

Sistema de carga modular y aterrizaje de precisión para enjambre de drones

José Alberto Castro Villasana, José Eduardo Castro Villasana,
Ana Bárbara Quintero García

Una Tesis presentada a la Facultad de Ingeniería en Conformidad con los
Requisitos para el Grado de:
Ingeniero en Tecnologías Electrónicas y Robótica, Ingeniero en Mecatrónica



UDEM

Universidad de Monterrey
Departamento de Ingeniería y Tecnologías
San Pedro Garza García, México

30 de noviembre 2024

Asesor: Fermín Castro Aragón

Contents

1	Introducción	3
1.1	Contexto General y Motivación	3
1.1.1	Contexto del Problema	3
1.1.2	Importancia del Problema	4
1.2	Planteamiento del Problema	4
1.2.1	Definición Clara del Problema	4
1.2.2	Preguntas de Investigación	4
1.3	Objetivos	5
1.3.1	Objetivo General	5
1.3.2	Objetivos Específicos	5
1.4	Justificación	5
1.4.1	Relevancia del Proyecto	5
1.4.2	Impacto Potencial	6
1.5	Justificación	6
1.5.1	Relevancia del Proyecto	6
1.5.2	Impacto Potencial	7
1.6	Alcance y Limitaciones	7
1.6.1	Alcance del Trabajo	7
1.6.2	Limitaciones del Estudio	7
1.7	Estructura de la Tesis	8
2	Estado del Arte	9
2.1	Estado del Arte	9
2.2	Hardware	9
2.2.1	Drones	9
2.2.2	Charging Stations	10
2.2.3	Automatic Drawer Mechanism	11
2.2.4	Motors and Propellers	11
2.3	Electronics	13
2.3.1	Companion Computer	13
2.3.2	Flight Controllers	13
2.4	Software	13
2.4.1	FC Firmware	13
2.4.2	Ground Control Systems	15
2.4.3	Companion Computer Operative Systems	16
2.4.4	ROS2 Distributions	17
2.4.5	ROS2 Documentation	18

2.4.6	Computer Vision	20
2.4.7	What is an ArUco?	21
2.4.8	Aruco vs Embedded Aruco	22
2.4.9	Aruco Detection	23
3	Desarrollo	26
3.0.1	Diseño de la Estación de Carga	26
3.1	Ensamble de Estructura de la Estación de Carga	26
3.2	Armado de Mecanismo de Poleas	26
3.3	Electronica para movimiento del cajón	26
3.4	Programación para control de movimiento del cajón	26
3.5	Armado del Drone	26
4	Resultados	27
5	Conclusiones	28

Chapter 1

Introducción

1.1 Contexto General y Motivación

1.1.1 Contexto del Problema

En los últimos años, los drones han evolucionado para convertirse en herramientas esenciales en una amplia gama de industrias. Estos vehículos aéreos no tripulados varían en tamaño y capacidad, pero comparten la característica de acceder a áreas difíciles o peligrosas, lo que ha impulsado su uso en sectores como la inspección de infraestructuras, la agricultura de precisión y las operaciones de vigilancia y seguridad. A nivel comercial, los drones están diseñados para realizar tareas repetitivas de manera autónoma, lo que ha revolucionado la eficiencia en dichas áreas.

En el ámbito de la inspección de infraestructuras, los drones son utilizados para monitorear y evaluar el estado de torres eléctricas, aerogeneradores, líneas de transmisión y otras construcciones críticas, permitiendo detectar fallos estructurales sin la necesidad de exponer a los trabajadores a riesgos innecesarios [?]. En agricultura, los drones facilitan la gestión de cultivos, monitoreo de riego y la topografía, lo que permite a los agricultores optimizar la productividad y reducir costos operativos [?]. En vigilancia y seguridad, se emplean para monitorear incendios forestales, realizar labores de seguridad fronteriza y evaluar daños tras desastres naturales [?].

A pesar de sus múltiples ventajas, uno de los mayores desafíos en la operación continua de drones es la duración limitada de la batería. En aplicaciones profesionales, los drones suelen tener una autonomía de vuelo entre 20 y 30 minutos, lo cual resulta insuficiente para operaciones prolongadas como la inspección de grandes áreas o la vigilancia a largo plazo [?]. Además, las condiciones climáticas adversas pueden afectar significativamente el rendimiento y estabilidad de los drones en vuelo, lo que limita aún más su capacidad operativa [?]. Estos factores hacen necesario interrumpir las operaciones para recargar las baterías, lo cual afecta tanto la eficiencia como los costos operativos.

Para abordar esta limitación, se han diseñado estaciones de carga automáticas que permiten recargar las baterías sin intervención humana. Sin embargo, las estaciones actuales presentan varios inconvenientes: la mayoría no están diseñadas para gestionar múltiples drones de manera

simultánea, lo que limita su eficiencia en operaciones a gran escala, como las que utilizan enjambres de drones [?]. Además, el aterrizaje preciso sobre estas estaciones sigue siendo un reto importante, lo que puede afectar el éxito del proceso de carga [?].

1.1.2 Importancia del Problema

Con el creciente uso de drones en sectores como la agricultura, la logística y la vigilancia, la capacidad de mantener operaciones prolongadas y autónomas se ha vuelto crucial. Estos sectores demandan drones capaces de completar misiones largas sin interrupciones frecuentes para la recarga de baterías [?].

Aquí es donde entra en juego la necesidad de un sistema de carga modular. En este contexto, la modularidad se refiere a la posibilidad de montar varios sistemas de carga en un mismo espacio físico, lo que permite a múltiples drones recargarse simultáneamente. Esto optimiza el uso de los recursos y maximiza la eficiencia operativa de los enjambres de drones, al reducir los tiempos de espera entre recargas [?]. Esta solución es especialmente valiosa en aplicaciones que requieren la coordinación de varios drones en tiempo real, donde cualquier retraso en la recarga podría impactar negativamente en la misión.

Además, un sistema de aterrizaje preciso es clave para garantizar que los drones se posen correctamente sobre las estaciones de carga, lo que permite automatizar completamente el proceso y reducir la necesidad de intervención humana. Este tipo de solución tiene el potencial de mejorar la eficiencia y autonomía de las operaciones con drones, abriendo nuevas posibilidades para su adopción en áreas donde las limitaciones actuales de autonomía y recarga son un obstáculo [?].

1.2 Planteamiento del Problema

1.2.1 Definición Clara del Problema

Uno de los mayores desafíos para la operación continua y autónoma de enjambres de drones es la falta de estaciones de carga modulares y autónomas que permitan la recarga simultánea de múltiples drones. Actualmente, las soluciones de carga disponibles en el mercado presentan limitaciones significativas, como la imposibilidad de gestionar de manera eficiente la carga de varios drones al mismo tiempo, y que aseguren un aterrizaje preciso en un mismo espacio sin intervención humana. Estas limitaciones no solo disminuyen la autonomía de los drones, sino que también afectan la eficiencia operativa, ya que requieren intervenciones frecuentes para maniobrar los drones de manera manual y requiere mayor espacio en caso de utilizar las soluciones que se encuentran actualmente en el mercado.

1.2.2 Preguntas de Investigación

- ¿Cómo se puede desarrollar una estación de carga modular que permita la carga simultánea de múltiples drones? - ¿Qué tecnologías de localización y visión pueden ser implementadas para

asegurar un aterrizaje preciso? - ¿Cuales serían los recursos necesarios para implementar una estación de carga y aterrizaje de precisión en un enjambre de drones? - ¿Qué tipo de carga es la más adecuada para la carga de un dron en cuestión de eficiencia y rendimiento?

1.3 Objetivos

1.3.1 Objetivo General

Diseñar, manufacturar e instrumentar una base de carga inalámbrica para cuadricópteros de arquitectura abierta, y desarrollar un sistema de aterrizaje mediante el uso de sensores de localización y un sistema de visión, para asegurar una integración eficaz y segura entre el cuadricóptero y su sistema de carga.

1.3.2 Objetivos Específicos

- Realizar un diseño CAD para la base de carga inalámbrica, asegurando que sea compatible con cuadricópteros de arquitectura abierta y que cumpla con los requisitos de eficiencia y seguridad para la carga de baterías.
- Manufacturar la base de carga para el enjambre de drones.
- Adaptar el diseño actual del dron de arquitectura abierta a la base de carga.
- Instrumentar un cuadricóptero con onboard computer, flight controler, cámara de visión and sensores de localización.
- Programar un sistema de comunicación entre la base de carga y el cuadricóptero para la transferencia de datos relevantes para su interacción.
- Programar un sistema que permita controlar el drone desde una onboard computer.
- Programar un sistema de visión por computadora que detecte en tiempo real la pose de un Aruco Marker embebido en la base de carga.
- Complementar sistemas de control de motores con el sistema de vision por computadora para lograr un aterrizaje preciso.

1.4 Justificación

1.4.1 Relevancia del Proyecto

La operación continua y autónoma de enjambres de drones enfrenta varios obstáculos, entre ellos, la falta de soluciones eficientes de carga que permitan la recarga simultánea de múltiples drones

de manera autónoma y sin intervención humana. Aunque este proyecto no busca solucionar completamente este desafío, sí establece las bases fundamentales para desarrollar una solución viable. Al diseñar una estación de carga modular y autónoma con un sistema de aterrizaje preciso, este trabajo proporciona un punto de partida sobre el cual se pueden realizar futuras investigaciones y mejoras.

El proyecto es relevante para sectores como la logística, la vigilancia y la agricultura de precisión, donde la operación autónoma de drones puede generar grandes beneficios. Al presentar una solución modular que permite recargar varios drones en un mismo espacio físico, se sientan las bases para una futura implementación a mayor escala, lo que potencialmente podría reducir los costos operativos y mejorar la eficiencia de las operaciones [?]. Aunque no se pretende resolver el problema de manera completa en este trabajo, los avances presentados contribuirán al desarrollo de soluciones reales en el futuro.

1.4.2 Impacto Potencial

A pesar de que el sistema desarrollado en este proyecto no es una solución final, su implementación tiene el potencial de influir en futuras investigaciones y desarrollos en el campo de la autonomía de drones. Las tecnologías de aterrizaje preciso y carga simultánea que se exploran aquí podrían ser fundamentales para el diseño de soluciones más avanzadas y escalables, capaces de abordar las necesidades operativas de sectores que dependen cada vez más de los drones.

El impacto potencial de este proyecto se reflejará en la capacidad de inspirar investigaciones que mejoren la eficiencia de los enjambres de drones en aplicaciones como la entrega de mercancías, la vigilancia de grandes áreas y la optimización de recursos en la agricultura. Si bien las soluciones actuales enfrentan limitaciones, este trabajo ofrece una plataforma sólida sobre la cual se podrían implementar mejoras significativas en futuras etapas [?].

Memoria actualizada Gracias por la aclaración. Aquí tienes una versión ajustada de las secciones, teniendo en cuenta que el proyecto establece las bases para una solución real:

1.5 Justificación

1.5.1 Relevancia del Proyecto

La operación continua y autónoma de enjambres de drones enfrenta varios obstáculos, entre ellos, la falta de soluciones eficientes de carga que permitan la recarga simultánea de múltiples drones de manera autónoma y sin intervención humana. Aunque este proyecto no busca solucionar completamente este desafío, sí establece las bases fundamentales para desarrollar una solución viable. Al diseñar una estación de carga modular y autónoma con un sistema de aterrizaje preciso, este trabajo proporciona un punto de partida sobre el cual se pueden realizar futuras investigaciones y mejoras.

El proyecto es relevante para sectores como la logística, la vigilancia y la agricultura de precisión,

donde la operación autónoma de drones puede generar grandes beneficios. Al presentar una solución modular que permite recargar varios drones en un mismo espacio físico, se sientan las bases para una futura implementación a mayor escala, lo que potencialmente podría reducir los costos operativos y mejorar la eficiencia de las operaciones [?]. Aunque no se pretende resolver el problema de manera completa en este trabajo, los avances presentados contribuirán al desarrollo de soluciones reales en el futuro.

1.5.2 Impacto Potencial

A pesar de que el sistema desarrollado en este proyecto no es una solución final, su implementación tiene el potencial de influir en futuras investigaciones y desarrollos en el campo de la autonomía de drones. Las tecnologías de aterrizaje preciso y carga simultánea que se exploran aquí podrían ser fundamentales para el diseño de soluciones más avanzadas y escalables, capaces de abordar las necesidades operativas de sectores que dependen cada vez más de los drones.

El impacto potencial de este proyecto se reflejará en la capacidad de inspirar investigaciones que mejoren la eficiencia de los enjambres de drones en aplicaciones como la entrega de mercancías, la vigilancia de grandes áreas y la optimización de recursos en la agricultura. Si bien las soluciones actuales enfrentan limitaciones, este trabajo ofrece una plataforma sólida sobre la cual se podrían implementar mejoras significativas en futuras etapas [?].

1.6 Alcance y Limitaciones

1.6.1 Alcance del Trabajo

Este proyecto establece las bases para una solución modular y autónoma de recarga de drones. Se enfoca en el diseño, manufactura e instrumentación de una estación de carga que permita gestionar múltiples drones simultáneamente, utilizando tecnologías de visión por computadora y sensores de localización. El trabajo se llevará a cabo dentro de un entorno controlado y se validará en condiciones simuladas para asegurar que la solución es viable desde un punto de vista técnico.

Aunque el proyecto no resuelve el problema de la carga autónoma a gran escala, proporciona un primer paso hacia una solución más avanzada. Se presentará el diseño de un sistema de carga modular que permita recargar varios drones en el mismo espacio, sentando así las bases para desarrollos posteriores, y físicamente se realizará un prototipo funcional de uno de los módulos de carga para un drone. Las pruebas se realizarán en un cuadricóptero de arquitectura abierta con algunas modificaciones en el diseño que ayudarán a la carga y a la localización de los sensores de visión y localización. [?].

1.6.2 Limitaciones del Estudio

Este estudio se limita a la validación técnica del sistema en un entorno controlado y con un tipo de drone específico (cuadricóptero de arquitectura abierta). No se abordará la escalabilidad a otros

tipos de drones ni su implementación en condiciones climáticas adversas. Además, el proyecto no incluye pruebas prolongadas ni la implementación en aplicaciones comerciales.

Las tecnologías de visión y localización implementadas también estarán limitadas a las condiciones de prueba, y su precisión en escenarios más complejos no será evaluada en este trabajo. El objetivo es demostrar la viabilidad técnica de los componentes clave, más que ofrecer una solución final completamente funcional [?].

1.7 Estructura de la Tesis

El capítulo 2 presentará una revisión detallada del estado del arte en sistemas de carga para drones abarcando temas de mecánica, electrónica y programación. El capítulo 3 describirá la metodología empleada para el desarrollo del prototipo. El capítulo 4 discutirá los resultados obtenidos durante las pruebas, y el capítulo 5 incluirá las conclusiones y recomendaciones para futuros trabajos.

Chapter 2

Estado del Arte

2.1 Estado del Arte

El propósito de este capítulo es brindar una visión clara y detallada de los trabajos e investigaciones previas en el campo relacionado con la carga de drones. Se presenta una revisión de la literatura, las tecnologías existentes y las limitaciones identificadas que justifican la necesidad de la investigación propuesta.

2.2 Hardware

2.2.1 Drones

Types of Drones

Drones available in the market vary widely based on their intended applications, features, and prices. The following are the main types of drones that can be found:

Consumer Drones These drones are intended for recreational use and typically come equipped with a camera for capturing photos and videos. They are relatively affordable, with prices ranging from \$200 to \$1,500 depending on their features. Popular models include DJI's Phantom and Mavic series.

Commercial Drones These drones are designed for specific commercial purposes, such as surveying, mapping, inspection, and agriculture. They usually feature specialized sensors, like multispectral cameras, and have advanced software for planning and analyzing missions. Their price can range from \$2,000 to \$20,000. Examples include DJI's Matrice series and Parrot's Anafi.

Industrial Drones Industrial drones are high-performance devices used in demanding applications, such as power line inspections, oil and gas facility monitoring, and cargo transport. These drones are highly customizable and have long flight times, strong payload capabilities, and robust

structures. Prices vary widely but generally exceed \$20,000, depending on the customization and capabilities.

Racing Drones Racing drones are built for speed, agility, and performance in drone racing competitions. They are lightweight, feature powerful motors, and are capable of speeds up to 150 km/h. Prices start around \$300 and can exceed \$2,000 for high-end models used by professional racers.

Military Drones Military drones, also known as Unmanned Aerial Vehicles (UAVs), are used for surveillance, reconnaissance, and combat missions. These drones are the most sophisticated, featuring stealth capabilities, long endurance, and powerful payloads. Costs vary significantly, often running into millions of dollars.

The type of drone chosen depends on the specific requirements of the task, such as the payload capacity, flight time, range, and available budget.

2.2.2 Charging Stations

Types of Charging Stations

There are several types of charging stations for drones currently available in the market, each designed to meet specific needs depending on the type of drone, its intended application, and operational environment. The most common types of drone charging stations include:

Contact-Based Charging Stations These are among the most widely used solutions in the market, typically relying on metal contacts between the drone and the station to transfer electrical energy. Contact-based charging stations are relatively inexpensive and are suited for various commercial applications, including surveying, surveillance, and agriculture. They require precise landing, as proper alignment is necessary to establish the connection.

Wireless Charging Stations Wireless charging, also known as inductive charging, uses electromagnetic fields to transfer power to the drone without physical contact. While this technology provides more flexibility in terms of landing, it is less efficient compared to contact-based charging and has higher production costs. It is often utilized in specialized environments where drones must be kept operational without manual intervention, such as security monitoring or in areas where contact charging is impractical.

Battery Swap Stations These stations are designed to swap out depleted drone batteries with fully charged ones. The process is automated, allowing for continuous drone operation with minimal downtime. This approach is typically used in high-demand applications, such as drone delivery services and industrial inspections, where the quick turnaround is crucial.

Solar-Powered Charging Stations Solar-powered stations use solar panels to generate energy, which is stored and used to charge the drone. They are especially popular in remote areas where there is no direct access to power grids. These stations are environmentally friendly and allow drones to operate in off-grid locations, making them ideal for environmental monitoring and research.

Overall, the choice of charging station depends on the intended application of the drone, the available infrastructure, and the required operational efficiency.

2.2.3 Automatic Drawer Mechanism

Types of Mechanisms

The automatic drawer mechanism in the charging station for drone storage is similar to the mechanisms found in other automatic storage solutions. Here, we'll discuss the main mechanisms that could be used in this context, including the advantages and disadvantages of each. The mechanism chosen for this project was the belt and V-pulley due to its easy accessibility and low cost.

Pistons Pistons are a common choice for automated movement in many types of machinery. They use compressed air or hydraulic fluid to extend and retract, making them well-suited for precise and strong movements. For an automatic drawer for drones, pistons could provide a robust and stable means to open and close the drawer. However, this approach may have higher costs and complexity compared to other mechanisms, as it requires pneumatic or hydraulic infrastructure and regular maintenance.

Belt and Toothed Pulley This mechanism uses a toothed belt and pulley to move the drawer. It offers greater precision compared to V-belts because the toothed design prevents slipping, making it a suitable choice for environments where precise positioning is critical. However, the increased complexity and cost of toothed belts make them less accessible, and they may require more maintenance due to the increased friction between the teeth.

Belt and V-Pulley This mechanism involves a belt shaped like a V and a pulley, which ensures good friction and prevents slipping while moving the drawer. The belt and V-pulley mechanism is simple, inexpensive, and easy to source, which made it the choice for this project. It offers a balance between precision and accessibility, being a practical solution for the needs of an automatic drone storage drawer. Its main disadvantage is that it may lack the precise positioning capabilities of toothed pulleys, but for the current application, this is acceptable.

2.2.4 Motors and Propellers

Motor Selection

The selection of motors and propellers for a drone is based primarily on the total weight that the drone must carry, which determines the thrust required. Brushless motors are commonly used in

drones due to their efficiency, reliability, and power-to-weight ratio. The specific motors chosen for this project are A2212 1000KV Brushless Motors, which are popular for their balance between power and efficiency.

The thrust required for a drone must be approximately double the total weight of the drone to ensure stable flight. For a drone weighing 2 kg, the motors must provide at least 4 kg of thrust in total. The A2212 1000KV motors are capable of providing sufficient thrust when paired with appropriate propellers, making them suitable for small- to medium-sized drones. The propellers should be chosen based on the motor's specifications, balancing thrust, efficiency, and maneuverability.

Comparison with Other Motors extbf2216 900KV Motors: These motors are slightly larger than the A2212 and provide more torque, making them suitable for heavier drones or those needing more stability. They are often used with larger propellers, which allows for increased thrust but decreases agility.

extbf1806 2300KV Motors: These motors are designed for smaller, lighter drones or racing drones, providing higher RPMs and increased speed. They are not suitable for drones carrying heavy payloads, as they prioritize speed over thrust.

The choice of motor is ultimately determined by the drone's intended purpose. The A2212 1000KV motors offer a good compromise for general use, providing enough thrust for stability while maintaining efficiency.

2.3 Electronics

2.3.1 Companion Computer

Modelos Disponibles

Configuración de Hardware

Sistemas Operativos Soportados

2.3.2 Flight Controllers

Modelos Disponibles

Why use a Flight Controller insted of just a Companion Computer?

Protocolos de Comunicación entre FC y CC

2.4 Software

2.4.1 FC Firmware

Firmware PX4

El firmware PX4 ha sido una pieza fundamental en el desarrollo de sistemas de control de vuelo para drones y vehículos no tripulados, destacándose por su naturaleza de código abierto y su flexibilidad para diversas aplicaciones. Según la documentación oficial, *“PX4 is a powerful open source autopilot flight stack running on the NuttX RTOS”* [1]. Este sistema es ampliamente reconocido por su arquitectura modular, que permite la integración de nuevos sensores, actuadores y algoritmos de control, facilitando adaptaciones específicas para cumplir con los requisitos de diferentes proyectos [1].

Entre sus características más destacadas, se encuentra su capacidad para soportar una amplia variedad de tipos de vehículos. En este sentido, PX4 *“supports many different vehicle frames/types, including: multicopters, fixed-wing aircraft (planes), VTOLs (hybrid multicopter/fixed-wing), ground vehicles, and underwater vehicles”* [1]. Esta capacidad de soportar múltiples configuraciones es esencial para el diseño de sistemas modulares, ya que permite aplicar el mismo marco de trabajo a diversas plataformas con modificaciones mínimas.

Además, PX4 es parte integral de un ecosistema más amplio que incluye estaciones de control en tierra como QGroundControl, Mission Planner y hardware específico como Pixhawk. La documentación menciona que *“PX4 is a core part of a broader drone platform that includes the QGroundControl ground station, Pixhawk hardware, and MAVSDK for integration with companion computers, cameras, and other hardware using the MAVLink protocol”* [1]. Esta integración asegura una comunicación fluida entre los diferentes componentes del sistema, lo cual es crucial para el monitoreo en tiempo real y la planificación de misiones.

La comunidad activa de desarrolladores que respalda a PX4 contribuye a su desarrollo continuo, proporcionando soporte y mejoras en sus características. Esta colaboración refuerza su capacidad para adaptarse a diferentes aplicaciones y asegura que los usuarios puedan implementar *“robust and deep integration with companion computers and robotics APIs such as ROS 2 and MAVSDK”* [1].

Finalmente, PX4 ofrece *“flexible and powerful flight modes and safety features”* que son vitales para proyectos que requieren maniobras complejas, como el aterrizaje autónomo preciso en estaciones de carga [1]. Estas funcionalidades avanzadas de control y su capacidad de integrarse con tecnologías de visión por computadora y sensores externos consolidan a PX4 como una herramienta clave en el desarrollo de tecnologías de vanguardia para vehículos autónomos.

Firmware ArduPilot

ArduPilot es un software de código abierto que se ejecuta en una amplia variedad de hardware, permitiendo la creación y uso de sistemas autónomos de vehículos no tripulados para aplicaciones pacíficas. Según la documentación oficial, *“ArduPilot provides a comprehensive suite of tools suitable for almost any vehicle and application. As an open source project, it is constantly evolving based on rapid feedback from a large community of users”* [2]. Esta flexibilidad y capacidad de adaptación han convertido a ArduPilot en una herramienta esencial para proyectos de automatización y robótica que buscan una solución versátil y robusta.

Uno de los aspectos más destacados de ArduPilot es que, aunque no fabrica hardware, *“ArduPilot firmware works on a wide variety of different hardware to control unmanned vehicles of all types”* [2]. Esto significa que puede integrarse con distintos tipos de controladores, sensores y dispositivos periféricos, transformando prácticamente cualquier máquina móvil en un vehículo autónomo con la simple adición de un paquete de hardware adecuado.

El firmware es el código que se ejecuta en el controlador, y la elección del tipo de firmware depende del vehículo y la misión. Como se menciona en la documentación, *“You choose the firmware to match your vehicle and mission: Copter, Plane, Rover, Sub, or Antenna Tracker”* [2]. Esta versatilidad permite a los desarrolladores seleccionar la configuración más apropiada para sus necesidades específicas, optimizando así el proceso de desarrollo y operación.

ArduPilot también se complementa con estaciones de control en tierra (GCS), que funcionan como la interfaz entre el usuario y el controlador del vehículo. *“Various versions of GCS software are available for Windows, Linux, and Android platforms”*, y estas estaciones permiten configurar, probar y afinar el vehículo, además de planificar y ejecutar misiones autónomas [2]. Una de las herramientas más completas en este ámbito es Mission Planner, descrita como *“a full-featured GCS supported by ArduPilot”*, que ofrece interacción mediante clics, scripts personalizados y simulación [2].

2.4.2 Ground Control Systems

QGroundControl

QGroundControl (QGC) es una estación de control en tierra ampliamente utilizada que ofrece un control de vuelo completo y la configuración de vehículos equipados con los firmwares PX4 y ArduPilot. La documentación oficial destaca que “*QGroundControl provides full flight control and vehicle setup for PX4 or ArduPilot powered vehicles*” [3]. Esta plataforma se caracteriza por su facilidad de uso para principiantes, al tiempo que ofrece un soporte robusto y avanzado para usuarios más experimentados, permitiéndoles aprovechar al máximo las capacidades de sus vehículos no tripulados.

Entre las principales características de QGroundControl, se incluye la configuración y ajuste total de vehículos propulsados por ArduPilot y PX4. Además, “*Flight support for vehicles running PX4 and ArduPilot (or any other autopilot that communicates using the MAVLink protocol)*” es una funcionalidad esencial que asegura su compatibilidad con cualquier sistema de piloto automático que utilice el protocolo MAVLink [3]. Esto lo convierte en una herramienta versátil para gestionar diversas configuraciones de vehículos y misiones.

QGroundControl también permite la planificación de misiones autónomas con una visualización detallada de mapas de vuelo que muestran la posición del vehículo, la trayectoria de vuelo, los puntos de referencia (waypoints) y los instrumentos del vehículo. “*Mission planning for autonomous flight*” y la visualización de la misión en un mapa de vuelo enriquecido proporcionan a los operadores una comprensión precisa y visual del entorno y las operaciones en curso [3].

Por último, una de las ventajas más notables de QGC es su soporte para la gestión de múltiples vehículos, permitiendo a los operadores controlar y supervisar más de un dron simultáneamente. Esta funcionalidad es crucial en aplicaciones avanzadas como operaciones de enjambre y misiones coordinadas de múltiples vehículos [3].

Mission Planner

Mission Planner es una estación de control en tierra (GCS) desarrollada para gestionar vehículos equipados con el firmware ArduPilot. Según la documentación oficial, “*Mission Planner is a full-featured GCS supported by ArduPilot*” [2]. Esta herramienta proporciona a los usuarios la capacidad de configurar, supervisar y controlar sus vehículos de forma eficiente y avanzada, siendo adecuada tanto para usuarios novatos como para expertos.

Una de las funcionalidades más notables de Mission Planner es su capacidad para la planificación de misiones autónomas, lo que permite a los operadores definir rutas de vuelo de manera sencilla a través de una interfaz intuitiva. Esta funcionalidad hace que Mission Planner sea fundamental para el desarrollo y ajuste de misiones complejas. Además, la posibilidad de realizar simulaciones y analizar los datos después de cada vuelo proporciona una visión integral del rendimiento del vehículo.

Mission Planner incluye herramientas de configuración y ajuste de parámetros que facilitan la

personalización del vehículo para adaptarlo a distintas aplicaciones. La capacidad de *“tuning and testing vehicle parameters directly from the GCS”* permite a los operadores optimizar el firmware para ajustarse a las necesidades específicas del vehículo y mejorar su desempeño [2].

Otra ventaja significativa es la presentación de datos de telemetría en tiempo real, que ofrece información sobre la posición, velocidad, estado de la batería y otros parámetros críticos, ayudando a garantizar la seguridad y efectividad de las operaciones de vuelo.

Figure 2.1: Interfaz de Mission Planner mostrando la planificación de una misión.

2.4.3 Companion Computer Operative Systems

Existen múltiples opciones de companion computers que pueden emplearse en el desarrollo de proyectos relacionados con vehículos no tripulados. Entre las más populares se encuentran: Raspberry Pi, Nvidia Jetson Nano, Odroid, entre otros [?]. En este proyecto, se ha elegido una Raspberry Pi 4 por su combinación de bajo costo, versatilidad y facilidad de uso. A continuación, se detallan las características principales de los sistemas operativos compatibles con la Raspberry Pi 4 y 5.

Ubuntu

Ubuntu es un sistema operativo de código abierto basado en Linux que ha ganado popularidad por su estabilidad y amplia comunidad de soporte. Según la documentación oficial de Ubuntu, este sistema es *“designed for security, reliability, and ease of use”* [4]. En el contexto de los companion computers para drones y otros vehículos autónomos, Ubuntu se utiliza frecuentemente debido a su compatibilidad con herramientas de robótica como ROS (Robot Operating System), lo que facilita la integración y desarrollo de software avanzado de control y automatización.

Ubuntu ofrece soporte para arquitecturas ARM, lo que permite su instalación y ejecución en dispositivos como la Raspberry Pi 4 y 5. Esta capacidad es fundamental para proyectos que requieren procesamiento local eficiente, manejo de datos de sensores y comunicación en tiempo real. Adicionalmente, la flexibilidad de Ubuntu permite la personalización de su entorno para adaptarse a las necesidades específicas del proyecto, ya sea para ejecutar nodos de control de vuelo o procesamiento de imágenes en tiempo real.

Figure 2.2: Ubuntu OS

Raspberry Pi OS

Raspberry Pi OS, es el sistema operativo oficial desarrollado y optimizado para dispositivos Raspberry Pi. La documentación de Raspberry Pi OS lo describe como *“a Debian-based operating system specifically tuned for the Raspberry Pi hardware”* [5]. Su principal ventaja es su optimización para

el hardware de Raspberry Pi, lo que garantiza un rendimiento óptimo y un uso eficiente de los recursos disponibles.

Raspbian incluye una serie de herramientas preinstaladas que facilitan el desarrollo y prototipado, haciendo que sea una opción preferida para proyectos educativos y de investigación. La compatibilidad con Python y otras bibliotecas de programación facilita la implementación de scripts y software necesarios para el control de drones y otras aplicaciones robóticas.

En comparación con otros sistemas operativos, Raspbian es ligero y permite un arranque rápido, lo cual es beneficioso en escenarios donde se requiere un inicio rápido del sistema. Además, la comunidad activa de usuarios de Raspberry Pi proporciona recursos y asistencia que resultan invaluable para la resolución de problemas y la optimización del software.

Figure 2.3: Raspberry Pi OS

2.4.4 ROS2 Distributions

Las distribuciones de ROS 2 (Robot Operating System 2) proporcionan entornos estandarizados para el desarrollo de aplicaciones robóticas, cada una adaptada a diferentes necesidades y capacidades de hardware. Una de las ventajas de ROS 2 es su capacidad para facilitar la comunicación entre distribuciones mediante el uso de topics, lo que permite que nodos en distintas versiones de ROS 2 se comuniquen y colaboren de manera efectiva en un mismo proyecto [6]. En la siguiente sección se abordarán más detalles sobre el funcionamiento general de ROS 2 y su importancia en proyectos de robótica.

En el contexto de companion computers como la Raspberry Pi, las distribuciones específicas de ROS 2 se eligen basándose en la versión de hardware y la compatibilidad de software. La Raspberry Pi 4, por ejemplo, es adecuada para ejecutar la distribución ROS 2 Humble [?], mientras que la Raspberry Pi 5 puede correr la distribución más reciente, ROS 2 Jazzy Jalisco [6]. A continuación, se describen las principales características de cada distribución y algunas diferencias importantes.

Distribución Humble

La distribución ROS 2 Humble es conocida por su estabilidad y soporte a largo plazo (LTS), lo que la convierte en una opción popular para proyectos que requieren fiabilidad y consistencia en el tiempo. Esta versión es ideal para proyectos que utilizan hardware como la Raspberry Pi 4, ya que está optimizada para correr de manera eficiente en dispositivos con arquitecturas ARM [?].

Características principales de ROS 2 Humble:

Soporte a largo plazo (LTS), garantizando actualizaciones y soporte extendido. Mayor estabilidad en la comunicación entre nodos. Compatibilidad con una amplia gama de paquetes y bibliotecas de ROS. Recomendado para proyectos de investigación y aplicaciones de largo plazo.

Distribución Jazzy

La distribución ROS 2 Jazzy Jalisco es una de las más recientes y está diseñada para aprovechar las capacidades avanzadas del hardware más moderno, como la Raspberry Pi 5. Esta versión incluye mejoras en rendimiento y nuevas funcionalidades que facilitan el desarrollo de aplicaciones robóticas más complejas y eficientes [?].

Características principales de ROS 2 Jazzy Jalisco:

Introducción de características experimentales y optimizaciones de rendimiento. Mejoras en la latencia y el manejo de la comunicación entre nodos. Integración mejorada con nuevas herramientas de simulación y depuración. Adecuado para desarrolladores que buscan utilizar las últimas funcionalidades de ROS 2 en proyectos avanzados. Ambas distribuciones comparten la capacidad de usar el middleware DDS (Data Distribution Service) para la comunicación entre nodos, permitiendo la creación de sistemas distribuidos y escalables. La elección entre ROS 2 Humble y Jazzy Jalisco dependerá de las especificaciones de hardware disponibles y las necesidades específicas del proyecto.

Figure 2.4: ROS 2 Humble y ROS 2 Jazzy Jalisco

2.4.5 ROS2 Documentation

ROS 2 (Robot Operating System 2) es un conjunto de bibliotecas y herramientas que permiten a los desarrolladores crear aplicaciones de robótica. Es una evolución de ROS 1, diseñada para abordar limitaciones en términos de escalabilidad, seguridad y soporte a sistemas en tiempo real. ROS 2 ofrece un marco de trabajo robusto que facilita la implementación de sistemas distribuidos y permite la comunicación entre múltiples nodos a través de una arquitectura basada en el middleware DDS (Data Distribution Service) [6]. A continuación, se detallan los beneficios y la estructura de ROS 2.

Benefits of ROS2

ROS 2 introduce mejoras significativas que lo hacen más adecuado para aplicaciones avanzadas de robótica. Entre los principales beneficios se encuentran:

- Multithreading y manejo en tiempo real: ROS 2 permite la ejecución de múltiples nodos en paralelo, lo que mejora el rendimiento en aplicaciones complejas.
- Comunicación mediante topics, nodos y servicios: Facilita la creación de sistemas distribuidos donde los nodos se comunican de forma eficiente.
- Archivos de lanzamiento (launchfiles): Simplifican la ejecución de múltiples nodos y configuraciones mediante un solo archivo.

- Baja latencia y comunicación rápida: La arquitectura basada en DDS proporciona una comunicación confiable y de baja latencia, esencial para aplicaciones críticas.
- Diferentes tipos de mensajes: Soporte para mensajes personalizados y definidos por el usuario.

Table 2.1: Beneficios clave de ROS 2

Beneficio	Descripción
Multithreading	Permite la ejecución de nodos en paralelo, optimizando el rendimiento
Uso de topics	Comunicación asincrónica entre nodos para el intercambio de datos
Nodos y servicios	Los nodos son unidades de ejecución que interactúan mediante servicios y topics
Archivos de lanzamiento (launchfiles)	Permiten la configuración y ejecución de múltiples nodos simultáneamente
Comunicación rápida	Basada en DDS, proporciona comunicación de baja latencia
Mensajes personalizados	Soporte para definir y usar mensajes específicos para aplicaciones personalizadas

ROS2 Architecture

Dentro de la documentación oficial de ROS 2 se habla de que un aspecto central de esta arquitectura son los nodos, que actúan como las unidades básicas de procesamiento en un sistema de ROS 2. Los nodos se comunican entre sí mediante topics, que permiten un intercambio de información asincrónico y eficiente.

Los mensajes en ROS 2 definen la estructura de los datos que se transmiten a través de topics. Estos mensajes pueden ser de tipos predefinidos o personalizados, adaptándose a las necesidades de la aplicación. Los servicios, por otro lado, permiten una comunicación sincrónica entre nodos, donde un nodo solicita una acción específica y otro nodo responde con el resultado.

Los archivos de lanzamiento, o launchfiles, son documentos que permiten ejecutar múltiples nodos y configurar parámetros de forma simultánea. Facilitan la gestión de sistemas complejos, ya que centralizan la configuración en un solo archivo, lo que simplifica la puesta en marcha de aplicaciones robóticas [6].

Finalmente, ROS 2 admite la configuración y uso de parámetros, que son variables accesibles desde los nodos para ajustar comportamientos en tiempo de ejecución sin necesidad de recompilar el código.

Figure 2.5: Estructura de ROS 2, mostrando la interacción de nodos, topics, servicios y launchfiles.

2.4.6 Computer Vision

La visión por computadora es un campo de la programación que permite a los sistemas interpretar y procesar información visual del mundo que los rodea. Es esencial para aplicaciones que requieren el análisis de imágenes y videos en tiempo real, como la navegación autónoma de vehículos, detección de objetos, inteligencia artificial, etc.

OpenCV Library

OpenCV (Open Source Computer Vision Library) es una biblioteca de código abierto ampliamente utilizada en la industria y en la investigación académica para el desarrollo de aplicaciones de visión por computadora. De acuerdo con la documentación oficial, OpenCV es *“an open-source computer vision and machine learning software library containing more than 2500 optimized algorithms”* [7]. Esta biblioteca proporciona herramientas para tareas como el procesamiento de imágenes, reconocimiento de objetos, seguimiento de movimiento, entre otras, y es compatible con múltiples lenguajes de programación como Python, C++ y más.

OpenCV es conocida por su flexibilidad y facilidad de uso, lo que la hace popular tanto entre principiantes como entre expertos en el área de la visión por computadora. La biblioteca incluye funciones de alto rendimiento que pueden ser ejecutadas en tiempo real, haciendo que sea una opción ideal para proyectos de robótica y sistemas autónomos [?].

Camera Calibration

La calibración de cámaras es un proceso crucial en la visión por computadora, necesario para corregir distorsiones inherentes a las lentes de las cámaras y obtener mediciones precisas del entorno. Este proceso implica la obtención de parámetros intrínsecos y extrínsecos que permiten mapear las coordenadas 2D de una imagen a coordenadas 3D en el mundo real.

Figure 2.6: Tipos de distorsiones de cámaras

Según la documentación de OpenCV, *“Camera calibration is the process of estimating the parameters of the lens and the image sensor of an imaging device”* [?]. OpenCV proporciona herramientas y funciones que permiten realizar este proceso de manera eficiente mediante la detección de patrones en imágenes, como tableros de ajedrez o círculos.

El flujo típico de calibración de una cámara con OpenCV incluye capturar imágenes de un patrón conocido, identificar los puntos de interés en cada imagen y utilizar algoritmos de optimización para calcular los parámetros de la cámara. Estos parámetros incluyen la distancia focal, el punto principal, y los coeficientes de distorsión radial y tangencial [?].

Además, se pueden utilizar los resultados de la calibración para corregir la distorsión en imágenes y videos, mejorando así la precisión de las aplicaciones de visión por computadora y el uso para calcular la pose o posiciones de algunos objetos de interés. El uso de una cámara calibrada es

fundamental en proyectos que requieren un análisis espacial preciso, como la navegación autónoma de robots y drones.

Figure 2.7: Patrón de tablero de ajedrez utilizado en la calibración de cámaras con OpenCV.

2.4.7 What is an ArUco?

ArUco, cuyo nombre proviene de la combinación de "Artificial" y "Uco" (por la Universidad de Córdoba, donde se desarrolló), es una biblioteca de código abierto ampliamente reconocida en el ámbito de la visión por computadora para la detección de marcadores fiduciales en imágenes. Esta tecnología es fundamental para la estimación de la pose de la cámara con respecto a los marcadores cuando la cámara ha sido previamente calibrada. De acuerdo con la documentación, "*ArUco is an OpenSource library for detecting squared fiducial markers in images*" [8]. La detección de estos marcadores es crucial en aplicaciones que requieren una estimación precisa de la posición y orientación de objetos en el espacio tridimensional.

Historia de los Marcadores Aruco

Los marcadores ArUco se originaron como una solución para superar las limitaciones de otras tecnologías de detección de patrones, colores o figuras. El objetivo era desarrollar una técnica que proporcionara alta fiabilidad incluso bajo oclusiones parciales y condiciones de iluminación variables. Los primeros estudios se centraron en la generación automática de marcadores con un diseño que asegurara su unicidad y facilidad de detección. Estos marcadores están compuestos por un patrón binario rodeado de un borde negro, lo que mejora su visibilidad y robustez en diferentes condiciones de iluminación [8].

Figure 2.8: ArUco example

Aplicaciones Comunes

Los marcadores ArUco se utilizan en una variedad de aplicaciones, que incluyen la calibración de cámaras, la realidad aumentada, y el control y navegación de robots y drones. Una de las ventajas de usar ArUco es su capacidad para actuar como puntos de referencia en entornos 3D, permitiendo a los sistemas de visión por computadora calcular la pose de la cámara. Según la documentación, "*Markers can be used as 3D landmarks for camera pose estimation*" [?]. Esta característica hace que los marcadores sean esenciales en sistemas de seguimiento y posicionamiento donde la precisión es crítica.

Figure 2.9: ArUco pose estimation

Formatos de Marcadores

Los marcadores ArUco están compuestos por un borde negro externo y una región interna que codifica un patrón binario único. Dependiendo del diccionario que se esté utilizando, el número de bits en el marcador varía, lo que afecta la probabilidad de confusión con otros marcadores y la distancia de detección. Una mayor resolución de los marcadores permite que estos se detecten desde distancias más lejanas, pero puede requerir mayor procesamiento [8].

La biblioteca ArUco también soporta la creación de diccionarios personalizados, lo que permite a los desarrolladores adaptar los marcadores a las necesidades específicas de sus proyectos. *“The design of a dictionary is important since the idea is that their markers should be as different as possible to avoid confusions”* [?]. Esta flexibilidad es especialmente útil en proyectos donde es crucial mantener la unicidad y fiabilidad de la detección de marcadores en entornos complejos.

Figure 2.10: Imagen explicación de bits en ArUco

2.4.8 Aruco vs Embedded Aruco

Los marcadores ArUco y los marcadores Embedded ArUco (e-ArUco) son tecnologías utilizadas en visión por computadora para tareas de detección y estimación de pose. Aunque ambos comparten una base común en cuanto a su diseño y algoritmos de detección, tienen diferencias importantes que los hacen adecuados para distintas aplicaciones, especialmente en el contexto de operaciones de alta precisión y entornos desafiantes.

Diferencias Principales

La principal diferencia entre los marcadores ArUco tradicionales y los marcadores Embedded ArUco radica en la optimización de estos últimos para detecciones de alta precisión en un rango amplio de distancias. Según Khazetdinov et al. (2021), *“a new type of fiducial marker called embedded ArUco (e-ArUco) was developed specially for a task of robust marker detection for a wide range of distances”* [9]. Los e-ArUco están diseñados para mantener su detectabilidad y precisión en escenarios donde los marcadores ArUco estándar podrían no ser tan efectivos, como cuando se requiere una precisión milimétrica en aplicaciones de aterrizaje de UAVs.

Otra diferencia significativa es que los marcadores e-ArUco están diseñados para mejorar la robustez de la detección, minimizando los errores que podrían surgir por condiciones de iluminación cambiantes y oclusiones parciales. Estos marcadores se basan en los algoritmos de detección de ArUco, lo que permite su implementación sin cambios sustanciales en los sistemas existentes basados en ArUco [9].

Casos de Uso

Los marcadores ArUco tradicionales como se mencionó en la sección anterior, se utilizan comúnmente en aplicaciones de realidad aumentada, estimación de poses y navegación de robots y drones. Estos marcadores son versátiles y se pueden adaptar a diversas aplicaciones que no requieren una precisión extrema, siendo ideales para proyectos de investigación y desarrollo.

Por otro lado, los marcadores Embedded ArUco (e-ArUco) están específicamente diseñados para situaciones donde la precisión es más crítica. Un ejemplo destacado es el uso en el aterrizaje de UAVs (vehículos aéreos no tripulados), donde se necesita una detección precisa y confiable a diferentes distancias. En un estudio realizado por Khazetdinov et al., tuvo como resultado “*an average landing accuracy was 2.03 cm with a standard deviation of 1.53 cm*” al usar marcadores e-ArUco y un algoritmo de aterrizaje implementado en ROS y probado en el simulador Gazebo [9]. Esta capacidad hace que los e-ArUco sean ideales para entornos donde se requiere una precisión milimétrica, como en operaciones de aterrizaje autónomo de alta precisión.

Figure 2.11: Representación de un marcador Embedded ArUco utilizado en el aterrizaje de precisión de UAVs.

2.4.9 Aruco Detection

La detección de marcadores ArUco es un proceso esencial en la visión por computadora que permite la identificación y estimación de la pose de los marcadores en imágenes. La siguiente sección detalla los algoritmos de detección, la implementación en OpenCV y los parámetros que afectan la precisión de la detección.

Algoritmos de Detección

La detección de los marcadores ArUco se basa en algoritmos de visión por computadora que identifican contornos y patrones específicos en las imágenes. Según la documentación de OpenCV, el proceso de detección comienza con la identificación de cuadrados en la imagen y la verificación de si estos contienen un patrón binario válido correspondiente a un marcador ArUco [?]. El algoritmo implementado en OpenCV utiliza la segmentación de contornos y la detección de bordes para encontrar regiones cuadradas que luego se comprueban para determinar si coinciden con los patrones en el diccionario de ArUco.

Una vez que se identifica un marcador, el algoritmo calcula la pose de la cámara en relación con el marcador utilizando la proyección inversa. Este proceso es particularmente importante en aplicaciones donde se necesita calcular la posición y orientación de la cámara para la navegación y el control de robots y drones.

Implementación en OpenCV

OpenCV ofrece una implementación robusta para la detección de marcadores ArUco a través del módulo `cv::aruco`. La función principal para la detección es `cv::aruco::detectMarkers`, que se encarga de identificar los marcadores en una imagen y devolver sus esquinas y los IDs correspondientes. La documentación de OpenCV resalta que *“the function detects the markers and returns their IDs and corner positions in the image”* [?].

El siguiente ejemplo muestra cómo usar OpenCV para detectar marcadores ArUco en Python:

```
import cv2
import cv2.aruco as aruco

# Cargar la imagen
image = cv2.imread('image_path.jpg')

# Definir el diccionario de ArUco
aruco_dict = aruco.Dictionary_get(aruco.DICT_6X6_250)

# Detectar los marcadores
corners, ids, _ = aruco.detectMarkers(image, aruco_dict)

# Dibujar los marcadores detectados
if ids is not None:
    aruco.drawDetectedMarkers(image, corners, ids)
```

Este código permite la detección y visualización de los marcadores ArUco en una imagen de una forma muy simple, devolviendo las posiciones de las esquinas y los IDs de los marcadores detectados.

Parámetros de Precisión

La precisión de la detección de marcadores ArUco depende de varios factores, incluidos la calidad de la imagen, el tamaño del marcador, y los parámetros de calibración de la cámara. Según la documentación de OpenCV, la calibración precisa de la cámara es fundamental para minimizar los errores en la estimación de la pose [?]. Entre los parámetros que afectan la detección se encuentran:

- **Distorsión de la lente:** La corrección de la distorsión de la lente mejora la precisión de la detección.
- **Resolución del marcador:** Marcadores con mayor resolución permiten una detección más precisa a distancias más largas, pero requieren más recursos de procesamiento.
- **Iluminación y contraste:** La detección puede verse afectada por condiciones de iluminación variables, por lo que es importante que la imagen tenga un buen contraste entre el marcador y el fondo.

Figure 2.12: Detección de marcadores ArUco con OpenCV mostrando los IDs y las esquinas detectadas.

Chapter 3

Desarrollo

3.0.1 Diseño de la Estación de Carga

3.1 Ensamble de Estructura de la Estación de Carga

3.2 Armado de Mecanismo de Poleas

3.3 Electronica para movimiento del cajón

3.4 Programación para control de movimiento del cajón

3.5 Armado del Drone

Chapter 4

Resultados

Chapter 5

Conclusiones

Bibliography

- [1] PX4 Documentation, “PX4 is a powerful open source autopilot flight stack running on the NuttX RTOS.” Disponible en: . [*Último acceso: mes, año*].
- [2] ArduPilot Documentation, “ArduPilot firmware works on a wide variety of different hardware to control unmanned vehicles of all types.” Disponible en: <https://ardupilot.org/ardupilot/index.html>. [*Último acceso: mes, año*].
- [3] QGroundControl Documentation, “QGroundControl provides full flight control and vehicle setup for PX4 or ArduPilot powered vehicles.” Disponible en: <https://docs.qgroundcontrol.com/en/>. [*Último acceso: mes, año*].
- [4] Ubuntu Documentation, “Designed for security, reliability, and ease of use.” Disponible en: <https://ubuntu.com>. [*Último acceso: mes, año*].
- [5] Raspberry Pi OS Documentation, “A Debian-based operating system specifically tuned for the Raspberry Pi hardware.” Disponible en: <https://www.raspberrypi.org/documentation/raspbian/>. [*Último acceso: mes, año*].
- [6] ROS 2 Documentation, “Benefits and architecture of ROS 2 for distributed systems in robotics.” Disponible en: <https://docs.ros.org/en/>. [*Último acceso: mes, año*].
- [7] OpenCV Documentation, “An open-source computer vision and machine learning software library containing more than 2500 optimized algorithms.” Disponible en: <https://docs.opencv.org/>. [*Último acceso: mes, año*].
- [8] ArUco Documentation, “ArUco is an OpenSource library for detecting squared fiducial markers in images.” Disponible en: . [*Último acceso: mes, año*].
- [9] Khazetdinov, M., et al., “Embedded ArUco Markers for UAV Landing: High Accuracy Detection in Wide Distance Range,” 2021. DOI: [DOI number or URL if available].