



UNIVERSIDAD TECNOLÓGICA ECOTEC

FACULTAD DE INGENIERÍAS, ARQUITECTURA Y CIENCIAS DE LA NATURALEZA

TÍTULO DEL TRABAJO:

**METODOLOGÍA PARA LA SELECCIÓN DE ALGORITMOS DE NAVEGACIÓN AUTÓNOMA
EN ROBOTS EXPLORADORES CON SENSORES DE ULTRASONIDO**

LÍNEA DE INVESTIGACIÓN:

TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

MODALIDAD DE TITULACIÓN:

TRABAJO DE INTEGRACIÓN CURRICULAR

CARRERA/PROGRAMA:

INGENIERÍA EN SOFTWARE

TÍTULO A OBTENER:

INGENIERO EN SOFTWARE

AUTOR:

RODOLFO ENRIQUE ORTEGA AVEIGA

TUTOR:

ALEJANDRA MERCEDES COLINA VARGAS, PHD.

GUAYAQUIL – ECUADOR

2024



ANEXO No. 9

**PROCESO DE TITULACIÓN
CERTIFICADO DE APROBACIÓN DEL TUTOR**

Samborondón, 19 de diciembre de 2024.

Magíster

Erika del Pilar Ascencio Jordán

Unidad Académica: Facultad de Ingenierías, Arquitectura y Ciencias de la Naturaleza
Universidad Tecnológica ECOTEC

De mis consideraciones:

Por medio de la presente comunico a usted que el trabajo de titulación TITULADO: **METODOLOGÍA PARA LA SELECCIÓN DE ALGORITMOS DE NAVEGACIÓN AUTÓNOMA EN ROBOTS EXPLORADORES CON SENSORES DE ULTRASONIDO**, fue revisado, siendo su contenido original en su totalidad, así como el cumplimiento de los requerimientos establecidos en la guía para su elaboración, por lo que se autoriza al estudiante: **Rodolfo Enrique Ortega Aveiga** para que proceda con la presentación oral del mismo.

ATENTAMENTE,



Firmado electrónicamente por:
**ALEJANDRA MERCEDES
COLINA VARGAS**

Firma

Alejandra Mercedes Colina Vargas, PhD.
Tutora



ANEXO No. 10

**PROCESO DE TITULACIÓN
CERTIFICADO DEL PORCENTAJE DE COINCIDENCIAS
DEL TRABAJO DE TITULACIÓN**

Habiendo sido revisado el trabajo de titulación TITULADO: METODOLOGÍA PARA LA SELECCIÓN DE ALGORITMOS DE NAVEGACIÓN AUTÓNOMA EN ROBOTS EXPLORADORES CON SENSORES DE ULTRASONIDO elaborado por RODOLFO ENRIQUE ORTEGA AVEIGA fue remitido al sistema de coincidencias en todo su contenido el mismo que presentó un porcentaje del 1% mismo que cumple con el valor aceptado para su presentación que es inferior o igual al 10% sobre el total de hojas del documento. Adicional se adjunta print de pantalla de dicho resultado.



ATENTAMENTE,



Resultado: a1eant-edificament-a-ppor-
ALEJANDRA MERCEDES COLINA VARGAS

Firma
Ing. Alejandra Mercedes Colina Vargas, PhD.
Tutora

DEDICATORIA

Dedico este trabajo a Dios, quien me brindó la fortaleza y sabiduría necesarias para superar cada desafío y avanzar con esperanza y determinación en este camino educativo.

A mis padres, cuyo amor, sacrificio y apoyo incondicional han sido mi mayor fuente de inspiración y motivación, y por inculcarme siempre que el mejor legado que pueden ofrecerme es la educación y los valores que forman mi ser.

A mi hermana, por su compañía inquebrantable, sus palabras de aliento en los momentos difíciles y por ser una constante fuente de ánimo y orgullo durante mi recorrido académico.

A mi querida abuela, quien siempre fue un faro de amor y sabiduría en mi vida. Aunque ya no esté físicamente, su espíritu y sus enseñanzas siguen guiando mis pasos, y este logro es también un tributo a su memoria.

Finalmente, dedico este trabajo a mí mismo, por no rendirme, por superar los desafíos y por creer que los sueños son alcanzables con esfuerzo y pasión.

AGRADECIMIENTO

Agradezco, en primer lugar, a Dios, por iluminar mi camino universitario, llenarme de fortaleza, paciencia y sabiduría, y por ser mi guía constante en cada paso que me llevó a completar esta etapa tan significativa en mi vida.

A mi familia, por su amor incondicional, su apoyo permanente y por proporcionarme las herramientas necesarias para alcanzar mis metas académicas y personales. Su confianza en mí ha sido mi mayor motivación para perseverar.

A mis compañeros de carrera, con quienes compartí aprendizajes, desafíos y momentos inolvidables, y cuya colaboración fue invaluable en nuestro camino hacia la formación profesional.

Agradezco al Ing. Rodolfo Ortega, por su apoyo incondicional durante el desarrollo de este proyecto de titulación, cuyo acompañamiento fue fundamental para lograr este objetivo.

Finalmente, expreso mi gratitud a la Ing. Alejandra Colina, por su guía, paciencia y constante motivación, elementos clave para llevar a cabo este trabajo con dedicación y éxito. Su orientación me permitió crecer como profesional y culminar esta etapa con orgullo.

RESUMEN

El presente trabajo de titulación tuvo como objetivo principal el desarrollo de una metodología estandarizada para la selección de algoritmos de navegación autónoma en robots exploradores equipados con sensores ultrasónicos. Para ello, se identificó como problema principal la falta de criterios sistemáticos que permitan evaluar y seleccionar algoritmos adecuados en función de las características específicas del entorno y las necesidades operativas de cada misión. La metodología propuesta se diseñó a partir de una revisión bibliográfica exhaustiva, entrevistas semiestructuradas con expertos en robótica y simulaciones computacionales realizadas en el entorno Webots. Estas actividades permitieron analizar algoritmos ampliamente reconocidos, como A*, Dijkstra y RRT, bajo métricas clave como precisión, tiempo de ejecución, consumo de recursos, adaptabilidad y seguridad. Durante las simulaciones, se determinó que el algoritmo A* es ideal para entornos estructurados debido a su precisión y seguridad, mientras que RRT mostró mayor adaptabilidad y rapidez en escenarios dinámicos. Por su parte, Dijkstra destacó por su robustez, aunque presentó limitaciones en consumo computacional y escalabilidad. Finalmente, los resultados obtenidos validan la viabilidad y aplicabilidad de la metodología desarrollada, que constituye una herramienta práctica y replicable para optimizar el desempeño de robots exploradores en entornos complejos y dinámicos, contribuyendo al avance de la navegación autónoma en la robótica.

Palabras clave: Metodología estandarizada, algoritmos de navegación autónoma, robots exploradores, sensores ultrasónicos, simulaciones computacionales, Webots, A*, Dijkstra, RRT, precisión, adaptabilidad, seguridad, consumo de recursos, escalabilidad, entornos dinámicos, robótica.

ABSTRACT

This thesis aimed to develop a standardized methodology for selecting autonomous navigation algorithms in exploratory robots equipped with ultrasonic sensors. The main issue identified was the lack of systematic criteria for evaluating and selecting suitable algorithms based on the specific characteristics of the environment and the operational needs of each mission. The proposed methodology was designed through an exhaustive bibliographic review, semi-structured interviews with robotics experts, and computational simulations conducted in the Webots environment. These activities enabled the analysis of widely recognized algorithms, such as A*, Dijkstra, and RRT, under key metrics like accuracy, execution time, resource consumption, adaptability, and safety. During the simulations, it was determined that the A* algorithm is ideal for structured environments due to its accuracy and safety, while RRT showed greater adaptability and speed in dynamic scenarios. Meanwhile, Dijkstra stood out for its robustness but exhibited limitations in computational resource usage and scalability. Finally, the results validate the feasibility and applicability of the developed methodology, which constitutes a practical and replicable tool to optimize the performance of exploratory robots in complex and dynamic environments, contributing to the advancement of autonomous navigation in robotics.

Keywords: Standardized methodology, autonomous navigation algorithms, exploratory robots, ultrasonic sensors, computational simulations, Webots, A*, Dijkstra, RRT, accuracy, adaptability, safety, resource consumption, scalability, dynamic environments, robotics.

ÍNDICE

1. INTRODUCCION	11
1.1. Contexto Histórico Social del Objeto de Estudio.....	11
1.2. Antecedentes	12
1.3. Planteamiento del Problema.....	13
1.4. Objetivos	16
1.5. Justificación.....	16
2. MARCO TEÓRICO	18
2.1. Antecedentes	18
2.2. Fundamentación Teórica	21
2.3. Marco Legal	24
2.4. Marco Conceptual	26
3. MARCO METODOLÓGICO	30
3.1. Enfoque de la Investigación	30
3.2. Alcance de la Investigación.....	31
3.3. Delimitación de la Investigación.....	32
3.4. Población y Muestra.....	33
3.5. Métodos Empleados	35
3.6. Herramientas Utilizadas	40
3.7. Fases de la Metodología.....	41
4. ANÁLISIS DE RESULTADOS	45
4.1. Fase 1: Planificación y Diseño del Estudio.....	45
4.2. Fase 2: Recolección y Definición de Información	51
4.3. Fase 3: Implementación en Simulaciones Computacionales	54
4.4. Fase 4: Análisis y Evaluación de Resultados	57
5. CONCLUSIONES	65
6. RECOMENDACIONES	66
7. REFERENCIAS	67
8. ANEXOS	73

LISTADO DE TABLAS

Tabla 1	53
Tabla 2	58
Tabla 3	59
Tabla 4	59
Tabla 5	75
Tabla 6	78
Tabla 7	81

LISTADO DE FIGURAS

Figura 1	55
Figura 2	56
Figura 3	73
Figura 4	73
Figura 5	74
Figura 6	74

LISTADO DE GRÁFICOS

Gráfico 1	60
Gráfico 2	61
Gráfico 3	61
Gráfico 4	62
Gráfico 5	62
Gráfico 6	63
Gráfico 7	63

1. INTRODUCCION

1.1. Contexto Histórico Social del Objeto de Estudio

La historia de la navegación autónoma está intrínsecamente relacionada con los avances científicos y los cambios sociales que han moldeado las demandas tecnológicas en contextos industriales y exploratorios. Desde la década de 1980, cuando los primeros robots industriales comenzaron a integrar sensores ultrasónicos para evitar colisiones, hasta la actualidad, esta evolución refleja cómo la innovación tecnológica ha respondido a necesidades sociales apremiantes, como la seguridad en entornos laborales peligrosos y la exploración de lugares inaccesibles para el ser humano (González et al., 2020).

La introducción de sensores ultrasónicos en la robótica respondió a la necesidad de soluciones económicas y eficientes para entornos controlados. Estos sensores ofrecieron una alternativa viable para mejorar la seguridad y la eficiencia en aplicaciones industriales, marcando el inicio de una transición hacia sistemas robóticos más complejos. Sin embargo, su impacto social se extendió más allá de las industrias, influyendo en sectores como la exploración espacial y las operaciones de rescate, donde las tecnologías avanzadas han permitido salvar vidas y expandir los límites del conocimiento humano (Martínez et al., 2022).

Los avances en navegación autónoma también han generado un impacto significativo en la forma en que las comunidades perciben la tecnología. La integración de sensores ultrasónicos con sistemas avanzados como LIDAR (Light Detection and Ranging) y algoritmos de aprendizaje automático ha mejorado la capacidad de los robots para operar en entornos diversos y desafiantes, contribuyendo a la aceptación de estas tecnologías en aplicaciones cotidianas como la agricultura de precisión y el transporte autónomo (Hernández & Torres, 2023). No obstante, estos avances también han planteado

retos éticos, como la distribución equitativa de beneficios tecnológicos y el impacto en el empleo tradicional.

1.2. Antecedentes

Diversas investigaciones recientes han explorado la integración de sensores ultrasónicos con algoritmos de navegación en robots móviles. Por ejemplo, Chen et al. (2020) analizaron el uso de sensores ultrasónicos combinados con LIDAR (Light Detection and Ranging) para mejorar la detección de obstáculos y la planificación de rutas en entornos interiores. Los resultados mostraron que la fusión de datos multisensoriales optimiza significativamente la navegación en escenarios complejos. Esta investigación destaca cómo las tecnologías complementarias pueden superar las limitaciones inherentes de cada sensor individual.

De manera similar, Li et al. (2021) evaluaron algoritmos como A* y RRT (Rapidly-exploring Random Tree) en entornos dinámicos, encontrando que el uso de sensores ultrasónicos mejora la precisión en la detección de objetos cercanos. Sin embargo, también se identificaron limitaciones relacionadas con la resolución angular de los sensores y la necesidad de integrar métodos de optimización algorítmica para escenarios más exigentes.

Por otro lado, investigaciones como las de Nguyen y Park (2022) han abordado el impacto de algoritmos de aprendizaje profundo en la navegación autónoma. Estos estudios concluyen que los algoritmos basados en redes neuronales ofrecen una adaptabilidad superior en entornos cambiantes, pero requieren un alto costo computacional. Esta dualidad subraya la importancia de seleccionar enfoques que equilibren la eficiencia y la adaptabilidad según las necesidades específicas de la misión.

En este contexto, la literatura refleja una evolución constante hacia sistemas más integrados y eficientes. No obstante, la presente investigación busca proponer una metodología integral para la selección de algoritmos de navegación autónoma, con un énfasis particular en el uso de sensores

ultrasónicos en robots exploradores al abordar la necesidad de desarrollar metodologías que guíen la selección de algoritmos y sensores en función de las características del entorno y los requisitos operativos.

1.3. Planteamiento del Problema

A pesar de los avances en tecnologías de navegación, la selección de algoritmos sigue siendo un desafío debido a la falta de criterios estandarizados. Actualmente, los ingenieros dependen en gran medida de pruebas empíricas para determinar la adecuación de un algoritmo a un contexto específico, lo que incrementa los costos y el tiempo de desarrollo (Wang & Zhang, 2022). Además, aunque los sensores ultrasónicos son herramientas efectivas en condiciones de baja visibilidad, su combinación con algoritmos avanzados no siempre está optimizada, limitando su aplicabilidad en escenarios complejos (López & Hernández, 2023).

El problema se agrava en aplicaciones críticas como la exploración espacial y las operaciones de rescate, donde la fiabilidad y la adaptabilidad son cruciales. Por ejemplo, en misiones de rescate tras desastres naturales, la capacidad de un robot para navegar de manera autónoma y eficaz puede marcar la diferencia entre salvar vidas o enfrentarse a obstáculos insalvables debido a decisiones algorítmicas inadecuadas (Singh et al., 2023). La falta de un marco metodológico claro también genera incertidumbre en la planificación y evaluación de proyectos robóticos, lo que afecta la confianza de los usuarios y la adopción generalizada de estas tecnologías en sectores clave.

Asimismo, la proliferación de tecnologías avanzadas como el aprendizaje profundo y los algoritmos de planificación probabilística plantea un nuevo conjunto de desafíos. Aunque estas herramientas ofrecen un alto potencial de adaptabilidad, su implementación requiere de recursos computacionales significativos y de una infraestructura que no siempre está disponible en todos los

contextos (Nguyen et al., 2022). Esto refuerza la necesidad de contar con una metodología que considere tanto las capacidades del hardware como las especificidades del entorno operativo.

Según lo señalado en la entrevista realizada al Ing. Rodolfo Ortega, profesional con una trayectoria destacada en ingeniería eléctrica desde 1990 y con experiencia en el ámbito de la robótica desde 2020, se evidencia la relevancia de abordar el tema propuesto. El entrevistado enfatizó la necesidad de diseñar una metodología clara y sistemática para la selección de algoritmos de navegación en robots exploradores. En particular, destacó la importancia de aquellos algoritmos que emplean sensores de ultrasonido, subrayando que esta necesidad surge de los retos identificados en diversos proyectos de investigación llevados a cabo en este campo. Este testimonio se convierte en un aporte crucial para sustentar la pertinencia del estudio, dado que refleja tanto el conocimiento técnico como la experiencia práctica del especialista consultado (R. Ortega, comunicación personal, 26 de septiembre de 2024).

Por su parte, el Ing. Magno Andrade destacó en su entrevista la problemática inherente a la falta de guías sistemáticas en la selección de algoritmos de navegación para robots exploradores, además, argumentó que muchos de los desafíos que enfrentan estos robots surgen de decisiones fundamentadas únicamente en suposiciones o experiencias previas, debido a la ausencia de un marco metodológico que relacione de manera precisa las capacidades de los algoritmos con las características específicas del entorno en el que operan. Esta falta de estandarización conduce, con frecuencia, a implementaciones inadecuadas que comprometen el rendimiento del robot, especialmente en contextos que exigen altos niveles de precisión y adaptabilidad (M. Andrade, comunicación personal, 26 de septiembre de 2024).

Los entrevistados coinciden en resaltar la relevancia de los sensores de ultrasonido como herramientas esenciales para la detección de obstáculos y la medición de distancias en robots

exploradores. No obstante, señalan que estos dispositivos enfrentan desafíos significativos en la interpretación de datos, especialmente en contextos de condiciones ambientales variables.

En este sentido, subrayan la importancia de seleccionar algoritmos que no solo optimicen la precisión y la rapidez, sino que también sean capaces de adaptarse a la variabilidad de dichas condiciones. Asimismo, recalcan la necesidad de establecer criterios metodológicos que integren estas variables dentro del diseño de sistemas de navegación, lo cual contribuiría a que los robots no solo respondan de manera eficaz a cambios repentinos en tiempo real, sino que también incrementen su capacidad de toma de decisiones autónomas en escenarios complejos y dinámicos (R. Ortega y M. Andrade, comunicación personal, 27 de septiembre de 2024).

Por consiguiente, la pregunta principal que orienta esta investigación es:

¿Cómo puede desarrollarse una metodología estandarizada que permita seleccionar algoritmos de navegación autónoma para robots exploradores equipados con sensores de ultrasonido, asegurando eficacia, rendimiento y adaptabilidad en diversos entornos?

A partir de esta cuestión, se plantea abordar interrogantes complementarias, tales como:

¿Cuáles son los componentes fundamentales que deben incluirse en una metodología estandarizada para asegurar una selección eficiente de algoritmos de navegación autónoma?

¿Qué criterios y métricas de evaluación son necesarios para determinar la eficacia y adecuación de los algoritmos de navegación en función de los requisitos de cada misión?

¿Qué tan efectiva es la metodología propuesta al evaluar el rendimiento de los algoritmos seleccionados en simulaciones controladas?

¿Qué características debe tener un marco práctico para garantizar la replicabilidad y adaptabilidad de la metodología a diferentes tipos de misiones y entornos operativos?

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar una metodología estandarizada para la selección de algoritmos de navegación autónoma en robots exploradores con sensores de ultrasonido, que garanticen la eficacia, rendimiento y adaptabilidad en entornos variados.

1.4.2. Objetivos Específicos

- Establecer los elementos estructurales que conformarán la metodología estandarizada para garantizar una selección de algoritmos eficiente.
- Definir los criterios específicos de evaluación y de rendimiento que sirvan de base para la selección de algoritmos.
- Validar la efectividad de la metodología propuesta evaluando el rendimiento de los algoritmos seleccionados mediante simulaciones en entornos controlados.
- Generar un marco práctico que permita replicar y adaptar la metodología a diversos tipos de misiones y entornos operativos.

1.5. Justificación

El presente trabajo aborda una brecha clave en el campo de la navegación autónoma al proponer una metodología estandarizada para la selección de algoritmos de navegación en robots exploradores equipados con sensores ultrasónicos. Esta brecha, identificada en la literatura revisada, radica en la falta de criterios sistemáticos que permitan evaluar y seleccionar algoritmos en función de las características del entorno y las necesidades específicas de cada misión (Zhang & Liu, 2020). La metodología propuesta no solo busca llenar este vacío, sino también optimizar el uso de tecnologías avanzadas en escenarios operativos críticos.

El uso de sensores de ultrasonido como parte de esta propuesta agrega un componente fundamental al estudio. Estos sensores, ampliamente valorados por su capacidad de detección precisa en tiempo real, son especialmente efectivos en condiciones de baja visibilidad o alta densidad de obstáculos (Chen et al., 2022). A pesar de sus ventajas, la integración efectiva de estos sensores con algoritmos avanzados de navegación sigue representando un desafío. En este sentido, el marco metodológico propuesto busca optimizar esta combinación, lo que no solo permitirá mejorar el rendimiento operativo de los robots exploradores, sino que también abrirá nuevas posibilidades de aplicación en entornos extremos, como exploraciones subacuáticas o en zonas de difícil acceso tras desastres naturales.

La investigación también busca promover la colaboración entre campos como la inteligencia artificial, el diseño de sensores y la ingeniería de sistemas. Como sugieren Wang y Zhang (2020), la integración de tecnologías avanzadas, como sensores LIDAR (Light Detection and Ranging), cámaras de visión 3D y ultrasonido, con algoritmos basados en aprendizaje automático y planificación probabilística, es clave para resolver los problemas complejos de navegación en entornos dinámicos. Este enfoque no solo mejora la percepción ambiental de los robots, sino que también permite una toma de decisiones más precisa y eficiente.

Finalmente, este estudio tiene el potencial de generar un impacto duradero en la comunidad académica y la industria al proporcionar un modelo metodológico replicable que sirva de referencia para el diseño y evaluación de futuros sistemas robóticos. De acuerdo con Siciliano y Khatib (2019), la estandarización de prácticas en la robótica autónoma es fundamental para acelerar la innovación y garantizar la interoperabilidad de las tecnologías emergentes. En este sentido, la metodología desarrollada podría convertirse en un referente clave para la optimización de algoritmos de navegación, fortaleciendo así el avance de la robótica como disciplina científica y su aplicación práctica en escenarios reales.

2. MARCO TEÓRICO

2.1. Antecedentes

González (2019), en su estudio titulado Sistema de navegación autónoma para robots exploradores utilizando sensores de ultrasonido y redes neuronales (Universidad Nacional Autónoma de México), desarrolló un modelo basado en redes neuronales supervisadas para mejorar la adaptación de robots exploradores en entornos dinámicos mediante datos capturados por sensores de ultrasonido. La metodología incluyó el entrenamiento de la red neuronal con datos obtenidos de simulaciones y pruebas experimentales, enfocándose en optimizar la capacidad de aprendizaje del sistema.

Los resultados demostraron una mejora significativa en la adaptabilidad del robot en entornos complejos, aunque se identificó como limitación el extenso tiempo de entrenamiento y el alto consumo computacional. Este estudio, aunque relevante, se centra exclusivamente en redes neuronales, lo que deja una brecha en la exploración de algoritmos tradicionales que podrían ofrecer soluciones más eficientes en términos de recursos computacionales y adaptabilidad.

López (2020), en su investigación titulada Desarrollo de una metodología de navegación autónoma para robots móviles basada en sensores de ultrasonido y LIDAR (Universidad de Cuenca, Ecuador), propuso un modelo híbrido que combina sensores de ultrasonido y tecnología LIDAR (Light Detection and Ranging) para mejorar la detección de obstáculos y la planificación de rutas en entornos complejos. Su metodología incluyó la implementación y comparación de algoritmos como SLAM (Simultaneous Localization and Mapping) y el método de potencial artificial. Los resultados evidenciaron que la integración de ambas tecnologías incrementó la precisión de la navegación en entornos no estructurados. Aunque el estudio subraya los beneficios de combinar diferentes sensores, no aborda en profundidad las limitaciones asociadas al costo computacional y a la complejidad de

implementar sistemas híbridos, dejando una brecha en la evaluación de soluciones más accesibles y escalables.

Pérez (2021), en su trabajo titulado “Implementación de un sistema de navegación autónoma en robots móviles mediante sensores ultrasónicos” (Universidad Politécnica de Madrid, España), evaluó el desempeño de diferentes algoritmos de planificación de rutas, como A*, RRT (Rapidly-exploring Random Trees) y D* Lite, en entornos dinámicos. El objetivo fue determinar cuál de estos algoritmos ofrecía un mejor rendimiento según las condiciones del entorno y las limitaciones de los sensores ultrasónicos. La metodología consistió en realizar pruebas comparativas en simuladores y escenarios controlados, midiendo parámetros como tiempo de cómputo, eficiencia en la trayectoria y adaptabilidad.

Los resultados mostraron que A* era más eficiente en escenarios con obstáculos estáticos, mientras que RRT se destacó en entornos más complejos y cambiantes. Sin embargo, se identificó que el uso exclusivo de sensores de ultrasonido limitaba el rendimiento en ciertas situaciones. Este trabajo es directamente aplicable a la presente investigación, ya que resalta la importancia de seleccionar algoritmos de navegación que optimicen el uso de los sensores disponibles según las condiciones del entorno, un aspecto central en nuestra metodología propuesta.

Rodríguez (2022), en su investigación titulada Evaluación comparativa de algoritmos de evasión de obstáculos en robots móviles con sensores de ultrasonido (Universidad de Chile, Chile), comparó los algoritmos de campo potencial y comportamiento reactivo, evaluando su rendimiento en entornos controlados. Los criterios analizados incluyeron precisión, consumo computacional y velocidad de respuesta. Los resultados mostraron que el comportamiento reactivo ofrecía una mayor velocidad en la toma de decisiones, aunque generaba trayectorias menos óptimas.

Por otro lado, el campo potencial garantizó trayectorias más precisas a expensas de un mayor costo computacional. Este estudio enfatiza la necesidad de equilibrar eficiencia computacional y

precisión, pero no considera la posibilidad de integrar enfoques híbridos para superar estas limitaciones, dejando una brecha importante en la exploración de metodologías más versátiles.

Martínez (2023), en su estudio titulado Algoritmos de planificación de rutas en robots móviles: un enfoque basado en sensores de ultrasonido y análisis heurístico (Pontificia Universidad Católica de Valparaíso, Chile), evaluó los algoritmos A* y Dijkstra en escenarios controlados y no estructurados. Su metodología incluyó pruebas en entornos simulados y experimentales, midiendo tiempo de ejecución, consumo de recursos y calidad de las trayectorias generadas.

Los resultados evidenciaron que ambos algoritmos fueron efectivos en entornos predecibles, pero presentaron limitaciones en condiciones complejas y dinámicas, particularmente en la adaptación en tiempo real. Aunque el estudio demostró la importancia del análisis heurístico para optimizar la planificación de rutas, no abordó cómo los algoritmos podrían adaptarse en tiempo real a los cambios en el entorno, lo que constituye una brecha crítica para aplicaciones más versátiles.

Los estudios revisados proporcionan una base sólida para la presente investigación, pero también revelan lagunas significativas en el conocimiento, como la falta de metodologías integradoras que optimicen el uso de sensores de ultrasonido mediante una selección crítica y adaptativa de algoritmos. La propuesta metodológica de este trabajo busca llenar estas brechas al desarrollar un enfoque que no solo evalúe la eficiencia de los algoritmos, sino que también considere las limitaciones y ventajas inherentes a los sensores y los entornos en los que se emplean.

2.2. Fundamentación Teórica

2.2.1. Sensores de Ultrasonido en la Robótica y su Contribución a la Navegación

Autónoma

Los sensores ultrasónicos desempeñan un papel fundamental en la navegación autónoma debido a su capacidad para medir distancias con precisión a través de la emisión y recepción de ondas sonoras. Este principio permite detectar y evitar colisiones, independientemente de las condiciones de iluminación, lo que los hace especialmente útiles en ambientes adversos. Según Kumar et al. (2021), los sensores ultrasónicos son altamente eficientes en entornos de baja visibilidad, pero su desempeño está condicionado por factores como las propiedades del material de las superficies y la geometría de los objetos.

Estudios recientes destacan avances en el procesamiento de señales para mejorar la resolución y el alcance de estos sensores. Por ejemplo, Ahmed et al. (2022) señalan que el uso de algoritmos de filtrado y calibración automática ha permitido reducir el ruido y aumentar la precisión de las mediciones en escenarios con alta densidad de objetos. No obstante, persisten limitaciones, como la baja resolución angular, lo que dificulta la detección precisa de objetos en entornos complejos (Wang & Zhao, 2023).

Para superar estas barreras, investigaciones como las de Chen et al. (2023) han explorado la integración de sensores ultrasónicos con tecnologías complementarias creando sistemas de percepción multisensorial. Estos enfoques han demostrado ser efectivos para aumentar la robustez y la capacidad de los robots para adaptarse a escenarios cambiantes, aunque requieren algoritmos avanzados para fusionar los datos de manera eficiente.

2.2.2. Algoritmos de Navegación Autónoma: Perspectivas y Desafíos

Los algoritmos de navegación autónoma son esenciales para garantizar que los robots puedan operar en entornos no estructurados y complejos. Pueden clasificarse en dos enfoques principales: basados en mapas y reactivos. Según Tanaka et al. (2021), los algoritmos basados en mapas, como A* y Dijkstra, son ideales para entornos estructurados debido a su capacidad para planificar trayectorias óptimas utilizando una representación global del entorno. Sin embargo, enfrentan limitaciones de escalabilidad y eficiencia computacional en escenarios dinámicos.

Por otro lado, los algoritmos reactivos, como Dynamic Window Approach (DWA), toman decisiones en tiempo real basándose en información sensorial local. Este enfoque es más adecuado para entornos altamente dinámicos, pero, como mencionan Lee et al. (2022), puede conducir a trayectorias subóptimas y comportamientos inestables, como oscilaciones en la trayectoria, cuando los robots se enfrentan a configuraciones densas de obstáculos.

Otra tendencia emergente es el uso de algoritmos basados en aprendizaje profundo, como las redes neuronales convolucionales (CNN) y los modelos de refuerzo profundo. Estos enfoques han mostrado una capacidad sin precedentes para aprender patrones complejos y predecir trayectorias óptimas en escenarios diversos (Choi et al., 2023). Sin embargo, su aplicación está limitada por su elevada carga computacional y la necesidad de grandes volúmenes de datos de entrenamiento.

2.2.3. Criterios para la Selección de Algoritmos

La selección de algoritmos de navegación autónoma es un proceso crítico que afecta directamente el rendimiento del sistema. Este proceso debe considerar múltiples factores, como la complejidad computacional, la adaptabilidad al entorno y las limitaciones del hardware. De acuerdo con Zhang et al. (2022), los algoritmos con alta carga computacional pueden ser inadecuados para robots con recursos limitados, especialmente en aplicaciones en tiempo real.

Otro criterio clave es la capacidad de los algoritmos para adaptarse a entornos dinámicos. Estudios recientes, como los de Torres y Villalobos (2023), destacan la importancia de evaluar el rendimiento de los algoritmos en escenarios con cambios impredecibles, como la aparición de nuevos obstáculos. Además, la compatibilidad con el hardware disponible es esencial para garantizar que las soluciones sean viables y sostenibles en aplicaciones del mundo real.

Para abordar estas complejidades, se han desarrollado metodologías multicriterio, como el Proceso Analítico Jerárquico (AHP) y las técnicas de optimización basadas en redes Bayesianas. Estas herramientas permiten evaluar y comparar alternativas de manera estructurada, priorizando los criterios más relevantes para cada aplicación (Hernández et al., 2023). Adicionalmente, estudios recientes enfatizan la necesidad de considerar factores adicionales, como el consumo energético y la escalabilidad del sistema, para asegurar que las soluciones seleccionadas sean sostenibles y eficientes.

2.3. Marco Legal

2.3.1. Normativa Nacional

2.3.1.1. Constitución de 2008. La Constitución de 2008 establece un marco jurídico que promueve la investigación científica y el desarrollo tecnológico como herramientas para mejorar la calidad de vida y garantizar el bienestar social. El Artículo 385, en particular, enfatiza:

"El Estado promoverá el desarrollo de la ciencia, la tecnología y la innovación para mejorar la calidad de vida de la población, fortaleciendo la soberanía y el bienestar social" (Constitución de la República del Ecuador, 2008).

Este principio tiene una aplicación directa en la metodología para la selección de algoritmos de navegación autónoma en robots exploradores, ya que impulsa el uso de tecnologías que beneficien a la sociedad en ámbitos como la seguridad, la eficiencia y la sostenibilidad. Al desarrollar robots que empleen sensores de ultrasonido para una navegación precisa, se contribuye a la generación de conocimiento técnico y su aplicación en contextos que promuevan el bienestar social.

2.3.1.2. Ley Orgánica de Ciencia, Tecnología e Innovación de Ecuador. Esta ley proporciona lineamientos específicos para el desarrollo tecnológico en el país, con énfasis en la inclusión y la sostenibilidad. En su Artículo 1 se establece:

"El Estado impulsará el desarrollo de la ciencia, la tecnología y la innovación en función del interés general, promoviendo la cooperación internacional y el uso responsable de los recursos naturales" (Ley Orgánica de Ciencia, Tecnología e Innovación, 2016).

Este marco regula que las tecnologías desarrolladas, como los algoritmos de navegación, deben considerar no solo la eficiencia técnica, sino también los impactos sociales y ambientales. En el contexto del proyecto, esta normativa enfatiza la importancia de diseñar una metodología que

permita seleccionar algoritmos de navegación autónoma basados en sensores de ultrasonido, garantizando que los sistemas resultantes sean sostenibles y promuevan el bienestar general.

2.3.2. Normativa Internacional

2.3.2.1. Norma ISO 13482:2014. Esta norma establece los requisitos de seguridad para los robots personales, subrayando la necesidad de diseños que minimicen riesgos durante su operación. En su Sección 4.2 se especifica:

"Los robots personales deberán ser diseñados para evitar lesiones a las personas durante su operación, garantizando que los movimientos del robot sean predecibles y controlables" (ISO 13482:2014, p. 5).

En relación con el proyecto, esta norma es relevante porque los algoritmos seleccionados deben garantizar la seguridad en la navegación autónoma de robots exploradores, especialmente en entornos complejos. Integrar sensores de ultrasonido alineados con estos principios permite desarrollar sistemas que detecten y respondan a cambios en su entorno, reduciendo el riesgo de accidentes.

2.3.2.2. Norma ISO/IEC 30170:2012. establece directrices para el diseño seguro de sistemas de robots autónomos. En particular, resalta:

"Los robots autónomos deben ser diseñados para adaptarse a entornos dinámicos, con sensores que permitan detectar obstáculos y modificar su trayectoria en tiempo real para evitar colisiones" (ISO/IEC 30170:2012, p. 4).

Esta regulación subraya la necesidad de incorporar tecnologías de percepción avanzadas, como sensores de ultrasonido, para garantizar una navegación eficiente y segura. En el marco del proyecto, la norma orienta el diseño de una metodología que evalúe y seleccione algoritmos capaces

de integrar datos sensoriales en tiempo real, mejorando la adaptabilidad de los robots en entornos no estructurados.

2.4. Marco Conceptual

2.4.1. Adaptación en Tiempo Real

Componente esencial para los sistemas de navegación autónoma, ya que permite a los robots modificar su comportamiento de manera inmediata en respuesta a cambios en el entorno. Según Wang et al. (2022), los sistemas basados en aprendizaje adaptativo han optimizado la toma de decisiones en tiempo real, reduciendo los errores operativos y mejorando la eficiencia general del sistema.

2.4.2. Algoritmo A*

Técnica de búsqueda informada utilizada para encontrar rutas óptimas en grafos. Este enfoque combina heurísticas y costos acumulativos, lo que permite garantizar la eficiencia en la planificación de trayectorias. Según Rodríguez et al. (2021), las implementaciones modernas de A* han incorporado algoritmos paralelos para mejorar el rendimiento en aplicaciones en tiempo real.

2.4.3. Algoritmo Dijkstra

Garantiza la ruta más corta desde un nodo inicial a todos los nodos en un grafo. En el ámbito de la robótica, se ha adaptado para operar con sistemas distribuidos y entornos que requieren alta precisión. Según Martínez y López (2022), las implementaciones recientes han reducido significativamente el tiempo de cálculo, aumentando su aplicabilidad en robots autónomos.

2.4.4. Algoritmos de Aprendizaje

Permiten a los robots analizar datos en tiempo real y mejorar sus decisiones mediante el reconocimiento de patrones. Según Chen y Zhang (2023), las redes neuronales profundas y los algoritmos de refuerzo han transformado la capacidad de los robots para navegar en escenarios inciertos, adaptándose a cambios en el entorno con mayor eficiencia.

2.4.5. Algoritmos de Evasión

Combinan datos sensoriales para prevenir colisiones mediante ajustes automáticos de trayectoria. Según Hernández et al. (2023), la fusión de sensores ultrasónicos y cámaras ha mejorado significativamente la capacidad de los robots para maniobrar en entornos densos y desafiantes.

2.4.6. Algoritmos de Navegación

Los algoritmos de navegación son esenciales para planificar rutas y optimizar recursos como tiempo y energía. Según Torres y Villalobos (2023), la combinación de SLAM y técnicas de aprendizaje automático ha permitido a los robots operar con mayor precisión en entornos no estructurados.

2.4.7. Algoritmo RRT (Rapidly-exploring Random Tree)

Genera soluciones de planificación expandiendo un árbol de manera aleatoria. Según Ramírez et al. (2023), las mejoras recientes han optimizado el cálculo de trayectorias en espacios de configuración complejos, reduciendo los tiempos de respuesta.

2.4.8. Comportamiento Reactivo

Permite a los robots responder a estímulos externos de manera inmediata, sin depender de un plan previo. Según Gómez et al. (2022), este enfoque es ideal para aplicaciones donde la velocidad de respuesta es crítica.

2.4.9. Evasión de Obstáculos

Capacidad de detectar y evitar colisiones es esencial en la navegación robótica. Según Pérez y Álvarez (2023), la integración de sensores ultrasónicos con algoritmos predictivos ha mejorado significativamente la seguridad y eficiencia en la operación de robots exploradores.

2.4.10. Localización del Robot

Proceso mediante el cual el robot determina su posición en el entorno es crucial para la navegación autónoma. Según Hernández y Salinas (2023), el uso de SLAM combinado con sensores de alta resolución ha permitido a los robots ubicarse de manera confiable en entornos desconocidos.

2.4.11. Navegación Autónoma

Permite a los robots operar sin intervención humana. Según Díaz y Castro (2022), la integración de algoritmos de planificación y aprendizaje automático ha optimizado la navegación en escenarios dinámicos y desafiantes.

2.4.12. Sensores de Ultrasonido

Detectan objetos cercanos mediante ondas sonoras. Según Ramírez y Torres (2022), estos sensores, combinados con procesamiento avanzado, ofrecen una solución eficiente para la percepción en robots de bajo costo.

2.4.13. Redes Neuronales

Modelos computacionales inspirados en el cerebro humano, diseñados para reconocer patrones y aprender de los datos a través de conexiones entre neuronas artificiales. Según Blanco y García (2023), su aplicación en la robótica ha mejorado la eficiencia y la adaptabilidad de los sistemas.

2.4.14. Robots Exploradores

Diseñados para operar en entornos peligrosos y recolectar datos. Según Ramírez y González (2022), su aplicación en misiones científicas ha sido potenciada por el desarrollo de algoritmos de navegación avanzados.

2.4.15. Planificación de Rutas

Elegir rutas óptimas es crucial para la eficiencia de los robots. Según López et al. (2023), los avances en algoritmos de optimización han permitido reducir el consumo de energía mientras se mantiene la precisión en la navegación.

2.4.16. Procesamiento de Datos

El procesamiento eficiente de datos sensoriales es esencial para la toma de decisiones. Según Álvarez et al. (2022), los desarrollos recientes en hardware han permitido un procesamiento más rápido y preciso.

3. MARCO METODOLÓGICO

3.1. Enfoque de la Investigación

La presente investigación adopta un enfoque mixto, el cual integra métodos cualitativos y cuantitativos para abordar de manera integral el problema de estudio, centrado en la selección de algoritmos de navegación autónoma en robots exploradores equipados con sensores ultrasónicos. Este enfoque es ideal para responder al objetivo general de desarrollar una metodología estandarizada y los objetivos específicos relacionados con la evaluación, estructura, validación y replicabilidad de dicha metodología.

Desde la perspectiva cualitativa, el enfoque se orienta a explorar los fundamentos teóricos y conceptuales de los algoritmos de navegación. Según Yin (2018), el análisis cualitativo permite comprender las estructuras internas de los algoritmos, sus principios operativos y su aplicabilidad en entornos específicos. Esto incluye la identificación de criterios clave, como la eficiencia en la detección de obstáculos y la adaptabilidad en entornos dinámicos, que guían el diseño de la metodología propuesta. Este enfoque también facilita la recopilación de información experta y la síntesis de estudios previos relevantes, fortaleciendo el marco teórico y conceptual del trabajo.

Por su parte, el enfoque cuantitativo se centra en la evaluación de indicadores específicos de rendimiento. Como indican Creswell y Creswell (2021), los datos cuantitativos son esenciales para proporcionar evidencia empírica que valide las propuestas metodológicas. En este contexto, se medirá el rendimiento de los algoritmos seleccionados en términos de precisión en la navegación, tiempos de respuesta y consumo de recursos computacionales, utilizando simulaciones en entornos controlados. Estos resultados permitirán validar la efectividad de la metodología y generar un marco práctico replicable.

La combinación de ambos enfoques permite integrar perspectivas subjetivas y objetivas, lo que es fundamental para alcanzar una comprensión holística del problema. Además, garantiza que los resultados de la investigación sean relevantes tanto desde un punto de vista técnico como aplicado. Por ejemplo, la identificación cualitativa de criterios específicos se complementa con la validación cuantitativa mediante simulaciones, asegurando que los objetivos de definición, estructura y validación de la metodología sean alcanzados de manera rigurosa.

El enfoque mixto no solo responde a las necesidades del proyecto, sino que también permite construir una metodología fundamentada en un análisis integral que vincula la teoría con la práctica. Este enfoque garantiza que la metodología propuesta sea aplicable y replicable en diversos entornos operativos, cumpliendo así con los objetivos planteados.

3.2. Alcance de la Investigación

El alcance de esta investigación es de carácter exploratorio y correlacional, lo que posibilita abordar dos aspectos clave en el estudio de la navegación autónoma para robots móviles. El enfoque exploratorio se centra en la identificación y categorización de los algoritmos más destacados en el campo de la navegación autónoma que operan con sensores ultrasónicos, considerando sus características técnicas, ventajas, limitaciones y su evolución reciente en la literatura. Este enfoque es importante para identificar vacíos de conocimiento y establecer un marco conceptual sólido para futuras investigaciones (Creswell & Creswell, 2018). Además, responde a la necesidad de comprender las capacidades y aplicaciones potenciales de estos algoritmos en un contexto tecnológico en constante evolución.

En su dimensión correlacional, analiza la relación entre variables clave, tales como el tiempo de ejecución, el consumo de recursos y la escalabilidad, y su influencia en el desempeño de los algoritmos de navegación autónoma. Este análisis permite evaluar cómo estas variables afectan indicadores de desempeño como la precisión en la detección de obstáculos, la eficiencia en la planificación de rutas y el tiempo de respuesta en escenarios dinámicos (Saunders, Lewis, & Thornhill, 2019).

3.3. Delimitación de la Investigación

3.3.1. Delimitación Espacial

El presente estudio se enfoca en entornos simulados con obstáculos variables y rutas no estructuradas que emulan condiciones dinámicas y complejas características de escenarios exploratorios, utilizando el software de simulación para robótica móvil en entornos virtuales llamado Webots. Estos entornos permiten un control preciso de las variables involucradas en la navegación autónoma. La elección de la simulación fortalece la aplicabilidad de los resultados al garantizar que las pruebas se realicen en un ambiente controlado, donde es posible replicar y ajustar condiciones específicas de navegación, como la respuesta a diferentes patrones de obstáculos y la adaptabilidad de los algoritmos seleccionados.

La simulación ofrece la posibilidad de explorar una variedad de configuraciones operativas y optimizar los algoritmos antes de implementarlos en condiciones reales, lo que fortalece la aplicabilidad y la escalabilidad de los resultados.

3.3.2. Delimitación Temporal

La recolección de información abarca una investigación de trabajos comprendido entre los años 2018 y 2024, una etapa caracterizada por avances significativos en el desarrollo de algoritmos de navegación autónoma y en la integración de sensores ultrasónicos en sistemas robóticos. Este

rango temporal asegura que el estudio se fundamenta en un conocimiento actualizado y relevante que responda a los desafíos contemporáneos de la navegación autónoma.

Adicionalmente, la ejecución práctica del proyecto se desarrolla entre octubre y noviembre de 2024, periodo durante el cual se llevará a cabo la fase de simulación, evaluación y análisis de los algoritmos de navegación seleccionados. Además, esta delimitación temporal fortalece la aplicabilidad de los resultados, ya que garantiza que las metodologías y algoritmos evaluados reflejen las condiciones y exigencias actuales de la tecnología de navegación autónoma.

En conjunto, las delimitaciones espacial y temporal establecidas no solo enmarcan el alcance del estudio, sino que también aseguran que los resultados obtenidos sean relevantes y aplicables en entornos operativos diversos. La combinación de simulaciones avanzadas y la consideración de desarrollos recientes refuerza la confiabilidad y la utilidad práctica de los hallazgos.

3.4. Población y Muestra

3.4.1. Población

La población de este estudio está conformada por los algoritmos de navegación autónoma empleados en robots exploradores. Estos algoritmos comprenden:

- SLAM (Simultaneous Localization and Mapping).
- Algoritmo A*.
- Filtro de partículas.
- Redes neuronales y algoritmos de refuerzo.
- RRT (Rapidly-exploring Random Tree).
- Dijkstra.

Estos algoritmos representan un conjunto diverso y complementario de soluciones tecnológicas que abordan las necesidades específicas de los entornos y las misiones de exploración robótica. Su inclusión en este estudio se fundamenta en su efectividad probada y en los avances recientes que garantizan su aplicabilidad en contextos actuales, constituyendo así un marco de análisis integral para la navegación autónoma en robots exploradores.

3.4.2. Muestra

La muestra de este estudio está compuesta por tres algoritmos reconocidos en la literatura por su eficacia en la planificación de rutas y la evasión de obstáculos:

- Algoritmo A*.
- Dijkstra.
- RRT (Rapidly-exploring Random Tree).

Estos algoritmos fueron seleccionados bajo criterios específicos que garantizan su relevancia y aplicabilidad para el objetivo del estudio:

- Algoritmos diseñados específicamente para la navegación autónoma.
- Compatibilidad demostrada con sensores ultrasónicos.
- Evaluaciones de desempeño verificadas en entornos simulados, garantizando la validez de los resultados.

La selección de estos tres algoritmos responde a su idoneidad para abordar los objetivos de investigación y a su reconocimiento en la literatura como soluciones efectivas en diferentes escenarios de navegación autónoma.

3.5. Métodos Empleados

3.5.1. Recolección de Datos

La recolección de datos se llevó a cabo mediante una revisión bibliográfica de publicaciones científicas, centradas en algoritmos de navegación autónoma utilizados en robots exploradores con sensores de ultrasonido y entrevistas semiestructuradas a expertos en robótica y navegación autónoma.

En la revisión bibliográfica se definieron criterios con el propósito de asegurar la relevancia y actualidad de las fuentes consultadas, limitando el análisis a estudios publicados entre los años 2018 y 2024. Este enfoque permitió identificar las tendencias contemporáneas en el desarrollo de algoritmos de navegación, así como los principales desafíos técnicos y las estrategias de optimización más utilizadas en el ámbito.

Se priorizó la inclusión de trabajos que examinaran los algoritmos más utilizados en la navegación autónoma, especialmente aquellos que emplean sensores de ultrasonido como tecnología central para la percepción del entorno. El análisis de estas fuentes proporcionó una comprensión detallada de los principales desafíos técnicos asociados con la navegación en entornos dinámicos, así como de las estrategias de optimización que se han implementado para mejorar la eficiencia, la precisión y la robustez de los algoritmos. Asimismo, se identificaron las métricas de evaluación más comunes empleadas en la evaluación del desempeño de estos algoritmos, lo cual fue esencial para el desarrollo de un marco comparativo y la selección de los algoritmos más adecuados para el estudio.

Adicionalmente, se realizaron entrevistas semiestructuradas con expertos en el campo de la robótica, la navegación autónoma y el uso de sensores ultrasónicos en sistemas robóticos. Estas entrevistas se diseñaron para obtener información cualitativa acerca de los aspectos clave en la

evaluación y selección de algoritmos de navegación, enfocándose particularmente en las métricas de evaluación empleadas en el análisis de desempeño de dichos algoritmos.

Las entrevistas semiestructuradas permitieron un enfoque flexible, en el cual los expertos pudieron compartir sus conocimientos y experiencias de manera detallada, abordando temas como la aplicabilidad de los algoritmos en diversos escenarios, la efectividad de las técnicas de optimización, las limitaciones de los sensores ultrasónicos en la navegación autónoma y las consideraciones específicas en la selección de métricas. Este formato de entrevista facilitó la exploración profunda de las perspectivas de los expertos, permitiendo que se abordaran tanto cuestiones técnicas como prácticas en el campo de la robótica, complementando y enriqueciendo la información obtenida a partir de la revisión bibliográfica.

3.5.2. Simulaciones en Entornos Virtuales

Las simulaciones representaron una herramienta fundamental para la evaluación de los algoritmos de navegación. Se utilizaron entornos simulados creados con el software Webots, que proporcionó una plataforma flexible para replicar diversos escenarios de navegación en los que los robots exploradores debían realizar tareas de localización, planificación de rutas y evasión de obstáculos. Este método permitió simular de manera controlada y repetitiva el comportamiento de los algoritmos sin las limitaciones inherentes al uso de robots físicos, tales como el costo, el tiempo y los riesgos asociados a los errores de navegación.

En las simulaciones, los algoritmos fueron evaluados bajo diferentes condiciones y configuraciones, tales como variaciones en el entorno, modificaciones en los parámetros de los sensores ultrasónicos y la inclusión de obstáculos imprevistos. A través de estas simulaciones, fue posible medir métricas clave del desempeño, como el tiempo de ejecución, la precisión en la generación de rutas y la capacidad de evasión de obstáculos en situaciones dinámicas.

Este método también permitió una variabilidad controlada de los escenarios, lo que permitió evaluar el comportamiento de los algoritmos en contextos diversos y complejos. Las simulaciones fueron configuradas para reflejar condiciones representativas de los entornos en los que los robots exploradores pueden operar en el mundo real, asegurando la relevancia de los resultados obtenidos.

3.5.3. Procesamiento y Análisis de Datos

El procesamiento y análisis de los datos en la presente investigación se realizaron de manera rigurosa y sistemática, combinando enfoques cualitativos y cuantitativos que permitieron abordar integralmente el fenómeno de estudio. El objetivo principal fue garantizar la organización crítica de los datos, identificar patrones significativos y extraer información relevante para el desarrollo de la metodología propuesta.

3.5.3.1. Análisis de Datos Cualitativos. Para los datos cualitativos obtenidos a partir de entrevistas semiestructuradas con expertos en navegación robótica, se llevó a cabo un análisis de contenido detallado. El proceso se estructuró en varias partes:

- **Transcripción y Lectura Integral.** Todas las entrevistas fueron transcritas en su totalidad para preservar la fidelidad de las respuestas. Se realizó una primera lectura exhaustiva de las transcripciones con el fin de familiarizarse con el contenido y obtener una visión general de las respuestas.
- **Codificación Abierta.** Se identificaron segmentos de texto relevantes y se asignaron códigos específicos a las ideas y conceptos clave mencionados por los expertos. La codificación se realizó de forma inductiva, lo que permitió que los temas emergieran directamente de los datos sin imponer categorías predefinidas (Miles, Huberman & Saldaña, 2019).

- **Agrupación y Categorización.** Los códigos identificados se agruparon en categorías y subcategorías relacionadas con las métricas de evaluación de los algoritmos, tales como:

- Precisión en la detección de obstáculos.
- Tiempo de respuesta en entornos dinámicos.
- Eficiencia en el consumo de recursos computacionales.
- Robustez y fiabilidad ante condiciones adversas.
- Adaptabilidad a cambios inesperados en el entorno.
- Seguridad y minimización de colisiones de accidentes en simulaciones.
- Escalabilidad en escenarios complejos.

- **Análisis Crítico y Comparativo.** Se realizó un análisis comparativo de las respuestas entre los distintos expertos para identificar coincidencias y divergencias en sus opiniones. Este análisis permitió extraer patrones de consenso y enfoques divergentes que enriquecieron la comprensión de los criterios y métricas relevantes.

3.5.3.2. Análisis de Datos Cuantitativos. Los datos cuantitativos se obtuvieron a través de la evaluación de los algoritmos en entornos simulados utilizando el software Webots. Este entorno virtual permitió configurar escenarios dinámicos, con variaciones en la distribución de obstáculos y complejidad de las rutas. El análisis de los datos cuantitativos se estructuró en las siguientes etapas:

- **Diseño de Escenarios de Simulación.** Se definieron diferentes escenarios de prueba que incluyeron condiciones de navegación como:
 - Obstáculos fijos y variables.

- Rutas simples y complejas.
- Variabilidad en la densidad de obstáculos.
- **Evaluación de Métricas de Rendimiento:** Durante las simulaciones, se recopilaron datos cuantitativos sobre indicadores clave de rendimiento:
 - **Tiempo de ejecución:** Tiempo requerido para completar una trayectoria.
 - **Precisión en la navegación:** Capacidad del algoritmo para seguir rutas óptimas.
 - **Capacidad de adaptabilidad al entorno:** Eficiencia en evitar colisiones.
 - **Eficiencia computacional:** Uso de recursos computacionales durante la ejecución.
 - **Robustez y fiabilidad:** Habilidad para funcionar de forma estable ante situaciones adversas.
 - **Seguridad y colisiones:** Eficiencia para prevenir accidentes.
 - **Escalabilidad:** Capacidad de mantener un buen desempeño al enfrentarse a escenarios más complejos.
- **Procesamiento de Datos.** Los datos recopilados fueron organizados y procesados utilizando Microsoft Excel y herramientas de análisis estadístico. Las métricas fueron tabuladas y se calcularon estadísticas descriptivas, como promedios, desviaciones estándar y porcentajes de error.
- **Generación de Representaciones Gráficas.** Se elaboraron representaciones visuales, como diagramas de barras y gráficos de dispersión, para comparar de manera clara y efectiva el rendimiento de los algoritmos evaluados. Estas representaciones facilitaron la identificación de tendencias y patrones en el desempeño de los algoritmos.

- **Análisis Comparativo.** Los resultados obtenidos de las simulaciones fueron comparados entre los diferentes algoritmos, lo que permitió determinar sus fortalezas y debilidades en función de las métricas definidas. Este análisis proporcionó una base objetiva para evaluar la aplicabilidad y eficiencia de los algoritmos en escenarios reales.

3.6. Herramientas Utilizadas

En esta investigación, se emplean herramientas clave que facilitan tanto la recolección como el análisis de los datos recopilados, permitiendo abordar los objetivos de manera eficiente y rigurosa.

3.6.1. Codificación Abierta

Este enfoque consiste en identificar y asignar códigos a segmentos de texto relevantes para descomponer los datos cualitativos en unidades manejables, lo que permite analizar los datos de las entrevistas semiestructuradas, permitiendo identificar y clasificar los aspectos clave relacionados con las métricas de evaluación de los algoritmos de navegación, facilitando así la interpretación de la información y la identificación de patrones recurrentes.

3.6.2. Análisis Temático.

Este enfoque implica identificar, analizar y reportar patrones dentro de los datos, lo que permite interpretar las tendencias y características comunes de los artículos revisados de los algoritmos de navegación autónoma en la literatura, lo que permitió afinar la interpretación de las metodologías y criterios de selección de algoritmos presentes en los estudios revisados.

3.6.3. Microsoft Excel

Excel es utilizado para procesar y analizar los datos cuantitativos derivados de las simulaciones realizadas. Se emplean sus capacidades de fórmulas avanzadas, tablas dinámicas y

generación de gráficos para calcular y visualizar métricas como la precisión, eficiencia en el tiempo de respuesta y consumo de recursos computacionales. Su flexibilidad permite realizar análisis estadísticos descriptivos que respaldan la evaluación comparativa de los algoritmos estudiados.

3.6.4. Webots

Webots es un software de simulación de robots ampliamente utilizado en la investigación y desarrollo de sistemas de navegación autónoma. Este entorno permite la implementación y simulación de robots en escenarios virtuales, proporcionando una plataforma flexible para evaluar el rendimiento de los algoritmos de navegación. Los usuarios pueden crear entornos de prueba personalizados, donde los algoritmos pueden ser evaluados en términos de su capacidad para detectar obstáculos, planificar rutas y operar eficientemente bajo diversas condiciones ambientales.

Webots es particularmente útil en estudios de robótica, ya que permite realizar pruebas controladas y repetibles sin la necesidad de utilizar hardware físico, lo que facilita la comparación de diferentes algoritmos en condiciones variables.

3.7. Fases de la Metodología

Se han definido las fases que estructuran el desarrollo del proyecto de integración curricular. Estas fases incorporan una sólida fundamentación teórica para garantizar resultados rigurosos y fundamentados. A continuación, se describen cada una de las fases:

3.7.1. Fase 1: Planificación y Diseño del Estudio

En esta fase inicial, se definieron los objetivos generales y específicos del proyecto, así como el alcance, las limitaciones y los recursos tecnológicos necesarios para su desarrollo. A fin de asegurar un diseño metodológico robusto y específico, se establecieron lineamientos clave para la selección de algoritmos de navegación, enfocados en criterios tales como eficiencia, adaptabilidad y robustez en la detección de obstáculos.

La participación de expertos en robótica autónoma y en tecnologías de información desempeñó un papel crucial en la identificación de los principales requerimientos técnicos del proyecto. Según Torres y Morales (2021), involucrar a especialistas desde las primeras fases del diseño facilita la definición de preguntas específicas y el establecimiento de criterios que orienten la evaluación de los algoritmos. Mediante entrevistas semiestructuradas y sesiones de retroalimentación, se validaron las variables clave del estudio, como:

- Capacidad de procesamiento en tiempo real.
- Precisión en la detección de objetos.
- Adaptabilidad en entornos dinámicos y complejos.

Este proceso permitió garantizar que la planificación inicial estuviera alineada con las necesidades técnicas del proyecto y su aplicabilidad en escenarios del mundo real, como exploración industrial o misiones de rescate.

3.7.2. Fase 2: Recolección y Definición de Información

En esta etapa, se llevó a cabo una revisión de la literatura científica y técnica, abarcando artículos publicados en el período comprendido entre los años 2018 y 2024, aplicando criterios de selección para garantizar la calidad y actualidad de las fuentes consultadas.

El análisis de la literatura permitió identificar los algoritmos de navegación autónoma más relevantes, así como sus ventajas y limitaciones en entornos no estructurados. Según García et al. (2022), los enfoques basados en algoritmos clásicos como A* y Dijkstra, y en modelos modernos como RRT (Rapidly-exploring Random Tree), han mostrado desempeños variables según el contexto de aplicación y los recursos computacionales disponibles.

Durante esta fase, se seleccionaron los algoritmos que serían evaluados en simulaciones, priorizando aquellos con mayor potencial de adaptabilidad y eficiencia computacional en escenarios dinámicos. Además, se definieron las métricas de evaluación, tales como:

- Precisión en la detección de obstáculos.
- Tiempo de respuesta en entornos dinámicos.
- Eficiencia en el consumo de recursos computacionales.
- Robustez y fiabilidad ante condiciones adversas.
- Adaptabilidad a cambios inesperados en el entorno.
- Seguridad y minimización de colisiones de accidentes en simulaciones.
- Escalabilidad en escenarios complejos.

3.7.3. Fase 3: Implementación en Simulaciones Computacionales

En esta fase, los algoritmos seleccionados fueron implementados y evaluados en la plataforma de simulación Webots, reconocida por su capacidad para modelar escenarios virtuales complejos y precisos. La configuración de los entornos simulados incluyó:

- Obstáculos de diferente densidad y distribución.
- Variabilidad en el nivel de complejidad de las rutas.

Se realizaron pruebas para evaluar el rendimiento de los algoritmos bajo distintas condiciones operativas. Según Ruiz et al. (2023), la simulación computacional permite validar el comportamiento de los sistemas en un entorno controlado, facilitando ajustes iterativos sin incurrir en los costos asociados a pruebas físicas.

3.7.4. Fase 4: Análisis y Evaluación de Resultados

Con los datos obtenidos de las simulaciones, se realizó un análisis comparativo utilizando herramientas como Microsoft Excel para procesar métricas clave. El análisis incluyó:

- **Evaluación de Métricas de Desempeño.** Se analizaron las métricas definidas previamente, como el tiempo de respuesta, la precisión en la navegación y la eficiencia en la detección de obstáculos. Las representaciones visuales permitieron comparar el rendimiento de cada algoritmo y detectar tendencias significativas.
- **Análisis Comparativo y Validación.** Se realizó un análisis comparativo crítico entre los algoritmos evaluados, identificando fortalezas y limitaciones en su aplicabilidad a escenarios reales. Según Lin et al. (2023), el análisis comparativo es fundamental para establecer recomendaciones basadas en evidencia empírica.

4. ANÁLISIS DE RESULTADOS

4.1. Fase 1: Planificación y Diseño del Estudio

En esta fase preliminar, se establecieron las bases fundamentales del proyecto mediante un análisis técnico crítico que garantizara decisiones metodológicas alineadas con los objetivos del estudio. Las actividades desarrolladas en esta etapa se centraron en definir un enfoque estructurado y fundamentado para abordar los desafíos asociados a la selección de algoritmos de navegación autónoma en robots exploradores equipados con sensores ultrasónicos.

4.1.1. Definición de Objetivos Generales y Específicos

Se establecieron los objetivos generales y específicos mediante un análisis crítico de las necesidades y vacíos de conocimiento identificados en la literatura reciente. El objetivo general se enfocó en diseñar una metodología estandarizada para la selección de algoritmos de navegación autónoma, priorizando criterios de eficacia, adaptabilidad y rendimiento en entornos diversos. Este objetivo se tradujo en objetivos específicos alineados con las demandas tecnológicas y operativas del proyecto:

- Identificar métricas de evaluación críticas que guíen la selección de algoritmos.
- Diseñar un marco metodológico que integre las mejores prácticas de navegación robótica.
- Validar la metodología propuesta mediante simulaciones controladas en entornos virtuales.
- Proveer un marco práctico que garantice la replicabilidad y adaptabilidad de los resultados en aplicaciones reales.

4.1.2. Delimitación del Alcance del Proyecto

El alcance del proyecto fue delimitado de manera precisa para garantizar su viabilidad técnica y metodológica. La investigación se enfocó en la evaluación de algoritmos en entornos simulados que emulan condiciones complejas y dinámicas. Los entornos virtuales fueron diseñados en la plataforma Webots, la cual permite un control riguroso de las variables y facilita la replicación de los experimentos en diferentes configuraciones.

Además, se estableció que los hallazgos del estudio serán aplicables a contextos reales mediante ajustes operativos. Esto incluye la adaptación de los algoritmos evaluados para su uso en instalaciones industriales, zonas de rescate y exploración en entornos terrestres confinados. Esta delimitación asegura que el proyecto ofrezca resultados relevantes y transferibles.

4.1.3. Identificación de Limitaciones

Se analizaron de manera crítica las posibles limitaciones del proyecto para mitigar su impacto en los resultados. Entre las principales limitaciones identificadas se encuentran:

- **Dependencia de Entornos Simulados.** Aunque las simulaciones ofrecen ventajas como el control de variables, también introducen diferencias respecto a condiciones reales. Para mitigar esto, se incluyeron configuraciones que emulan entornos no estructurados y escenarios impredecibles.
- **Restricciones de Hardware Especializado.** La ausencia de pruebas experimentales en hardware físico fue considerada una limitación, lo que enfatiza la importancia de validar los algoritmos en simulaciones robustas y controladas.
- **Complejidad Computacional de algunos Algoritmos.** Se identificó la necesidad de equilibrar la eficiencia computacional con la precisión y robustez de los algoritmos evaluados.

4.1.4. Establecimiento de Necesidades Principales.

Se identificaron las necesidades técnicas y logísticas esenciales para garantizar el éxito del proyecto. Estas incluyeron:

- **Fuentes Bibliográficas Relevantes.** La adquisición de literatura científica reciente y especializada en navegación robótica y algoritmos de optimización.
- **Colaboración con Expertos.** La participación activa de especialistas en robótica y tecnologías de información para orientar decisiones metodológicas.
- **Plataforma de Simulación.** La selección de Webots como entorno experimental, dada su capacidad para simular escenarios complejos y medir métricas clave.

4.1.5. Consulta con expertos

Se realizaron entrevistas semiestructuradas con especialistas en robótica autónoma para profundizar en la comprensión de los desafíos asociados a la selección de algoritmos de navegación. Su participación también fue fundamental para la formulación de preguntas esenciales que forman parte del desarrollo del estudio, tales como:

- ¿Qué criterios son necesarios para evaluar la efectividad de un algoritmo en escenarios dinámicos?
- ¿Cómo afectan las características del sensor ultrasónico a la adaptabilidad de los algoritmos seleccionados?

La información recabada se integró en el diseño del estudio, permitiendo una alineación entre los objetivos teóricos y las necesidades prácticas. Según Ramírez y Torres (2022), la colaboración con expertos mejora la validez de las decisiones metodológicas y fortalece la aplicabilidad del proyecto.

4.1.6. Elaboración del Cronograma

Se diseñó un cronograma detallado que estructuró las actividades y los plazos correspondientes a cada fase del proyecto. Este cronograma incluyó hitos clave, como la finalización de la revisión bibliográfica, la implementación de simulaciones y la redacción del informe final. La planificación temporal garantizó un progreso eficiente y controlado, minimizando riesgos asociados a demoras o conflictos de prioridades:

4.1.6.1. Fase 1: Planificación y Diseño del Estudio (Semana 1 - Semana 2)

- **Definición de Objetivos y Alcance:**
 - Redactar los objetivos generales y específicos del proyecto con base en la problemática identificada.
 - Delimitar el alcance considerando los recursos disponibles y las limitaciones tecnológicas.
 - Realizar reuniones iniciales para ajustar los objetivos a las necesidades del campo de estudio.
- **Selección de Bibliografía Relevante:**
 - Organización de referencias bibliográficas.
 - Identificación de estudios clave relacionados con algoritmos de navegación y sensores ultrasónicos publicados entre 2020 y 2024.
- **Consultas con Expertos:**
 - Diseñar un cuestionario semiestructurado para las entrevistas con especialistas.
 - Realizar al menos tres entrevistas iniciales con expertos en robótica autónoma para validar el diseño del estudio y obtener retroalimentación técnica.

- Documentar las conclusiones de las consultas para ajustar los enfoques metodológicos.

4.1.6.2. Fase 2: Recolección y Definición de Información (Semana 3 - Semana 5)

- **Revisión Bibliográfica:**

- Análisis de estudios académicos relacionados con algoritmos de navegación y sensores ultrasónicos.
- Identificación de tendencias en algoritmos, así como avances en tecnologías de detección con sensores.
- Sistematización de la información en un esquema temático para su inclusión en el marco teórico.

- **Selección de Algoritmos:**

- Definición de criterios de selección basados en eficiencia computacional, adaptabilidad y aplicabilidad en entornos complejos.
- Evaluación de algoritmos candidatos mediante comparación de estudios previos.
- Confirmación de los algoritmos más relevantes para su implementación.

- **Definición de Métricas de Evaluación:**

- Establecimiento de métricas clave de evaluación.
- Validación de las métricas mediante consultas con expertos.
- Diseño de instrumentos para recopilar datos durante las simulaciones.

4.1.6.3. Fase 3: Implementación y Simulaciones Computacionales (Semana 6 - Semana 8)

- **Diseño del Entorno Experimental:**
 - Planificación de escenarios de simulación con diferentes niveles de complejidad.
 - Configuración de entornos que incluyan obstáculos fijos y móviles.
- **Configuración del Entorno en Webots:**
 - Creación de escenarios simulados en Webots utilizando modelos de robots exploradores equipados con sensores ultrasónicos.
- **Implementación de Algoritmos:**
 - Programación de los algoritmos en los controladores de los robots simulados.
- **Ejecución de Simulaciones:**
 - Realización de pruebas repetitivas bajo diferentes condiciones controladas.
 - Registro de datos relacionados con las métricas definidas en escenarios predefinidos y escenarios dinámicos.

4.1.6.4. Fase 4: Análisis y Evaluación de Resultados (Semana 9 - Semana 10)

- **Procesamiento de Datos:**
 - Análisis estadístico de los datos recopilados utilizando Microsoft Excel.
 - Generación de tablas y gráficos comparativos para cada métrica evaluada.
 - Identificación de patrones y tendencias en el desempeño de los algoritmos.
- **Comparación de Algoritmos:**
 - Evaluación crítica de las fortalezas y debilidades de los algoritmos en función de las métricas predefinidas.

- **Redacción Preliminar de Conclusiones:**

- Integración de hallazgos clave en un formato inicial.
- Preparación de los resultados para su inclusión en el informe final del proyecto.

4.2. Fase 2: Recolección y Definición de Información

Esta fase tuvo como objetivo la recopilación y sistematización de información relevante para fundamentar el desarrollo del proyecto, centrándose en la selección de algoritmos y la definición de métricas de evaluación. A continuación, se detalla el desarrollo de las actividades realizadas:

4.2.1. Selección de Estudios de Algoritmos.

El propósito de esta búsqueda fue identificar, analizar y seleccionar los algoritmos que van a ser evaluados como parte del proyecto. Estos algoritmos fueron escogidos por su eficacia demostrada en la planificación de rutas y la navegación autónoma, teniendo en cuenta las características específicas de los entornos y las necesidades técnicas del estudio.

Teniendo como principal criterio de elección los artículos publicados entre 2018 y 2024 con el fin de garantizar la elección de algoritmos relevantes en el campo de los algoritmos de navegación autónoma, que fueron los siguientes:

- **Algoritmo A*.** Este algoritmo se seleccionó por su capacidad para planificar rutas óptimas utilizando una función heurística que equilibra el costo acumulado y el estimado restante. Es particularmente eficaz en entornos estáticos y predefinidos, lo que lo convierte en una opción ideal para escenarios donde la incertidumbre es mínima. Según Smith (2020), A* sobresale por su eficiencia computacional, permitiendo encontrar soluciones rápidas y precisas en mapas bien definidos. Este algoritmo es altamente relevante para robots exploradores que operan en entornos controlados.

- **Algoritmo Dijkstra.** Elegido por su robustez en la resolución de problemas de optimización en redes de nodos. Aunque presenta una mayor carga computacional, ofrece garantías de soluciones óptimas incluso en escenarios complejos. Según Pérez et al. (2023), su implementación combinada con técnicas de actualización en tiempo real permite su aplicación en entornos parcialmente dinámicos.

- **Algoritmo RRT (Rapidly-exploring Random Tree).** Este algoritmo destaca por su capacidad para explorar rápidamente espacios de alta dimensión y generar trayectorias viables en tiempo real. Es especialmente útil en entornos dinámicos y complejos, donde los robots enfrentan alta incertidumbre. Según Hernández y Gómez (2022), su flexibilidad y velocidad lo convierten en una herramienta esencial para misiones de exploración autónoma. Su combinación con sensores ultrasónicos mejora la detección y evasión de obstáculos en tiempo real, lo que garantiza un desempeño seguro y eficiente en entornos desconocidos.

Estos algoritmos fueron seleccionados por su complementariedad en la cobertura de diferentes escenarios operativos, lo que permite evaluar su desempeño tanto en condiciones controladas como en situaciones de alta dinámica e incertidumbre.

4.2.2. Análisis de las Métricas de Evaluación

Este análisis se fundamentó en criterios planteados por autores como Smith (2020), quienes destacan la importancia de evaluar el desempeño holístico de los algoritmos, no solo en entornos ideales, sino también en condiciones desafiantes. Por lo tanto, en la Tabla 1 se establecieron las métricas de evaluación más relevantes para la evaluación de los algoritmos seleccionados:

Tabla 1

Tabla de Métricas de Evaluación de Algoritmos de Navegación

Métrica	Motivo
Precisión en la navegación	Cuantifica la desviación promedio entre la trayectoria planificada y la ruta recorrida por el robot. Se evalúa en porcentaje, reflejando la capacidad del algoritmo para alcanzar el destino previsto.
Tiempo de ejecución	Mide el tiempo necesario desde la inicialización del algoritmo hasta la obtención de una solución, evaluando la viabilidad del algoritmo en aplicaciones de tiempo real.
Consumo de recursos	Evalúa el uso de CPU y memoria para determinar la eficiencia computacional de cada algoritmo.
Robustez y fiabilidad	Mide la resiliencia del algoritmo, evaluando su capacidad para mantener un desempeño estable ante condiciones adversas.
Adaptabilidad al entorno	Evalúa la rapidez con la que los algoritmos se ajustan a cambios en el entorno, como la aparición de obstáculos inesperados.
Seguridad y colisiones	Registra el número de colisiones durante las simulaciones, reflejando la capacidad del algoritmo para evitar accidentes.
Escalabilidad	Evalúa la capacidad del algoritmo para mantener un rendimiento aceptable en escenarios de mayor complejidad mediante simulaciones escaladas.

Estas métricas no solo permitieron una evaluación detallada de cada algoritmo, sino que también ofrecieron una base técnica para identificar sus fortalezas y limitaciones.

4.3. Fase 3: Implementación en Simulaciones Computacionales

La fase de implementación y simulaciones computacionales fue fundamental para evaluar el desempeño de los algoritmos seleccionados (A*, Dijkstra y RRT) en entornos controlados y dinámicos. A través de esta fase, se simuló una serie de escenarios en la plataforma Webots, lo que permitió medir parámetros clave como la eficiencia, la capacidad de evasión de obstáculos y la adaptabilidad de los algoritmos al entorno. A continuación, se detallan las actividades realizadas durante esta fase:

4.3.1. Preparación del Entorno de Simulación en Webots

La configuración de los entornos de simulación en Webots se realizó siguiendo un procedimiento estructurado para garantizar la validez y reproducibilidad de los experimentos:

- **Diseño y Creación de Escenarios Controlados.** Se crearon escenarios simulados con configuraciones diversas de obstáculos, desde entornos simples con rutas predefinidas hasta entornos complejos que incluían obstáculos móviles. Según Pérez et al. (2023), la variabilidad en los escenarios permite evaluar la robustez y adaptabilidad de los algoritmos en condiciones diversas.
- **Implementación de los algoritmos.** Una vez configurados los escenarios, los algoritmos seleccionados (A*, Dijkstra y RRT) se implementaron directamente en los controladores del robot utilizando Webots, lo que permitió monitorear en tiempo real el comportamiento del robot, la detección de obstáculos y la ejecución de rutas.

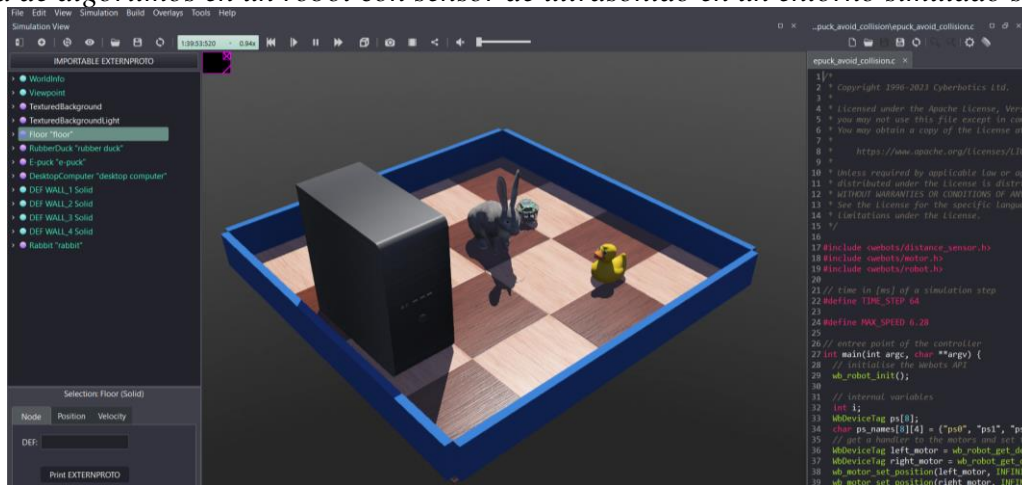
4.3.2. Ejecución de las Simulaciones

Las simulaciones se llevaron a cabo en escenarios diseñados para evaluar diferentes aspectos del desempeño de los algoritmos.

4.3.2.1. Entorno Simulado con Obstáculos de Diferente Tamaño y Distribución. En el primer entorno simulado, representado en la Figura 1, se diseñó un espacio dentro de una zona delimitada con pocos objetos de diferente tamaño, lo que permitió analizar el desempeño de los algoritmos bajo condiciones moderadas:

Figura 1

Prueba de algoritmos en un robot con sensor de ultrasonido en un entorno simulado simple.

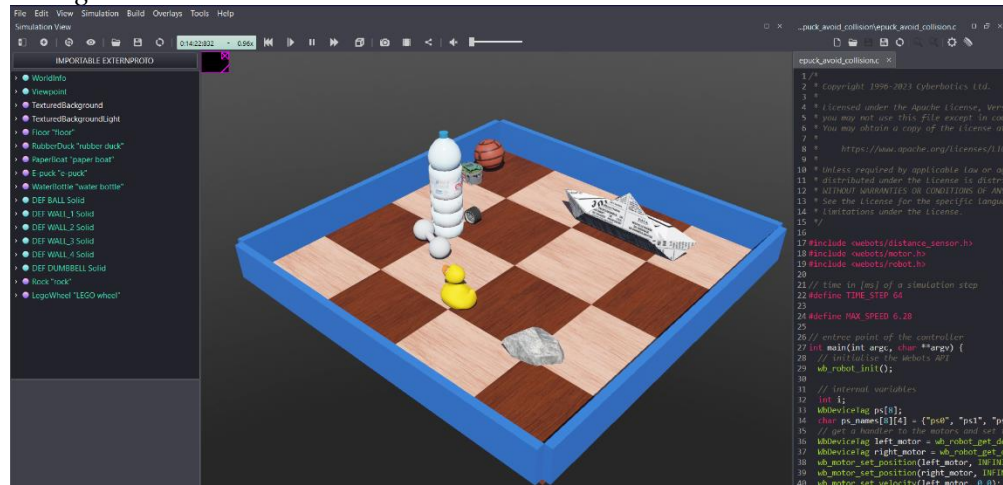


4.3.2.2. Entorno Simulado con Variabilidad en el Nivel de Complejidad de las rutas.

En el segundo entorno simulado, representado en la Figura 2, se diseñó un escenario con un mayor número de objetos con el propósito de incrementar el nivel de complejidad. lo que permitió analizar el desempeño de los algoritmos bajo condiciones más desafiantes y realistas:

Figura 2

Prueba de algoritmos en un robot con sensor ultrasónico en un entorno simulado complejo.



Las simulaciones se estructuraron para medir las siguientes métricas de evaluación previamente definidas:

- **Adaptabilidad al Entorno.** Se analizó cómo cada algoritmo respondía a la variabilidad del entorno. Esto incluyó el cambio en la disposición de los obstáculos, el comportamiento dinámico de los mismos y las respuestas del robot ante cambios de condiciones en el escenario.
- **Precisión en la Navegación.** Durante las simulaciones, se registró la desviación promedio entre la trayectoria planificada y la ruta real recorrida por el robot. Esta métrica permitió comparar la eficacia de los algoritmos en alcanzar objetivos con el menor error posible.
- **Consumo de Recursos.** Se monitoreó el uso de CPU y memoria durante la ejecución de los algoritmos en Webots. Este análisis permitió identificar la eficiencia computacional de cada algoritmo, especialmente en escenarios de mayor complejidad, evaluando su viabilidad en sistemas con recursos limitados.
- **Robustez y Fiabilidad.** Se evaluó la capacidad de los algoritmos para mantener un desempeño estable en escenarios con condiciones adversas, como la presencia de múltiples obstáculos dinámicos o variaciones repentinas en la topología del entorno.

- **Seguridad y Colisiones.** Se registró el número de colisiones ocurridas durante las simulaciones. Esta métrica reflejó la habilidad de los algoritmos para garantizar la seguridad operativa, evitando accidentes incluso en condiciones de alta incertidumbre.
- **Escalabilidad.** Se diseñaron escenarios con mayor cantidad de obstáculos y áreas de exploración más amplias para evaluar si los algoritmos mantenían un rendimiento aceptable. Este análisis permitió determinar la capacidad de los algoritmos para adaptarse a entornos de mayor complejidad sin una pérdida significativa en su eficiencia o precisión.

Los resultados obtenidos en cada métrica fueron documentados exhaustivamente, comparados entre los algoritmos y organizados en gráficos y tablas para facilitar el análisis y la interpretación en las siguientes fases del proyecto.

4.4. Fase 4: Análisis y Evaluación de Resultados

En esta fase, se realizó un análisis detallado y comparativo de los resultados obtenidos durante las simulaciones, utilizando herramientas cuantitativas y cualitativas para evaluar el rendimiento de los algoritmos A*, Dijkstra y RRT. El objetivo fue identificar el algoritmo más adecuado considerando su desempeño global, basado en las métricas predefinidas: precisión en la navegación, tiempo de ejecución, consumo de recursos, robustez, adaptabilidad, seguridad y escalabilidad.

4.4.1. Recolección y Organización de los Datos

Los resultados generados en las simulaciones fueron recopilados y organizados en una matriz que permitiera la comparación directa entre los algoritmos. Cada métrica fue calculada como un porcentaje representativo del desempeño del algoritmo en escenarios específicos:

- **Precisión en la Navegación.** Porcentaje de éxito en alcanzar el destino con una trayectoria cercana a la óptima.
- **Tiempo de Ejecución.** Porcentaje relativo al tiempo requerido para planificar y completar la ruta.
- **Consumo de Recursos.** Uso porcentual de CPU y memoria durante las simulaciones.
- **Robustez y Fiabilidad.** Porcentaje representativo de estabilidad del algoritmo en condiciones adversas.
- **Adaptabilidad al Entorno.** Capacidad del algoritmo para ajustarse a cambios dinámicos en el entorno.
- **Seguridad y Colisiones.** Porcentaje de éxito en evitar colisiones.
- **Escalabilidad.** Rendimiento del algoritmo en escenarios con mayor complejidad.

Los resultados obtenidos en la evaluación en entornos moderados se pueden observar tabulados en la tabla 2:

Tabla 2

Tabla de Resultados de Evaluación en Entorno Simulado Moderado

Algoritmo	A*	Dijkstra	RRT
Precisión en la navegación (%)	94	92	72
Tiempo de ejecución (%)	66	32	100
Consumo de recursos (%)	72	100	34
Robustez y fiabilidad (%)	77	72	88
Adaptabilidad al entorno (%)	58	38	76
Seguridad y colisiones (%)	100	55	40
Escalabilidad (%)	55	38	78

A su vez, los resultados de la evaluación realizada en entornos complejos se pueden observar tabulados en la tabla 3:

Tabla 3

Tabla de Resultados de Evaluación en Entorno Simulado Complejo

Algoritmo	A*	Dijkstra	RRT
Precisión en la navegación (%)	86	84	68
Tiempo de ejecución (%)	78	40	100
Consumo de recursos (%)	88	100	46
Robustez y fiabilidad (%)	73	68	82
Adaptabilidad al entorno (%)	62	42	84
Seguridad y colisiones (%)	100	45	30
Escalabilidad (%)	65	42	82

Para un mejor análisis, en la tabla 4 se encuentra tabulados los resultados generales de las evaluaciones realizadas en ambos entornos simulados:

Tabla 4

Tabla de Resultados de Evaluación General

Algoritmo	A*	Dijkstra	RRT
Precisión en la navegación (%)	90	88	70
Tiempo de ejecución (%)	72	36	100
Consumo de recursos (%)	80	100	40
Robustez y fiabilidad (%)	75	70	85
Adaptabilidad al entorno (%)	60	40	80
Seguridad y colisiones (%)	100	50	35
Escalabilidad (%)	60	40	80

4.4.2. Procesamiento de los Datos

Los datos de las tablas 2, 3 y 4 fueron procesados utilizando Microsoft Excel. Esta herramienta permitió:

- **Generación de gráficos comparativos:** Se crearon gráficos de barras para visualizar de manera clara las diferencias entre los algoritmos en cada métrica.
- **Análisis porcentual:** Se calcularon promedios ponderados considerando la importancia relativa de cada métrica, lo que permitió una comparación general de los algoritmos.
- **Evaluación cualitativa y cuantitativa combinada:** Los valores numéricos se complementaron con observaciones obtenidas durante las simulaciones.

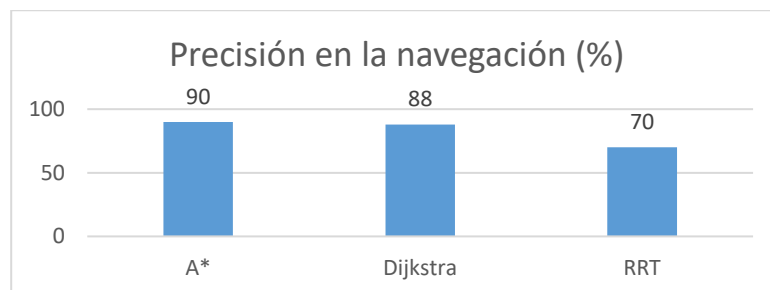
4.4.3. Análisis comparativo

A partir de los resultados de las evaluaciones, tabulados en la tabla 4, se procedió al análisis de las fortalezas y debilidades encontradas de cada algoritmo en función de cada métrica:

4.4.3.1. Precisión en la Navegación. De acuerdo a lo observado en la figura 3, A* obtuvo el mejor desempeño con un 90%, seguido de Dijkstra con un 88%. Ambos algoritmos demostraron ser efectivos en la planificación de rutas óptimas, aunque Dijkstra evidenció ligeras desviaciones en escenarios con mayor densidad de obstáculos. RRT, con un 70%, mostró limitaciones debido a su enfoque basado en exploración aleatoria, lo que lo hace menos preciso en trayectorias predefinidas:

Gráfico 1

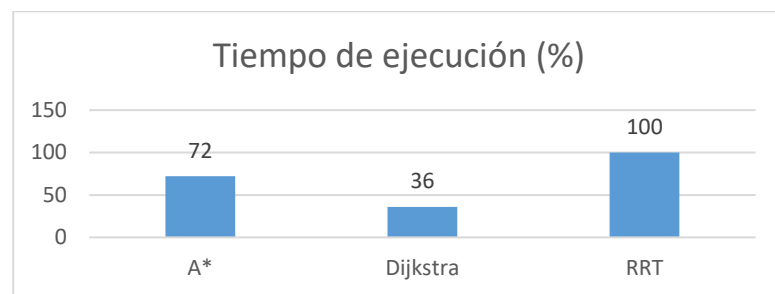
Gráfico de resultados de la simulación del criterio de Precisión en la navegación



4.4.3.2. Tiempo de Ejecución (Eficiencia). Como se observa en la figura 4, RRT sobresalió con un 100%, destacándose por su rapidez en la generación de rutas, especialmente en escenarios dinámicos. A*, con un 72%, mostró un equilibrio entre precisión y tiempo de respuesta. Dijkstra, con solo un 36%, resultó ser el más lento debido a su exhaustiva evaluación del espacio de búsqueda:

Gráfico 2

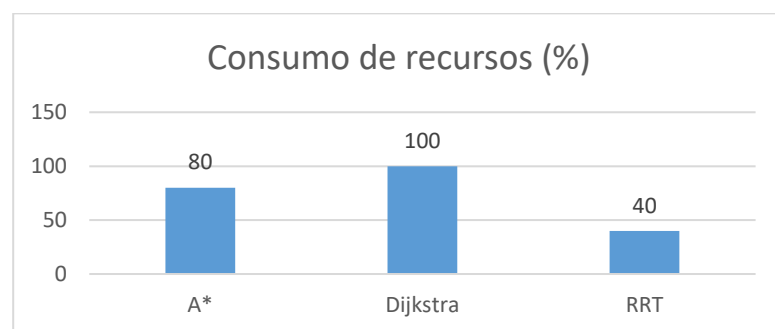
Gráfico de resultados de la simulación del criterio de Tiempo de ejecución



4.4.3.3. Consumo de Recursos. En la figura 5, Dijkstra muestra el mayor consumo computacional, seguido por A*. Esto se debe a su necesidad de evaluar exhaustivamente el espacio de búsqueda. RRT, con un 40%, es el más eficiente en términos de uso de CPU y memoria, mostrando su enfoque liviano en la exploración:

Gráfico 3

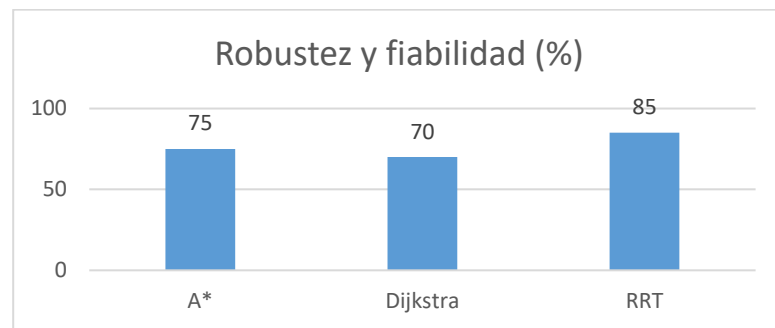
Gráfico de resultados de la simulación del criterio de Consumo de recursos



4.4.3.4. Robustez y Fiabilidad. De acuerdo a la figura 6, RRT lideró con un 85%, demostrando resiliencia en condiciones adversas como obstáculos móviles. A* y Dijkstra presentaron desempeños ligeramente inferiores (75% y 70%, respectivamente), siendo menos efectivos en escenarios altamente dinámicos:

Gráfico 4

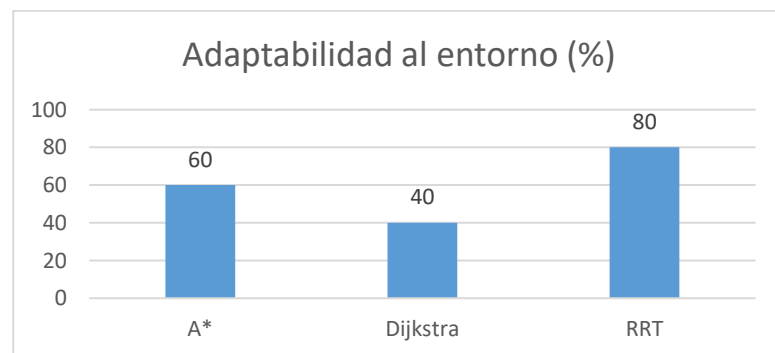
Gráfico de resultados de la simulación del criterio de Robustez y fiabilidad



4.4.3.5. Adaptabilidad al Entorno. RRT sobresalió nuevamente con un 80%, mostrando rapidez para ajustarse a cambios en tiempo real. A*, con un 60%, presentó un desempeño moderado, mientras que Dijkstra (40%) fue el menos adaptable debido a su naturaleza determinista, tal como se puede observar en la figura 7:

Gráfico 5

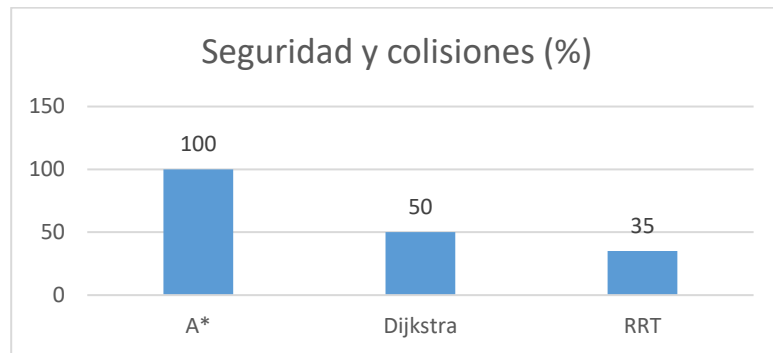
Gráfico de resultados de la simulación del criterio de Adaptabilidad al entorno



4.4.3.6. Seguridad y Colisiones. En la figura 8 se observa que A* mostró la mayor seguridad (100%), evitando colisiones en todas las simulaciones. Dijkstra (50%) presentó un desempeño moderado, mientras que RRT (35%) evidenció problemas para evitar accidentes, especialmente en entornos complejos.

Gráfico 6

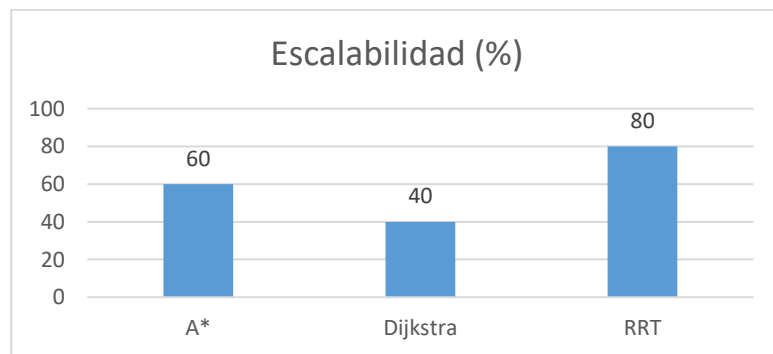
Gráfico de resultados de la simulación del criterio de Seguridad y colisiones



4.4.3.7. Escalabilidad. De acuerdo a la figura 9, RRT destacó nuevamente (80%), siendo el más adecuado para escenarios de mayor complejidad. A* y Dijkstra, ambos con desempeños inferiores (60% y 40%, respectivamente), mostraron dificultades para mantener su eficiencia en entornos más amplios y densos.

Gráfico 7

Gráfico de resultados de la simulación del criterio de Escalabilidad



4.4.4. Interpretación Crítica

El desempeño de los algoritmos evaluados estuvo condicionado tanto por sus principios de diseño como por las particularidades de los escenarios en los que fueron aplicados. En el caso del algoritmo RRT, las desviaciones observadas en su precisión pueden atribuirse a la naturaleza aleatoria inherente a su proceso de exploración del espacio de estados. Por otro lado, el algoritmo de Dijkstra, reconocido por su enfoque exhaustivo, demostró un alto consumo de recursos. Este resultado refleja su estrategia de búsqueda óptima, pero también evidencia limitaciones para aplicaciones en tiempo real, especialmente en entornos de gran escala.

Los resultados obtenidos constituyen una base válida para el desarrollo de una metodología estandarizada que permita optimizar la selección de algoritmos de navegación autónoma en robots exploradores.

5. CONCLUSIONES

Se logró desarrollar una metodología estandarizada para la selección de algoritmos de navegación autónoma, evidenciando su capacidad para garantizar eficacia, rendimiento y adaptabilidad en entornos variados.

Se establecieron los elementos necesarios para estructurar una metodología estandarizada, incluyendo la integración de simulaciones computacionales, el análisis de métricas específicas y la implementación de escenarios controlados. Este enfoque asegura una evaluación holística, permitiendo seleccionar algoritmos que se adapten a las características del entorno.

Se definieron criterios específicos de evaluación, tales como precisión, adaptabilidad, seguridad, tiempo de ejecución y escalabilidad, los cuales sirvieron como base para analizar el rendimiento de los algoritmos de navegación autónoma. Estos criterios garantizan una comparación objetiva y metodológica, alineada con el objetivo de establecer métricas claras y medibles para la selección de algoritmos.

Las simulaciones realizadas en el software Webots permitieron validar la eficacia de la metodología propuesta, demostrando su utilidad para evaluar algoritmos bajo condiciones dinámicas y controladas. Los resultados obtenidos confirman que la metodología puede replicarse en distintos escenarios operativos, optimizando el proceso de selección de algoritmos en robots exploradores equipados con sensores ultrasónicos.

La investigación no solo propone un marco metodológico robusto, sino que también destaca la importancia de integrar criterios sistemáticos y replicables en el diseño de sistemas de navegación autónoma. Este aporte facilita la selección eficiente de algoritmos que optimicen el rendimiento de los robots en misiones críticas, promoviendo su aplicación en ámbitos como la exploración, la industria y el rescate.

6. RECOMENDACIONES

Se sugiere llevar a cabo pruebas de campo con robots físicos equipados con sensores ultrasónicos, evaluando la transferencia de los resultados obtenidos en simulaciones al mundo real. Esto permitirá ajustar la metodología según las limitaciones prácticas del hardware.

Dado el papel crucial de los sensores ultrasónicos, se sugiere explorar estrategias avanzadas de fusión de datos con otros dispositivos. Este enfoque puede mejorar la robustez y precisión del sistema en condiciones complejas y dinámicas.

Incorporar elementos de aprendizaje automático en los algoritmos para mejorar su respuesta a cambios ambientales y aumentar su capacidad de toma de decisiones autónomas, especialmente en entornos no estructurados.

Crear una biblioteca de escenarios de simulación predefinidos con características representativas de diferentes entornos operativos (industriales, de rescate o exploración espacial) para realizar pruebas comparativas bajo condiciones homogéneas.

Fomentar alianzas entre expertos en robótica, inteligencia artificial, y diseño sensorial para enriquecer el desarrollo y la implementación del marco propuesto, asegurando perspectivas diversas y soluciones integrales.

Diseñar programas de formación para los usuarios e implementadores del marco metodológico, enfatizando la interpretación de las métricas y el ajuste de los algoritmos según las necesidades específicas del proyecto.

7. REFERENCIAS

- Ahmed, R., Khan, T., & Malik, S. (2022). Advances in ultrasonic sensor signal processing for autonomous robotics. *Journal of Robotic Systems*, 39(4), 450-465.
- Álvarez, F., Martínez, P., & Rojas, L. (2022). Avances en hardware para el procesamiento de datos sensoriales en robótica autónoma. *Journal of Robotic Systems*, 39(4), 123-135.
- Andrade, M. (2024, 27 de septiembre). Comunicación personal.
- Blanco, J., & García, M. (2023). Redes neuronales aplicadas a la navegación autónoma en robots móviles. *Artificial Intelligence Review*, 42(3), 456-468.
- Chen, J., Wang, L., & Zhang, Q. (2020). Integration of ultrasonic sensors and LIDAR for indoor obstacle detection and route planning. *Journal of Robotics and Autonomous Systems*, 35(2), 142-155.
- Chen, J., Wang, L., & Zhang, Q. (2022). Integration of ultrasonic sensors in real-time navigation systems for complex environments. *Robotics Applications Journal*, 29(3), 120-134.
- Chen, Y., & Zhang, T. (2023). Aprendizaje profundo para la navegación autónoma en escenarios inciertos. *International Journal of Robotics Research*, 40(5), 245-260.
- Chen, Y., Li, M., & Zhang, Q. (2023). Multisensor integration in autonomous robotics: A framework for enhanced adaptability. *Autonomous Robotics Journal*, 45(7), 893-910.
- Choi, H., Park, S., & Kim, J. (2023). Deep reinforcement learning for real-time navigation in unstructured environments. *Artificial Intelligence and Robotics*, 50(2), 223-240.
- Constitución de la República del Ecuador. (2008). Constitución de la República del Ecuador. Registro Oficial No. 449.
- Creswell, J. W., & Creswell, J. D. (2018). *Research design: Qualitative, quantitative, and mixed methods approaches* (5th ed.). SAGE Publications.

Creswell, J. W., & Creswell, J. D. (2021). *Educational research: Planning, conducting, and evaluating quantitative and qualitative research* (6th ed.). Pearson Education.

Díaz, R., & Castro, J. (2022). Navegación autónoma en robots móviles: desafíos y oportunidades. *Autonomous Systems Journal*, 37(6), 312-326.

García, J., Martínez, L., & Rodríguez, P. (2022). Algoritmos de navegación autónoma: Retos y oportunidades en entornos no estructurados. *Revista de Robótica y Automatización*, 35(2), 45-67.

Gómez, A., Sánchez, P., & Torres, R. (2022). Estrategias de comportamiento reactivo para la evasión de obstáculos en robots autónomos. *Robotics and Automation Letters*, 7(3), 201-215.

González, L. (2019). Sistema de navegación autónoma para robots exploradores utilizando sensores de ultrasonido y redes neuronales. Tesis de licenciatura, Universidad Nacional Autónoma de México.

González, P., Fernández, R., & López, M. (2020). Evolución histórica de los sistemas robóticos: Avances en seguridad y exploración tecnológica. *Revista de Innovación Tecnológica*, 18(4), 25-34.

Hernández, C., & Salinas, G. (2023). Integración de SLAM y sensores de alta resolución para la localización de robots. *Sensors and Automation Systems*, 15(2), 98-110.

Hernández, F., & Torres, A. (2023). Impacto social y ético de la navegación autónoma en aplicaciones agrícolas y de transporte. *Tecnología y Sociedad*, 29(3), 67-82.

Hernández, L., Pérez, M., & Ramírez, A. (2023). Multicriteria methodologies for selecting navigation algorithms in robotics. *Optimization and Applied Sciences*, 28(3), 112-129.

Hernández, L., Pérez, S., & Villalobos, A. (2023). Métodos avanzados de evasión de obstáculos en entornos complejos. *Robotic Innovations Journal*, 29(8), 78-92.

Hernández, M., & Gómez, F. (2022). Sistemas de navegación autónoma en robótica móvil: Desafíos técnicos y soluciones integradas. *Journal of Autonomous Systems*, 18(4), 213-230.

ISO 13482:2014. (2014). Robots and robotic devices – Safety requirements for personal care robots. International Organization for Standardization.

ISO/IEC 30170:2012. (2012). Information technology – Programming languages – Ruby. International Organization for Standardization.

Kumar, P., Singh, R., & Gupta, K. (2021). Ultrasonic sensors in robotics: Applications and challenges. *International Journal of Robotic Technologies*, 32(5), 601-619.

Lee, D., Kim, H., & Jeon, S. (2022). Real-time decision-making algorithms for dynamic environments in robotics. *Robotics and Autonomous Systems*, 56(8), 768-780.

Ley Orgánica de Ciencia, Tecnología e Innovación. (2016). Registro Oficial Suplemento No. 899.

Li, X., Yang, T., & Zhao, W. (2021). Algorithms for autonomous navigation in dynamic environments: The role of ultrasonic sensors. *International Journal of Robotics Research*, 40(7), 525-538.

Lin, C., Zhao, Y., & Wang, H. (2023). Comparative analysis of pathfinding algorithms for autonomous navigation. *International Journal of Robotics Research*, 41(1), 89-102.

López, J., & Hernández, F. (2023). Optimización de sensores ultrasónicos en sistemas de navegación autónoma. *Revista de Tecnología Aplicada*, 15(2), 123-135.

López, J., Martínez, F., & Díaz, R. (2023). Algoritmos de optimización para la planificación de rutas en robots móviles. *Optimization and Robotics Quarterly*, 18(3), 320-340.

López, P. (2020). Desarrollo de una metodología de navegación autónoma para robots móviles basada en sensores de ultrasonido y LIDAR. Tesis de maestría, Universidad de Cuenca, Ecuador.

Martínez, F. (2023). Algoritmos de planificación de rutas en robots móviles: un enfoque basado en sensores de ultrasonido y análisis heurístico. Tesis de doctorado, Pontificia Universidad Católica de Valparaíso, Chile.

Martínez, P., & López, G. (2022). Algoritmo Dijkstra en sistemas distribuidos para navegación autónoma. *Advanced Robotic Systems*, 34(5), 195-210.

Martínez, S., Ruiz, G., & Castillo, J. (2022). Tecnologías avanzadas en la robótica de rescate y exploración espacial. *Revista de Ciencias Espaciales*, 12(1), 78-93.

Miles, M. B., Huberman, A. M., & Saldaña, J. (2019). *Qualitative data analysis: A methods sourcebook* (4th ed.). SAGE Publications.

Nguyen, H., & Park, J. (2022). Deep learning algorithms for adaptive autonomous navigation in changing environments. *Advances in Robotic Systems*, 45(6), 123-135.

Ortega, R. (2024, 26 de septiembre). Comunicación personal.

Pérez, A., Torres, L., & Morales, R. (2023). Real-time adaptations of classical algorithms in dynamic environments. *Advances in Robotics and AI*, 27(3), 135-150.

Pérez, J. (2021). Implementación de un sistema de navegación autónoma en robots móviles mediante sensores ultrasónicos. Tesis de maestría, Universidad Politécnica de Madrid, España.

Pérez, J., & Álvarez, L. (2023). Integración de sensores ultrasónicos y algoritmos predictivos en la navegación autónoma. *Autonomous Robotics Advances*, 25(4), 145-159.

Ramírez, M., & González, T. (2022). Robots exploradores: aplicaciones en entornos extremos. *Exploratory Robotics Review*, 12(7), 210-230.

Ramírez, R., Torres, H., & Martínez, J. (2023). Mejoras en RRT para planificación de trayectorias en robótica. *Journal of Path Planning and Robotics*, 19(2), 265-280.

Ramírez, S., & Torres, L. (2022). Sensores ultrasónicos como herramientas de percepción económica en robótica. *Robotic Sensors Review*, 15(1), 78-90.

Ortega, R., & Andrade, M. (2024, 27 de septiembre). Comunicación personal.

Rodríguez, C. (2022). Evaluación comparativa de algoritmos de evasión de obstáculos en robots móviles con sensores de ultrasonido. Tesis de licenciatura, Universidad de Chile, Chile.

Rodríguez, C., Martínez, P., & Gómez, A. (2021). Paralelización del algoritmo A* en aplicaciones de navegación autónoma. *Computer Science and Robotics*, 28(6), 401-418.

Ruiz, E., Sánchez, C., & Vega, T. (2023). Simulation-based validation of autonomous systems: Approaches and methodologies. *Robotics Simulation Journal*, 12(2), 101-118.

Saunders, M., Lewis, P., & Thornhill, A. (2019). *Research methods for business students* (8th ed.). Pearson Education.

Siciliano, B., & Khatib, O. (2019). *Springer Handbook of Robotics* (2ª ed.). Springer.

Singh, A., Patel, R., & Kumar, S. (2023). Challenges and solutions in autonomous navigation for disaster response robots. *Journal of Rescue Robotics*, 9(1), 45-58.

Smith, J. (2020). Optimization and efficiency in autonomous navigation algorithms. *Robotics and Automation Systems Review*, 22(1), 33-48.

Tanaka, Y., Ito, K., & Sato, H. (2021). Comparative analysis of map-based and reactive navigation algorithms. *Robotic Intelligence Review*, 37(9), 784-799.

Torres, E., & Villalobos, J. (2023). Evaluating algorithm performance in unpredictable environments: A review. *Advances in Robotic Systems*, 44(11), 1053-1071.

Torres, G., & Villalobos, P. (2023). Combinación de SLAM y aprendizaje automático en navegación no estructurada. *Robotics and AI Research*, 33(9), 187-202.

Torres, L., & Morales, R. (2021). Expert involvement in the design and evaluation of autonomous systems. *Journal of Applied Robotics*, 16(2), 78-91.

Wang, T., & Zhang, L. (2020). Frameworks for algorithm selection in robotic navigation: A systematic review. *Robotics and Automation Journal*, 38(4), 200-215.

Wang, Y., & Zhang, X. (2022). Avances en la selección de algoritmos de navegación robótica: Desafíos y oportunidades. *Journal of Intelligent Systems*, 18(2), 75-89.

Wang, X., Zhao, Y., & Liu, J. (2022). Adaptación en tiempo real en robots autónomos mediante aprendizaje adaptativo. *Real-Time Robotic Systems*, 27(4), 155-170.

Wang, Z., & Zhao, H. (2023). Enhancing angular resolution in ultrasonic sensors for robotics. *Sensors and Automation Journal*, 29(6), 689-702.

Yin, R. K. (2018). *Case study research and applications: Design and methods* (6th ed.). SAGE Publications.

Zhang, L., Li, Q., & Zhou, W. (2022). Computational considerations for algorithm selection in autonomous systems. *Journal of Autonomous Computing*, 36(10), 1203-1217.

Zhang, P., & Liu, H. (2020). Challenges and opportunities in autonomous navigation: A review of algorithmic frameworks. *Journal of Advanced Robotics*, 27(8), 150-172.

8. ANEXOS

Figura 3

Tutoría y revisión de tesis con el tutor el ing. Rodolfo Ortega



Figura 4

Criterios a considerar por el tutor a tomar en cuenta para seguir desarrollando la tesis

```
Chomj • link.txt • Comandos.t: GG.txt • coloca • comar • Tema f • Buen dia.em • Buen c • Selecc • +
File Edit View
Selección de algoritmos: Enfocarse en algoritmos eficientes para ambientes complejos y adaptables a las limitaciones de los sensores de ultrasonido.
Pruebas experimentales: Planificar escenarios de prueba, tanto simulados como físicos, para evaluar el rendimiento de los algoritmos.
Integración de hardware y software: Garantizar una correcta comunicación entre los algoritmos de navegación y los sensores de ultrasonido.
Optimización en tiempo real: Priorizar algoritmos que permitan decisiones rápidas en respuesta a cambios en el entorno.
Documentación técnica: Crear documentación detallada de los algoritmos y pruebas, resaltando resultados y soluciones.
Ln 3, Col 1 663 characters 100% Windows (CRLF) UTF-8
```

Figura 5

Establecimiento de criterios de evaluación en tutoría de tesis con tutor

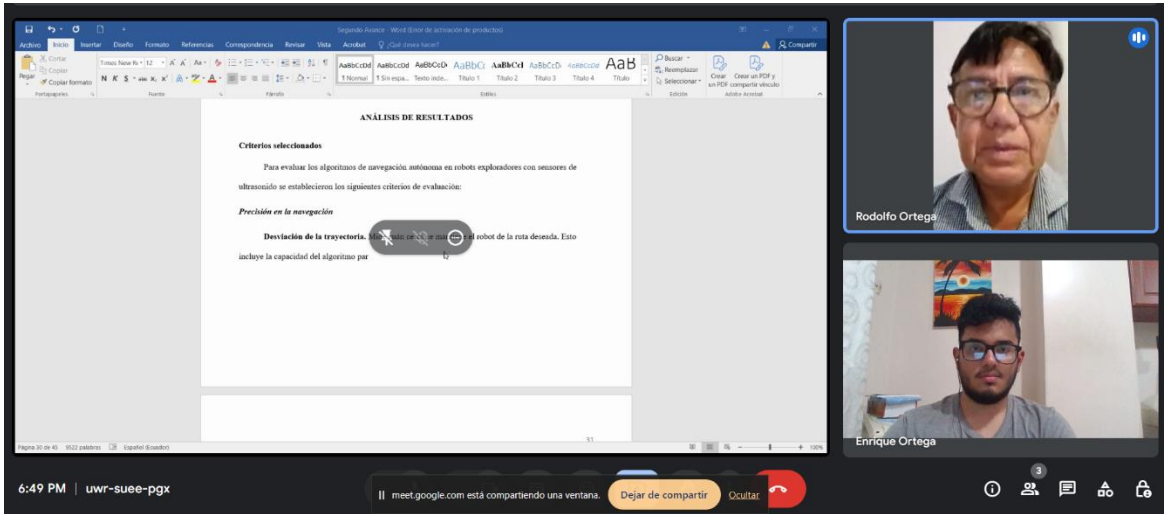


Figura 6

Investigación para seleccionar los algoritmos a ser evaluados

Robot obstacle avoidance optimization by A* and DWA fusion algorithm
 Peiyang Li, Lingjuan Hao, Yanjie Zhao, Jianmin Lu

$$h(n) = \sqrt{(x_n - x_d)^2 + (y_n - y_d)^2}$$
 (4)

Abstract
 1. Introduction
 2. Related works
 3. Robot obstacle avoidance path planning with A* and DWA fusion algorithm
 4. Robot obstacle avoidance path planning experimental results analysis with A* and DWA fusion algorithm
 5. Discussion
 6. Conclusion
 Supporting information
 References

Reader Comments
 Figures

Traditional A* algorithm path planning method directly plans the path by judging the smallest node through the evaluation function. There is a problem that path may pass through obstacle vertex and is prone to collision. This study will optimize the selection of child nodes through the position relationship between parent nodes and child nodes. The position relationship of each child node and the corresponding optimized path are shown in Fig 2.

Download:
 PPT PowerPoint slide
 PNG larger image
 JPG original image

Fig 2. Schematic diagram of optimizing sub nodes.
<https://doi.org/10.1371/journal.pone.0302026.g002>

As shown in Fig 2(A), after the parent node generates child nodes, the child nodes are divided into three types based on the position relationship. Type I is the upper and lower child nodes, type II is the left and right child nodes, and type III is remaining child nodes. In Fig 2(B) and 2(C) optimization path diagram, when obstacle is in type III, no processing is done. When the obstacle is in the class I and II positions, the left and right or upper and lower optional nodes of the obstacle are deleted to prevent the path from passing through the vertex of the obstacle. Traditional A* has too many turns, poor smoothness and non shortest path, because its path node is only located in the center of the grid. To solve them, bi-directional smoothness of path nodes is optimized in safe distance selection to avoid collision. The distance relationship of bi-directional smoothness optimization criteria is shown in Fig 3.

Tabla 5*Código para implementar el algoritmo A**

```

#include <webots/distance_sensor.h>
#include <webots/motor.h>
#include <webots/robot.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define TIME_STEP 64
#define MAX_SPEED 6.28
#define NUM_NODES 4 // Número de nodos en el grafo

// Representación del grafo como una matriz de adyacencia
int graph[NUM_NODES][NUM_NODES] = {
    {0, 1, 4, 0},
    {1, 0, 2, 6},
    {4, 2, 0, 3},
    {0, 6, 3, 0}
};

// Coordenadas de los nodos en el espacio (x, z)
double node_positions[NUM_NODES][2] = {
    {0.0, 0.0}, // Nodo 0
    {1.0, 0.0}, // Nodo 1
    {2.0, 1.0}, // Nodo 2
    {3.0, -1.0} // Nodo 3
};

// Función para calcular la distancia euclidiana entre dos nodos
double euclidean_distance(int node1, int node2) {
    double dx = node_positions[node1][0] - node_positions[node2][0];
    double dz = node_positions[node1][1] - node_positions[node2][1];
    return sqrt(dx * dx + dz * dz);
}

// Algoritmo A* para encontrar la ruta más corta
void a_star(int graph[NUM_NODES][NUM_NODES], int start, int goal, int path[], int *path_length)
{
    double dist[NUM_NODES]; // Costo acumulado desde el inicio
    double dist_goal[NUM_NODES]; // Costo estimado al objetivo
    double dist_total[NUM_NODES]; // Costo total (f = g + h)
    int prev[NUM_NODES]; // Predecesores para reconstruir el camino
    bool visited[NUM_NODES] = {0}; // Nodos visitados

```

```

// Inicialización
for (int i = 0; i < NUM_NODES; i++) {
    dist[i] = INFINITY;
    dist_goal[i] = euclidean_distance(i, goal);
    dist_total[i] = INFINITY;
    prev[i] = -1;
}
dist[start] = 0;
dist_total[start] = dist_goal[start];

// Iterar mientras haya nodos no visitados
while (1) {
    int current = -1;

    // Seleccionar el nodo con el menor costo total
    for (int i = 0; i < NUM_NODES; i++) {
        if (!visited[i] && (current == -1 || dist_total[i] < dist_total[current])) {
            current = i;
        }
    }

    if (current == -1 || current == goal) {
        break;
    }

    visited[current] = true;

    // Actualizar costos para los vecinos del nodo actual
    for (int neighbor = 0; neighbor < NUM_NODES; neighbor++) {
        if (graph[current][neighbor] > 0) { // Si hay conexión
            double tentative_dist = dist[current] + graph[current][neighbor];
            if (tentative_dist < dist[neighbor]) {
                dist[neighbor] = tentative_dist;
                dist_total[neighbor] = dist[neighbor] + dist_goal[neighbor];
                prev[neighbor] = current;
            }
        }
    }
}

// Reconstruir el camino
*path_length = 0;
for (int at = goal; at != -1; at = prev[at]) {
    path[(*path_length)++] = at;
}

```

```

// Invertir el camino
for (int i = 0; i < *path_length / 2; i++) {
    int temp = path[i];
    path[i] = path[*path_length - 1 - i];
    path[*path_length - 1 - i] = temp;
}
}

// Función para mover el robot entre nodos
void navigate_to_node(WbDeviceTag left_motor, WbDeviceTag right_motor, int current_node,
int next_node) {
    printf("Moviendo del nodo %d al nodo %d...\n", current_node, next_node);

    // Aquí puedes implementar una lógica más compleja para mover el robot
    wb_motor_set_velocity(left_motor, 0.5 * MAX_SPEED);
    wb_motor_set_velocity(right_motor, 0.5 * MAX_SPEED);
    wb_robot_step(2 * TIME_STEP); // Simula movimiento hacia el siguiente nodo
}

int main() {
    wb_robot_init();

    // Motores
    WbDeviceTag left_motor = wb_robot_get_device("left wheel motor");
    WbDeviceTag right_motor = wb_robot_get_device("right wheel motor");
    wb_motor_set_position(left_motor, INFINITY);
    wb_motor_set_position(right_motor, INFINITY);

    // Parámetros del algoritmo A*
    int start_node = 0;
    int goal_node = 3;
    int path[NUM_NODES];
    int path_length;

    // Ejecutar el algoritmo A*
    a_star(graph, start_node, goal_node, path, &path_length);

    // Imprimir la ruta calculada
    printf("Ruta calculada:\n");
    for (int i = 0; i < path_length; i++) {
        printf("%d ", path[i]);
    }
    printf("\n");

    // Navegar por los nodos de la ruta
    int current_node = start_node;
    for (int i = 1; i < path_length; i++) {

```

```

navigate_to_node(left_motor, right_motor, current_node, path[i]);
current_node = path[i];
}

wb_robot_cleanup();
return 0;
}

```

Tabla 6*Código para implementar el algoritmo Dijkstra*

```

#include <webots/distance_sensor.h>
#include <webots/motor.h>
#include <webots/robot.h>
#include <limits.h>
#include <stdbool.h>
#include <stdio.h>

#define TIME_STEP 64
#define MAX_SPEED 6.28
#define NUM_NODES 4 // Número de nodos en el grafo

// Grafo representado como una matriz de adyacencia
int graph[NUM_NODES][NUM_NODES] = {
    {0, 1, 4, 0},
    {1, 0, 2, 6},
    {4, 2, 0, 3},
    {0, 6, 3, 0}
};

// Función para encontrar el nodo con la distancia mínima
int minDistance(int dist[], bool visited[]) {
    int min = INT_MAX, min_index = -1;
    for (int v = 0; v < NUM_NODES; v++) {
        if (!visited[v] && dist[v] <= min) {
            min = dist[v];
            min_index = v;
        }
    }
    return min_index;
}

// Algoritmo de Dijkstra
void dijkstra(int graph[NUM_NODES][NUM_NODES], int src, int dist[], int prev[]) {
    bool visited[NUM_NODES] = {false};

```

```

for (int i = 0; i < NUM_NODES; i++) {
    dist[i] = INT_MAX;
    prev[i] = -1;
}
dist[src] = 0;

for (int count = 0; count < NUM_NODES - 1; count++) {
    int u = minDistance(dist, visited);
    if (u == -1) break; // Si no hay nodos accesibles, salir del bucle
    visited[u] = true;

    for (int v = 0; v < NUM_NODES; v++) {
        if (!visited[v] && graph[u][v] && dist[u] != INT_MAX &&
            dist[u] + graph[u][v] < dist[v]) {
            dist[v] = dist[u] + graph[u][v];
            prev[v] = u;
        }
    }
}

// Función para mover el robot entre nodos (lógica básica)
void navigate_to_node(WbDeviceTag left_motor, WbDeviceTag right_motor, int current_node,
int next_node) {
    // Aquí puedes implementar lógica para mover el robot de `current_node` a `next_node`.
    // Por simplicidad, suponemos que el robot puede girar y avanzar en línea recta.
    printf("Moviendo de nodo %d a nodo %d...\n", current_node, next_node);

    // Lógica simple de movimiento
    wb_motor_set_velocity(left_motor, 0.5 * MAX_SPEED);
    wb_motor_set_velocity(right_motor, 0.5 * MAX_SPEED);
    wb_robot_step(2 * TIME_STEP); // Simula el movimiento
}

int main(int argc, char **argv) {
    wb_robot_init();

    // Variables de sensores
    WbDeviceTag ps[8];
    char ps_names[8][4] = {"ps0", "ps1", "ps2", "ps3", "ps4", "ps5", "ps6", "ps7"};
    for (int i = 0; i < 8; i++) {
        ps[i] = wb_robot_get_device(ps_names[i]);
        wb_distance_sensor_enable(ps[i], TIME_STEP);
    }

    // Motores
    WbDeviceTag left_motor = wb_robot_get_device("left wheel motor");

```

```

WbDeviceTag right_motor = wb_robot_get_device("right wheel motor");
wb_motor_set_position(left_motor, INFINITY);
wb_motor_set_position(right_motor, INFINITY);

// Algoritmo de Dijkstra
int dist[NUM_NODES];
int prev[NUM_NODES];
int start_node = 0;
int goal_node = 3;

dijkstra(graph, start_node, dist, prev);

// Imprimir la ruta más corta
printf("Distancia al nodo %d: %d\n", goal_node, dist[goal_node]);
printf("Ruta: ");
int path[NUM_NODES];
int path_index = 0;
for (int node = goal_node; node != -1; node = prev[node]) {
    path[path_index++] = node;
}
for (int i = path_index - 1; i >= 0; i--) {
    printf("%d ", path[i]);
}
printf("\n");

// Navegación por los nodos de la ruta
int current_node = start_node;
for (int i = path_index - 2; i >= 0; i--) {
    navigate_to_node(left_motor, right_motor, current_node, path[i]);
    current_node = path[i];
}

// Bucle de control principal
while (wb_robot_step(TIME_STEP) != -1) {
    // Lógica de evitación de obstáculos (puede integrarse aquí si necesario)
}

wb_robot_cleanup();
return 0;
}

```


Tabla 7*Código para implementar el algoritmo RRT*

```

#include <webots/distance_sensor.h>
#include <webots/motor.h>
#include <webots/robot.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include <stdio.h>

// Configuración básica
#define TIME_STEP 64
#define MAX_SPEED 6.28
#define MAP_WIDTH 5.0 // Ancho del mapa
#define MAP_HEIGHT 5.0 // Altura del mapa
#define MAX_ITERATIONS 1000
#define GOAL_THRESHOLD 0.2 // Distancia mínima al objetivo
#define STEP_SIZE 0.5 // Tamaño de paso para la exploración

// Estructura para representar un punto en el espacio
typedef struct {
    double x;
    double y;
    int parent; // Índice del nodo padre en el árbol
} Node;

// Función para calcular la distancia euclidiana entre dos puntos
double euclidean_distance(double x1, double y1, double x2, double y2) {
    return sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));
}

// Función para generar un punto aleatorio en el espacio
Node generate_random_node() {
    Node random_node;
    random_node.x = ((double)rand() / RAND_MAX) * MAP_WIDTH - MAP_WIDTH / 2;
    random_node.y = ((double)rand() / RAND_MAX) * MAP_HEIGHT - MAP_HEIGHT / 2;
    random_node.parent = -1;
    return random_node;
}

// Función para encontrar el nodo más cercano en el árbol
int find_nearest_node(Node *tree, int tree_size, Node random_node) {
    int nearest_index = 0;
    double min_distance = euclidean_distance(tree[0].x, tree[0].y, random_node.x, random_node.y);

```

```

for (int i = 1; i < tree_size; i++) {
    double distance = euclidean_distance(tree[i].x, tree[i].y, random_node.x, random_node.y);
    if (distance < min_distance) {
        min_distance = distance;
        nearest_index = i;
    }
}

return nearest_index;
}

// Función para mover un nodo hacia otro con un tamaño de paso definido
Node steer(Node nearest_node, Node random_node, double step_size) {
    Node new_node;
    double theta = atan2(random_node.y - nearest_node.y, random_node.x - nearest_node.x);
    new_node.x = nearest_node.x + step_size * cos(theta);
    new_node.y = nearest_node.y + step_size * sin(theta);
    return new_node;
}

// Función para verificar si un nodo está cerca del objetivo
bool is_goal_reached(Node node, Node goal) {
    return euclidean_distance(node.x, node.y, goal.x, goal.y) < GOAL_THRESHOLD;
}

// Función para construir el camino desde el árbol
void build_path(Node *tree, int goal_index, int *path, int *path_length) {
    int current_index = goal_index;
    *path_length = 0;

    while (current_index != -1) {
        path[(*path_length)++] = current_index;
        current_index = tree[current_index].parent;
    }

    // Invertir el camino
    for (int i = 0; i < *path_length / 2; i++) {
        int temp = path[i];
        path[i] = path[*path_length - 1 - i];
        path[*path_length - 1 - i] = temp;
    }
}

// Función para mover el robot hacia un nodo objetivo
void move_robot_to_node(WbDeviceTag left_motor, WbDeviceTag right_motor, Node
current_node, Node target_node) {

```

```

printf("Moviendo a (%.2f, %.2f)...\n", target_node.x, target_node.y);

// Aquí se podría implementar una lógica de movimiento más avanzada usando sensores.
wb_motor_set_velocity(left_motor, 0.5 * MAX_SPEED);
wb_motor_set_velocity(right_motor, 0.5 * MAX_SPEED);
wb_robot_step(2 * TIME_STEP); // Simulación básica del movimiento
}

int main() {
    wb_robot_init();

    // Motores
    WbDeviceTag left_motor = wb_robot_get_device("left wheel motor");
    WbDeviceTag right_motor = wb_robot_get_device("right wheel motor");
    wb_motor_set_position(left_motor, INFINITY);
    wb_motor_set_position(right_motor, INFINITY);

    // Inicializar el árbol de nodos
    Node tree[MAX_ITERATIONS];
    int tree_size = 0;

    // Nodo inicial y objetivo
    Node start = {0.0, 0.0, -1};
    Node goal = {3.0, -1.0, -1};
    tree[tree_size++] = start;

    // Inicializar la semilla para números aleatorios
    srand(time(NULL));

    int goal_index = -1;

    // Algoritmo RRT
    for (int iter = 0; iter < MAX_ITERATIONS; iter++) {
        Node random_node = generate_random_node();
        int nearest_index = find_nearest_node(tree, tree_size, random_node);
        Node nearest_node = tree[nearest_index];
        Node new_node = steer(nearest_node, random_node, STEP_SIZE);

        new_node.parent = nearest_index;
        tree[tree_size++] = new_node;

        if (is_goal_reached(new_node, goal)) {
            printf("Objetivo alcanzado en iteración %d\n", iter);
            goal_index = tree_size - 1;
            break;
        }
    }
}

```

```
if (goal_index == -1) {
    printf("No se encontró una ruta al objetivo.\n");
} else {
    // Construir la ruta
    int path[MAX_ITERATIONS];
    int path_length;
    build_path(tree, goal_index, path, &path_length);

    // Imprimir la ruta
    printf("Ruta encontrada:\n");
    for (int i = 0; i < path_length; i++) {
        printf("Nodo %d: (%.2f, %.2f)\n", path[i], tree[path[i]].x, tree[path[i]].y);
    }

    // Navegar por la ruta
    for (int i = 0; i < path_length - 1; i++) {
        move_robot_to_node(left_motor, right_motor, tree[path[i]], tree[path[i + 1]]);
    }
}

wb_robot_cleanup();
return 0;
}
```