

Evidencia 1. Reporte escrito del proyecto y reflexión individual

1st Brisa Itzel Reyes Castro
Departamento de Ingeniería
Tec de Monterrey Querétaro
Querétaro, Mexico
A01710395@tec.mx

2nd Diego Quezada Colorado
Departamento de Ingeniería
Tec de Monterrey Querétaro
Querétaro, México
A01710024@tec.mx

3rd Emanuel Josué Vega González
Departamento de Ingeniería
Tec de Monterrey Querétaro
Querétaro, México
A01710366@tec.mx

4nd Aldrick Victor Tadeo Arellano
Departamento de Ingeniería
Tec de Monterrey Querétaro
Querétaro, México
A01710105@tec.mx

Abstract—La navegación en robótica móvil es la tecnología que capacita a los robots para desplazarse de forma autónoma de un punto a otro, tanto en entornos de interior como de exterior. Este proceso implica conocer su posición actual, construir un mapa del entorno, planificar la ruta a seguir y controlar sus movimientos para llegar al destino deseado, mientras ejecutan la tarea para la que han sido diseñados. Este proyecto presenta el desarrollo de un sistema de navegación autónoma para un robot Puzzlebot utilizando ROS2, integrando localización basada en marcadores ArUco y algoritmos de evasión de obstáculos Bug0/Bug2 dentro de un entorno de simulación en Gazebo.

I. OBJETIVOS

- Implementar un sistema de localización basado en marcadores ArUco para mejorar la estimación de pose del robot.
- Integrar un filtro EKF para fusionar información de navegación y reducir errores acumulativos de localización.
- Desarrollar algoritmos de evasión de obstáculos Bug0 y Bug2 que permitan al robot desplazarse hacia objetivos específicos evitando colisiones.
- Diseñar una arquitectura modular en ROS2 basada en nodos independientes para facilitar la integración del sistema.
- Evaluar el desempeño del sistema mediante simulaciones realizadas en Gazebo y visualización en RViz.
- Analizar el comportamiento del robot frente a distintos escenarios con obstáculos.

II. INTRODUCCIÓN

En el presente proyecto se desarrolló un sistema de navegación autónoma para el robot Puzzlebot, integrando sensores LiDAR, detección visual mediante marcadores ArUco y un filtro de Kalman extendido (EKF) con el propósito de permitir que el robot alcanzara una meta de manera autónoma dentro de un entorno simulado. Para ello, se utilizó Gazebo como plataforma de simulación, permitiendo recrear escenarios con

obstáculos y validar el comportamiento del robot en condiciones controladas.

El sistema de navegación se basó en algoritmos reactivos, particularmente Bug2, el cual permitió al robot desplazarse hacia una coordenada objetivo siguiendo una línea de navegación previamente definida. Cuando el LiDAR detectaba obstáculos en el trayecto, el robot cambiaba temporalmente a un modo de seguimiento de pared para rodearlos y posteriormente retomar la trayectoria hacia la meta. Asimismo, la implementación de marcadores ArUco permitió mejorar la precisión de localización, mientras que el filtro EKF contribuyó a reducir errores acumulativos de pose mediante la fusión de información proveniente de distintos sensores. Como resultado, fue posible desarrollar un sistema de navegación autónoma funcional y validarlo mediante diferentes escenarios dentro de Gazebo.

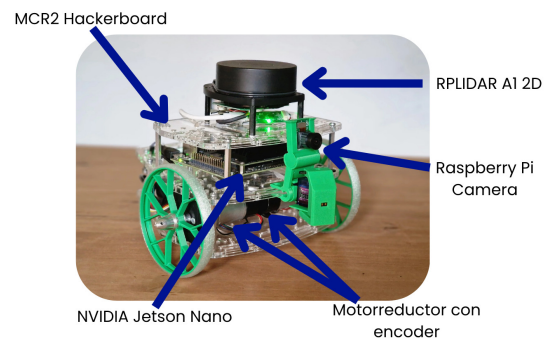


Fig. 1. Puzzlebot



Fig. 2. Maze

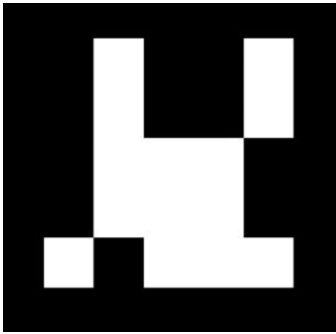


Fig. 3. Aruco

III. ESTADO DEL ARTE

La navegación autónoma en robots móviles ha sido un área ampliamente estudiada debido a su relevancia en aplicaciones de automatización, exploración, logística y manufactura. Diversas investigaciones han propuesto metodologías para resolver problemas relacionados con localización, evasión de obstáculos y planeación de trayectorias en entornos dinámicos y parcialmente conocidos.

Uno de los enfoques más utilizados en robótica móvil es el empleo de sistemas de localización basados en odometría. Sin embargo, estos métodos presentan limitaciones debido a errores acumulativos provocados por ruido sensorial y deslizamiento del robot, lo que afecta la precisión de la estimación de posición con el paso del tiempo. Para reducir estos errores se han implementado técnicas de fusión sensorial, siendo el Extended Kalman Filter (EKF) una de las más utilizadas debido a su capacidad para combinar diferentes fuentes de información y mejorar la estimación del estado del sistema.

Por otra parte, la visión computacional ha adquirido gran importancia dentro de los sistemas autónomos de navegación.

En particular, los marcadores ArUco se han convertido en una solución eficiente para tareas de localización y reconocimiento espacial, debido a que permiten identificar patrones visuales únicos y estimar posiciones relativas mediante cámaras. Su implementación en robótica facilita la corrección de errores generados por odometría, permitiendo mejorar la estabilidad del sistema de navegación.

En cuanto a evasión de obstáculos, los algoritmos Bug son considerados métodos clásicos de navegación reactiva. El algoritmo Bug0 consiste en dirigir el robot hacia la meta y, en caso de detectar un obstáculo, rodearlo hasta encontrar nuevamente una trayectoria libre hacia el objetivo. Este método destaca por su simplicidad computacional y facilidad de implementación.

Por otro lado, el algoritmo Bug2 introduce una estrategia más estructurada mediante una línea de referencia denominada m-line, que conecta el punto inicial con la meta. Cuando el robot encuentra un obstáculo, lo rodea hasta regresar a dicha línea, retomando posteriormente el desplazamiento hacia el objetivo. Esto permite reducir trayectorias innecesarias y mejorar la eficiencia del recorrido.

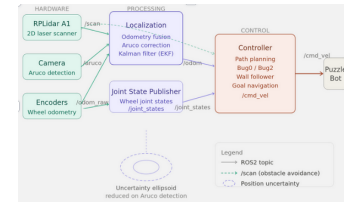


Fig. 4. Diagrama funcionamiento de puzzlebot

IV. MATERIALES Y MÉTODOS

La metodología del proyecto se dividió en varias etapas orientadas a la integración progresiva de los módulos de navegación.

A. Etapa 1: Configuración del entorno de simulación

Inicialmente se configuró el entorno de simulación en Gazebo, integrando el modelo del Puzzlebot y definiendo un escenario con obstáculos que permitiera evaluar el desempeño del sistema de navegación autónoma.

Asimismo, se utilizó RViz para visualizar la información generada por sensores y nodos internos del sistema, incluyendo trayectoria, estimación de pose y detección de obstáculos.

B. Etapa 2: Implementación de localización con ArUco

Se implementó un sistema de visión computacional basado en marcadores ArUco para mejorar la localización del robot. La cámara del Puzzlebot detectaba los marcadores presentes en el entorno y estimaba la posición relativa respecto a ellos.

La detección visual permitió corregir errores acumulativos provenientes de odometría, mejorando la precisión de navegación.

C. Etapa 3: Integración del filtro EKF

Posteriormente se implementó un filtro EKF encargado de fusionar la información obtenida mediante odometría y marcadores ArUco. Este proceso permitió generar una estimación de pose más estable y menos susceptible al ruido sensorial.

La integración del filtro mejoró significativamente la capacidad del robot para mantener trayectorias más cercanas a la meta establecida.

D. Etapa 4: Implementación de algoritmos Bug0 y Bug2

Para la evasión de obstáculos se implementaron algoritmos reactivos de navegación.

En Bug0, el robot avanzaba directamente hacia la meta y únicamente modificaba su trayectoria al detectar obstáculos mediante LiDAR, rodeándolos hasta recuperar una ruta libre.

En Bug2 se añadió una estrategia basada en una trayectoria de referencia entre el origen y la meta, permitiendo recorridos más organizados y reduciendo desplazamientos innecesarios.

E. Etapa 5: Integración del sistema completo

Finalmente, se integraron todos los módulos en una arquitectura ROS2 modular, permitiendo la comunicación entre nodos de percepción, localización y control de movimiento.

La arquitectura incluyó los siguientes componentes:

- Detección de ArUco
- Localización mediante EKF
- Controlador de movimiento
- *Wall follower*
- Generación de trayectorias
- Evasión de obstáculos

Esta integración permitió evaluar el desempeño global del robot en escenarios simulados.

V. DISCUSIÓN Y ANÁLISIS DE RESULTADOS

Los resultados obtenidos permiten afirmar que la integración entre localización visual y algoritmos de navegación reactiva mejora significativamente el desempeño autónomo de un robot móvil en entornos simulados.

La utilización de marcadores ArUco demostró ser una alternativa efectiva para corregir errores de localización derivados de la odometría. Sin embargo, su efectividad depende de diversos factores, tales como la orientación de la cámara, la distancia de detección y la visibilidad del marcador dentro del entorno.

Por otra parte, el filtro de Kalman extendido (EKF) mostró ventajas importantes al fusionar diferentes fuentes de información, permitiendo reducir la incertidumbre y estabilizar la estimación de pose. Esto tuvo un impacto positivo en la capacidad del robot para seguir trayectorias más consistentes.

Asimismo, el análisis comparativo entre los algoritmos Bug0 y Bug2 evidenció diferencias relevantes en el desempeño de navegación. Mientras que Bug0 resultó funcional en escenarios simples, Bug2 presentó un comportamiento más eficiente al optimizar la recuperación de la trayectoria hacia el objetivo, reduciendo recorridos innecesarios.

No obstante, también se identificaron ciertas limitaciones del sistema, principalmente relacionadas con la sensibilidad de parámetros de control, la detección inconsistente de marcadores y el ruido presente en el entorno de simulación. Estos factores pueden afectar la precisión del desplazamiento y la estabilidad general del sistema.

En términos generales, los resultados obtenidos validan la viabilidad de implementar estrategias híbridas de navegación que integren percepción visual, fusión sensorial y algoritmos reactivos, logrando mejorar la autonomía y robustez del Puzzlebot dentro de entornos controlados. Comparando ambos algoritmos de navegación, Bug0 resultó ser más simple y fácil de implementar, aunque menos eficiente en escenarios con obstáculos complejos. En contraste, Bug2 presentó un mejor comportamiento al minimizar rutas innecesarias mediante el seguimiento de una trayectoria de referencia. Uno de los principales desafíos encontrados durante el desarrollo fue la calibración de parámetros de navegación, particularmente relacionados con velocidad angular, tolerancias de detección y comportamiento cerca de obstáculos. Estos ajustes fueron necesarios para evitar colisiones y mejorar la estabilidad general del sistema.

VI. DESARROLLO DE EXPERIMENTO

La Figura 5 presenta la visualización general del sistema dentro de RViz, donde se muestran los distintos elementos utilizados durante la navegación del Puzzlebot.

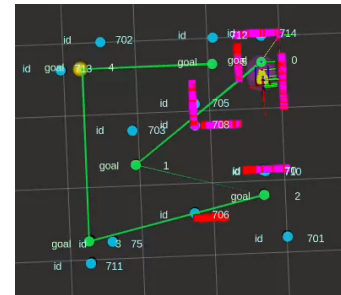


Fig. 5. Visualización general del sistema en RViz.

Como se observa en la Figura 5, el sistema incluye diversos elementos de monitoreo y percepción necesarios para la navegación autónoma:

- **Goals (verde):** representan los puntos objetivo definidos para la trayectoria del robot. Estos puntos sirven como referencia para que el controlador de navegación determine la dirección de desplazamiento.
- **Marcadores ArUco (azul):** representan referencias visuales distribuidas dentro del entorno. Cada marcador posee un identificador único (*id*) utilizado para la estimación de pose del robot.
- **Puzzlebot (amarillo):** representa la posición actual estimada del robot dentro del entorno de simulación.
- **Transformaciones TF:** se visualizan los marcos de referencia asociados a las articulaciones y componentes del

robot, permitiendo observar la relación espacial entre sensores, base móvil y sistema global de coordenadas.

- **Elipsoide de confianza (morado):** ubicado debajo del robot, representa la incertidumbre asociada a la estimación de pose generada mediante el filtro EKF. Un tamaño mayor indica mayor incertidumbre en la localización.
- **Lecturas del LiDAR (rosa fuerte y rojo):** corresponden a los puntos detectados por el sensor de distancia, utilizados para identificar obstáculos y ejecutar los algoritmos de evasión Bug0 y Bug2.

Adicionalmente, el entorno de RViz se encuentra dividido en una cuadrícula de referencia donde cada cuadro corresponde aproximadamente a un área de 1×1 metro, facilitando el análisis espacial del desplazamiento del robot.

A. Análisis del error de localización

Uno de los principales objetivos experimentales fue analizar el comportamiento del sistema de localización antes y después de detectar un marcador ArUco.

La Figura 6 muestra el estado del robot antes de realizar la corrección visual mediante ArUco.

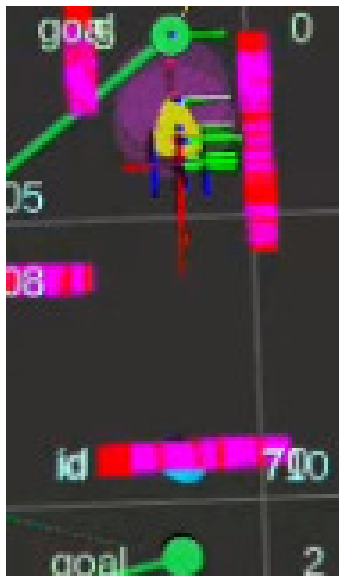


Fig. 6. Error de localización antes de detectar un marcador ArUco.

En esta etapa, el robot presenta un error acumulado de localización generado principalmente por incertidumbre en odometría y ruido sensorial. Esto provoca una estimación menos precisa de la pose, reflejada en un **elipsoide de confianza más amplio**, indicando una mayor incertidumbre sobre la posición real del robot dentro del entorno.

Asimismo, se observa que la orientación estimada puede presentar ligeras desviaciones respecto a la posición esperada, lo que puede impactar negativamente en la precisión de navegación y seguimiento de trayectoria.

Posteriormente, al detectar un marcador ArUco, el sistema realiza una corrección de pose mediante la información visual obtenida por la cámara y fusionada a través del filtro EKF.

La Figura 7 muestra el comportamiento del sistema después de la corrección visual.

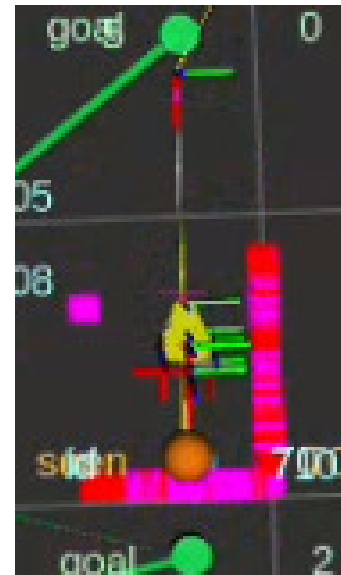


Fig. 7. Corrección del error de localización después de detectar un marcador ArUco.

Como puede observarse en la Figura 7, después de detectar un marcador ArUco se produce una reducción considerable del error de localización. Esto se evidencia mediante la disminución del tamaño del **elipsoide de confianza**, indicando que el sistema presenta una menor incertidumbre respecto a la pose estimada del robot.

La corrección visual permitió ajustar tanto la posición como la orientación del Puzzlebot, mejorando la estabilidad del sistema de navegación y reduciendo desviaciones durante el desplazamiento hacia las metas establecidas.

Estos resultados validan la importancia de integrar marcadores ArUco junto con un filtro EKF, ya que permiten compensar errores acumulativos provenientes de odometría y mejorar significativamente la precisión del sistema de navegación autónoma. “`latex id=goalsresults`”

B. Configuración de Goals y Evaluación de Precisión de Navegación

Con el propósito de evaluar el desempeño del sistema de navegación autónoma, se definieron múltiples puntos objetivo (*goals*) distribuidos estratégicamente dentro del entorno de simulación. Estos puntos permitieron analizar la capacidad del Puzzlebot para desplazarse de forma autónoma hacia ubicaciones específicas mientras evitaba obstáculos y corregía errores de localización.

La Tabla I muestra las coordenadas cartesianas de los *goals* definidos dentro del entorno de navegación. Cada punto fue utilizado como una meta de desplazamiento para evaluar el comportamiento del robot en diferentes trayectorias y configuraciones espaciales.

Durante el proceso experimental, el robot navegó entre distintos pares de objetivos utilizando los algoritmos Bug0 y

Bug2, integrados con el sistema de localización basado en marcadores ArUco y el filtro EKF.

TABLE I
COORDENADAS DE LOS GOALS DEFINIDOS EN EL ENTORNO DE SIMULACIÓN

Goal ID	x (m)	y (m)
0	0.36	-0.27
1	1.70	-2.04
2	2.18	-0.30
3	2.72	-2.72
4	0.36	-2.72
5	0.36	-0.94

Para validar la precisión del sistema, se midió la distancia final entre la base del robot y el *goal* objetivo al concluir cada segmento de navegación. La Tabla II presenta las distancias residuales registradas para diferentes trayectorias ejecutadas durante el experimento.

TABLE II
DISTANCIA RESIDUAL ENTRE LA BASE DEL ROBOT Y EL GOAL AL FINALIZAR CADA TRAYECTO

Initial Goal ID	Final Goal ID	Distance (m)
0	1	0.01
1	2	0.10
2	3	0.08
3	4	0.09
4	5	0.07
0	4	0.08
5	1	0.01
0	2	0.09

A partir de los resultados obtenidos, se observó que el sistema alcanzó un nivel adecuado de precisión durante la navegación autónoma. El error residual promedio registrado al finalizar los recorridos fue cercano a los **9.5 cm**, valor que se definió como el *threshold* final de aceptación para considerar un objetivo como alcanzado.

Este resultado demuestra que la integración entre localización visual mediante marcadores ArUco, el filtro EKF y los algoritmos de evasión Bug0/Bug2 permitió mantener un error relativamente bajo durante el desplazamiento del robot, incluso en presencia de obstáculos dentro del entorno simulado.

Asimismo, el comportamiento observado evidenció que el robot fue capaz de corregir desviaciones acumulativas de localización, permitiendo aproximarse de forma consistente a las metas establecidas.

C. Distribución de marcadores ArUco

Con el fin de proporcionar referencias visuales para la corrección de pose, se distribuyeron múltiples marcadores ArUco dentro del entorno de simulación. La ubicación de dichos marcadores fue seleccionada estratégicamente para maximizar la cobertura visual del robot durante su navegación.

La Tabla III presenta las coordenadas espaciales de los marcadores empleados durante el experimento.

La disposición de estos marcadores permitió al sistema detectar referencias visuales durante distintos segmentos de navegación, reduciendo errores acumulativos de odometría y mejorando la precisión global de localización del Puzzlebot.

TABLE III
POSICIONES DE LOS MARCADORES ARUCO DENTRO DEL ENTORNO DE SIMULACIÓN

Marker ID	x (cm)	y (cm)
70	184.0	-30.0
705	90.0	-120.0
708	119.0	-121.0
706	239.5	-126.0
701	276.5	0.0
702	0.0	-246.0
703	123.0	-207.0
75	274.0	-240.0
710	185.5	-27.0
711	302.7	-271.0
712	0.0	-92.0
713	36.5	-302.0
714	0.0	-26.0

VII. CONCLUSION

Aldrick:

Este proyecto me permitió reconocer que la navegación autónoma en robótica móvil no depende únicamente de implementar algoritmos, sino de comprender cómo interactúan la percepción, la localización, la fusión de sensores y el control dentro de un sistema completo. A lo largo del desarrollo aprendí la importancia de trabajar con herramientas como ROS2, Gazebo, RViz, marcadores ArUco, odometría, EKF y algoritmos de evasión de obstáculos, ya que cada una aporta una parte esencial para que el robot pueda estimar su posición, tomar decisiones y desplazarse de manera más confiable. Una de las principales novedades fue entender cómo la fusión de información permite reducir errores acumulativos y mejorar la estabilidad del sistema, aunque también quedó claro que existen oportunidades de mejora en la calibración de parámetros, la sincronización entre nodos, la detección de marcadores y la validación en escenarios más complejos. Estos aprendizajes abren oportunidades importantes para trabajos futuros, como probar el sistema en un robot físico, integrar métodos de navegación más robustos, optimizar la planeación de trayectorias y aplicar mejores prácticas de gestión de proyectos para distribuir responsabilidades, riesgos y criterios de avance desde etapas tempranas. En general, el proyecto deja una base técnica valiosa para seguir desarrollando soluciones robóticas autónomas más precisas, escalables y aplicables a entornos reales.

Diego:

Este proyecto representó una experiencia muy enriquecedora tanto desde la perspectiva técnica como desde la gestión de equipos. Más allá de los resultados obtenidos en la implementación del sistema de navegación autónoma, permitió comprender la importancia de integrar diferentes componentes de robótica móvil, como la percepción del entorno, la localización, la planeación de movimiento y el control, para lograr un comportamiento autónomo confiable. Durante el desarrollo se fortalecieron habilidades relacionadas con el uso de herramientas y tecnologías especializadas, así como la capacidad de análisis y resolución de problemas en entornos complejos.

Adicionalmente, desempeñar el rol de Project Manager me permitió desarrollar competencias de liderazgo, organización y coordinación del trabajo, asegurando el cumplimiento de objetivos, la distribución de tareas y el seguimiento del avance del proyecto. Esta experiencia ayudó a comprender de manera más clara la relevancia de una gestión efectiva para alcanzar resultados exitosos en proyectos de ingeniería.

Si bien existen oportunidades de mejora, como la optimización de parámetros, la validación en escenarios más exigentes y la incorporación de estrategias de navegación más avanzadas, el proyecto proporciona una base sólida para futuras extensiones y aplicaciones en entornos reales. En términos generales, disfruté mucho participar en este proyecto, ya que me permitió combinar conocimientos técnicos con habilidades de gestión, aprender de los desafíos encontrados durante el desarrollo y colaborar en la construcción de una solución que integra conceptos fundamentales de la robótica autónoma. Y sobre todo por el hecho de que no fue un proyecto más, sino el proyecto integrador de nuestra carrera universitaria.

Emmanuel:

Este proyecto me permitió reconocer que la navegación autónoma en robótica móvil no depende únicamente de implementar algoritmos, sino de comprender cómo interactúan la percepción, la localización, la fusión de sensores y el control dentro de un sistema completo. A lo largo del desarrollo aprendí la importancia de trabajar con herramientas como ROS2, Gazebo, RViz, marcadores ArUco, odometría, EKF y algoritmos de evasión de obstáculos, ya que cada una aporta una parte esencial para que el robot pueda estimar su posición, tomar decisiones y desplazarse de manera más confiable. Una de las principales novedades fue entender cómo la fusión de información permite reducir errores acumulativos y mejorar la estabilidad del sistema, aunque también quedó claro que existen oportunidades de mejora en la calibración de parámetros, la sincronización entre nodos, la detección de marcadores y la validación en escenarios más complejos. Estos aprendizajes abren oportunidades importantes para trabajos futuros, como probar el sistema en un robot físico, integrar métodos de navegación más robustos, optimizar la planeación de trayectorias y aplicar mejores prácticas de gestión de proyectos para distribuir responsabilidades, riesgos y criterios de avance desde etapas tempranas. En general, el proyecto deja una base técnica valiosa para seguir desarrollando soluciones robóticas autónomas más precisas, escalables y aplicables a entornos reales.

Brisa:

Trabajar en este proyecto me dejó claro que la diferencia entre un algoritmo que funciona en papel y uno que funciona en un robot real está en los detalles: los valores infinitos del LiDAR, la fricción del motor, la latencia del sensor, los parámetros que hay que ajustar una y otra vez hasta encontrar el balance correcto. Esa brecha entre teoría y hardware es donde realmente se aprende. También comprendí que en un sistema robótico ningún componente trabaja solo — el wall-follower depende de que el LiDAR esté bien filtrado, el Bug2

depende de que la odometría sea confiable, y todo depende de que los parámetros estén bien calibrados para el entorno específico. Hacia adelante, me quedo con la necesidad de ser más sistemática al hacer pruebas: definir qué se va a medir, registrar los resultados y cambiar una variable a la vez, porque en hardware improvisar cuesta tiempo que no siempre se tiene.

VIII. PROJECT MANAGEMENT

Antes del desarrollo e implementación del sistema de navegación autónoma, se realizó una fase de planeación del proyecto con el objetivo de definir las actividades, etapas y recursos necesarios para cumplir con los objetivos establecidos. Durante esta etapa se identificaron los componentes principales del sistema, se establecieron mini retos de desarrollo y se organizó el trabajo de manera progresiva para facilitar la integración de módulos y la validación de resultados. Como resultado de esta planificación, se elaboró una estructura de desglose del trabajo (WBS) que permitió distribuir y dar seguimiento a las actividades del proyecto.

El Work Breakdown Structure (WBS) del proyecto permitió organizar el desarrollo del sistema de navegación autónoma del Puzzlebot mediante la división del trabajo en diferentes etapas y mini retos. Se estructuraron actividades relacionadas con el modelado del robot en ROS2, el modelo cinemático, visualización multi-robot, manejo de incertidumbre, evasión de obstáculos, filtros probabilísticos y la integración final del sistema de navegación autónoma. Esta organización facilitó la planificación, seguimiento y validación progresiva de cada componente del proyecto hasta lograr un sistema funcional.

hyperref

Carpeta de Google Drive

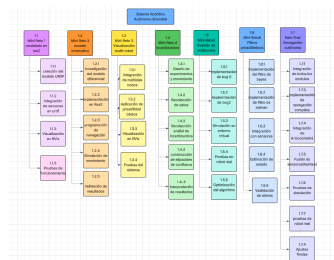


Fig. 8. Work Breakdown Structure (WBS) del proyecto

IX. REPOSITORIO DEL PROYECTO

El código fuente, implementación de algoritmos y archivos relacionados con el desarrollo del sistema de navegación autónoma del Puzzlebot se encuentran disponibles en el siguiente repositorio de GitHub:

Repositorio Puzzlebot-Bremdial

REFERENCES

- [1] Robotnik, "Navegación autónoma y robots móviles," 2025. [En línea]. Disponible en: <https://robotnik.eu/es/navegacion-autonoma-y-robots-moviles/>. Consultado: 12-jun-2026.
- [2] Automatic Addison, "The Bug2 Algorithm for Robot Motion Planning," 2020. [En línea]. Disponible en: <https://automaticaddison.com/the-bug2-algorithm-for-robot-motion-planning/>. Consultado: 12-jun-2026.