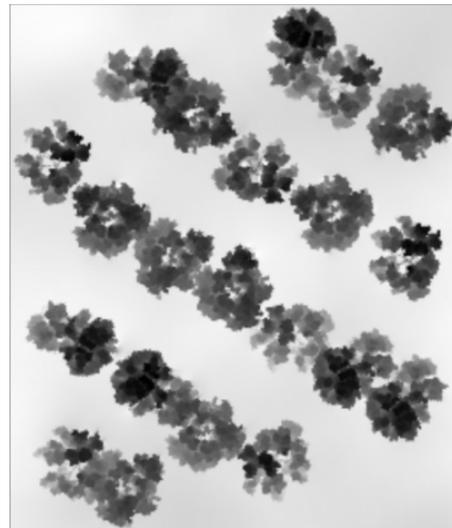
<sup>2</sup>La Facultad de Ingeniería posee este equipo

## 4.2. Generación del ortomosaico

Para evaluar la generación del mapa utilizamos varios mundos, para el caso de los de la figura 4.1 obtuvimos los ortomosaicos de la figura 4.5, observamos que estos mapas fueron generados correctamente.



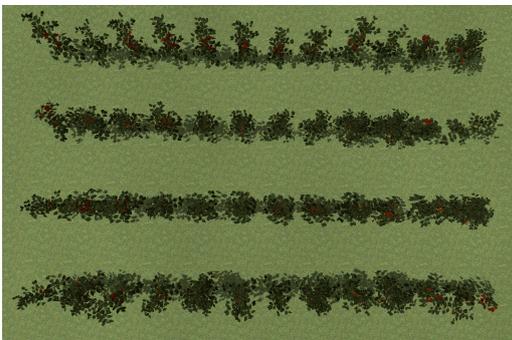
(a) Ortomosaico generado para 4.1a



(b) Ortomosaico generado para 4.1b

Figura 4.5: Error en generación del mapa

Otra prueba realizada fue con un mundo con plantas de tomates (4.6) y uno con árboles de manzanas pequeños (4.7), ambos con una altura máxima menor a 1 metro. Como estos cultivos eran de tamaño menor a los anteriores, se eligió volar a menor altura (5 metros).



(a) Imagen del cultivo original en RGB



(b) Mapa generado por ODM

Figura 4.6: Generación de mapa para cultivo de tomates



(a) Imagen del cultivo original en RGB



(b) Mapa generado por ODM

Figura 4.7: Generación de mapa para cultivo de manzanas

Como se observa en las figuras, hay limitaciones en la generación de estos mapas. Esto podría ser por la altura de los árboles, ya que cada árbol ocupa menos píxeles en las imágenes capturadas, reduciendo la información. Otro motivo podría ser que los árboles estaban algo separados entre sí dentro de la misma hilera, y al no tener copas de gran tamaño, podría generar variaciones significativas en las distintas alturas capturadas en las imágenes de profundidad.

### 4.3. Generación del recorrido entre hileras

El objetivo de esta prueba es cuantificar la desviación entre una ruta manual óptima entre las hileras y la ruta generada automáticamente. Se busca que la solución obtenida por el sistema se asemeje lo más posible a la ruta que consideramos óptima visualmente.

Para llevar a cabo esta evaluación, se desarrolló un algoritmo en Python utilizando la biblioteca OpenCV. Este algoritmo emplea la función [SSIM](#) (Structural Similarity Index), una métrica que compara dos imágenes y devuelve una imagen con las diferencias detectadas y una puntuación de similitud. La puntuación SSIM varía entre 0 y 1, donde un valor de 1 indica que las imágenes son idénticas y valores inferiores reflejan un mayor grado de diferencia (Z. Wang et al., 2004). Para esta prueba, se compararon las rutas generadas sin la superposición de los cultivos.

Se eligió SSIM porque considera la estructura espacial de las trayectorias. Esto es fundamental para evaluar rutas, donde la forma del recorrido es más importante que las diferencias puntuales en cada lugar, logrando capturar si una ruta mantiene la lógica estructural de la navegación entre las hileras.

Para estas pruebas establecimos un valor mínimo de SSIM de 0.75. Este valor fue seleccionado para asegurar que las rutas generadas por el sistema no difirieran en más de una cuarta parte de una solución óptima, lo que valida la robustez de nuestro enfoque para diversos tipos de trayectorias.

#### 4.3.1. Cultivos horizontales

Como se aprecia en la figura [4.8](#), ambas rutas son visualmente muy similares. El puntaje SSIM resultante de 0.875 confirma este parecido, indicando que la ruta generada por el sistema se asemeja de manera significativa a la trayectoria óptima, lo cual valida el cumplimiento de nuestro objetivo.

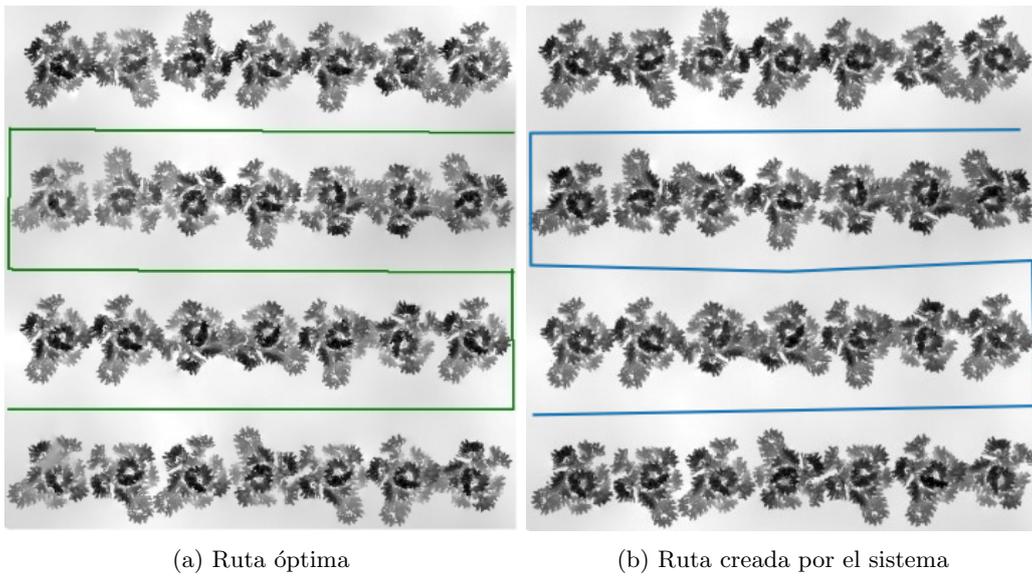


Figura 4.8: Comparación de rutas en un cultivo en horizontal

#### 4.3.2. Cultivos oblicuos

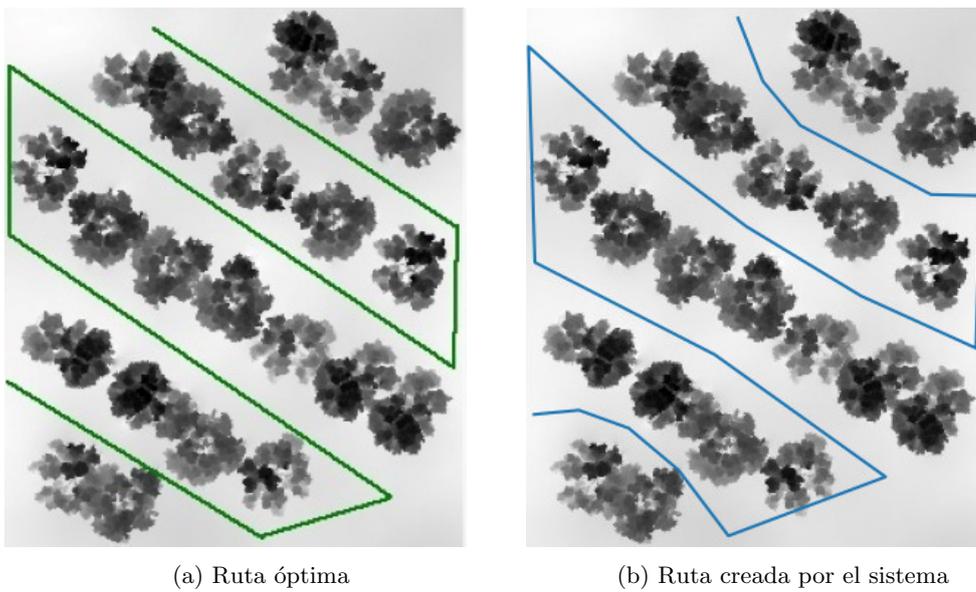


Figura 4.9: Comparación de rutas en un cultivo en diagonal

Para el caso de la figura 4.9, el puntaje SSIM resultante fue de 0.779. Aunque este valor es ligeramente inferior al caso 4.8, sigue indicando una similitud notable entre la ruta óptima y la generada por el sistema. Aun así, ambas comparaciones superaron el umbral de 0.75 SSIM establecido para la prueba.

### 4.4. Vuelo entre hileras

Para evaluar el funcionamiento del vuelo entre hileras, también realizamos 50 ejecuciones sobre cada cultivo de la figura 4.1. En este caso, no presentaremos las gráficas sobre la altura porque se mantiene en el margen previsto por la experimentación realizada anteriormente (4.1.1).

Como se observa en la figura 4.10, los caminos realizados por las ejecuciones del sistema son similares al recorrido calculado, por lo que podemos decir que realiza un recorrido correctamente. Para el cultivo 4.1a, la longitud del recorrido planificado es de 74,56m; y el promedio del tiempo de ejecución es de 61,59s con una desviación estándar de 0,97s. En cambio, para el cultivo 4.1b

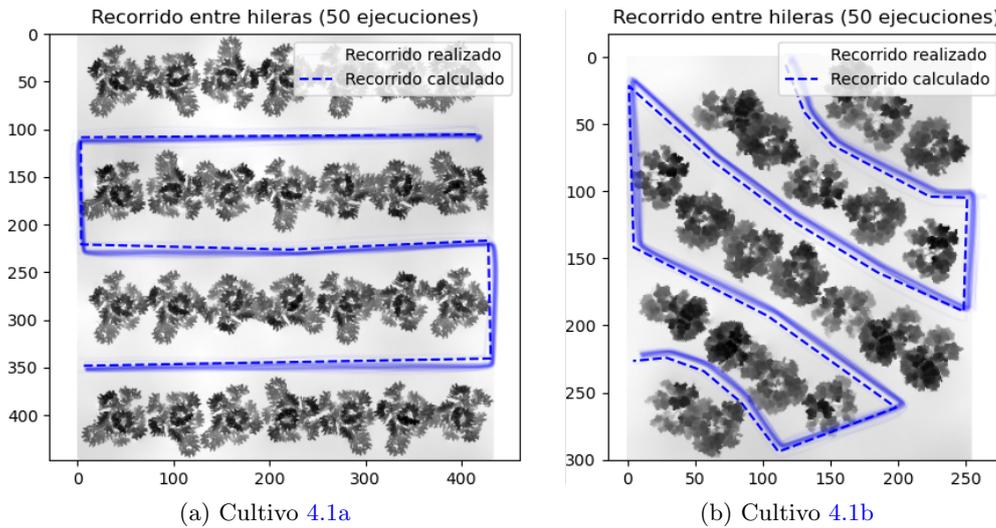


Figura 4.10: Camino entre hileras en cada cultivo

la longitud del recorrido es de  $57,06m$ , con un promedio del tiempo de ejecución de  $58,58s$  y la desviación estándar es de  $3,5s$ .

La leve diferencia entre los tiempos de ejecución y la gran diferencia en la longitud del recorrido se debe a que el recorrido más corto posee una mayor cantidad de vértices, cuando el dron alcanza un nuevo vértice, disminuye su velocidad para lograr alcanzarlo con una mejor precisión.

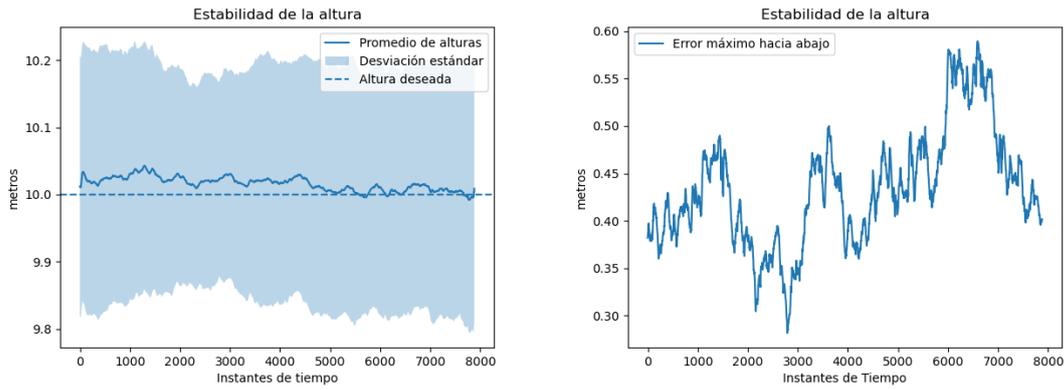
En cuanto al mundo real, tomando como referencia el cultivo mencionado anteriormente, que su largo de hilera es  $60m$  y tiene  $23$  hileras de árboles (dispuestas de manera paralela al borde de la imagen), por lo que se tendrían que hacer  $21$  pasadas en el recorrido. Tomando estos datos como base, tenemos que el recorrido sobre este cultivo sería de  $1260m$ , tomando en cuenta el tiempo que tarda en realizar el recorrido para el cultivo [4.1a](#), el tiempo que tardaría en ejecutarse para este cultivo es de  $1040s$  ( $17$  minutos). Por lo tanto, se tiene un indicador sobre la viabilidad de realizar el vuelo en un escenario real.

## 4.5. Ejecuciones con viento

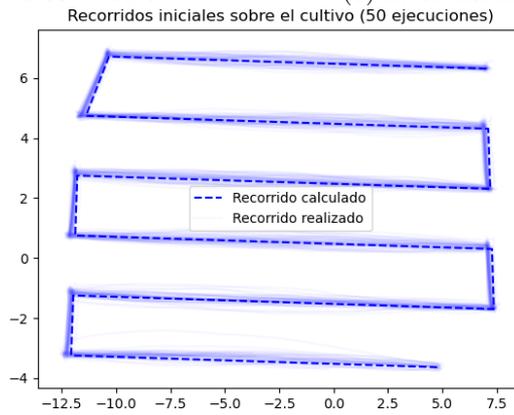
Para evaluar el desempeño en una simulación más real se realizaron  $50$  ejecuciones con viento para el cultivo [4.1b](#). El viento se configuró con un vector de velocidad constante  $(6, 6, 6)$ , lo que representa una velocidad de  $6m/s$  en cada eje del espacio ( $X$ ,  $Y$  y  $Z$ ). Esta configuración genera una magnitud total de aproximadamente  $10,39m/s$ , equivalente a unos  $37,4km/h$ . Este valor es coherente con condiciones reales de operación, ya que drones como el *Matrice 350 RTK* pueden tolerar vientos de hasta  $12m/s$  (DJI, [2025](#)).

En la figura [4.11](#) se presentan los resultados obtenidos para este cultivo. Se observa que la altura generalmente varía entre  $9,73$  y  $10,24$  metros, teniendo un máximo hacia abajo de  $0,49m$ , siendo menor a cuando realizamos las pruebas sin viento. Esto puede deberse al funcionamiento de PX4 que contrarresta el viento aumentando la inclinación del dron y volando a una mayor altura en promedio.

En este caso, realizar el recorrido lleva un promedio de  $180,42 \pm 8,9$  segundos, un valor que es aproximadamente un minuto mayor sobre el tiempo calculado anteriormente. Además, se cubre un  $m^2$  en  $0,94$  segundos. A pesar de esta diferencia de tiempo, se observa que el recorrido se mantiene cercano al tiempo estimado inicialmente, conservando las distancias entre las hileras de manera adecuada.



(a) Promedio de alturas - viento (b) Error máximo hacia abajo - viento



(c) Recorridos realizados - viento

Figura 4.11: Resultados obtenidos en mundo 4.1b con viento

También realizamos 50 ejecuciones para el vuelo entre hileras, cuyos resultados se presentan en la figura 4.12. En este caso, observamos que el recorrido se realiza más cerca de las hileras que cuando no hay viento, tardando  $60,10 \pm 1,3$  segundos en promedio ( $\approx 2$  segundos más que sin viento).

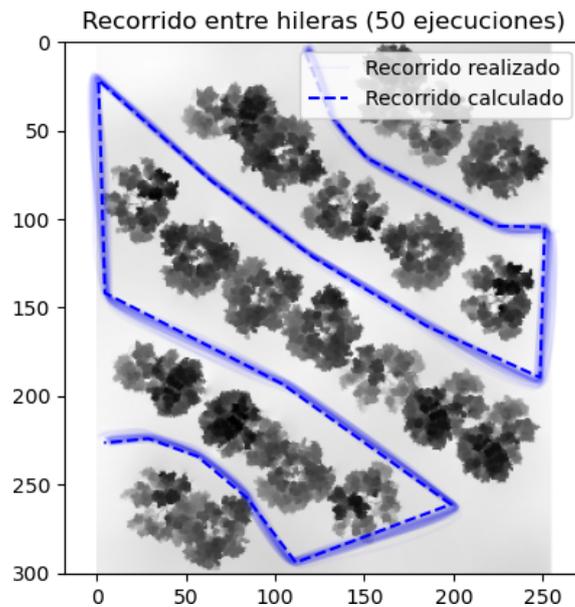


Figura 4.12: Vuelo entre hileras - viento

Generalizando los datos obtenidos anteriormente, para un cultivo real de  $5400m^2$  con una velocidad del viento de  $10,39m/s$ , el vuelo de reconocimiento tardaría 84,6 minutos. Realizando nuevamente la adaptación descrita en 4.1.3 reducimos el tiempo hasta 42,3 minutos. El vuelo entre

hileras pasaría a tardar 20 minutos en este nuevo escenario, por lo que el sistema seguiría siendo viable con viento.

## Capítulo 5

# Conclusiones y Trabajo Futuro

El desarrollo de este proyecto permitió explorar y validar una solución autónoma en un entorno simulado para la inspección de cultivos utilizando drones realizando planificación de rutas y procesamiento de imágenes. En este capítulo se presentan las conclusiones alcanzadas, una reflexión crítica sobre el proceso y posibles líneas de trabajo que podrían continuarse en el futuro para mejorar y expandir el sistema propuesto.

### 5.1. Contribuciones

Como aporte fundamental, se desarrolló una arquitectura que en principio permite la navegación autónoma de drones para la inspección agrícola, con capacidad para capturar y procesar imágenes georreferenciadas que facilitan el análisis detallado de los cultivos. Para facilitar la reproducibilidad y el seguimiento del proyecto, el código fuente está disponible públicamente en el siguiente repositorio de GitLab: <https://gitlab.fing.edu.uy/pginspcultivos>.

### 5.2. Conclusiones

Los resultados obtenidos en la sección anterior permitieron evidenciar que los objetivos principales planteados inicialmente fueron alcanzados en un entorno simulado. Se logró desarrollar un sistema funcional capaz de integrar diversas tecnologías para generar recorridos sobre cultivos en los que los árboles están dispuestos de forma recta, ya sea en orientación horizontal, vertical u oblicua. Estos recorridos se generan mediante vuelos autónomos simulados, cuyas trayectorias fueron diseñadas para asegurar una cobertura completa del terreno, sin solapamientos innecesarios ni colisiones con el cultivo. La fiabilidad de esta solución se confirma a través de los resultados experimentales, donde los promedios de altura y las trayectorias realizadas se mantuvieron dentro de los parámetros definidos, garantizando que no se presentaran riesgos de colisión.

El sistema fue creado para que se adapte a diferentes tipos de cultivos, variando parámetros clave, como la distancia entre las filas y la orientación de las hileras. La experimentación realizada en simulación con distintas configuraciones demostró que la solución ajusta dinámicamente la planificación de trayectorias, sugiriendo una mayor robustez que deberá ser validada en condiciones reales.

Además, es crucial destacar la importancia de que el dron cuente con una cámara de altura incorporada, ya que el contraste en las imágenes, generado por la diferencia de alturas entre las hileras de árboles y el suelo, resultó fundamental para diferenciar ambos elementos. Esto evita tener que lidiar con variaciones de color del suelo que pudieran parecer similares a las del cultivo. De este modo, se optimiza el procesamiento y se evita el uso de algoritmos adicionales de corrección de colorimetría, lo que simplifica la carga computacional y mejora la eficiencia general del sistema.

En resumen, el desarrollo de la solución representa un primer paso hacia un avance en la mejora de la eficiencia y sostenibilidad de las operaciones agrícolas, representando un paso fundamental hacia la automatización de la detección de malezas, con potencial para optimizar recursos, aumentar la eficiencia operativa y disminuir costos.

### 5.3. Dificultades encontradas

Durante el desarrollo del proyecto nos enfrentamos a una serie de dificultades que complicaron considerablemente el avance del trabajo. Una de las principales barreras fue la falta de experiencia previa en robótica por parte del equipo. Esto implicó una curva de aprendizaje pronunciada, ya que fue necesario dedicar una cantidad significativa de tiempo a adquirir conocimientos sobre los distintos componentes tecnológicos utilizados, los cuales eran completamente nuevos para nosotros.

Asimismo, la instalación y configuración de las tecnologías involucradas resultó ser una tarea compleja. Nos vimos afectados en múltiples ocasiones por actualizaciones de software que modificaban el comportamiento del código, lo que nos obligó a realizar constantes ajustes en el entorno de trabajo para mantener la compatibilidad entre las herramientas.

Otro desafío importante fue la escasa documentación disponible para resolver problemas técnicos muy específicos. Si bien muchas de las herramientas empleadas cuentan con comunidades activas, eran escasos los casos en que se utilizaban de manera integrada, como lo hicimos en este proyecto.

Finalmente, uno de los obstáculos más significativos fue la generación del ortomosaico, aspecto que ya ha sido abordado en detalle en capítulos anteriores. La búsqueda de un software que cumpliera con los requisitos del proyecto demandó varios meses de prueba y error. A pesar de evaluar diversas alternativas, no logramos una solución confiable para todos los tipos de cultivo. Particularmente para las plantaciones donde el tamaño de los árboles era muy pequeño o estaban separados entre sí dentro de la misma hilera, los resultados obtenidos fueron insatisfactorios, incluso luego de contactar con los equipos de soporte de las herramientas principales empleadas (ODM y PX4), sin recibir una respuesta concluyente que resolviera el problema.

### 5.4. Trabajos a futuro

En primer lugar, sería bueno mejorar el algoritmo utilizado para la detección de caminos entre hileras para lograr una mayor robustez y funcione bien aunque haya lugares en las hileras donde falten árboles y no dependa tanto de las copas de los árboles. Para esto se podría intentar seguir con los enfoques de detectar una línea representativa de la hilera y trazar paralelas por los espacios entre ellas.

Luego, se deberían realizar pruebas exhaustivas en condiciones reales para validar la efectividad del sistema propuesto. Incluyendo diferentes tipos de cultivos, con distintos tipos de terreno y condiciones meteorológicas.

Una vez validado el sistema en condiciones reales, habiendo cumplido con el objetivo principal del proyecto (desarrollar la infraestructura para la planificación de trayectorias), uno de los próximos pasos consiste en la integración de un sistema de análisis basado en inteligencia artificial que permita identificar regiones con alta probabilidad de presencia de malezas, logrando el objetivo más amplio de detección automática de malezas. La idea es desarrollar una herramienta capaz de generar una malla de probabilidad sobre cada ortomosaico generado, marcando con mayor intensidad aquellas áreas donde se sospecha la aparición de malezas. Para ello, podría utilizarse un modelo de segmentación entrenado específicamente con imágenes etiquetadas de cultivos locales. Sin embargo, la falta de modelos preentrenados adecuados, sumada al desafío de construir un conjunto de datos propio dentro de los plazos establecidos, impidió concretar esta fase durante el presente trabajo.

Como trabajo extra, se investigaron diversas herramientas y se identificó a YOLO<sup>1</sup> como uno de los modelos más prometedores para la detección de malezas. YOLO es una herramienta rápida y precisa y es utilizada en aplicaciones de monitoreo en campo donde se requiere detección en tiempo real. Además, existen versiones preentrenadas en conjuntos de datos como COCO (Ultralytics, 2023) o PlantVillage (Palacio, 2021), que podrían adaptarse a la detección de malezas. Este enfoque permitiría realizar una detección rápida y eficaz, adecuándose a las necesidades de un sistema autónomo en un entorno agrícola.

Por otro lado, consideramos que una de las principales áreas de mejora sería el desarrollo o la integración de un software más flexible y preciso para la generación de mapas de alturas, que sea capaz de adaptarse a diferentes tipos de cultivos. Esto podría implicar la utilización de software comercial como Pix4D, que no solo ofrece una mayor capacidad en comparación con ODM, sino que también proporciona una mayor variedad de configuraciones y un soporte técnico dedicado para resolver cualquier tipo de duda o inconveniente.

---

<sup>1</sup>YOLO (You Only Look Once) es un modelo de detección de objetos en imágenes y videos que predice las ubicaciones de los objetos en una sola pasada

# Bibliografía

- Abdulsalam, M., Ahiska, K., & Aouf, N. (2023). A novel UAV-integrated deep network detection and relative position estimation approach for weeds. *Proceedings of the Institution of Mechanical Engineers, Part G*, 237(10), 2211-2227. <https://doi.org/10.1177/09544100221150284>
- Ajayi, O. G., Ashi, J., & Guda, B. (2023). Performance evaluation of YOLO v5 model for automatic crop and weed classification on UAV images. *Smart Agricultural Technology*, 5, 100231. <https://doi.org/https://doi.org/10.1016/j.atech.2023.100231>
- Analytics, N. (2022). *Ultimate Guide: Voronoi Diagrams in Geometric Algorithms*. <https://www.numberanalytics.com/blog/ultimate-guide-voronoi-diagrams-geometric-algorithms>
- Bouguettaya, A., Zarzour, H., Kechida, A., & Taberkit, A. M. (2023). A survey on deep learning-based identification of plant and crop diseases from UAV-based aerial images. *Cluster Computing*, 26, 1297-1317. <https://doi.org/10.1007/s10586-022-03627-x>
- Castelar Wembers, C., Pflughaupt, J., Moshagen, L., Kurenkov, M., Lewejohann, T., & Schildbach, G. (2024). LiDAR-based automated UAV inspection of wind turbine rotor blades. *Journal of Field Robotics*, 41(4), 1116-1132. <https://doi.org/https://doi.org/10.1002/rob.22309>
- de Investigación Agropecuaria (INIA), I. N. (2020). *Situación actual de malezas problemáticas en campo natural y propuestas para su combate* (Serie Técnica N.º AD-420) (Informe elaborado para la Mesa de Ganadería sobre Campo Natural (MGAP)). INIA Uruguay. <https://ainfo.inia.uy/digital/bitstream/item/15860/1/ad-420.pdf>
- Dellepiane, A. Q., Kaspariy, T. E., & Latasa, M. A. G. (2023). *Capítulo 6: Malezas en sistemas ganaderos y agrícolas* (Informe técnico N.º Aportes científicos y tecnológicos, Capítulo 6) (En: García-Inza, G.P.; Paruelo, J.M.; Zoppolo, R. (Eds.), \*Aportes científicos y tecnológicos del INIA-Uruguay a las trayectorias agroecológicas\*). Instituto Nacional de Investigación Agropecuaria (INIA) Uruguay. <https://www.inia.uy/sites/default/files/publications/2024-10/Capitulo6Malezasensistemasganaderosyagricolas.pdf>
- DJI. (2025). *Matrice 350 RTK*. <https://enterprise.dji.com/matrice-350-rtk/specs>
- Gao, M., Yang, F., Wei, H., & Liu, X. (2023). Automatic Monitoring of Maize Seedling Growth Using Unmanned Aerial Vehicle-Based RGB Imagery. *Remote Sensing*, 15(14). <https://doi.org/10.3390/rs15143671>
- Gokool, S., Mahomed, M., Clulow, A., Sibanda, M., Kunz, R., Naiken, V., & Mabhaudhi, T. (2024). Exploring the Potential of Remote Sensing to Facilitate Integrated Weed Management in Smallholder Farms: A Scoping Review. *Drones*, 8(3). <https://doi.org/10.3390/drones8030081>
- International Civil Aviation Organization (ICAO). (2011, abril). *Circular 328-AN/190: Unmanned Aircraft Systems (UAS)* (Circular N.º 328-AN/190) (Revised and available in multiple languages). International Civil Aviation Organization. <https://www.icao.int/meetings/uas/documents/circular%20328-en.pdf>
- Istiak, M. A., Syeed, M. M., Hossain, M. S., Uddin, M. F., Hasan, M., Khan, R. H., & Azad, N. S. (2023). Adoption of Unmanned Aerial Vehicle (UAV) imagery in agricultural management: A systematic literature review. *Ecological Informatics*, 78, 102305. <https://doi.org/https://doi.org/10.1016/j.ecoinf.2023.102305>
- Kim, C., Lim, P.-C., & Kim, T. (2023). ESTIMATION OF SENSOR OFFSETS FOR A UAV PLATFORM USING TIEPOINTS ONLY. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-1/W2-2023, 1707-1713. <https://doi.org/10.5194/isprs-archives-XLVIII-1-W2-2023-1707-2023>
- Kim, D., & Moon, I. (2024). Scheduling-location problem with drones. *International Transactions in Operational Research*, 31(5), 2850-2874. <https://doi.org/https://doi.org/10.1111/itor.13423>
- Krestenitis, M., Raptis, E. K., Kapoutsis, A. C., Ioannidis, K., Kosmatopoulos, E. B., & Vrochidis, S. (2024). Overcome the Fear Of Missing Out: Active sensing UAV scanning for precision

- agriculture. *Robotics and Autonomous Systems*, 172, 104581. <https://doi.org/https://doi.org/10.1016/j.robot.2023.104581>
- Meesaragandla, S., Jagtap, M. P., Khatri, N., Madan, H., & Vadduri, A. A. (2024). Herbicide spraying and weed identification using drone technology in modern farms: A comprehensive review. *Results in Engineering*, 21, 101870. <https://doi.org/https://doi.org/10.1016/j.rineng.2024.101870>
- Ministerio de Ganadería, A. y. P. (2025). *Situación actual de las malezas problemáticas y propuestas para su combate a nivel nacional* [Accedido: 2025-01-05]. <https://www.gub.uy/ministerio-ganaderia-agricultura-pesca/politicas-y-gestion/situacion-actual-se-malezas-problematicas-propuestas-para-su-combate-nacional>
- Ministerio de Ganadería, Agricultura y Pesca (MGAP). (2023). *Situación actual de malezas problemáticas y propuestas para su combate nacional*. <https://www.gub.uy/ministerio-ganaderia-agricultura-pesca/politicas-y-gestion/situacion-actual-se-malezas-problematicas-propuestas-para-su-combate-nacional>
- Open Robotics. (2022). *ROS 2 Documentation*. <https://docs.ros.org/en/humble>
- Ortataş, F. N., Ulutaş, H., ŞahİN, M. E., & Çİftçİ, F. (2023). Autonomous Mapping and Spraying in Precision Agriculture using Unmanned Ground Vehicles. *2023 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 1-5. <https://doi.org/10.1109/ASYU58738.2023.10296767>
- Palacio, S. (2021). *PlantVillage for Object Detection (YOLO)*. <https://www.kaggle.com/datasets/sebastianpalaciob/plantvillage-for-object-detection-yolo/data>
- Pizard, S., Acerenza, F., Casella, V., Moreno, S., García, R., Lezama, J., & Vallespir, D. (2019). *Conceptos de ingeniería de software basada en evidencias: Versión 2*. Consultado el 4 de mayo de 2025, desde <https://www.colibri.udelar.edu.uy/jspui/handle/20.500.12008/25557>
- Pranaswi, D., Jagtap, M. P., Shinde, G. U., Khatri, N., Shetty, S., & Pare, S. (2024). Analyzing the synergistic impact of UAV-based technology and knapsack sprayer on weed management, yield-contributing traits, and yield in wheat (*Triticum aestivum* L.) for enhanced agricultural operations. *Computers and Electronics in Agriculture*, 219, 108796. <https://doi.org/https://doi.org/10.1016/j.compag.2024.108796>
- PX4 Autopilot. (2024). *Gazebo Vehicles*. [https://docs.px4.io/main/en/sim\\_gazebo\\_gz/vehicles.html#x500-quadrotor-with-gimbal-front-facing](https://docs.px4.io/main/en/sim_gazebo_gz/vehicles.html#x500-quadrotor-with-gimbal-front-facing)
- PX4 Autopilot. (2025). *ROS 2 User Guide*. [https://docs.px4.io/main/en/ros2/user\\_guide.html](https://docs.px4.io/main/en/ros2/user_guide.html)
- Ramer, U. (1972). An Iterative Procedure for the Polygonal Approximation of Plane Curves. *Computer Graphics and Image Processing*, 1(3), 244-256.
- Raptis, E. K., Englezos, K., Kypris, O., Krestenitis, M., Kapoutsis, A. C., Ioannidis, K., Vrochidis, S., & Kosmatopoulos, E. B. (2023). CoFly: An automated, AI-based open-source platform for UAV precision agriculture applications. *SoftwareX*, 23, 101414. <https://doi.org/https://doi.org/10.1016/j.softx.2023.101414>
- Soludrones. (2024). *¿Cómo funcionan las cámaras de los Drones?* <https://www.soludrones.cl/camaras-drones-4k-alta-resolucion>
- Ultralytics. (2023). *Documentación de Datasets COCO para detección de objetos*. <https://docs.ultralytics.com/es/datasets/detect/coco/>
- Verdú, C. M. J. (2023). *Enabling Rolling Shutter Optical Camera Communication Using Artificial Intelligence: Towards Widespread Adoption and Dual Use of Cameras as Receivers* [PhD thesis]. Universidad de Las Palmas de Gran Canaria. <https://acedacris.ulpgc.es/handle/10553/121553>
- Wang, Y., Duan, C., Huang, X., Zhao, J., Zheng, R., & Li, H. (2023). Task-Driven Path Planning for Unmanned Aerial Vehicle-Based Bridge Inspection in Wind Fields. *Fluids*, 8(12). <https://doi.org/10.3390/fluids8120321>
- Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4), 600-612. <https://doi.org/10.1109/TIP.2003.819861>

# Anexo A

## A.1. ROS 2

ROS (Robot Operating System) es un sistema de código abierto que ofrece un conjunto de bibliotecas y herramientas para construir aplicaciones robóticas. Estas permiten integrar elementos físicos en los robots (sensores, cámaras) y algoritmos, permitiendo la implementación de sistemas robóticos complejos. ROS 2 es una versión que mejora aspectos fundamentales de la arquitectura original, proporcionando mayor robustez, seguridad y rendimiento.

La arquitectura de ROS 2 se basa en nodos independientes que se comunican a través de varios tipos de mensajes como por ejemplo servicios y tópicos. Los servicios permiten una comunicación sincrónica basada en RPC y los tópicos son una forma de comunicación asincrónica entre nodos que permite que puedan intercambiar datos.

Para el desarrollo del proyecto se utilizó ROS 2 Humble Hawksbill. Esta distribución soporta varias plataformas, como Ubuntu 22.04 o Windows 10. En nuestro caso utilizamos Ubuntu 22.04. (Open Robotics, 2022)

# Glosario

**bag** Herramienta que permite grabar y reproducir simulaciones de ROS2, almacenando datos de mensajes. [13](#)

**middleware** Software que actúa como intermediario para que diferentes aplicaciones se comuniquen entre sí. [30](#)

**nodo** Unidad de ejecución de ROS2, es un programa independiente que realiza una tarea específica. [13](#)

**ortomosaico** Imagen detallada y georreferenciada de un lugar, creada mediante la unión de fotos de alta resolución tomadas desde drones u otras plataformas aéreas. [19](#)

**RGBD** Tipo de imagen que combina información de color (RGB) y profundidad (D). Cada píxel contiene los valores de color en los canales rojo, verde y azul y un valor que indica la distancia de ese punto respecto al sensor. [10](#)

**RTK** Técnica de corrección diferencial del GPS que permite obtener una geolocalización de alta precisión, con márgenes de error en el orden de los centímetros. [32](#)

**sqlite3** Sistema de gestión de bases de datos relacionales. [13](#)

**SSIM** Índice de similitud estructural que compara dos imágenes basándose en la similitud estructural percibida, considerando luminancia, contraste y estructura. Su valor va de -1 a 1, donde 1 indica imágenes idénticas. [37](#)

**tópico** Forma de comunicación entre nodos que permite que los nodos puedan intercambiar datos. Los nodos pueden publicar (enviar) información a un tema, y otros nodos pueden suscribirse. [18](#)