



REPÚBLICA DE PANAMÁ

MINISTERIO DE EDUCACIÓN

INSTITUTO SUPERIOR TECNOLÓGICO DEL CLAUSTRO GÓMEZ

**TRABAJO FINAL DE INVESTIGACIÓN PARA OPTAR POR EL TÍTULO DE
TÉCNICO SUPERIOR EN PROGRAMACIÓN EN INFORMÁTICA**

**CONTROL DE ROBOTS AUTÓNOMOS USANDO PROGRAMACIÓN EN
PYTHON Y ROS**

ELABORADO POR: MANUEL ENRIQUE TUÑÓN ON - 4-799-795

Julio 2024

Índice

I.	RESUMEN	3
II.	INTRODUCCIÓN	4
III.	JUSTIFICACIÓN	6
IV.	OBJETIVOS	7
V.	MARCO TEÓRICO	8
VI.	METODOLOGÍA	9
VII.	DESARROLLO O CUERPO DEL TRABAJO.....	11
VIII.	CONCLUSIÓN	13
IX.	RECOMENDACIONES.....	14
X.	BIBLIOGRAFÍA	15

I. RESUMEN

El control de robots autónomos es un campo en constante evolución que ha adquirido una gran importancia en diversas aplicaciones, desde la automatización industrial hasta la exploración espacial y la asistencia en entornos domésticos. La capacidad de los robots para operar de manera autónoma, interpretar su entorno y tomar decisiones en tiempo real depende en gran medida de los algoritmos y sistemas de control que se implementan. En este contexto, esta tesina se centra en el desarrollo y control de robots autónomos utilizando programación en Python y el sistema operativo para robots (ROS), herramientas ampliamente utilizadas en la comunidad robótica por su versatilidad, modularidad y capacidad de integración con diversos sensores y actuadores.

El estudio tiene como objetivo principal diseñar e implementar algoritmos de navegación y control para robots autónomos, aprovechando las capacidades de Python y ROS para la comunicación entre módulos, el procesamiento de datos sensoriales y la planificación de trayectorias. Se busca desarrollar un marco de trabajo que permita a los robots desplazarse en entornos estructurados y no estructurados, optimizando su capacidad de percepción y toma de decisiones. Para ello, se emplean sensores como LiDAR, cámaras RGB-D y encoders, que proporcionan información del entorno en tiempo real y permiten la implementación de estrategias de navegación autónoma basadas en visión artificial y aprendizaje automático.

La metodología seguida en esta investigación combina un enfoque teórico y experimental. Inicialmente, se realiza un análisis detallado del estado del arte en robótica autónoma, explorando las herramientas disponibles en Python y ROS y su aplicabilidad en el desarrollo de sistemas de control. Posteriormente, se diseña una arquitectura modular que integra diferentes componentes, como la localización y mapeo simultáneo (SLAM), la planificación de trayectorias y la evasión de obstáculos. Para la validación de los algoritmos, se emplean entornos de simulación en Gazebo, donde se prueban distintos escenarios antes de implementar las soluciones en robots físicos, asegurando así su correcto funcionamiento y optimización.

Los resultados obtenidos evidencian la eficacia de Python y ROS para el desarrollo de robots autónomos, permitiendo una comunicación eficiente entre los distintos módulos y facilitando la integración de algoritmos avanzados de control y percepción. Se observa que los robots equipados con estrategias de visión artificial y redes neuronales mejoran su capacidad para reconocer objetos y evitar colisiones en entornos dinámicos. Asimismo, la flexibilidad de ROS permite la reutilización de código y la escalabilidad de los sistemas, favoreciendo su aplicación en diferentes tipos de robots y escenarios.

Como conclusión, esta tesina demuestra que la combinación de Python y ROS representa una solución poderosa y accesible para el desarrollo de sistemas de control de robots autónomos. La modularidad y capacidad de personalización de estas herramientas permiten la implementación de algoritmos cada vez más sofisticados, con aplicaciones en múltiples sectores. Se sugiere como línea futura de investigación la exploración de modelos más avanzados de aprendizaje profundo para mejorar la autonomía y adaptabilidad de los robots, así como la integración de arquitecturas multi-robot que permitan la cooperación entre múltiples unidades en entornos complejos.

II. INTRODUCCIÓN

El control de robots autónomos ha emergido como un campo de gran relevancia en la robótica moderna, con aplicaciones en diversas áreas como la automatización industrial, la exploración espacial, la asistencia médica, la agricultura de precisión y la logística. Un robot autónomo es un sistema capaz de percibir su entorno, tomar decisiones y ejecutar acciones sin intervención humana directa, lo que requiere la implementación de algoritmos avanzados de control, navegación y aprendizaje.

Para lograr un control eficiente de estos robots, es fundamental contar con herramientas de programación y software especializado que permitan la integración de sensores, actuadores y módulos de toma de decisiones. En este contexto, Python y el sistema operativo para robots (ROS) han demostrado ser soluciones robustas y flexibles para el desarrollo de

sistemas autónomos. Python es un lenguaje de programación ampliamente utilizado en el ámbito de la inteligencia artificial y el procesamiento de datos, mientras que ROS proporciona un entorno modular y escalable para la comunicación y gestión de procesos en robots.

Esta tesina explora la implementación de algoritmos de control y navegación para robots autónomos utilizando Python y ROS, abordando aspectos fundamentales como la percepción del entorno mediante sensores (LiDAR, cámaras, IMU), la planificación de trayectorias, la localización y mapeo simultáneo (SLAM) y la toma de decisiones basada en inteligencia artificial. A través de este estudio, se busca desarrollar soluciones eficientes para el control de robots móviles, mejorando su capacidad de adaptación a entornos dinámicos y su autonomía operativa.

La implementación de estos algoritmos no solo permite mejorar la precisión y eficiencia del movimiento de los robots autónomos, sino que también facilita su aplicación en entornos complejos y cambiantes. La combinación de sensores avanzados con modelos de inteligencia artificial optimiza la percepción del entorno, permitiendo una mejor toma de decisiones en tiempo real. Además, la integración de ROS proporciona una arquitectura flexible y modular que facilita la experimentación con diferentes enfoques de control y navegación, promoviendo la innovación en el desarrollo de sistemas robóticos autónomos.

Este estudio no solo aporta conocimientos teóricos y prácticos sobre el uso de Python y ROS en robótica, sino que también sienta las bases para futuras investigaciones y aplicaciones en el ámbito de la automatización y la inteligencia artificial. La posibilidad de extender estos desarrollos a otros tipos de robots, como drones o vehículos autónomos, amplía el alcance de esta investigación y abre nuevas oportunidades en sectores clave como el transporte, la seguridad y la exploración. Con los avances tecnológicos actuales, el control de robots autónomos sigue evolucionando, y este trabajo contribuye al desarrollo de soluciones más inteligentes, adaptativas y eficientes en el campo de la robótica moderna.

III. JUSTIFICACIÓN

El desarrollo de robots autónomos representa un avance significativo en múltiples disciplinas, ya que permite la automatización de tareas complejas que requieren precisión, rapidez y adaptabilidad. En la industria, estos sistemas mejoran la eficiencia en líneas de producción y logística, reduciendo costos operativos y optimizando recursos. En la medicina, los robots autónomos son utilizados en cirugías asistidas y en el cuidado de pacientes, proporcionando soluciones innovadoras en la atención sanitaria. En la agricultura, permiten la siembra, el riego y la recolección automatizada, incrementando la productividad y reduciendo el impacto ambiental.

A pesar de los avances tecnológicos, el desarrollo de robots autónomos sigue enfrentando desafíos en términos de control, percepción y toma de decisiones. La capacidad de los robots para operar de manera autónoma depende en gran medida de la precisión de sus algoritmos y de la integración eficiente de sus sensores y actuadores. Python y ROS han surgido como herramientas fundamentales en este campo, ya que facilitan la implementación de modelos de control y aprendizaje, permitiendo el desarrollo de soluciones escalables y de código abierto.

Esta investigación es relevante porque contribuye a la comprensión y aplicación de estas herramientas en el desarrollo de sistemas robóticos avanzados. Además, busca mejorar la accesibilidad a tecnologías de robótica autónoma, promoviendo su aplicación en distintos sectores. La combinación de Python y ROS permite un enfoque modular y flexible que puede adaptarse a diversas plataformas robóticas, fomentando la innovación y el desarrollo de nuevas aplicaciones en el ámbito de la robótica.

Otro aspecto relevante en el desarrollo de robots autónomos es su impacto en la movilidad y el transporte. Los vehículos autónomos, por ejemplo, están revolucionando la industria automotriz al reducir la necesidad de intervención humana y mejorar la seguridad vial mediante sistemas avanzados de percepción y navegación. Además, los robots de entrega autónoma están siendo implementados en ciudades inteligentes para agilizar el comercio

electrónico y reducir la congestión vehicular. La implementación de estos sistemas no solo optimiza el tiempo y los costos operativos, sino que también disminuye el impacto ambiental al hacer un uso más eficiente de los recursos energéticos y reducir las emisiones de carbono.

Otro desafío crucial en este campo es la seguridad y la confiabilidad de los sistemas robóticos en entornos dinámicos. Aunque los algoritmos de control y navegación han avanzado significativamente, la interacción de los robots con humanos y otros sistemas sigue siendo un área de investigación activa. Se requieren métodos más robustos para la detección y prevención de fallos, así como protocolos de seguridad que minimicen el riesgo de errores en la toma de decisiones del robot. La combinación de aprendizaje profundo con sistemas de control tradicionales puede mejorar la capacidad de los robots para responder de manera efectiva a situaciones imprevistas, aumentando su autonomía y eficiencia en entornos no estructurados.

Finalmente, la educación y la capacitación en robótica autónoma son fundamentales para seguir impulsando el desarrollo de esta tecnología. La accesibilidad a herramientas como Python y ROS permite que más investigadores, ingenieros y estudiantes puedan experimentar con sistemas robóticos sin la necesidad de grandes inversiones en hardware especializado. Iniciativas de código abierto y comunidades de desarrollo colaborativo juegan un papel clave en la democratización de la robótica, facilitando la creación de nuevas soluciones y el intercambio de conocimientos a nivel global. En este contexto, la presente investigación aporta una base sólida para futuras innovaciones en el diseño y control de robots autónomos, promoviendo su integración en múltiples disciplinas y sectores productivos.

IV. OBJETIVOS

- **Objetivo general**
 - Diseñar e implementar algoritmos de control y navegación para robots autónomos utilizando programación en Python y ROS, optimizando su capacidad de percepción, toma de decisiones y desplazamiento en entornos dinámicos.

- **Objetivos específicos**

- Explorar y analizar las herramientas proporcionadas por Python y ROS para el desarrollo de sistemas de control y navegación en robots autónomos, identificando sus ventajas y limitaciones.
- Desarrollar algoritmos de percepción basados en sensores como LiDAR, cámaras y encoders, permitiendo la interpretación del entorno y la generación de mapas en tiempo real.
- Implementar estrategias de navegación y control de movimiento, empleando técnicas de localización y planificación de trayectorias para optimizar el desplazamiento autónomo de los robots.

V. MARCO TEÓRICO

El desarrollo de robots autónomos se basa en múltiples disciplinas como la robótica, la inteligencia artificial, la visión por computadora y el control de sistemas dinámicos. Uno de los fundamentos clave es el sistema operativo para robots (ROS), una plataforma de software de código abierto diseñada para la programación modular y la comunicación entre los distintos componentes de un robot. ROS facilita la integración de sensores, motores y algoritmos de navegación, permitiendo que los robots interactúen con su entorno de manera eficiente. En paralelo, el lenguaje Python se ha convertido en una herramienta esencial en el desarrollo de software para robótica debido a su simplicidad, versatilidad y compatibilidad con bibliotecas especializadas como OpenCV para visión artificial, TensorFlow y PyTorch para aprendizaje automático, y NumPy para el procesamiento de datos numéricos.

Desde una perspectiva histórica, el campo de la robótica autónoma ha evolucionado significativamente desde los primeros autómatas hasta los actuales sistemas avanzados de inteligencia artificial. Investigaciones previas han explorado diversas estrategias de control y navegación, como los algoritmos de localización y mapeo simultáneo (SLAM), que permiten a un robot construir un mapa de su entorno mientras determina su posición dentro de él. Asimismo, los modelos de control basado en comportamiento, propuestos por Brooks

(1986), han sido fundamentales para el desarrollo de robots que responden de manera reactiva a estímulos del entorno sin depender exclusivamente de modelos predefinidos. En la literatura reciente, se han desarrollado enfoques híbridos que combinan planificación probabilística con aprendizaje automático, mejorando la toma de decisiones autónoma en entornos dinámicos y no estructurados.

Teóricamente, el control de robots autónomos se fundamenta en modelos matemáticos y algoritmos computacionales que rigen la percepción y la acción del robot. La teoría del control óptimo y los filtros de Kalman son ampliamente utilizados para mejorar la precisión en la localización y navegación. Por otro lado, la inteligencia artificial ha permitido el desarrollo de sistemas de aprendizaje reforzado que optimizan la capacidad de los robots para adaptarse a su entorno. Investigaciones recientes han demostrado que la combinación de redes neuronales profundas con técnicas de SLAM mejora significativamente la autonomía de los robots en tareas complejas como la exploración y la manipulación de objetos. En este contexto, la presente investigación se apoya en estos marcos teóricos para diseñar e implementar un sistema de control de robots autónomos eficiente y adaptable a diversas aplicaciones.

VI. METODOLOGÍA

Esta investigación adopta un enfoque mixto, combinando métodos cuantitativos y cualitativos para analizar el desempeño y la eficiencia de los algoritmos de control en robots autónomos. A través del enfoque cuantitativo, se miden variables como la precisión en la localización, la eficiencia en la navegación y los tiempos de respuesta del sistema, utilizando métricas establecidas en el campo de la robótica. En paralelo, el enfoque cualitativo permite evaluar la experiencia del usuario y la usabilidad del sistema mediante observaciones y análisis interpretativo de los resultados obtenidos en pruebas experimentales y simulaciones. La combinación de estos enfoques proporciona una visión más integral del funcionamiento de los algoritmos desarrollados, facilitando la identificación de mejoras y la comparación con estudios previos en el área.

Para la recopilación de datos, se emplearon diversas fuentes primarias y secundarias. Las fuentes primarias incluyen pruebas experimentales realizadas en entornos simulados mediante Gazebo y pruebas en hardware real con plataformas robóticas compatibles con ROS, como el TurtleBot3. Durante estas pruebas, se registraron datos clave sobre la precisión en la navegación, la estabilidad del sistema y la eficiencia en la ejecución de tareas autónomas. Además, se llevaron a cabo entrevistas semiestructuradas con expertos en robótica y desarrolladores de software especializados en ROS, quienes brindaron información cualitativa sobre las ventajas y limitaciones del enfoque utilizado. Las fuentes secundarias se obtuvieron a través de una revisión de literatura en bases de datos científicas como IEEE Xplore, Springer y ScienceDirect, analizando estudios previos sobre algoritmos de control, localización y planificación de trayectorias en robots autónomos. También se revisaron manuales técnicos de ROS y Python, así como artículos que describen implementaciones exitosas de sistemas de control robótico en entornos similares.

El análisis de los datos obtenidos se llevó a cabo mediante técnicas estadísticas y herramientas computacionales. En las pruebas cuantitativas, se calcularon métricas como el error en la localización, la eficiencia en el tiempo de respuesta y la tasa de éxito en la navegación para evaluar el rendimiento de los algoritmos implementados. Estos datos fueron procesados con Python utilizando bibliotecas como NumPy, Pandas y Matplotlib, facilitando la visualización y el análisis de tendencias. Para el análisis cualitativo, se aplicó un enfoque de análisis de contenido, categorizando y comparando las respuestas de las entrevistas y las observaciones de las pruebas experimentales. Se identificaron patrones y opiniones recurrentes que permitieron evaluar la viabilidad del sistema y su potencial de mejora. Finalmente, los resultados obtenidos se contrastaron con estudios previos para validar la efectividad del enfoque propuesto, identificando posibles áreas de optimización en la implementación de algoritmos de control de robots autónomos basados en Python y ROS.

VII. DESARROLLO O CUERPO DEL TRABAJO

El desarrollo de robots autónomos utilizando Python y ROS requiere una comprensión profunda de los algoritmos de control, la percepción del entorno y la toma de decisiones en tiempo real. En este estudio, se implementaron y analizaron distintos enfoques de control para la navegación de un robot móvil, evaluando su desempeño en entornos simulados y en hardware real. Se exploraron métodos de localización y mapeo, planificación de trayectorias y evitación de obstáculos, con el objetivo de mejorar la precisión y la eficiencia del desplazamiento del robot. La integración de sensores como cámaras y LiDAR permitió aumentar la capacidad del sistema para interpretar su entorno y tomar decisiones informadas en tiempo real.

Uno de los aspectos fundamentales del control de robots autónomos es la localización y navegación. Se implementaron algoritmos de localización probabilística como los filtros de partículas y los filtros de Kalman, los cuales permiten estimar la posición del robot con base en datos de sensores y modelos cinemáticos. Estos enfoques fueron evaluados en distintos escenarios, comparando su precisión y tiempo de procesamiento. Asimismo, se empleó el algoritmo SLAM para la generación de mapas en tiempo real, lo que permitió al robot desplazarse sin depender de mapas predefinidos. A través de pruebas experimentales, se identificó que la combinación de SLAM con técnicas de inteligencia artificial mejora la capacidad del robot para adaptarse a entornos dinámicos y desconocidos.

Otro desafío abordado fue la planificación de trayectorias y la evitación de obstáculos, elementos esenciales para la navegación autónoma. Se implementaron algoritmos como A* y Dijkstra para la planificación de rutas óptimas, así como métodos basados en potenciales artificiales y redes neuronales para evitar colisiones en tiempo real. Se realizaron pruebas comparativas entre estos métodos, observando que los algoritmos heurísticos ofrecen soluciones eficientes en términos de rapidez, mientras que los enfoques de aprendizaje profundo presentan una mayor adaptabilidad en entornos complejos. El análisis de los resultados permitió determinar que una combinación de planificación heurística con

modelos de predicción basados en redes neuronales mejora la eficiencia del sistema en escenarios dinámicos.

En la evaluación del sistema, se realizaron pruebas tanto en entornos simulados como en un entorno físico controlado. Se compararon métricas como el tiempo de llegada a destino, la cantidad de colisiones evitadas y el consumo de recursos computacionales. Los resultados demostraron que los algoritmos implementados permiten un control eficiente del robot, con mejoras significativas en la estabilidad y precisión de la navegación. Sin embargo, también se identificaron limitaciones, como el impacto del ruido en los sensores y la necesidad de mayor capacidad de cómputo en escenarios altamente dinámicos.

El análisis comparativo con estudios previos permitió validar la efectividad del enfoque utilizado, destacando la importancia de combinar múltiples técnicas para lograr un control robusto y adaptable. Se concluyó que la integración de Python y ROS proporciona una plataforma flexible y poderosa para el desarrollo de robots autónomos, facilitando la implementación de algoritmos avanzados y la experimentación en distintos escenarios. Esta investigación sienta las bases para futuras mejoras en la autonomía de robots móviles, con aplicaciones en sectores como la industria, la exploración y la asistencia en entornos urbanos.

Además, se exploró el impacto del hardware en el desempeño del sistema de control del robot autónomo. Se realizaron pruebas utilizando distintas configuraciones de sensores y procesadores, comparando el rendimiento en términos de velocidad de procesamiento y precisión en la navegación. Se observó que el uso de procesadores de alto rendimiento combinados con sensores de alta resolución mejora significativamente la capacidad del robot para interpretar su entorno y reaccionar en tiempo real. No obstante, también se evidenció que el costo y consumo energético de estos componentes pueden ser factores limitantes en aplicaciones donde la eficiencia y autonomía energética son prioritarias.

Por otro lado, la interoperabilidad entre los distintos módulos de software y hardware fue un desafío clave en el desarrollo del sistema. Se implementaron arquitecturas modulares en

ROS para facilitar la integración de nuevos algoritmos y la reutilización de componentes en distintos entornos. La comunicación entre los nodos de ROS permitió la sincronización eficiente de los sensores y actuadores, optimizando la toma de decisiones en el robot. Este enfoque modular no solo facilitó la implementación y prueba de distintos algoritmos, sino que también permitió la escalabilidad del sistema, abriendo posibilidades para su aplicación en robots de mayor complejidad y en entornos más exigentes.

VIII. CONCLUSIÓN

Las conclusiones de esta investigación reflejan la importancia del uso de Python y ROS en el desarrollo y control de robots autónomos, destacando su flexibilidad, eficiencia y aplicabilidad en diversas áreas. Se ha demostrado que la combinación de algoritmos de localización, planificación de trayectorias y evitación de obstáculos permite mejorar significativamente la autonomía y precisión de los robots móviles. A lo largo del estudio, se implementaron técnicas de localización probabilística, como los filtros de partículas y de Kalman, que ofrecieron resultados favorables en la estimación de la posición del robot en entornos dinámicos. Además, la integración de SLAM permitió la generación de mapas en tiempo real sin necesidad de contar con información previa del entorno, lo que otorga una ventaja considerable en aplicaciones de navegación en espacios desconocidos o cambiantes.

Uno de los hallazgos más relevantes es la efectividad de la planificación heurística combinada con redes neuronales en la navegación autónoma. Los algoritmos A* y Dijkstra, aunque eficientes en la planificación de rutas óptimas, presentan limitaciones en entornos altamente dinámicos. La inclusión de técnicas de inteligencia artificial mejoró la capacidad de adaptación del robot, optimizando la toma de decisiones en tiempo real y reduciendo el tiempo de respuesta ante cambios inesperados en el entorno. Asimismo, la implementación de sensores como cámaras y LiDAR permitió una mejor interpretación del entorno, facilitando la evitación de obstáculos de manera más precisa. Las pruebas realizadas en entornos simulados y físicos demostraron que el uso combinado de estos métodos mejora la eficiencia del robot, minimizando el margen de error y aumentando la estabilidad en la navegación.

Otro aspecto importante identificado en este estudio es la necesidad de optimizar los recursos computacionales para mejorar el rendimiento del sistema. Si bien los algoritmos implementados mostraron buenos resultados en términos de precisión y adaptabilidad, se observó que el procesamiento en tiempo real puede verse afectado por el alto consumo de recursos, especialmente en sistemas con hardware limitado. Esto sugiere la necesidad de explorar alternativas que reduzcan la carga computacional sin comprometer la calidad del control del robot. Además, se identificó que el ruido en los sensores puede afectar la precisión de la localización, lo que resalta la importancia de emplear técnicas de filtrado y calibración más avanzadas para minimizar estos efectos.

IX. RECOMENDACIONES

Se sugiere que futuras investigaciones se centren en la optimización de los algoritmos de control y navegación para reducir el consumo de recursos computacionales. Esto podría lograrse mediante la implementación de técnicas de compresión de datos y el uso de hardware especializado, como unidades de procesamiento neuronal (NPUs) o GPUs, que permitan acelerar el procesamiento de información sin afectar el desempeño del robot. También sería beneficioso explorar la integración de técnicas de aprendizaje por refuerzo, que podrían mejorar la toma de decisiones del robot en entornos altamente dinámicos y no estructurados, incrementando su capacidad de adaptación en diferentes escenarios.

Otra recomendación clave es la ampliación de las pruebas experimentales en entornos más complejos y realistas, con el fin de evaluar el desempeño del sistema en condiciones más exigentes. Si bien las simulaciones brindan un entorno controlado para el desarrollo y prueba de algoritmos, las pruebas en hardware real permiten identificar desafíos prácticos que pueden no estar presentes en un entorno virtual. Asimismo, se recomienda la integración de sistemas de comunicación en red para permitir el control colaborativo de múltiples robots, lo que podría ampliar las aplicaciones del sistema en áreas como la automatización industrial, la exploración de terrenos peligrosos y la asistencia en entornos urbanos.

Se recomienda que futuras investigaciones exploren el impacto de la integración de ROS 2, la versión más reciente de este framework, que ofrece mejoras en la comunicación distribuida, la seguridad y la escalabilidad de los sistemas robóticos. La adopción de ROS 2 permitiría desarrollar robots autónomos más robustos y eficientes, con capacidades avanzadas de comunicación y procesamiento distribuido. Con base en estos hallazgos y recomendaciones, se concluye que el desarrollo de robots autónomos mediante Python y ROS sigue siendo un campo de gran potencial, con múltiples oportunidades para la innovación y la mejora continua en el ámbito de la robótica autónoma.

X. BIBLIOGRAFÍA

- Boal Martín-Larrauri, J., Collado de la Guerra, E., Cubillo Llanes, D., Giannetti, R., Martínez Fariña, A., Sánchez Molina, R., & Tordesillas Torres, J. (2024). Robots móviles autónomos.
- Morales Pareja, M. (2021). Gestión de robots autónomos y colaborativos mediante ROS y computación en la nube.
- Sotelo Castillo, D. (2023). Programación, Control e Implementación de un Robot Móvil de Seguimiento Reactivo basado en Visión Artificial y ROS.
- Cortes, D. S. M., Guzmán, L. E. S., & Vivas, A. V. (2024). Sistema de supervisión aplicado al robot de agricultura de precisión CERES para el cuidado de los cultivos agrícolas. *Ingenierías USBMed*, 15(2).
- García Ulloa, I. A., & Nieves Guerrero, W. J. (2024). *Desarrollo de un robot autónomo con sistema de mapeo y detección de obstáculos mediante sensor LiDAR* (Bachelor's thesis).