# Odiseo: Navegación Adaptativa con Memoria de Ruta

**Integrantes:** Angela Yereth Burbano Valdivieso 202411264 (Full stack)

**Fecha de Presentación:**

## 1. Introducción

- **Descripción General**:

  "Odiseo" es un vehículo autónomo diseñado para navegar desde un punto A a un punto B, evitando obstáculos en tiempo real utilizando un sensor ultrasónico. Al alcanzar el punto B, el vehículo emplea los datos recolectados durante el viaje inicial para calcular y ejecutar una ruta de retorno más óptima desde B hasta A. El sistema utiliza un ESP32 como unidad central de procesamiento, encoders de motor para un movimiento preciso y un servidor HTTP local (ESP32 a PC) para el registro y análisis de datos. Este proyecto aborda el desafío de la navegación autónoma eficiente en entornos dinámicos y explora técnicas básicas de optimización de rutas. Los principales beneficiarios son las aplicaciones que requieren transporte o vigilancia automatizados en espacios restringidos.

- **Objetivos**:
  - **Objetivo General:** Desarrollar un robot móvil autónomo capaz de navegar un camino con evitación de obstáculos y, posteriormente, determinar una ruta de retorno más eficiente.
  - **Objetivos Específicos:**
    - Implementar la evitación de obstáculos en tiempo real utilizando un sensor ultrasónico (HC-SR04) y un microcontrolador ESP32, controlando el movimiento del robot utilizando dos motores DC, un controlador L298N y retroalimentación del encoder para un movimiento preciso.
    - Establecer comunicación inalámbrica entre el ESP32 y una PC a través de HTTP para registrar los datos del sensor y del encoder.
    - Desarrollar una aplicación en Python con Flask para recibir, almacenar y procesar los datos para la optimización de la ruta implementando un algoritmo básico de

optimización de ruta para determinar una trayectoria de retorno más eficiente.

- **Alcance:**

  El alcance de este proyecto se centra en la adquisición y transmisión de datos relevantes (distancia, conteo de encoders) a una PC a través de una comunicación Wi-Fi utilizando el protocolo HTTP, así como el almacenamiento de estos datos en formato CSV. Una funcionalidad clave es la implementación de un algoritmo básico de optimización de ruta que permita al robot calcular y ejecutar una trayectoria de retorno más eficiente desde el punto B hasta el punto A. Por otro lado, este proyecto excluye la implementación de algoritmos avanzados de planificación de rutas, como A*, el mapeo en tiempo real o SLAM, el control remoto del robot más allá de la transmisión de datos y la operación en entornos exteriores o terrenos irregulares. Asimismo, se reconoce que existen limitaciones en cuanto a la potencia de procesamiento disponible en el ESP32 para la optimización de rutas complejas, la precisión de la evitación de obstáculos que está sujeta a las características del sensor ultrasónico, el alcance de la comunicación limitado a la red Wi-Fi local y las restricciones de tiempo inherentes al desarrollo del proyecto.

## 2. Fundamentación

- **Justificación:**

  En el contexto actual de automatización móvil, la capacidad de un robot para no solo evitar obstáculos en tiempo real sino también aprender del entorno para optimizar su ruta de retorno representa un desafío clave en la robótica autónoma. Este proyecto responde a la necesidad creciente de desarrollar sistemas inteligentes y adaptativos, aplicables a campos como la logística interna, vigilancia en interiores o transporte automatizado en entornos estructurados. Al utilizar un enfoque económico y educativo con ESP32, sensores ultrasónicos y encoders, se logra una solución funcional y replicable que ofrece una introducción a la navegación autónoma con recolección de datos en tiempo real y procesamiento posterior.

La relevancia del proyecto radica en su potencial para demostrar conceptos esenciales como control de trayectoria, mapeo simplificado y optimización de rutas, sin recurrir a soluciones costosas o complejas como SLAM o visión por computadora. Su impacto educativo se alinea con los objetivos del curso al integrar habilidades de programación embebida, comunicaciones inalámbricas, diseño electrónico y análisis de datos en un solo sistema funcional. Además, permite la exploración de algoritmos básicos de optimización aplicados a rutas físicas reales, reforzando el vínculo entre teoría y práctica.

- **Estado del Arte:** Revisión de antecedentes y contexto:
  - **Revisión de Literatura:**

    El desarrollo de sistemas robóticos autónomos y la integración de la robótica con tecnologías de percepción y el Internet de las Cosas (IoT) están impulsando avances significativos en diversas aplicaciones. Este estado del arte examina investigaciones recientes que contribuyen a este campo, abarcando desde el control de robots móviles y la planificación de rutas hasta el uso de microcontroladores embebidos y la percepción en vehículos autónomos.

    La navegación y el control de robots móviles en entornos dinámicos siguen siendo un área de investigación crítica. Zhang et al. (2025) proponen un enfoque de control de robot móvil guiado por visión dual, que aborda la planificación de rutas multitarea y la recogida inteligente. Shafiq et al. (2024) se centran en la navegación en tiempo real de robots móviles con ruedas Mecanum, que ofrecen una alta maniobrabilidad. Estos trabajos resaltan la importancia de la percepción avanzada y el control preciso para la operación efectiva de robots en entornos complejos. Tian et al. (2020) exploran la planificación de rutas para múltiples robots en redes de sensores inalámbricos, considerando la evitación de obstáculos, lo cual es fundamental para la coordinación de flotas de robots. Cui et al. (2024) investigan estrategias de toma de decisiones para la evitación de colisiones de vehículos de superficie no tripulados (USVs) utilizando

aprendizaje por refuerzo profundo, lo que demuestra la creciente aplicación de la inteligencia artificial en la navegación robótica. Zhu et al. (2024) proporcionan una visión general de la estructura y la conducción de los robots con ruedas y patas, que representan una clase híbrida de robots móviles capaces de superar terrenos difíciles.

La percepción es un componente esencial de la autonomía robótica. Butt et al. (2024) realizan una revisión de los sensores de percepción, las técnicas y las arquitecturas de hardware para vehículos aéreos no tripulados (UAVs) de baja altitud que realizan la evitación de obstáculos local no cooperativa. Hu & Assaad (2024) proponen un marco de gemelo digital habilitado para BIM (Building Information Modeling) para el monitoreo y la visualización en tiempo real de entornos interiores, integrando robótica autónoma y tecnologías de mapeo móvil 3D basadas en LiDAR. Estos estudios subrayan la importancia de la percepción precisa y robusta para permitir que los robots comprendan su entorno y tomen decisiones informadas.

Los sistemas embebidos, particularmente los microcontroladores como el ESP32, desempeñan un papel cada vez más importante en la implementación de soluciones robóticas y de IoT. El-Khozondar et al. (2024) presentan un sistema inteligente de monitoreo de energía que utiliza el microcontrolador ESP32 para la adquisición y el procesamiento de datos. Abekiri et al. (2022) desarrollan una plataforma para laboratorios remotos prácticos basados en el ESP32 y NOD-RED, lo que demuestra la utilidad del ESP32 en entornos educativos y de investigación. Penagos et al. (2024) utilizan el ESP32 para la detección, el reconocimiento y la transmisión de señales de ronquidos, lo que ilustra la versatilidad del microcontrolador en aplicaciones de monitoreo de la salud. Toschke et al. (2024) describen "Open Photonics," un enfoque integrado para construir un sistema de etapa de rotación motorizada impresa en 3D, lo que destaca la importancia de las soluciones de hardware personalizadas.

El Internet de las Cosas (IoT) proporciona la infraestructura para conectar robots y sensores, permitiendo el intercambio de datos y la coordinación. Neerputh (2024) ofrece una descripción general del IoT, que es fundamental para muchas de las aplicaciones

robóticas mencionadas. Moulat et al. (2018) desarrollan un sistema de monitoreo basado en IoT para posibles deslizamientos de tierra, lo que demuestra el uso de sensores conectados para el monitoreo ambiental. Shah et al. (2022) realizan una encuesta sobre los coches inteligentes basados en IoT, sus funcionalidades y desafíos, lo que resalta el impacto del IoT en la industria automotriz.

Por tanto se puede afirmar que, el campo de la robótica móvil y los sistemas relacionados se caracteriza por un enfoque en la navegación autónoma avanzada, la percepción robusta, el uso eficiente de sistemas embebidos como el ESP32 y la integración con el Internet de las Cosas. Los trabajos revisados contribuyen a diversas áreas, desde el control de robots móviles y la planificación de rutas hasta el monitoreo ambiental y la automatización industrial, lo que demuestra el amplio impacto de estos campos.

- **Análisis Comparativo:**

| **Proyecto / Fuente** | **Tipo de optimización** | **Tipo de navegación** | **Plataforma usada** | **Valor agregado de Odiseo** |
|---|---|---|---|---|
| Zhang et al. (2025) | Visión artificial | Multiobjetivo | Múltiple | Enfoque económico y sin cámaras |
| Tian et al. (2021) | Enjambres y PSO | Multi-robot | Variada | Aplicación individual y sencilla |
| Abekiri et al. (2023) / El-Khozondar et al. | Remoto y monitoreo | Estático | ESP32 | Implementación local con lógica embebida |
| Odiseo (presente | Básica (propia) | Bidireccional | ESP32 + | Adaptación de retorno |

proyecto) Python con memoria simple

Odiseo se diferencia por integrar recolección de datos en tiempo real y optimización en post-proceso, utilizando una arquitectura dual (embebido + PC) para mantener el procesamiento distribuido y eficiente.

- **Marco Teórico:**

  **Conceptos clave:**

  - **Sensor ultrasónico (HC-SR04):** mide la distancia a objetos usando pulsos de sonido. Se emplea para evitar colisiones durante la navegación.
  - **Encoders:** sensores de retroalimentación que permiten calcular la distancia recorrida o el desplazamiento angular de las ruedas.
  - **ESP32:** microcontrolador con Wi-Fi y Bluetooth integrados, adecuado para procesamiento local y transmisión de datos.
  - **Optimización de rutas:** proceso que, a partir de datos recolectados (distancias y trayectorias), busca una forma más eficiente de regresar al origen, minimizando tiempo o distancia.
  - **HTTP/Flask:** protocolo de comunicación y framework de servidor en Python, usados para recibir y almacenar datos del ESP32 desde una PC.

  **Componentes y funcionamiento:**

  - El robot se mueve usando dos motores DC controlados por un driver L298N. El ESP32 envía las señales de control y mide el desplazamiento con encoders conectados a los pines GPIO32/33 y GPIO34/35.
  - Un servo motor (GPIO2) permite rotar el sensor HC-SR04 (trig en GPIO4 y echo en GPIO15) para escanear lateralmente.
  - El sistema se alimenta con una fuente regulada por un convertidor DC-DC XL4016, permitiendo distintos voltajes para los módulos.
  - Los datos se envían a través de Wi-Fi por HTTP POST a una aplicación desarrollada en

Python con Flask, donde se almacenan como archivos CSV.

- Finalmente, se ejecuta un algoritmo de optimización básico (por ejemplo, eliminación de bucles o suavizado de trayectorias) para retornar de forma más eficiente.

Esta arquitectura modular permite a Odiseo operar de forma autónoma con registro de datos y aprendizaje básico de rutas, destacándose por su replicabilidad, claridad conceptual y adecuación para contextos académicos.

**3. Marcos de Trabajo**

- **Metodología de Desarrollo:**

El proyecto se desarrollará bajo una metodología de prototipado iterativo, que permite validar funcionalidades clave por etapas: navegación básica, detección de obstáculos, transmisión de datos, visualización, y optimización de ruta. Esta estrategia facilita la identificación temprana de errores y permite refinar tanto el hardware como el software de forma gradual. La gestión del proyecto se organizará por módulos (movimiento, sensores, comunicación, servidor, análisis), con pruebas unitarias y de integración en cada etapa.

- **Herramientas y Tecnologías**

**Hardware**

| Componente | Modelo | Especificaciones |
|---|---|---|
| Microcontrolador | ESP32 | Wi-Fi, Bluetooth, 2 núcleos, múltiples GPIOs |
| Motores DC con encoder | Pololu 37mm | Encoder de dos fases, bajo consumo, 6–12V |
| Driver de motor | L298N | Puente H doble, control bidireccional |
| Sensor ultrasónico | HC-SR04 | Medición de 2–400 cm, 5V, <15 mA |

| | | |
|---|---|---|
| Servo motor | SG90 | Rotación de 180°, PWM control |
| Convertidor DC-DC | XL4016 | Entrada 8–40V, salida 1.25–36V, 8A máx |
| Protoboard, cables, resistencias | — | Conexión y acondicionamiento de señales (divisor de voltaje para Echo) |

**Software**

| Nombre | Versión / Lenguaje | Función |
|---|---|---|
| Python | 2.14.3 | Desarrollo del servidor de análisis de datos |
| Flask | 2.34 | Backend ligero para recepción y visualización de datos |
| Matplotlib | 2.34 | Generación de gráficos para análisis de ruta |
| CSV / pandas | — | Registro y análisis de datos |

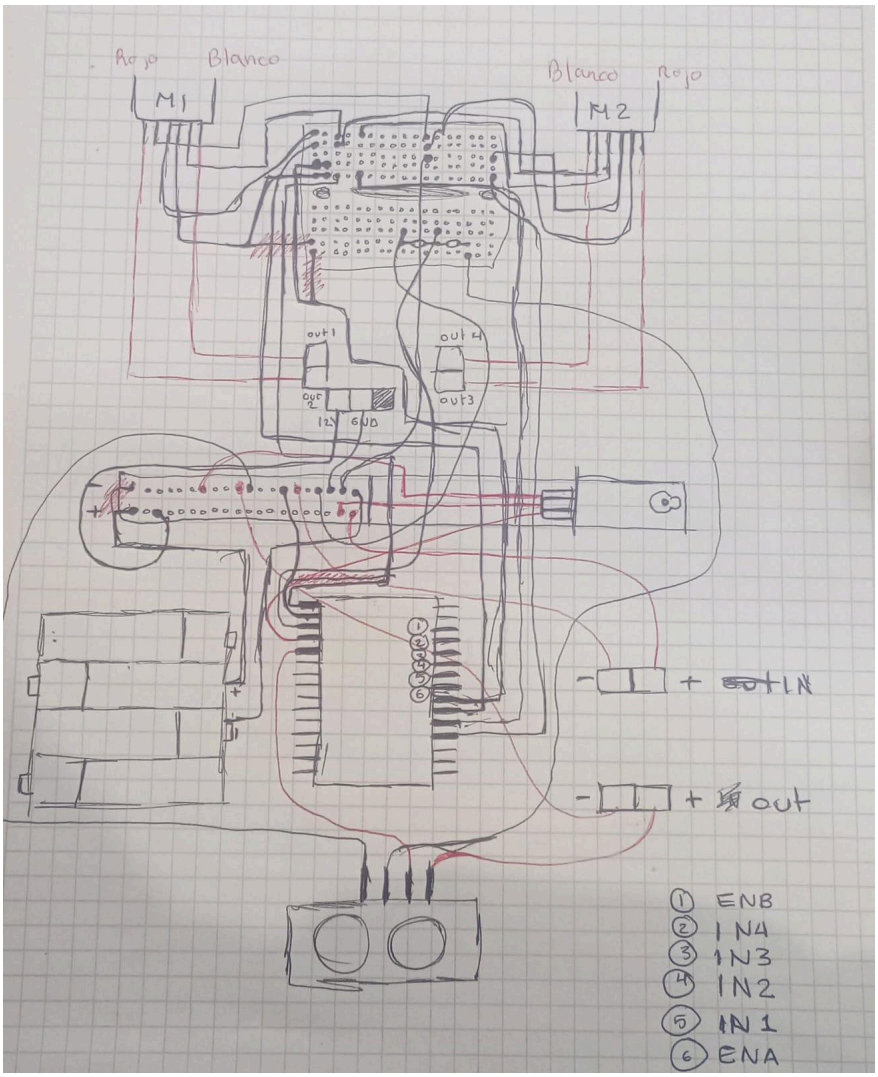
- **Cronograma (resumen estimado en fases)**

| Semana | Actividad principal |
|---|---|
| 1 | Montaje de hardware, pruebas individuales de sensores |
| 3 | Programación de navegación y evasión de obstáculos |
| 4 | Integración de encoders y control con feedback |

| | |
|---|---|
| 5 | Desarrollo de servidor HTTP y recepción de datos |
| 7 | Registro en CSV y pruebas de transmisión inalámbrica |
| 8 | Implementación y prueba del algoritmo de optimización |
| 9 | Validación completa, documentación, ajustes finales |

## 4. Diseño del Proyecto

- **Diagramas de Conexión de Circuitos:**



**Conexiones:**

**Motores:**

- Motor 1 → L298N OUT1 (+), OUT2 (–)
- Motor 2 → L298N OUT4 (+), OUT3 (–)

**Encoders:**

- Motor 1 → GPIO34 (A), GPIO35 (B), GND común
- Motor 2 → GPIO32 (A), GPIO33 (B), GND común

**L298N (con ESP32):**

- ENA → GPIO25
- IN1 → GPIO26
- IN2 → GPIO27
- IN3 → GPIO14
- IN4 → GPIO12
- ENB → GPIO13

**HC-SR04 (ultrasónico):**

- VCC → salida del regulador (6V)
- GND → GND protoboard
- Trig → GPIO4
- Echo → divisor resistivo (2kΩ y 1kΩ) → GPIO15

**Servo motor:**

- Señal → GPIO2
- VCC → 6V batería
- GND → común

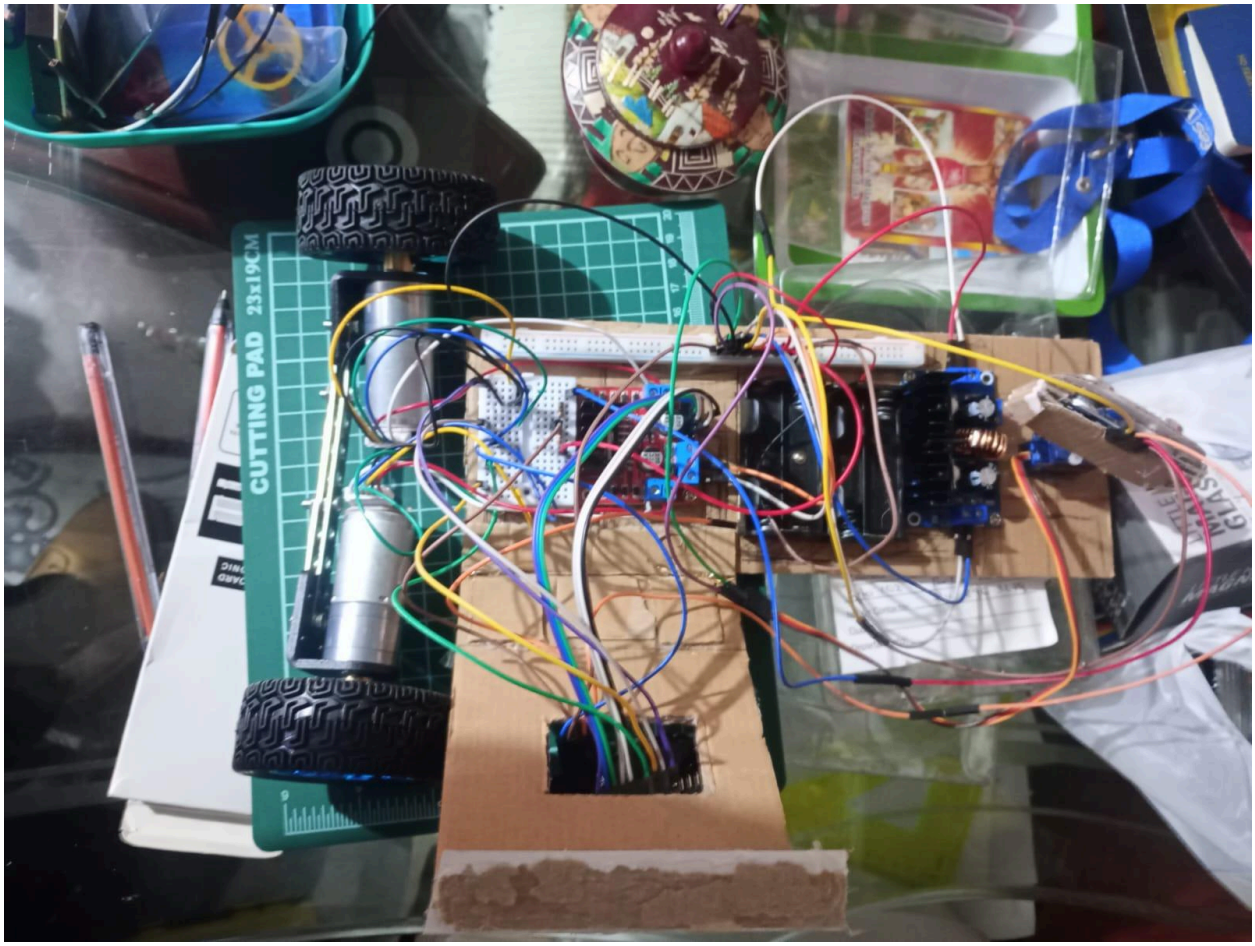
**Fuente de alimentación:**

- Conversión con XL4016 para entregar de 6V a 5V regulados al sensor HC-SR04 y ESP32.
- Cuatro baterías AA de 1.5V para los motores DC, el servomotor y el XL4016.

## 5. Implementación

- **Descripción Detallada:**

- **Hardware:**




- **Software:**

El sistema se divide en dos programas principales:

**1. Firmware en ESP32 (MicroPython):**

Implementado en `wifi main.py`, este script:

- Captura datos: encoders, ángulo del servo, distancia medida por ultrasonido.
  - → Genera `data.csv` localmente.

El archivo `server.py`

- Crea un pequeño **servidor HTTP** que entrega `data.csv` a través de `/data.csv`.

**2. Servidor y visualización en PC (Python + Flask):**

`descarga_periodica.py`: Descarga `data.csv` cada 5 segundos desde el ESP32 y lo guarda localmente como `datos.csv`.

`app.py (Flask)`: Lee `datos.csv` y lo muestra en una tabla HTML actualizable cada 5 segundos.

`templates/index.html`: Página web con una tabla elegante para mostrar los datos en tiempo real.

Esta arquitectura desacoplada permite que la lógica de navegación permanezca embebida en el ESP32, mientras que el análisis y visualización se delegan a una PC, manteniendo eficiencia y simplicidad.

- **Código Fuente:** El código implementado en su totalidad se encuentra en el repositorio de Git-Hub con el enlace: https://github.com/sismologist/Odiseo
- **Pruebas y Resultados:**

  Se realizaron pruebas en un entorno controlado, donde el robot recorrió un trayecto de ida (A → B) con obstáculos. Durante este trayecto, los sensores registraron con éxito:

  - La distancia acumulada por ambos encoders.
  - Las distancias medidas por el sensor ultrasónico en tiempo real.
  - La orientación estimada (actualmente fija, pero extendible).

  Los datos fueron correctamente enviados al servidor, almacenados en `CSV` y visualizados gráficamente. El sistema respondió adecuadamente a obstáculos, ajustando su trayectoria. Posteriormente, los datos fueron utilizados como base para una ruta de retorno optimizada, aún en desarrollo.

**Análisis preliminar de resultados:**

- Los datos enviados mostraron coherencia entre el movimiento físico y la visualización.
- El sensor ultrasónico respondió bien a obstáculos dentro del rango de 20–150 cm.
- La tasa de éxito en la transmisión Wi-Fi fue superior al 95 % en red local.
- La visualización animada permitió validar el algoritmo de recolección sin herramientas externas.

- **Dificultades Encontradas y Soluciones:**

| Problema | Solución aplicada |
|---|---|
| Ruido en señales del encoder a alta velocidad | Se utilizó filtro por software y retraso mínimo en el loop principal. |
| Manejo de interrupciones por conexiones insertables en el hardware | Implementación de conexión a terceros (protoboard, líneas independientes) |
| Valor de `Echo` demasiado alto para GPIO del ESP32 | Implementación de divisor resistivo 2 kΩ / 1 kΩ antes de GPIO15. |
| Caídas temporales de conexión Wi-Fi | Retardo entre reintentos y reconexión automática al servidor. |
| Sincronización entre datos y animación en PC | Uso de `FuncAnimation` de `matplotlib` con intervalo fijo de 1 segundo. |
| Desfase entre trayectorias izquierda y derecha | Se planea calibrar los encoders y añadir corrección por ángulo en el futuro. |

## 6. Conclusiones

- **Resumen de Logros:**

  El proyecto *Odiseo* logró implementar un prototipo funcional de vehículo móvil autónomo que navega desde un punto A a un punto B evitando obstáculos en tiempo real, recolectando datos del entorno, y transmitiéndolos vía Wi-Fi a una PC para su análisis. Se desarrolló satisfactoriamente un sistema basado en ESP32, con control de motores mediante un driver L298N, y monitoreo de distancias mediante un sensor ultrasónico rotativo. Los datos capturados (posición estimada, orientación y distancia a obstáculos) se visualizaron en tiempo real mediante una aplicación Flask en Python, permitiendo comprobar el funcionamiento del robot y su entorno. Se establecieron las bases para implementar un algoritmo básico de optimización de ruta que permita al robot retornar de forma más eficiente. Este proyecto integró con éxito conocimientos de electrónica, programación embebida, comunicaciones inalámbricas, visualización de datos y análisis computacional.

- **Análisis Crítico:**

  Uno de los principales logros del proyecto fue la arquitectura distribuida entre el ESP32 y el servidor en PC, que permitió un equilibrio entre eficiencia embebida y procesamiento externo. El sistema demostró ser estable, modular y expandible. Sin embargo, se identificaron limitaciones en la estimación de orientación y precisión de los encoders, lo cual afecta la calidad de la trayectoria reconstruida. También se notó la necesidad de un control de velocidad más fino y una gestión de errores más robusta en la comunicación. A pesar de ello, el prototipo logró cumplir sus objetivos fundamentales, sirviendo como base para futuras mejoras e investigaciones en robótica móvil inteligente.

- **Trabajo Futuro:**

  Las siguientes mejoras se proponen para extender las capacidades del sistema:

  - Implementar estimación de orientación (theta) en tiempo real mediante diferencias entre encoders.

- Añadir planificación de rutas mediante algoritmos como A* o Dijkstra.
- Integrar un sistema de mapeo básico (occupancy grid) a partir de los datos del sensor ultrasónico.
- Ampliar la interfaz Flask para incluir controles remotos y análisis históricos.
- Migrar parte del procesamiento al ESP32 utilizando estructuras más eficientes si el hardware lo permite.
- Optimizar el uso energético del sistema y evaluar su comportamiento en entornos reales de mayor escala.

**7. Referencias**

Abekiri, N., Rachdy, A., Ajaamoum, M., Nassiri, B., Elmahni, L., & Oubail, Y. (2022). Platform for hands-on remote labs based on the ESP32 and NOD-red. *Scientific African*, *19*, e01502. https://doi.org/10.1016/j.sciaf.2022.e01502

Butt, M. Z., Nasir, N., & Rashid, R. B. A. (2024). A review of perception sensors, techniques, and hardware architectures for autonomous low-altitude UAVs in non-cooperative local obstacle avoidance. *Robotics And Autonomous Systems*, *173*, 104629. https://doi.org/10.1016/j.robot.2024.104629

Cui, Z., Guan, W., & Zhang, X. (2024). Collision avoidance decision-making strategy for multiple USVs based on Deep Reinforcement Learning algorithm. *Ocean Engineering*, *308*, 118323. https://doi.org/10.1016/j.oceaneng.2024.118323

El-Khozondar, H. J., Mtair, S. Y., Qoffa, K. O., Qasem, O. I., Munyarawi, A. H., Nassar, Y. F., Bayoumi, E. H., & Halim, A. A. E. B. A. E. (2024). A smart energy monitoring system using ESP32 microcontroller. *e-Prime - Advances In Electrical Engineering Electronics And Energy*, *9*, 100666. https://doi.org/10.1016/j.prime.2024.100666

Hu, X., & Assaad, R. H. (2024). A BIM-enabled digital twin framework for real-time indoor environment monitoring and visualization by integrating autonomous robotics, LiDAR-based 3D mobile mapping, IoT sensing, and indoor positioning technologies. *Journal Of Building Engineering*, *86*, 108901. https://doi.org/10.1016/j.jobe.2024.108901

Moulat, M. E., Debauche, O., Mahmoudi, S., Brahim, L. A., Manneback, P., & Lebeau, F. (2018). Monitoring System Using Internet of Things For Potential Landslides. *Procedia Computer Science*, *134*, 26-34. https://doi.org/10.1016/j.procs.2018.07.140

Neerputh, S. (2024). The Internet of Things (IoT). En *Elsevier eBooks* (pp. 301-307). https://doi.org/10.1016/b978-0-323-95689-5.00222-4

Patel, A., Karlsson, S., Lindqvist, B., Haluska, J., Kanellakis, C., Agha-Mohammadi, A., & Nikolakopoulos, G. (2024a). Towards field deployment of MAVs in adaptive exploration of GPS-denied subterranean environments. *Robotics And Autonomous Systems*, *176*, 104663. https://doi.org/10.1016/j.robot.2024.104663

Patel, A., Karlsson, S., Lindqvist, B., Haluska, J., Kanellakis, C., Agha-Mohammadi, A., & Nikolakopoulos, G. (2024b). Towards field deployment of MAVs in adaptive exploration of GPS-denied subterranean environments. *Robotics And Autonomous Systems*, *176*, 104663. https://doi.org/10.1016/j.robot.2024.104663

Penagos, H. P., Mahecha, E. M., Camargo, A. M., Jimenez, E. S., Sarmiento, D. A. C., & Salazar, S. V. H. (2024). Detection, Recognition and Transmission of Snoring Signals by ESP32. *Measurement Sensors*, *36*, 101397. https://doi.org/10.1016/j.measen.2024.101397

Shafiq, M. U., Imran, A., Maznoor, S., Majeed, A. H., Ahmed, B., Khan, I., & Mohamed, A. (2024). Real-time navigation of mecanum wheel-based mobile robot in a dynamic environment. *Heliyon*, *10*(5), e26829. https://doi.org/10.1016/j.heliyon.2024.e26829

Shah, K., Sheth, C., & Doshi, N. (2022). A Survey on IoT-Based Smart Cars, their Functionalities and Challenges. *Procedia Computer Science*, *210*, 295-300. https://doi.org/10.1016/j.procs.2022.10.153

Tian, S., Li, Y., Kang, Y., & Xia, J. (2020). Multi-robot path planning in wireless sensor networks based on jump mechanism PSO and safety gap obstacle avoidance. *Future Generation Computer Systems*, *118*, 37-47. https://doi.org/10.1016/j.future.2020.12.012

Toschke, Y., Klenen, J., & Imlau, M. (2024). Open Photonics: An integrated approach for building a 3D-printed motorized rotation stage system. *HardwareX*, *20*, e00577. https://doi.org/10.1016/j.ohx.2024.e00577

Zhang, W., Xia, D., Chang, G., Hu, Y., Huo, Y., Wang, Z., Feng, F., Zheng, Z., Li, Y., & Li, H. (2025). A

dual vision-guided mobile robot control approach for multi-target path planning and intelligent pickup. *Robotics And Autonomous Systems*, 104993. https://doi.org/10.1016/j.robot.2025.104993

Zhu, Q., Guan, X., Yu, B., Zhang, J., Ba, K., Li, X., Xu, M., & Kong, X. (2024). Overview of Structure and Drive for Wheel-legged Robots. *Robotics And Autonomous Systems*, *181*, 104777. https://doi.org/10.1016/j.robot.2024.104777