



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN MECATRÓNICA
TRABAJO DE INTEGRACIÓN CURRICULAR

TEMA:

“TELEMETRÍA PARA ROBOTS MÓVILES MEDIANTE LORAWAN Y ROS”

Trabajo de titulación previo a la obtención del título de Ingeniera en
Mecatrónica

Línea de investigación: Prototipos industriales

AUTOR:

Alisson Gabriela Chandi Flores

DIRECTOR:

Dr. Carlos Xavier Rosero Chandi

Ibarra – Ecuador 2026



UNIVERSIDAD TÉCNICA DEL NORTE

DIRECCIÓN DE BIBLIOTECA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO	
APELLIDOS Y NOMBRES:	Chandi Flores Alisson Gabriela

DATOS DE LA OBRA	
TÍTULO:	“Telemetría para robots móviles mediante LoRaWAN y ROS”
AUTOR:	Chandi Flores Alisson Gabriela
FECHA:	16/04/2026
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	Ingeniera en Mecatrónica
DIRECTOR /ASESOR:	PhD. Carlos Xavier Rosero Chandi PhD. Brizeida Nohemí Gámez Aparicio

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 16 días del mes de abril de 2026

EL AUTOR:

Nombre: Chandi Flores Alisson Gabriela
CI: 1004652184

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 16 de abril de 2026

Carlos Xavier Rosero Chandi

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

Carlos Xavier Rosero Chandi

CI: 1002515821

APROBACIÓN DEL COMITÉ CALIFICADOR

El Comité Calificado del trabajo de Integración Curricular “ Telemetría para robots móviles mediante LoRaWAN y ROS” elaborado por Alisson Gabriela Chandi Flores, previo a la obtención del título de Ingeniera Mecatrónica , aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:

Carlos Xavier Rosero Chandi

CI: 1002515821

Brizeida Nohemí Gámez Aparicio

CI: 1758387383

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS	VIII
DEDICATORIA	X
AGRADECIMIENTO	XI
RESUMEN	XII
ABSTRACT	XIII
CAPÍTULO I: EL PROBLEMA	1
1.1 Planteamiento del problema	1
1.2 Objetivos	2
1.2.1 General	2
1.2.2 Específicos	2
1.3 Alcance y delimitación	2
1.4 Justificación	3
CAPÍTULO II: MARCO REFERENCIAL	4
2.1 Antecedentes	4
2.2 Marco Teórico	5
2.2.1 Robótica móvil	6
2.2.2 Telemetría	6
2.2.3 Tecnologías	8
2.2.4 Framework ROS	12
2.2.5 Integración de LoRaWAN con ROS	13
2.2.6 Variables críticas del robot	13
2.2.7 Servidor propietario	16
2.2.8 Sistema operativo	17
2.2.9 Almacenamiento de datos en la nube	18

2.2.10	Microcontrolador en el sistema de telemetría	18
CAPÍTULO III: MARCO METODOLÓGICO		20
3.1	Modelo de la investigación	20
3.2	Diseño de la investigación	21
3.2.1	Fase 1: Evaluación de las variables críticas	21
3.2.2	Fase 2: Diseño de la arquitectura para el sistema	21
3.2.3	Fase 3: Validación del sistema	24
CAPÍTULO IV: RESULTADOS		25
4.1	Arquitectura del sistema de telemetría	25
4.1.1	Módulo GPS	27
4.1.2	Microcomputador	29
4.1.3	Microcontrolador	33
4.1.4	Convertor RS485-LB	36
4.1.5	Servidor propietario	38
4.1.6	Servidor puente	39
4.1.7	Computador local	43
4.2	Pruebas de funcionamiento	50
4.2.1	Pruebas de funcionamiento en campo	53
4.3	Evaluación del rendimiento del sistema de comunicación LoRaWAN	54
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES		56
5.1	Conclusiones	56
5.2	Recomendaciones	57
5.3	Trabajo a futuro	57

Índice de figuras

2.1	Interpretación de la telemetría [19].	8
2.2	Modelo OSI de capas sobre las que se implementan LoRaWAN y LoRa [20].	9
2.3	Elementos de la arquitectura LoRaWAN [22].	10
2.4	Clases de nodos finales del protocolo LoRaWAN [22].	10
2.5	Diagrama de conexiones de 2 dispositivos en comunicación serial asíncrona [24].	11
4.1	Arquitectura del sistema de telemetría.	26
4.2	GY-GPS6MV2.	28
4.3	Pines del módulo GY-GPS6MV2 [41], [23].	29
4.4	Diagrama de flujo instalación ROS2.	32
4.5	Algoritmo del microcontrolador.	35
4.6	Divisor de voltaje.	36
4.7	Mapeo de pines del RS 485 LB de Dragino [44].	37
4.8	Política de Firewall.	40
4.9	Servidor Apache.	41
4.10	Estado del servicio PHP 8.0-FPM.	42
4.11	Diagrama de flujo del proceso de recepción y almacenamiento de datos en MySQL.	43
4.12	Estructura de la base de datos SQLite utilizada para el almacenamiento de datos de telemetría del robot móvil.	44
4.13	Estructura de las variables en Firebase.	46
4.14	Reglas para lectura y escritura de variables en Firebase.	47
4.15	Flujo de datos desde MySQL a Firebase mediante un script de Python.	47

4.16	Interfaz gráfica del sistema de telemetría.	48
4.17	Diagrama de flujo del monitoreo y actualización de datos.	49
4.18	Live data TTN.	50
4.19	Ejecución del script puente y verificación de transferencia de datos.	51
4.20	Variables monitoreadas.	51
4.21	Robot en pruebas de campo.	53
A.1.	Gráfica de latitud.	62
A.2.	Gráfica de longitud.	63
A.3.	Gráfica de velocidad.	63
A.4.	Gráfica del pico máximo del voltaje.	64
A.5.	Gráfica del pico mínimo del voltaje.	64
A.6.	Gráfica de la máxima derivada positiva de latitud.	65
A.7.	Gráfica de la máxima derivada positiva de velocidad.	65

Índice de tablas

2.1	Fundamentos del GPS [27].	14
2.2	Ventajas de usar TTN como servidor propietario [33].	17
2.3	Tipos de microcontroladores y sus características [38].	19
4.1	Características del GY-GPS6MV2 [41].	28
4.2	Configuración de pines [41].	29
4.3	Especificaciones técnicas de la NVIDIA Jetson Orin Nano [42].	30
4.4	Información general del microcontrolador Arduino UNO [43].	34
4.5	Comandos AT.	38
4.6	Información general del RS485-LB.	39
4.7	Características del servidor puente [45].	40
4.8	Parámetros de desempeño del enlace LoRaWAN.	54

DEDICATORIA

Quiero dedicar este trabajo, en primer lugar, a Dios, por ser mi guía constante, mi refugio en los momentos difíciles y la fuerza que me sostuvo cuando sentí que no podía más.

A mis padres, Graciela y Javier, quienes han sido el pilar fundamental de mi vida. Gracias por cada sacrificio, por cada palabra de aliento, por secar mis lágrimas en los momentos más duros y por no permitirme rendirme nunca. Este logro también es suyo, porque aunque la vida no les dio las mismas oportunidades, hicieron todo lo posible para que yo sí las tuviera. Hoy puedo decirles con el corazón en la mano: todo su esfuerzo valió la pena.

A mis hermanos, por su ejemplo, su apoyo incondicional y por cada palabra que me impulsó a seguir adelante cuando más lo necesitaba. A mis sobrinas, y con un amor inmenso a mi bebé, por estar siempre a mi lado, por acompañarme incluso en los momentos más difíciles, por ser ese apoyo silencioso pero tan importante que nunca me dejó caer.

A mis amigos, en especial a Antony, a quien quiero y admiro profundamente. Gracias por estar desde el inicio, por tu apoyo incondicional y por caminar conmigo en este proceso, hoy no solo es un logro mío, es un logro que compartimos juntos.

Finalmente, a todas las personas que formaron parte de esta increíble etapa, quienes de una u otra manera dejaron huella en este camino.

Alisson Gabriela Chandi Flores

AGRADECIMIENTO

Expreso mi más profundo agradecimiento a Dios, por ser mi guía, mi fortaleza y mi apoyo constante a lo largo de este camino, dándome la sabiduría y la perseverancia necesarias para culminar esta etapa tan importante de mi vida.

A mis padres, por su amor incondicional, su apoyo permanente y por ser el motor que me impulsa a seguir adelante. Gracias por creer en mí incluso en los momentos en los que yo dudaba, por sus sacrificios y por brindarme siempre lo mejor.

A mi tutor y asesora, PhD. Xavier Rosero y PhD. Brizeida Gámez, por su guía, paciencia y conocimientos compartidos durante el desarrollo de este trabajo. A la Universidad Técnica del Norte, por haber sido el espacio donde adquirí los conocimientos, experiencias y valores que hoy forman parte de mi desarrollo profesional.

Finalmente, agradezco a todas las personas que, de una u otra manera formaron parte de este hermoso proceso.

¡Gracias, totales!

Alisson Gabriela Chandi Flores

Resumen

El presente trabajo desarrolla e implementa un sistema de telemetría para un robot móvil de alto torque mediante la integración de tecnología LoRaWAN y el framework ROS 2. El objetivo es habilitar el monitoreo remoto y continuo de variables críticas asociadas al desempeño operativo del sistema robótico. La arquitectura diseñada se fundamenta en la plataforma NVIDIA Jetson Orin Nano como unidad principal de procesamiento y gestión del entorno ROS 2, la cual se encarga de la adquisición y estructuración de los datos. Además, se emplea un microcontrolador que actúa como puente de comunicación serial hacia un módulo LoRa, responsable de la transmisión inalámbrica. La transferencia de información se realiza a través de la infraestructura de The Things Network, esto permite el envío periódico de datos cada 20 segundos hacia un servidor remoto para su procesamiento y almacenamiento. La información recibida es gestionada en una base de datos SQLite y posteriormente es visualizada mediante una interfaz gráfica desarrollada en Python, esto facilita la supervisión en tiempo real, el análisis histórico y la evaluación del comportamiento dinámico del robot. Durante la fase de validación experimental, se verificó el funcionamiento integral del sistema y se comprobó la correcta adquisición, transmisión, recepción y registro de los datos en un entorno controlado. Los resultados demuestran la viabilidad técnica de integrar robótica móvil e Internet de las Cosas en una arquitectura modular y escalable.

Palabras clave: Telemetría, LoRaWAN, ROS 2, robot móvil.

Abstract

This work develops and implements a telemetry system for a high-torque mobile robot through the integration of LoRaWAN technology and the ROS 2 framework. The objective is to enable remote and continuous monitoring of critical variables associated with the operational performance of the robotic system. The proposed architecture is based on the NVIDIA Jetson Orin Nano platform as the main processing unit and ROS 2 management environment, which is responsible for data acquisition and structuring. In addition, a microcontroller is used as a serial communication bridge to a LoRa module, which is responsible for wireless transmission. Data transfer is carried out through The Things Network infrastructure. This allows the periodic transmission of data every 20 seconds to a remote server for processing and storage. The received information is managed in an SQLite database and later visualized through a graphical interface developed in Python. This facilitates real-time monitoring, historical analysis, and evaluation of the robot's dynamic behavior. During the experimental validation phase, the overall system performance was verified, confirming the correct acquisition, transmission, reception, and recording of data in a controlled environment. The results demonstrate the technical feasibility of integrating mobile robotics and the Internet of Things within a modular and scalable architecture.

Keywords: Telemetry, LoRaWAN, ROS 2, mobile robot.

CAPÍTULO I: EL PROBLEMA

1.1. Planteamiento del problema

Los robots móviles de alto torque están diseñados para ejecutar tareas que demandan gran fuerza de tracción, como el transporte y la manipulación de cargas en entornos industriales o logísticos [1]. Su eficacia se incrementa cuando incorporan tecnologías de monitoreo y comunicación que permiten una supervisión continua y la transmisión remota de datos operativos, lo que mejora la seguridad, la eficiencia y la capacidad de respuesta del sistema. En este contexto, la telemetría se configura como una tecnología clave, al posibilitar la recolección, procesamiento y envío de datos desde el robot hacia una estación remota, permitiendo monitorear variables críticas como temperatura, velocidad o voltaje [2].

En la Universidad Técnica del Norte se dispone actualmente de un prototipo funcional de robot móvil de alto torque, desarrollado como parte de proyectos de titulación en 2025. Este equipo ejecuta tareas básicas de navegación y evasión de obstáculos en entornos controlados. Sin embargo, carece de un sistema de telemetría que permita comunicar en tiempo real su estado operativo, lo que limita su uso en escenarios reales que requieren monitoreo remoto, análisis en línea o integración con sistemas distribuidos.

Esta ausencia de telemetría impide conocer el estado interno del robot durante su funcionamiento, lo que representa un riesgo potencial ante fallos no detectados a tiempo y reduce la capacidad de análisis de desempeño en condiciones reales. Si esta limitación no se resuelve, el uso del robot se mantendrá restringido a entornos experimentales, limitando su escalabilidad, autonomía y aplicabilidad práctica en contextos industriales y/o educativos avanzados.

Como solución, se plantea implementar un sistema de telemetría basado en tecnologías de comunicación de largo alcance y bajo consumo energético, integrado dentro de un entorno operativo modular. Esta arquitectura permitirá una comunicación eficiente entre los sensores del robot y una estación remota, facilitando el monitoreo en tiempo real y la gestión distribuida del sistema, con el objetivo de optimizar su funcionalidad y adaptarlo a condiciones reales de operación.

1.2. Objetivos

1.2.1. General

Implementar un sistema de telemetría para un robot móvil de alto torque mediante el uso de LoRaWAN y el framework ROS.

1.2.2. Específicos

- Evaluar las variables críticas del robot móvil de alto torque que requieren monitoreo remoto mediante telemetría.
- Diseñar una arquitectura de comunicación para el sistema de telemetría basada en LoRaWAN, compatible con el entorno operativo ROS.
- Validar el rendimiento del sistema de telemetría.

1.3. Alcance y delimitación

La finalidad de esta investigación es el desarrollo de un sistema de telemetría para un robot móvil de alto torque, utilizando tecnologías de comunicación inalámbrica de largo alcance tales como LoRaWAN, e integrando su funcionamiento dentro del entorno operativo de ROS.

El desarrollo iniciará con la selección e integración de sensores adecuados para el monitoreo de variables importantes del robot, posteriormente, se diseñará la arquitectura de comunicación basada en LoRaWAN, seleccionando módulos que estén disponibles en el mercado y que sean compatibles con el entorno técnico del robot.

Como parte final del alcance, se realizará la validación experimental del sistema de telemetría en campo, evaluando el rendimiento del enlace de datos, la versatilidad de la información transmitida y la autonomía energética del sistema.

1.4. Justificación

El desarrollo de un sistema de telemetría basado en LoRaWAN y ROS para un robot móvil de alto torque representa una contribución relevante para la ingeniería mecatrónica, al integrar tecnologías abiertas, escalables y de bajo costo, adaptadas a contextos locales con limitaciones de infraestructura.

Para el estudiante, el proyecto constituye una oportunidad formativa integral, al aplicar conocimientos en sistemas embebidos y robótica distribuida. Fortalece competencias técnicas importantes para el diseño de soluciones reales y replicables.

Para la comunidad, ofrece una alternativa accesible para monitoreo remoto en entornos sin conectividad convencional, alineándose con necesidades del sector productivo y educativo nacional.

En el ámbito de la ingeniería, promueve la integración de tecnologías como LoRaWAN y ROS 2 en sistemas autónomos, reforzando el desarrollo de soluciones modulares aplicables a robótica móvil, Internet de las Cosas y automatización distribuida.

CAPÍTULO II: MARCO REFERENCIAL

2.1. Antecedentes

Los robots móviles de alto torque se destacan por su capacidad para realizar tareas que requieren fuerza, autonomía y una estructura resistente, es por ello, que, en este tipo de aplicaciones, la incorporación de un sistema de telemetría resulta fundamental, ya que permite que el robot pueda enviar y recibir información a distancia, mejorando su funcionamiento.

A continuación, se presenta una revisión de algunos proyectos que sirven como base para el desarrollo de la investigación, los cuales han aportado ideas, soluciones y también han dejado en evidencia los desafíos que aún existen en el uso de robots móviles con sistemas de comunicación y monitoreo remoto.

Diversas investigaciones respaldan el uso de tecnologías de bajo costo y largo alcance en sistemas de monitoreo remoto. En [3], se desarrolla un sistema de telemetría y teleoperación en tiempo real en un entorno doméstico, demostrando la viabilidad de integrar soluciones accesibles basadas en IoT, aplicables también a plataformas móviles complejas. En [4], se implementa un sistema de coordinación para enjambres de UAV mediante LoRa y ROS, validando su eficacia en entornos sin infraestructura de red convencional, lo cual refuerza su aplicabilidad en robótica terrestre. De forma complementaria, [5] presenta una interfaz de validación remota para un robot hospitalario, confirmando la viabilidad de comunicación en tiempo real mediante arquitecturas cliente-servidor.

El estudio sistemático de [6] evidencia el crecimiento del IoT en sectores productivos, destacando el potencial de tecnologías como LoRaWAN para aplicaciones industriales, mientras que [7] analiza su impacto en la cadena de suministro, reforzando su utilidad en la transmisión de datos a larga distancia. En [8], se demuestra la eficacia de LoRa en una estación meteorológica IoT de bajo costo, alcanzando hasta 1.5 km de cobertura, lo cual valida su empleo en sistemas

distribuidos como los propuestos para robótica móvil.

La integración de ROS 2 en entornos industriales también ha sido explorada. En [9], se diseña una arquitectura centralizada para robots colaborativos con servicios en la nube, destacando la escalabilidad del framework y su compatibilidad con nodos de monitoreo. Asimismo, la tesis de [10] documenta el desarrollo electrónico de un robot móvil de alto torque utilizando ROS 2, proponiendo una base tecnológica directa para la incorporación de telemetría remota.

Trabajos como [11] y [12] desarrollan robots móviles autónomos para aplicaciones sanitarias, integrando visión artificial, sensores IoT y monitoreo remoto a través de plataformas en la nube. En [13] se implementa un sistema de mapeo 3D mediante visión artificial en un robot de alto torque, arquitectura que puede complementarse con telemetría remota para mejorar su operatividad. Estas experiencias refuerzan la factibilidad de sistemas similares aplicados a robots de mayor carga y complejidad.

En el ámbito de la detección ambiental, [14] presenta un sistema de telemetría embebido para incendios, útil como referencia para integrar sensores y módulos de transmisión en robots móviles. Por otro lado, [1] proporciona fundamentos de cinemática, dinámica y navegación para robots con ruedas, esenciales para interpretar variables relevantes a ser monitoreadas. Finalmente, los trabajos [15] y [16] exploran soluciones educativas y mecánicas aplicadas a robots móviles y manipuladores, cuyos principios pueden ser reutilizados para fortalecer el diseño y la integración del sistema propuesto.

Estos antecedentes demuestran que la combinación de LoRaWAN y ROS constituye una alternativa sólida para desarrollar un sistema de telemetría eficiente, modular y adaptable a las condiciones técnicas de un robot móvil de alto torque.

2.2. Marco Teórico

Este capítulo presenta los fundamentos teóricos necesarios para comprender el desarrollo del sistema de telemetría para robots móviles mediante LoRaWAN y ROS, en donde se abordan los conceptos clave relacionados con la adquisición y transmisión de datos en entornos móviles, así como las tecnologías que permiten su implementación eficiente.

2.2.1. Robótica móvil

La robótica combina mecánica, electrónica e informática para crear aparatos capaces de realizar tareas, que, con el tiempo, se ha desarrollado creando así la robótica móvil, que se enfoca en robots que pueden moverse por su entorno, ya sea de forma autónoma o controlada. Estos robots usan sensores para captar información del ambiente y actuadores para moverse o interactuar, asimismo, que pueden desplazarse con ruedas, orugas o patas, según el terreno o la aplicación, estos también se clasifican por su forma de navegación, es decir, algunos siguen rutas fijas y otros se adaptan según lo que detectan a su alrededor [17].

2.2.2. Telemetría

La telemetría es un sistema que permite la medición y transmisión de datos desde un equipo o proceso ubicado a distancia hacia una estación central, donde los datos son visualizados, almacenados o analizados. El término proviene del griego: tele (a distancia) y metron (medida), haciendo referencia a su principal función: medir variables físicas sin estar presente físicamente en el lugar donde ocurren [18].

En robótica, la telemetría se emplea para el monitoreo remoto de variables críticas del sistema, como la posición GPS, longitud, latitud, velocidad promedio del robot e incluso el voltaje interno de sus baterías, permitiendo una supervisión continua y en tiempo real.

Un sistema de telemetría está compuesto por distintos elementos funcionales que permiten la adquisición, transmisión y análisis remoto de datos. A continuación, se describen sus componentes y funciones principales:

Recopilación de datos

El primer paso es la obtención de información sobre variables clave del sistema, como temperatura, velocidad, nivel de voltaje o posición, los cuales permiten conocer con precisión el comportamiento del equipo, facilitando la toma de decisiones técnicas fundamentadas, especialmente en casos donde no es posible una inspección directa [19].

Uso de aparatos externos a los equipos

La recolección de información se realiza a través de transductores o sensores externos, sin necesidad de intervenir directamente en la estructura interna del equipo, asegurando que la

medición no afecte el funcionamiento del sistema original y permite una instalación rápida y segura [19].

Transmisión hacia un módulo emisor

Los datos obtenidos por los sensores son enviados a una unidad emisora que puede estar conectada por cable o funcionar de manera inalámbrica, esta unidad se encarga de codificar los datos y prepararlos para su envío [19].

Levantamiento de datos

Los valores medidos son transformados a unidades comprensibles y estandarizadas, como grados Celsius, kilómetros por hora, voltios o coordenadas GPS, facilitando su interpretación rápida tanto por el sistema como por los operadores remotos [19].

Comunicación a larga distancia

Una vez codificada, la información se transmite por redes inalámbricas como LoRaWAN, redes móviles o enlaces satelitales hacia un centro de monitoreo o estación remota [19].

Retroalimentación inmediata

Los datos recibidos permiten emitir respuestas casi instantáneas, ya sea de forma manual por parte de un operador o a través de órdenes automatizadas programadas en el sistema, este mecanismo de retroalimentación asegura una mayor eficiencia y reduce el riesgo ante eventos inesperados [19].

La Fig. 2.1 representa el funcionamiento general de un sistema de telemetría, en donde un equipo o vehículo equipado con sensores recopila datos operativos en tiempo real, como ubicación, velocidad, entre otros datos los cuales son digitalizados y transmitidos de forma inalámbrica a una central, donde se almacenan, procesan y analizan para facilitar la toma de decisiones. El proceso incluye la captura, transmisión, interpretación y resguardo de la información, permitiendo así una gestión remota eficiente, segura y basada en datos [19].



Fig. 2.1 Interpretación de la telemetría [19].

En robótica móvil distribuida, la telemetría permite además el análisis de desempeño, el diagnóstico temprano de fallos y la optimización energética. Su integración con tecnologías como LoRaWAN y ROS facilita la escalabilidad del sistema y la implementación de soluciones automatizadas más robustas [19].

Por estas razones, la telemetría es considerada una tecnología clave en el desarrollo de robots móviles modernos, no solo como herramienta de monitoreo, sino también como componente estratégico para la autonomía, eficiencia y seguridad del sistema.

2.2.3. Tecnologías

LoRa

LoRa (Long Range) es una tecnología de comunicación inalámbrica de largo alcance que opera en la capa física del modelo OSI (Open Systems Interconnection), esta tecnología fue desarrollada por la empresa francesa Cycleo, y más adelante adquirida por Semtech en el año 2012. Su principal característica es el uso de una modulación de espectro ensanchado (Spread Spectrum), el uso de este tipo de modulación permite una mejor tolerancia al ruido y de esta forma alcanzar largas distancias con un consumo muy bajo de energía, LoRa es el protocolo a Nivel de capa física (Capa OSI Nivel 1) como se muestra en la Fig. 2.2 [20].

Una de las razones por las que LoRa ha ganado relevancia en el ámbito del Internet de las Cosas (IoT) es su capacidad para funcionar en bandas de frecuencia sin licencia. Esta flexibilidad facilita la implementación de soluciones sin necesidad de depender de redes móviles o infraestructura compleja. Como tecnología de capa física, LoRa no se encarga de aspectos relacionados con la estructura de la red, seguridad o direccionamiento; su función principal es garantizar que la señal llegue con la menor pérdida posible y a largas distancias [21].

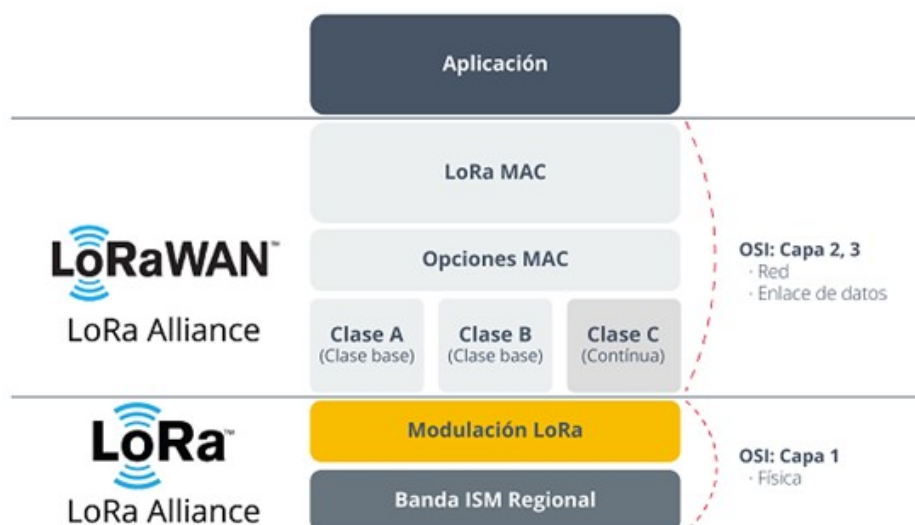


Fig. 2.2 Modelo OSI de capas sobre las que se implementan LoRaWAN y LoRa [20].

LoRaWAN

LoRaWAN (Long Range Wide Area Network) es el protocolo de red que complementa la tecnología física LoRa, y se encarga de la gestión de la comunicación entre dispositivos finales y servidores. Mientras que LoRa opera a nivel físico, LoRaWAN actúa sobre las capas 2 y 3 del modelo OSI como indica la Fig. 2.2, definiendo aspectos como el direccionamiento, autenticación, control de acceso y transporte de los datos. A diferencia de LoRa, LoRaWAN es un protocolo abierto, lo que permite que múltiples fabricantes desarrollen dispositivos compatibles entre sí, contribuyendo así a la reducción de costos y a la escalabilidad de las redes [20].

Además, una red LoRaWAN está compuesta por varios elementos clave como se indica en la Fig. 2.3:

- Nodos o dispositivos finales, que son sensores o actuadores encargados de recopilar o ejecutar información.
- Gateways, que reciben la señal LoRa de los nodos y la envían por Internet al servidor de red.
- Network Server, que administra la red, filtra duplicados y enruta los datos.
- Application Server, que procesa los datos y los pone a disposición de los usuarios o sistemas externos.

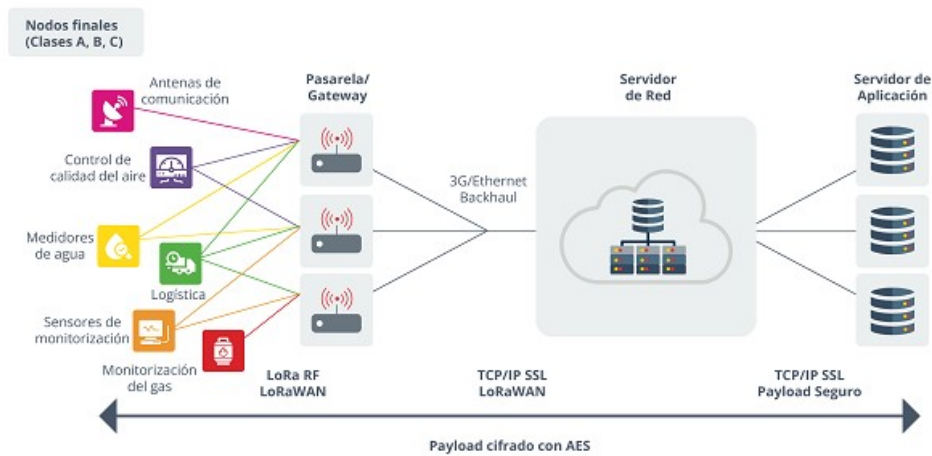


Fig. 2.3 Elementos de la arquitectura LoRaWAN [22].

Los dispositivos LoRaWAN pueden clasificarse en tres clases según su consumo y comportamiento como indica la Fig. 2.4:

- **Clase A:** dispositivos de muy bajo consumo, adecuados para sensores alimentados por batería que necesitan años de autonomía.
- **Clase B:** permiten sincronizar ventanas de recepción para recibir mensajes de forma más programada.
- **Clase C:** siempre están activos, permitiendo comunicaciones más rápidas, pero con un alto consumo energético.



Fig. 2.4 Clases de nodos finales del protocolo LoRaWAN [22].

Serial TTL

Comunicación serial a nivel TTL es una de las tecnologías más empleadas en sistemas electrónicos y embebidos durante las últimas décadas, debido a su simplicidad, confiabilidad y amplia disponibilidad de componentes en el mercado, esta tecnología utiliza niveles lógicos de voltaje típicamente comprendidos entre 0 V y 5 V para representar los estados binarios, permitiendo el intercambio de información entre dispositivos de forma directa y eficiente, asimismo de que su uso extendido facilita la integración de sensores, microcontroladores y módulos de comunicación dentro de sistemas IoT y robóticos [23].

La transmisión de datos mediante comunicación serial TTL se realiza de manera asíncrona, enviando la información bit a bit a través de un puerto serial, entre los parámetros más importantes que deben configurarse correctamente se encuentran la velocidad de transmisión, comúnmente establecida en 9600 baudios, el número de bits de datos por paquete, generalmente 8, los bits de paridad utilizados para detección de errores y los bits de parada que delimitan el inicio y fin de cada trama, adicionalmente, el control de flujo permite asegurar que el receptor procese los datos sin pérdidas, para este tipo de comunicación se emplean los pines TX y RX, los cuales permiten la transmisión y recepción de datos entre dispositivos, como se ilustra en la Fig. 2.5. Es así como, una solución eficiente para integrar señales TTL con redes de largo alcance es el uso de convertidores como el RS485-LB, capaz de adaptar la comunicación serial a la tecnología LoRaWAN [23].

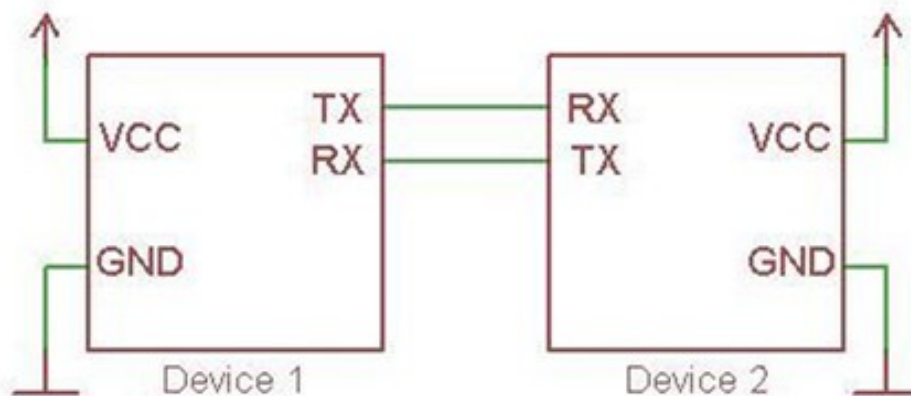


Fig. 2.5 Diagrama de conexiones de 2 dispositivos en comunicación serial asíncrona [24].

2.2.4. Framework ROS

ROS

ROS (Robot Operating System) es una plataforma de desarrollo de software de código abierto, el cual, desde su lanzamiento en 2007, ha sido adoptado globalmente por su capacidad de integrar múltiples funcionalidades, herramientas y bibliotecas en un entorno flexible y accesible para el prototipado y la construcción de soluciones robóticas avanzadas. Gracias a su enfoque modular, ROS permite que diferentes nodos, es decir, procesos independientes colaboren dentro de un mismo sistema, facilitando la integración de sensores, actuadores, algoritmos de navegación y control [25].

ROS 2

ROS 2 es la segunda versión del Sistema Operativo para Robots, desarrollado para abordar algunas de las limitaciones de la primera y ofrecer mejoras en varios aspectos [25], por ejemplo, en lo relativo a la comunicación, compatibilidad con otros sistemas operativos o en cuanto al rendimiento.

Fundamentales avances de ROS 2 frente a ROS 1:

- **Seguridad reforzada:** ROS 2 incorpora mecanismos de autenticación y cifrado para proteger la comunicación entre nodos, lo cual es esencial en aplicaciones sensibles o críticas [23].
- **Compatibilidad multiplataforma:** A diferencia de ROS 1, ROS 2 funciona de forma nativa en Linux, Windows y macOS, lo que amplía las posibilidades de desarrollo y despliegue [23].
- **Comunicación más robusta:** Utiliza Fast RTPS como middleware, basado en el estándar DDS, que mejora la eficiencia, confiabilidad y flexibilidad en la transmisión de datos entre nodos distribuidos [23].
- **Mejor rendimiento y escalabilidad:** La arquitectura de ROS 2 ha sido optimizada para operar en tiempo real y soportar sistemas más complejos y distribuidos [23].
- **Mayor soporte de lenguajes:** Además de C++ y Python, ROS 2 permite trabajar también con otros lenguajes como Rust, ofreciendo más opciones para los desarrolladores [23].

2.2.5. Integración de LoRaWAN con ROS

La integración de tecnologías de comunicación inalámbrica como LoRaWAN con sistemas robóticos basados en ROS permite extender las capacidades perceptivas de los robots más allá del alcance de sus sensores integrados. LoRaWAN es un protocolo diseñado para redes de largo alcance y bajo consumo energético, ideal para aplicaciones distribuidas en entornos amplios o donde se requiere autonomía energética prolongada. Esta arquitectura demuestra cómo ROS puede interoperar con redes IoT mediante protocolos como MQTT y LoRaWAN, facilitando el desarrollo de soluciones inteligentes y escalables para la robótica móvil o social [26]. La modularidad de ROS permite que estos datos sean utilizados por diversos nodos para análisis, decisiones autónomas o generación de alertas.

2.2.6. Variables críticas del robot

Se han definido algunas variables clave que deben ser monitoreadas para asegurar el correcto funcionamiento del robot en campo, estas variables permiten conocer su ubicación en tiempo real y el estado de su energía, lo cual es importante para la autonomía y seguridad del sistema.

Posición GPS

El sistema GPS (Sistema de Posicionamiento Global) es fundamental para conocer la ubicación geográfica del robot en tiempo real, este sistema se basa en una red de satélites que orbitan la Tierra, enviando señales que permiten determinar la posición, velocidad y tiempo del receptor GPS ubicado en el robot [27].

En el trabajo presente, se incorpora un módulo GPS que permite obtener las coordenadas de latitud y longitud, esta información se procesa dentro del sistema ROS y se transmite a través de LoRaWAN para fines de telemetría.

La Tabla 2.1 presenta los principales fundamentos técnicos del sistema GPS, los cuales son esenciales para comprender cómo este sistema permite determinar la posición geográfica del robot.

Tabla 2.1: Fundamentos del GPS [27].

Característica	Descripción
Satélites	El GPS está formado por una red de 24 satélites que giran alrededor de la Tierra a una altitud cercana a los 20.200 km.
Posición	El GPS emplea el principio de trilateración para calcular la ubicación de un receptor en la Tierra.
Efectos	Las señales del GPS pueden verse afectadas por la atmósfera terrestre, especialmente por la ionosfera y la troposfera, mientras viajan desde los satélites hasta los receptores.
Interferencia	La interferencia por trayectos múltiples ocurre cuando las señales GPS llegan al receptor por diferentes rutas, como la línea de visión directa y los reflejos de superficies cercanas, como edificios o el suelo.

Longitud y latitud

La latitud y la longitud son las dos coordenadas geográficas fundamentales utilizadas para ubicar cualquier punto sobre la superficie terrestre, estas coordenadas permiten determinar de manera precisa la posición de un objeto o lugar en el planeta.

La latitud hace referencia a la distancia de un punto respecto al ecuador, una línea imaginaria que divide la Tierra en dos hemisferios: el norte y el sur, esta se mide en grados, desde 0° en el ecuador hasta 90° en los polos, tanto hacia el norte como hacia el sur [28].

Por otro lado, la longitud se refiere a la distancia de un punto respecto al Primer Meridiano, que es una línea imaginaria que atraviesa el Observatorio Real de Greenwich, y se considera el meridiano 0° , la longitud se mide en grados, con un rango que va de 0° a 180° hacia el este y el oeste. A diferencia de la latitud, que es constante en distancia entre grados, la longitud cambia en cuanto más se aleja uno del ecuador, ya que las líneas de longitud convergen a medida que se acercan a los polos [28].

Ambas coordenadas, latitud y longitud, se combinan para formar una ubicación geográfica específica, conocida como coordenadas geográficas, y permiten saber dónde se encuentra el

robot en todo momento, esta información se puede transmitir mediante el protocolo LoRaWAN, que es ideal para enviar datos a largas distancias con bajo consumo de energía.

Velocidad promedio

Esta permite monitorear el rendimiento del robot, optimizar su navegación y generar alertas en tiempo real ante comportamientos anómalos, asimismo, esta información se puede obtener con alta precisión a partir de datos de posicionamiento geográfico, como los que proporciona un módulo GPS. El cálculo de la velocidad promedio se realiza mediante la adquisición de coordenadas geográficas (latitud y longitud) en intervalos regulares, medir la velocidad promedio no solo aporta datos esenciales para el control y monitoreo del robot, sino que también permite detectar condiciones de operación inusuales, identificar zonas de difícil desplazamiento o posibles fallos mecánicos. Además, tiene aplicaciones clave en la gestión de la energía del robot, ya que velocidades más altas suelen implicar mayor consumo [29].

Voltaje de la batería

El monitoreo del estado de la batería es esencial en los sistemas de telemetría, ya que garantiza una operación segura, eficiente y continua del sistema autónomo, asimismo, ayuda a prevenir la pérdida de comunicación, evitar descargas profundas y asegurar que el robot retorne a una estación base o punto de carga antes de agotar su energía, además, permite activar modos de operación de bajo consumo si el voltaje cae por debajo de un umbral crítico [30]. La integración de estos datos en la telemetría proporciona una visión completa del estado operativo del robot, mejorando la confiabilidad del sistema.

Análisis de picos y variaciones dinámicas

En sistemas de telemetría, además del valor instantáneo de las variables monitoreadas, es importante analizar sus valores extremos y su tasa de cambio en el tiempo.

Picos máximos y mínimos de voltaje: Un pico máximo corresponde al valor más alto registrado por la señal de voltaje en un intervalo determinado, mientras que un pico mínimo representa el valor más bajo, el análisis de estos valores permite evaluar la estabilidad del sistema de alimentación, detectar posibles sobretensiones o caídas críticas de energía y prevenir fallas en los componentes electrónicos [31].

Derivada de variables dinámicas: La derivada de una variable respecto al tiempo representa su tasa de cambio, en sistemas móviles, este análisis permite evaluar el comportamiento dinámico del sistema, el análisis de los valores máximos de estas derivadas permite identificar cambios bruscos en la trayectoria o en la dinámica de movimiento, contribuyendo a la evaluación del desempeño y estabilidad del sistema [32].

2.2.7. Servidor propietario

El servidor de red es el componente encargado de procesar, gestionar y almacenar los datos transmitidos por los gateways LoRaWAN, asegurando que la información recibida desde los nodos finales sea accesible, segura y útil para el monitoreo remoto del sistema, este elemento cumple un rol central dentro de la arquitectura LoRaWAN, ya que coordina la comunicación entre los dispositivos finales y las aplicaciones, simplificando la administración de redes de sensores distribuidos [33].

The Things Network

Entre las plataformas disponibles, The Things Network (TTN) constituye una solución eficiente y ampliamente utilizada para la gestión de redes LoRaWAN, TTN opera como servidor de red y servidor de aplicaciones, permitiendo la comunicación eficiente entre los nodos sensores, los gateways y los sistemas externos encargados del análisis y visualización de los datos como se explica en la Tabla 2.2 [34].

Tabla 2.2: Ventajas de usar TTN como servidor propietario [33].

Ventajas	Descripción
Red abierta y colaborativa	TTN es una plataforma de código abierto con una comunidad global que permite compartir infraestructura LoRaWAN y conocimientos técnicos.
Acceso a una red global de gateways	Existen miles de gateways comunitarios conectados a TTN alrededor del mundo, lo que proporciona una cobertura significativa.
Costos reducidos	La plataforma no requiere pago de suscripción para el uso básico del servidor de red LoRaWAN, reduciendo costos de despliegue.
Escalabilidad y flexibilidad	TTN está diseñado para crecer con el proyecto, permitiendo agregar gateways o dispositivos nuevos sin una infraestructura cerrada.
Soporte y recursos de la comunidad	Una comunidad activa ofrece documentación, foros y experiencias compartidas, facilitando la resolución de problemas y la implementación.
Integración con protocolos estándar	TTN permite integrar datos con servicios externos mediante webhooks, MQTT/HTTP, y APIs abiertas.

2.2.8. Sistema operativo

Los sistemas operativos permiten la gestión de recursos, la ejecución de procesos y la integración de múltiples herramientas de software, es común el uso de sistemas operativos basados en Linux gracias a su estabilidad, flexibilidad y compatibilidad con frameworks como ROS, ya que estos sistemas permiten la ejecución de nodos, servidores de comunicación y aplicaciones de monitoreo en tiempo real [35].

Ubuntu Linux

Ubuntu Linux es una distribución de código abierto muy utilizada, principalmente por ser el sistema operativo oficialmente soportado por ROS y ROS 2, su arquitectura permite una gestión eficiente de procesos, comunicación entre nodos y acceso a herramientas de desarrollo avanzadas [35].

2.2.9. Almacenamiento de datos en la nube

El almacenamiento de datos en la nube permite guardar grandes volúmenes de información de forma segura y accesible desde cualquier ubicación, en aplicaciones robóticas, el uso de servicios en la nube facilita el monitoreo remoto, el análisis histórico de datos y la escalabilidad del sistema ya que estas plataformas permiten centralizar la información proveniente de múltiples dispositivos, garantizando disponibilidad, respaldo y facilidad de integración con aplicaciones web o sistemas de análisis [36].

2.2.10. Microcontrolador en el sistema de telemetría

Los microcontroladores constituyen el núcleo de los sistemas de telemetría, ya que son los encargados de adquirir, procesar y transmitir los datos provenientes de los sensores instalados o por instalar dentro del robot móvil, estos dispositivos integran en un circuito la unidad de procesamiento, memoria y periféricos de entrada y salida, lo que permite una implementación compacta y eficiente en aplicaciones embebidas. Los microcontroladores cumplen funciones críticas como la lectura de sensores, el procesamiento básico de datos, la comunicación con módulos inalámbricos y la interacción con sistemas de mayor nivel, como estaciones base o frameworks robóticos [37].

Con el fin de identificar las características técnicas de los microcontroladores comúnmente empleados en sistemas de telemetría y robótica móvil, se presenta la Tabla 2.3, en la cual se comparan el Arduino Uno, Arduino Nano y el ESP32, considerando parámetros relevantes como capacidad de procesamiento, voltaje de operación, entradas y salidas, memoria y facilidad de uso, permitiendo contextualizar la selección del microcontrolador dentro del diseño del sistema propuesto [38].

Tabla 2.3: Tipos de microcontroladores y sus características [38].

Característica	Arduino Uno	Arduino Nano	Esp32
Microcontrolador	ATmega328P	ATmega328P	Tensilica Xtensa LX6
Frecuencia de reloj	16 MHz	16MHz	Hasta 240 MHz
Voltaje de operación	5V	5V	3.3 V
Entradas / Salidas digitales	14 (6PWM)	14 (6PWM)	Hasta 34
Entradas analógicas	6	8	Hasta 18
Memoria flash	32 KB	32 KB	Hasta 4 MB
Facilidad de uso	Alta	Alta	Media - Alta
Consumo energético	Bajo	Bajo	Medio

El Arduino Uno destaca por su facilidad de uso, robustez y comodidad para pruebas en laboratorio, lo que lo hace ideal para etapas de desarrollo y validación.

El Arduino Nano, aunque ofrece prestaciones similares, está orientado a aplicaciones donde el espacio es limitado.

Por su parte, el ESP32 presenta mayores capacidades de procesamiento y conectividad inalámbrica integrada, siendo adecuado para aplicaciones IoT, aunque con mayor complejidad de configuración y consumo energético.

CAPÍTULO III: MARCO METODOLÓGICO

3.1. Modelo de la investigación

El presente trabajo de integración curricular adopta un enfoque ingenieril, y una investigación aplicada ya que busca resolver un problema concreto [39], el cual es la ausencia de un sistema de telemetría en un robot móvil de alto torque, mediante el diseño e implementación de una solución tecnológica basada en LoRaWAN y ROS.

Complementariamente, se aplica una metodología de tipo documental, basada en la recopilación, análisis y organización de información proveniente de fuentes como artículos científicos, tesis, manuales técnicos y documentación oficial [40], de tecnologías como LoRaWAN y ROS, permitiendo establecer una base teórica sólida que respalde las decisiones técnicas del trabajo, de igual manera que facilite la selección de componentes y aporte referentes válidos para el diseño del sistema de telemetría.

Además, se incorpora una investigación de campo, dado que se realiza una observación directa del entorno de operación del robot móvil y se recopilan datos sobre sus requerimientos, funcionamiento y condiciones reales de uso [39]. Finalmente, el trabajo también es de tipo descriptivo, ya que se enfoca en detallar las características, componentes, arquitectura y funcionamiento del sistema de telemetría. Este tipo de investigación permite representar de manera clara y ordenada las soluciones tecnológicas desarrolladas, facilitando su comprensión y evaluación [30].

3.2. Diseño de la investigación

El desarrollo del sistema de telemetría se constituye en tres fases metodológicas: evaluación de variables, diseño del sistema y validación de este. Cada fase inspecciona actividades específicas orientadas al cumplimiento de los objetivos planteados, las cuales se registran y documentan de manera sistemática para garantizar su seguimiento y validación.

3.2.1. Fase 1: Evaluación de las variables críticas

Esta fase tiene como finalidad analizar el estado actual del sistema del robot y establecer cuáles son las variables internas que deben ser monitoreadas de forma remota para garantizar un funcionamiento seguro y eficiente.

Actividad 1.1: Identificación de los sensores ya integrados en el robot

Se realiza la revisión de la arquitectura electrónica existente del robot para determinar qué sensores están actualmente instalados, permitiendo establecer un punto de partida para el diseño del sistema de telemetría.

Actividad 1.2: Determinación de nuevas variables a medir

Se analiza el funcionamiento del robot para identificar qué variables nuevas requieren monitoreo.

Actividad 1.3: Investigación del uso de LoRaWAN y ROS para sistemas de telemetría en robots móviles

Se realiza una revisión bibliográfica y técnica para comprender cómo se ha implementado la comunicación LoRaWAN y el framework ROS en proyectos similares, esta investigación servirá como base teórica para el diseño del sistema.

3.2.2. Fase 2: Diseño de la arquitectura para el sistema

En esta fase se establece la solución técnica para el sistema de telemetría, seleccionando los componentes necesarios, además, se define el flujo de datos entre los dispositivos y se diseñan los nodos ROS que permitirán integrar la lectura de sensores con la transmisión de datos vía LoRaWAN.

Actividad 2.1: Definición de los componentes de la red LoRaWAN

Se seleccionan los módulos emisores y receptores de LoRaWAN que serán integrados al sistema, considerando compatibilidad, alcance, consumo energético y disponibilidad comercial.

Actividad 2.2: Selección de materiales, controladores y sensores

Se eligen los materiales ya sean controladores o sensores adecuados para el sistema de telemetría, considerando sus características, compatibilidad con LoRaWAN y ROS, precisión de medición y consumo energético, con el fin de asegurar un funcionamiento confiable y eficiente del sistema.

Actividad 2.3: Configuración de microcontroladores para capturar datos de sensores

Se programan microcontroladores para adquirir los datos de los sensores seleccionados y prepararlos para su transmisión a través del módulo LoRa.

Actividad 2.4: Verificación de funcionamiento mediante la placa auxiliar.

Se comprueba el correcto funcionamiento de los sensores y módulos antes de su integración final en el robot, por lo cual se utiliza una placa auxiliar que permite realizar pruebas individuales de lectura, transmisión y recepción de datos, garantizando que cada componente opere de forma estable y dentro de los parámetros esperados para posteriormente ser implementados en el robot.

Actividad 2.5: Comprobación de la correcta lectura y envío de datos de los sensores.

Se verifica que los sensores del sistema realicen lecturas apropiadas y que los datos sean enviados correctamente a través del módulo LoRaWAN hacia la red TTN, además se analiza la ausencia de errores en la recepción, asegurando una comunicación confiable entre los dispositivos y la plataforma de monitoreo.

Actividad 2.6: Configuración del decodificador de datos en TTN.

Se configura el decodificador en la plataforma The Things Network para interpretar correctamente los datos enviados por los nodos LoRa, es así que se define el formato del payload y se programan las funciones necesarias para convertir la información binaria en valores reales, permitiendo su correcta visualización y análisis en el servidor o sistema ROS.

Actividad 2.7: Acceso y monitoreo del sistema mediante RealVNC Viewer.

La verificación del funcionamiento de los webhooks configurados en The Things Network se realizó mediante el acceso remoto proporcionado por RealVNC Viewer, permitiendo observar en tiempo real la ejecución de los procesos encargados de la recepción y reenvío de los datos hacia Firebase.

Actividad 2.8: Pruebas de comportamiento en entorno simulado.

Se evalúa el funcionamiento del sistema de telemetría en un entorno simulado antes de su implementación real, para analizar el rendimiento, la estabilidad de comunicación y la respuesta del sistema ante posibles fallos o variaciones en las variables medidas.

Actividad 2.9: Creación de página web dentro del servidor apache.

Para la visualización remota de los datos de telemetría, se implementa una página web alojada en el servidor Apache, la cual funciona como una interfaz ligera que permite acceder a la información procesada por el sistema, mostrando de manera estructurada las variables recibidas desde la red LoRaWAN.

Actividad 2.10: Envío de datos dentro del servidor.

Se realiza mediante una estructura de comunicación que permite recibir, procesar y almacenar la información proveniente del nodo LoRaWAN, una vez que TTN transmite los paquetes decodificados, el servidor los recibe mediante un webhook, habilitando su integración con servicios locales como Apache, bases de datos o scripts en Python, además durante este proceso, los datos son organizados y enviados hacia los módulos responsables de visualización, análisis o almacenamiento permanente.

Actividad 2.11: Programación de nodos ROS para recibir, interpretar y publicar los datos.

Se desarrolla los nodos de ROS encargados de recibir los datos provenientes de TTN, interpretarlos según el formato establecido y publicarlos en los tópicos correspondientes, de esta manera, la información de los sensores se integra al sistema robótico, permitiendo su visualización y procesamiento.

3.2.3. Fase 3: Validación del sistema

Esta última fase tiene como objetivo comprobar el funcionamiento y desempeño del sistema de telemetría en condiciones reales, en donde, se realizan pruebas de campo para evaluar la estabilidad del enlace de comunicación, el alcance efectivo y la calidad de los datos transmitidos.

Actividad 3.1: Instalación de sensores y módulos LoRaWAN

Se integran físicamente los sensores seleccionados al robot, así como los módulos de comunicación LoRaWAN, asegurando su correcta conexión, protección y ubicación.

Actividad 3.2: Diagrama de la arquitectura final del sistema de comunicación.

Se elabora el diagrama que representa la arquitectura final del sistema de comunicación, en el cual se muestran los componentes principales como son los nodos sensores, microcontrolador, módulo LoRa, gateway, red TTN y servidor ROS.

Actividad 3.3: Medición de cobertura, estabilidad de datos y tiempos de transmisión

Se realizan pruebas de campo para evaluar cuánta distancia puede cubrir el sistema sin pérdida de señal, cuán estables son los datos recibidos y qué retardo existe entre la medición y su recepción en la estación.

Actividad 3.4: Análisis de resultados y planteamiento de parámetros de comunicación

Se procesan los datos obtenidos en las pruebas para detectar posibles mejoras y, a partir de los resultados, se ajustan parámetros.

CAPÍTULO IV: RESULTADOS

Este capítulo presenta el diseño e implementación del sistema de telemetría desarrollado para el robot móvil de alto torque, destacando la integración de los dispositivos y protocolos necesarios para lograr una transmisión eficiente y confiable de datos operativos. Para la adquisición de información, se emplea un módulo GPS encargado de registrar las coordenadas geográficas del robot, específicamente latitud y longitud, así como su velocidad promedio de desplazamiento. Estos datos son recibidos inicialmente por un microcomputador, donde se ejecutan aplicaciones desarrolladas en Python para la interpretación y procesamiento de las tramas de información. Posteriormente, los datos procesados son enviados mediante comunicación serial hacia un microcontrolador, el cual se encarga de estructurar la información y preparar el payload para su transmisión. Seguidamente, la información es transmitida a través de un módulo LoRaWAN RS485-LB hacia la red LoRaWAN, permitiendo establecer un enlace inalámbrico de largo alcance y bajo consumo energético. Una vez recibidos los paquetes de datos, el gateway los reenvía al servidor propietario The Things Network (TTN), donde son decodificados y gestionados mediante webhooks. Finalmente, los datos son transferidos hacia un servidor puente donde son almacenados en una base de datos MySQL. Posteriormente, mediante un script desarrollado en Python, la información es sincronizada con Firebase Realtime Database y consultada desde el computador local para su visualización y análisis en tiempo real.

4.1. Arquitectura del sistema de telemetría

La arquitectura del sistema de telemetría se estructura como un flujo distribuido que conecta los sensores del robot con el entorno operativo ROS mediante comunicación LoRaWAN. En este esquema, el módulo GPS instalado en el robot registra variables como la latitud, longitud y velocidad de desplazamiento, y transmite esta información hacia el microcomputador del sistema.

En el microcomputador se ejecutan aplicaciones desarrolladas en Python encargadas de in-

interpretar y procesar los datos recibidos, posteriormente, la información procesada es enviada mediante comunicación serial hacia un microcontrolador, el cual organiza los datos y genera el payload que será transmitido a través del módulo LoRaWAN RS485-LB.

Este módulo se encarga de enviar la información hacia un gateway LoRaWAN cercano, garantizando un enlace inalámbrico de largo alcance y bajo consumo energético. En la Fig. 4.1 se presenta la arquitectura general del sistema de telemetría propuesto.

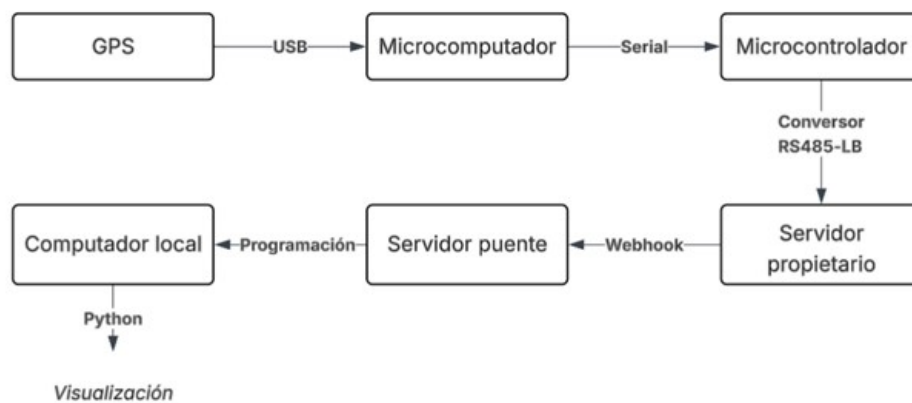


Fig. 4.1 Arquitectura del sistema de telemetría.

La arquitectura final del sistema de telemetría integra los procesos de adquisición, procesamiento, transmisión y visualización de los datos generados por el robot móvil de alto torque, su funcionamiento se basa en una arquitectura distribuida en la que intervienen el receptor GPS, el microcomputador, el microcontrolador, el servidor propietario, el servidor puente y el computador local.

El proceso inicia con la adquisición de datos de posicionamiento mediante el módulo GPS, el cual proporciona información de latitud y longitud del robot, estos datos son recibidos a través de una interfaz USB por el microcomputador, donde se ejecutan aplicaciones desarrolladas en Python encargadas de interpretar las tramas GPS, validar la información y calcular variables derivadas, como la velocidad promedio del robot.

Posteriormente, el microcomputador transmite los datos procesados al microcontrolador mediante comunicación serial, este dispositivo actúa como unidad de enlace, recibiendo la información estructurada y preparándola para su envío hacia el módulo de comunicación LoRaWAN. Adicionalmente, el microcontrolador gestiona el control del convertidor RS485-LB mediante el envío de comandos AT, los cuales permiten configurar el dispositivo, establecer los parámetros

de red y disparar la transmisión de los datos.

Una vez recibido el comando de transmisión desde el microcontrolador, el conversor RS485-LB encapsula la información en un payload LoRaWAN y la transmite de forma inalámbrica hacia el gateway más cercano. Posteriormente, el gateway reenvía los datos al servidor propietario The Things Network (TTN), donde estos son recibidos, decodificados y gestionados conforme a la configuración establecida para el dispositivo.

Finalmente, los datos provenientes del servidor propietario son enviados hacia un servidor puente, el cual permite recibir, procesar y almacenar la información en una base de datos, desde este servidor, el computador local accede a los datos mediante aplicaciones desarrolladas en Python, permitiendo su consulta, almacenamiento y visualización en tiempo real para el monitoreo de las variables de telemetría registradas por el robot móvil.

4.1.1. Módulo GPS

El Módulo GPS GY-GPS6MV2 se integra al sistema de telemetría con el fin de obtener la posición geográfica del robot, ya este proporciona coordenadas de longitud y latitud con alta estabilidad, lo que permite registrar el recorrido del robot y calcular variables derivadas como su velocidad promedio. En la Fig. 4.2 se presenta el sensor y en la Tabla 4.1 las especificaciones del mismo [41].



Fig. 4.2 GY-GPS6MV2.

Tabla 4.1: Características del GY-GPS6MV2 [41].

Característica	Descripción
Conectividad	UART (Serial TTL)
Tipo de mapa	Global
Voltaje	3.3 V – 5 V
Corriente	45 mA
Precisión de posicionamiento	± 2.5 m
Velocidad en baudios	9600 bps
Dimensiones del producto	25 mm \times 35 mm \times 6 mm

Configuración de pines del módulo GY-GPS6MV2

El módulo tiene cuatro pines de salida, como se muestra en la Fig. 4.3, la alimentación del módulo y la interfaz de comunicación se realizan a través de estos cuatro pines como se explica en la Tabla 4.2.

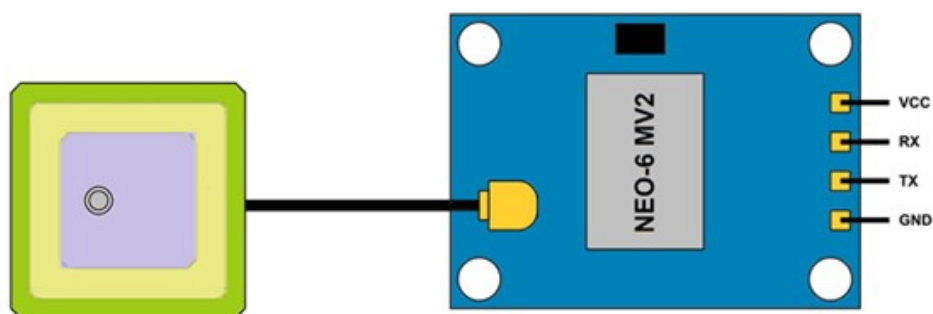


Fig. 4.3 Pines del módulo GY-GPS6MV2 [41], [23].

Tabla 4.2: Configuración de pines [41].

Pin del módulo	Descripción
VCC	Pin de alimentación positivo
RX	Pin de recepción UART
TX	Pin de transmission UART
GND	Tierra

4.1.2. Microcomputador

La plataforma de procesamiento o microcomputador utilizada en el sistema de telemetría es la NVIDIA Jetson Orin Nano, la cual actúa como estación central encargada de ejecutar los servicios de comunicación, procesamiento y visualización de los datos transmitidos por el robot móvil de alto torque, el dispositivo ejecuta el sistema operativo Ubuntu 22.04.5 LTS, sobre el cual se instalan y configuran las herramientas necesarias para la operación del sistema, incluyendo ROS 2, aplicaciones desarrolladas en Python para el procesamiento de datos y una base de datos local SQLite para el almacenamiento interno del sistema.

La Jetson Orin Nano recibe los datos provenientes del módulo GPS mediante conexión USB, donde se ejecutan aplicaciones desarrolladas en Python encargadas de interpretar las tramas NMEA, validar la información y calcular variables derivadas como la velocidad promedio del robot. La Jetson permite la ejecución simultánea de múltiples procesos, garantizando un funcionamiento estable y continuo del sistema de telemetría, permitiendo la ejecución de nodos ROS en tiempo real y el acceso remoto al sistema mediante herramientas de escritorio remoto [42].

En la Tabla 4.3 se presentan las principales especificaciones técnicas de la NVIDIA Jetson Orin Nano, las cuales justifican su selección como plataforma central de procesamiento del sistema

Tabla 4.3: Especificaciones técnicas de la NVIDIA Jetson Orin Nano [42].

Especificación	Descripción
Módulo	NVIDIA Jetson Orin Nano 8 GB
CPU	6-core Arm Cortex-A78AE v8.2 64-bit, 1.5 MB L2 + 4 MB L3
GPU	NVIDIA Ampere con 1024 CUDA cores y 32 Tensor cores
Memoria	8 GB LPDDR5 128-bit, 68 GB/s
Almacenamiento	Ranura microSD; soporte NVMe externo
Consumo energético	7 W a 15 W
Interfaces USB	4× USB 3.2 Gen2; 1× USB-C
Conectividad	1× Gigabit Ethernet
Cámara	2× MIPI CSI-2 22-pin
Display	DisplayPort 1.2
Expansión	Header de 40 pines
Dimensiones	100 mm × 79 mm × 21 mm

Instalación de ROS 2

Para la implementación del sistema de telemetría y la integración con el robot móvil de alto torque, se realiza la instalación y configuración del entorno de trabajo basado en ROS 2 Humble Hawksbill, el cual se ejecuta sobre el sistema operativo Ubuntu 22.04.5 LTS, debido a su compatibilidad nativa y soporte a largo plazo.

El proceso de instalación inicia con la verificación de la versión del sistema operativo mediante comandos del sistema, con el fin de asegurar que se cumpla con los requisitos estable-

cidos por ROS 2, y, en caso de no contar con la versión adecuada de Ubuntu, se procede a la instalación del sistema operativo correspondiente.

Una vez confirmada la versión del sistema operativo, se realiza la actualización del sistema y la instalación de dependencias básicas necesarias para la correcta descarga e instalación de ROS 2, posteriormente, se añade la clave de seguridad y el repositorio oficial de ROS 2, lo que permite acceder a los paquetes oficiales y mantener el sistema actualizado.

Con el repositorio configurado, se procede a la instalación de la distribución ROS 2 Humble Desktop, la cual incluye las herramientas fundamentales para el desarrollo, visualización y ejecución de nodos robóticos, finalizada la instalación, se configura el entorno de trabajo mediante la carga del archivo de configuración correspondiente, permitiendo el acceso a los comandos y paquetes de ROS 2 desde la terminal.

Adicionalmente, se evalúa la configuración permanente del entorno, incorporando la carga automática de ROS 2 en el archivo de inicialización del sistema, lo que facilita su uso en futuras sesiones, también se crea un espacio de trabajo dedicado, donde se compilan y gestionan los paquetes necesarios para el desarrollo del sistema de telemetría.

En la Fig. 4.4 se presenta el diagrama de flujo correspondiente al proceso de instalación y configuración del entorno ROS 2 utilizado en el desarrollo del sistema.

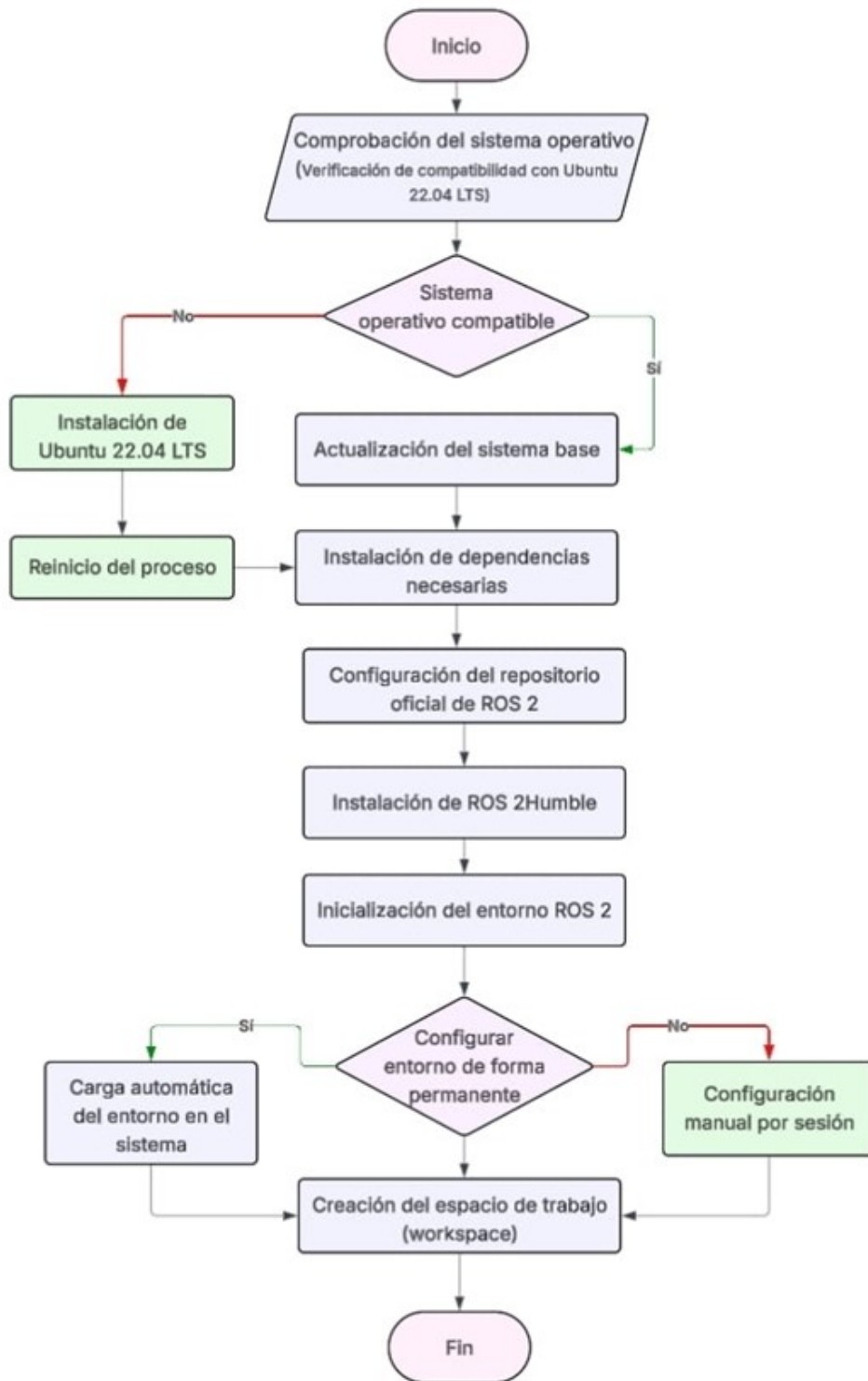


Fig. 4.4 Diagrama de flujo instalación ROS2.

4.1.3. Microcontrolador

El microcontrolador utilizado en el sistema de telemetría es el Arduino Uno, el cual cumple la función de unidad de procesamiento y enlace de comunicación de los datos generados por los sensores instalados en el robot móvil de alto torque, este dispositivo se encarga de recibir la información procesada por el microcomputador mediante comunicación serial, interpretar las tramas de datos, extraer las variables de interés y preparar la información para su posterior transmisión a través del módulo LoRaWAN.

El Arduino Uno opera como intermediario entre el microcomputador y el conversor RS485-LB, permitiendo la correcta estructuración del payload que será enviado hacia la red LoRaWAN., y, para ello, el microcontrolador ejecuta rutinas de lectura serial, validación de datos y formateo de la información, asegurando que las variables transmitidas cumplan con el esquema de codificación definido en el servidor The Things Network.

Asimismo, el microcontrolador gestiona los tiempos de muestreo y transmisión de los datos, estableciendo intervalos de envío de 20 segundos del sistema de telemetría, esta gestión permite mantener un equilibrio entre la frecuencia de actualización de la información y la eficiencia energética del sistema.

La elección del Arduino Uno se justifica por su facilidad de programación, estabilidad de operación y compatibilidad con múltiples sensores y módulos de comunicación, lo que facilita la integración con el sistema LoRaWAN y permite futuras ampliaciones del sistema de telemetría sin modificaciones significativas en la arquitectura general. En la Tabla 4.4 se muestran las especificaciones del microcontrolador.

Tabla 4.4: Información general del microcontrolador Arduino UNO [43].

Especificación Técnica	Descripción
Módulo	NATmega328P
Voltaje de operación	5 V
Entradas / Salidas digitales	14
Entradas analógicas	6
Corriente continua por pin	20 mA
Corriente CC para 3.3 V por pin	50 mA
Frecuencia de reloj	16 MHz
Comunicación	UART, I2C, SPI

Algoritmo de procesamiento y transmisión de datos

El microcontrolador Arduino Uno ejecuta un algoritmo autónomo encargado de la adquisición, procesamiento y preparación de las variables de telemetría antes de su transmisión mediante el módulo RS485-LB, en el cual, el sistema opera bajo un ciclo repetitivo, en el que se realiza la lectura de datos GPS recibidos por comunicación serial, el cálculo del voltaje de la batería mediante lectura analógica ADC, la detección de picos máximos y mínimos de voltaje, así como el cálculo de las máximas derivadas positivas de latitud y velocidad dentro de una ventana temporal de 10 segundos.

Posteriormente, el microcontrolador responde a solicitudes del módulo LoRaWAN enviando las variables correspondientes según el comando que el mismo recibe, además, se implementa un disparo automático de transmisión cada 20 segundos para garantizar la periodicidad del envío de datos. En la Fig. 4.5 se presenta el diagrama de flujo del algoritmo implementado en el microcontrolador.

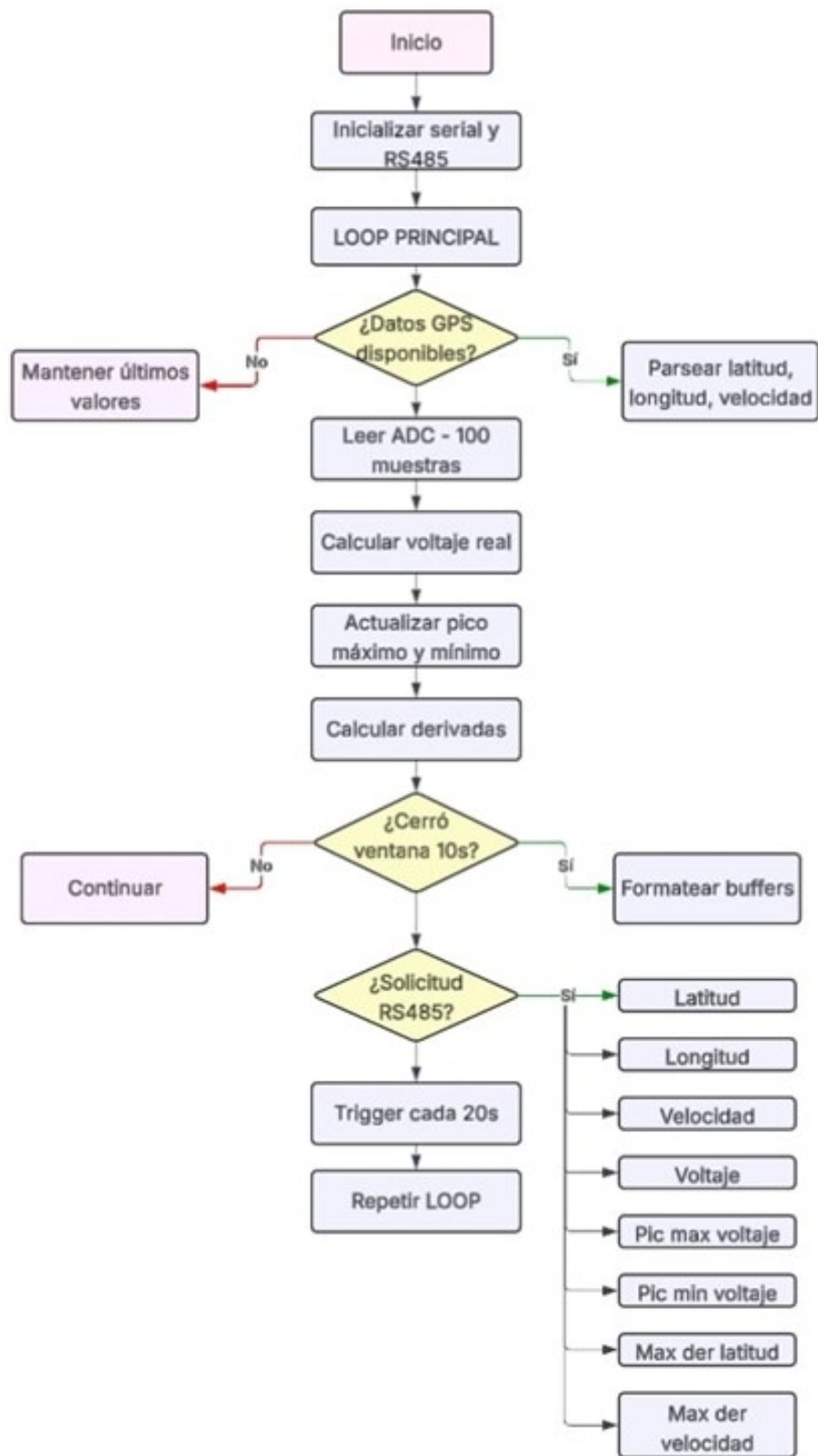


Fig. 4.5 Algoritmo del microcontrolador.

Sistema de medición de voltaje de baterías

El robot móvil emplea cuatro baterías de plomo-ácido de 12 V y 9 Ah configuradas en serie-paralelo para alimentar el sistema de potencia. Debido a que el Arduino Uno opera con un rango máximo de 5 V en sus entradas analógicas, se implementa un divisor resistivo de voltaje con resistencias $R1 = 82 \text{ k}\Omega$ y $R2 = 10 \text{ k}\Omega$, adicionalmente, se incorpora un diodo Zener de 5.1 V de protección para evitar daños por sobrevoltaje o polaridad inversa y permitiendo así reducir la tensión total del banco de baterías a un nivel seguro para el ADC del microcontrolador.

El voltaje real se calcula mediante la relación del divisor resistivo y el promedio de 100 muestras analógicas, lo que permite obtener una medición estable y reducir el ruido eléctrico del sistema, en la Fig. 4.6 se observa el divisor de voltaje realizado.

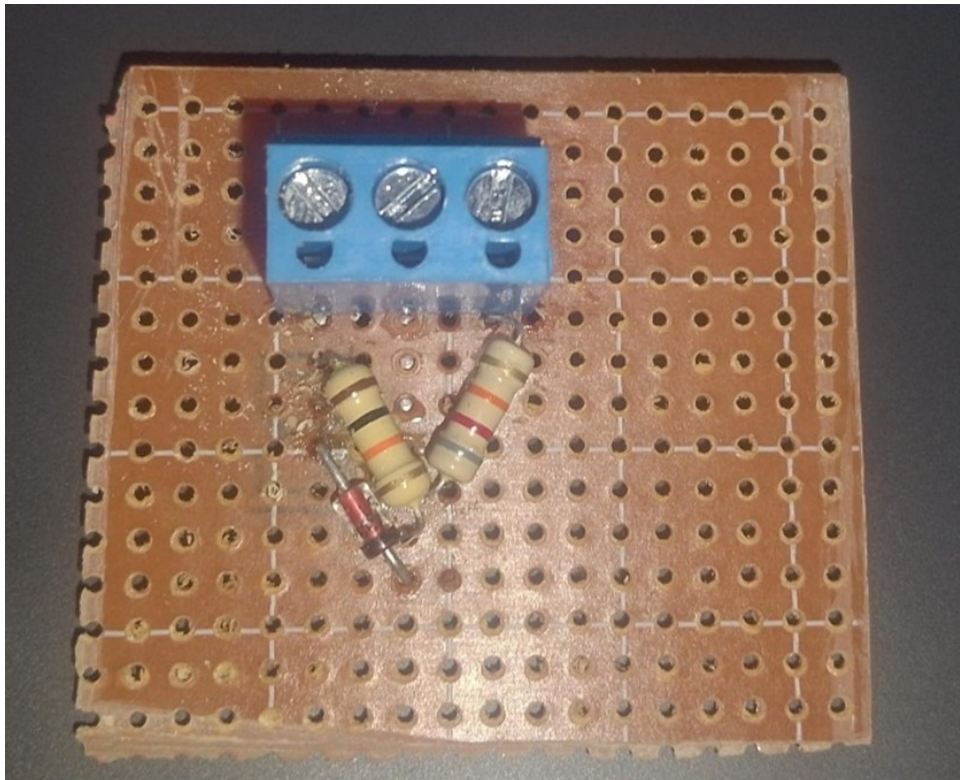


Fig. 4.6 Divisor de voltaje.

4.1.4. Conversor RS485-LB

El Dragino RS485-LB es un dispositivo encargado de convertir señales RS485 o UART a paquetes LoRaWAN, permitiendo integrar sensores cableados al sistema de telemetría mediante un enlace inalámbrico de largo alcance y bajo consumo energético, este módulo opera como un

nodo Clase A dentro de la red LoRaWAN, capaz de adquirir datos desde dispositivos externos, estructurarlos como payload y transmitirlos hacia un gateway para su posterior procesamiento en el servidor de red [23].

El RS485-LB admite sensores con interfaces RS485, UART de 3.3/5 V y señales de interrupción, entre sus ventajas se destacan su carcasa con protección IP67, su batería interna diseñada para operación prolongada, la posibilidad de configuración mediante Bluetooth Low Energy y la compatibilidad con actualizaciones OTA (Over The Air). Estas características permiten su instalación en exteriores o zonas de difícil acceso, garantizando la transmisión periódica y confiable de los datos hacia la infraestructura LoRaWAN. En la Fig. 4.7 se observa la presentación del módulo utilizado [23].

Además, su capacidad para funcionar como puente entre sensores cableados y redes LoRaWAN permite integrar dispositivos que no poseen comunicación inalámbrica nativa, ampliando la flexibilidad del sistema de telemetría. Gracias a ello, el módulo facilita la expansión futura del robot sin necesidad de modificar la arquitectura principal, por ello, el RS485-LB es un componente clave dentro del nodo final, asegurando la transmisión eficiente y segura de las variables obtenidas por los sensores hacia el ecosistema de monitoreo remoto. En la Fig. 4.7 se observan los pines que tiene el sensor.



Fig. 4.7 Mapeo de pines del RS 485 LB de Dragino [44].

Comandos AT

Los comandos AT son instrucciones en formato de texto utilizadas para configurar y controlar el funcionamiento interno del módulo RS485-LB, estos comandos se envían mediante una

interfaz serial y permiten modificar parámetros esenciales del dispositivo, consultar su estado operativo, ajustar modos de comunicación o definir el comportamiento del nodo dentro de la red LoRaWAN.

Para la configuración del RS485-LB se utilizó un conversor TTL–USB y una terminal serial, desde donde se enviaron los comandos necesarios para habilitar la transmisión LoRaWAN, registrar el dispositivo, definir el intervalo de envío y establecer el formato del payload. En la Tabla 4.5 se presentan los principales comandos utilizados durante el proceso de configuración.

Tabla 4.5: Comandos AT.

Comando AT	Descripción
ATZ	Reinicio del módulo
AT+FDR	Restablecer valores predeterminados de fábrica
AT+CFG	Mostrar configuración completa
AT+DEUI	Mostrar/configurar DevEUI
AT+APPEUI	Mostrar/configurar AppEUI
AT+APPKEY	Configurar AppKey

4.1.5. Servidor propietario

Como servidor propietario se utiliza The Things Network (TTN), dado que tiene una red pública, global y colaborativa basada en el protocolo LoRaWAN, es así que, se lleva a cabo el registro y autenticación tanto del gateway como del nodo final, garantizando su incorporación correcta a la red LoRaWAN. Para ello, es necesario crear una aplicación dentro de TTN, donde se gestiona el entorno operativo del dispositivo y se habilita el uso de webhooks para transmitir los datos hacia servidores externos, esta configuración es esencial para asegurar el funcionamiento completo del sistema IoT, permitiendo el flujo continuo de información desde los sensores hasta la plataforma de procesamiento, en la Tabla. 4.6 se observa la información general del RS485-LB en TTN.

Tabla 4.6: Información general del RS485-LB.

Comando AT	Descripción
End device ID	rs485-lb-robot
Frequency plan	United States 902-928 MHz, FSB 2 (used by TTN)
LoRaWAN version	LoRaWAN Specification 1.0.3
Regional Parameters version	RP001 Regional Parameters 1.0.3 revision A
Created at	Sep 19, 2025 13:51:27

4.1.6. Servidor puente

El servidor puente corresponde a un servidor web local encargado de recibir los datos transmitidos desde la plataforma TTN mediante el mecanismo de webhook, este servidor actúa como intermediario entre la red LoRaWAN y el sistema de almacenamiento.

Para su implementación se utilizó un entorno basado en Apache, configurado con una dirección IP local que permite la recepción de solicitudes HTTP enviadas automáticamente por TTN cada vez que se recibe un paquete de datos desde el nodo LoRaWAN.

Una vez recibida la información, los datos son procesados mediante scripts desarrollados en Python, los cuales se encargan de interpretar el contenido del mensaje y almacenarlo en una base de datos MySQL. Posteriormente, estos datos pueden ser consultados por el computador local y utilizados para su visualización en la interfaz gráfica desarrollada para el sistema de monitoreo.

Adicionalmente, se utiliza RealVNC como herramienta de acceso remoto para la gestión de una máquina virtual destinada a la supervisión y control del sistema de telemetría, su implementación permite establecer una conexión segura y estable con la instancia configurada para el procesamiento y monitoreo de datos, facilitando la administración remota del entorno de trabajo sin necesidad de acceso físico al equipo.

En la Tabla 4.7 se presentan las principales características técnicas del servidor puente RealVNC

Tabla 4.7: Características del servidor puente [45].

Especificación	Descripción
Software	RealVNC
Tipo de herramienta	Software de escritorio remoto
Protocolo	VNC (Virtual Network Computing)
Sistema operativo compatible	Ubuntu 22.04 LTS, Windows, macOS
Tipo de conexión	Red local (LAN) o Internet
Seguridad	Autenticación por usuario y contraseña; cifrado de extremo a extremo
Nombre	Xibernetiq-U55

Configuración de la base de datos MySQL – servidor web

Para el almacenamiento estructurado de la información recibida desde la red LoRaWAN, se instala el sistema gestor de bases de datos MySQL en la máquina virtual configurada con Ubuntu 22.04.5 LTS. Previo a la instalación, se verifica la configuración del firewall UFW, habilitando los puertos correspondientes al servicio HTTP y HTTPS, garantizando un entorno seguro de comunicación como se muestra en la Fig. 4.8.

```
xibernetiq@xibernetiq-U55:~$ sudo ufw status
[sudo] password for xibernetiq:
Status: active

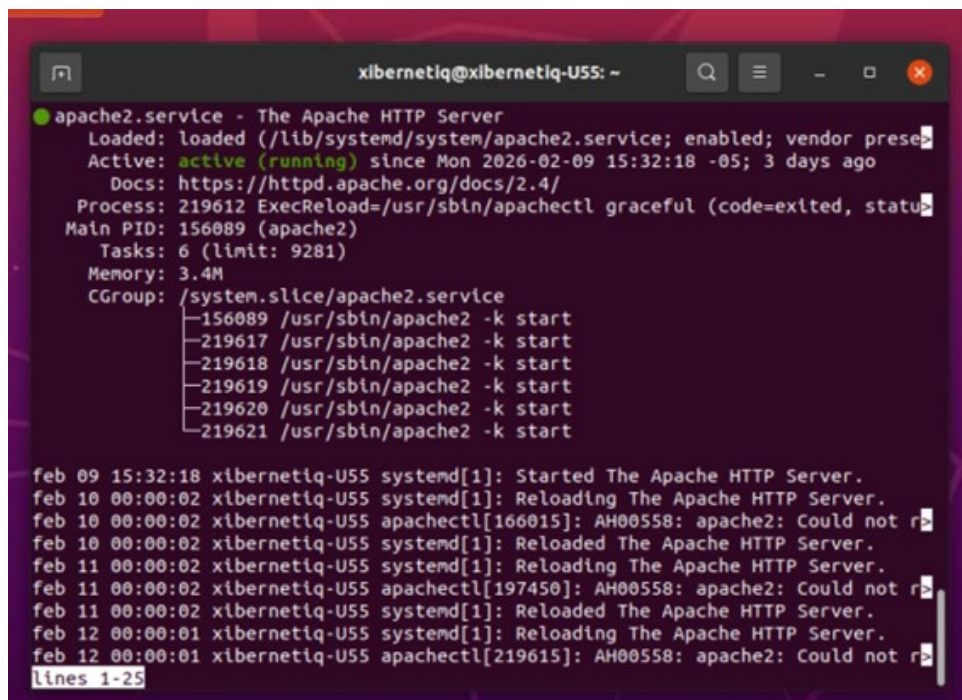
To Action From
--
22/tcp ALLOW Anywhere
Apache ALLOW Anywhere
80/tcp ALLOW Anywhere
443/tcp ALLOW Anywhere
80 ALLOW Anywhere
443 ALLOW Anywhere
22/tcp (v6) ALLOW Anywhere (v6)
Apache (v6) ALLOW Anywhere (v6)
80/tcp (v6) ALLOW Anywhere (v6)
443/tcp (v6) ALLOW Anywhere (v6)
80 (v6) ALLOW Anywhere (v6)
443 (v6) ALLOW Anywhere (v6)

xibernetiq@xibernetiq-U55:~$
```

Fig. 4.8 Política de Firewall.

Apache

En la consola de la máquina virtual se realizó la instalación del servidor web Apache HTTP Server, con el propósito de habilitar servicios de comunicación HTTP dentro del sistema de telemetría, este servidor permite gestionar solicitudes entrantes y establecer un canal de intercambio de datos entre aplicaciones externas, el Apache HTTP Server fue instalado sobre el sistema operativo Ubuntu 22.04.5 LTS configurado en la máquina virtual, utilizando el gestor de paquetes del sistema como se muestra en la Fig. 4.9



```
xibernetiq@xibernetiq-U55: ~  
● apache2.service - The Apache HTTP Server  
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese  
   Active: active (running) since Mon 2026-02-09 15:32:18 -05; 3 days ago  
     Docs: https://httpd.apache.org/docs/2.4/  
   Process: 219612 ExecReload=/usr/sbin/apachectl graceful (code=exited, statu  
   Main PID: 156089 (apache2)  
     Tasks: 6 (limit: 9281)  
    Memory: 3.4M  
   CGroup: /system.slice/apache2.service  
           └─156089 /usr/sbin/apache2 -k start  
             └─219617 /usr/sbin/apache2 -k start  
               └─219618 /usr/sbin/apache2 -k start  
                 └─219619 /usr/sbin/apache2 -k start  
                   └─219620 /usr/sbin/apache2 -k start  
                     └─219621 /usr/sbin/apache2 -k start  
  
feb 09 15:32:18 xibernetiq-U55 systemd[1]: Started The Apache HTTP Server.  
feb 10 00:00:02 xibernetiq-U55 systemd[1]: Reloading The Apache HTTP Server.  
feb 10 00:00:02 xibernetiq-U55 apachectl[166015]: AH00558: apache2: Could not r  
feb 10 00:00:02 xibernetiq-U55 systemd[1]: Reloaded The Apache HTTP Server.  
feb 11 00:00:02 xibernetiq-U55 systemd[1]: Reloading The Apache HTTP Server.  
feb 11 00:00:02 xibernetiq-U55 apachectl[197450]: AH00558: apache2: Could not r  
feb 11 00:00:02 xibernetiq-U55 systemd[1]: Reloaded The Apache HTTP Server.  
feb 12 00:00:01 xibernetiq-U55 systemd[1]: Reloading The Apache HTTP Server.  
feb 12 00:00:01 xibernetiq-U55 apachectl[219615]: AH00558: apache2: Could not r  
lines 1-25
```

Fig. 4.9 Servidor Apache.

Instalación y configuración de PHP 8.0

Para la implementación de los servicios web asociados al sistema de telemetría, se realiza la instalación del lenguaje de programación PHP en su versión 8.0, debido a su compatibilidad con las versiones actuales de librerías y sistemas gestores de bases de datos, las versiones de PHP desde la 8.0 en adelante requieren la incorporación de un repositorio adicional en Ubuntu para su correcta instalación.

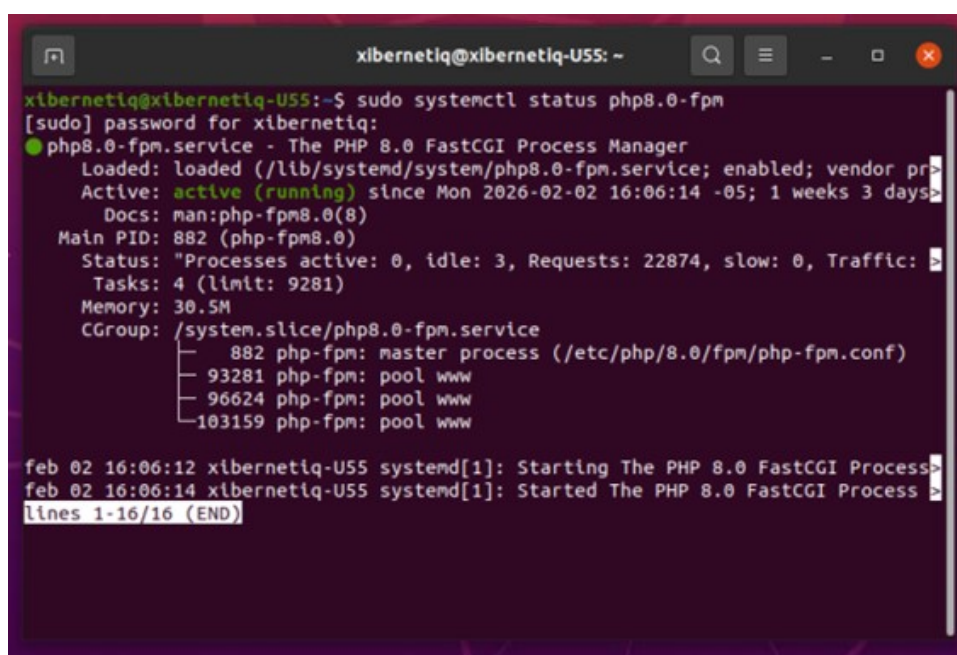
Se opta por utilizar una versión actualizada de PHP con el objetivo de evitar conflictos de dependencias con librerías modernas, especialmente aquellas utilizadas para la gestión de bases

de datos y servicios web.

Aunque algunos módulos básicos se instalan por defecto, fue necesario incluir módulos adicionales requeridos por la aplicación desarrollada: : php8.0-mysql y php8.0-json.

El módulo php8.0-mysql permite la conexión y manipulación de bases de datos MySQL, mientras que php8.0-json habilita el procesamiento y la codificación de datos en formato JSON, ampliamente utilizado en servicios web y APIs [46].

Finalmente, se verifica la correcta instalación y funcionamiento del servicio mediante la consulta del estado del sistema desde la consola, confirmando que el servicio se encuentra activo y en ejecución. En la Fig. 4.10 se muestra la salida correspondiente a la verificación del servicio PHP-FPM.



```
xibernetiq@xibernetiq-U55: ~  
xibernetiq@xibernetiq-U55:~$ sudo systemctl status php8.0-fpm  
[sudo] password for xibernetiq:  
● php8.0-fpm.service - The PHP 8.0 FastCGI Process Manager  
  Loaded: loaded (/lib/systemd/system/php8.0-fpm.service; enabled; vendor pr  
  Active: active (running) since Mon 2026-02-02 16:06:14 -05; 1 weeks 3 days  
    Docs: man:php-fpm8.0(8)  
  Main PID: 882 (php-fpm8.0)  
  Status: "Processes active: 0, idle: 3, Requests: 22874, slow: 0, Traffic: 0"  
    Tasks: 4 (limit: 9281)  
  Memory: 30.5M  
  CGroup: /system.slice/php8.0-fpm.service  
          └─ 882 php-fpm: master process (/etc/php/8.0/fpm/php-fpm.conf)  
             └─ 93281 php-fpm: pool www  
                └─ 96624 php-fpm: pool www  
                   └─ 103159 php-fpm: pool www  
  
feb 02 16:06:12 xibernetiq-U55 systemd[1]: Starting The PHP 8.0 FastCGI Process  
feb 02 16:06:14 xibernetiq-U55 systemd[1]: Started The PHP 8.0 FastCGI Process  
lines 1-16/16 (END)
```

Fig. 4.10 Estado del servicio PHP 8.0-FPM.

Procesamiento y almacenamiento de datos JSON

El servidor The Things Network transmite la información en formato JSON, el cual constituye un estándar ampliamente utilizado para la transferencia de datos estructurados a través de Internet, para la recepción y gestión de estos datos, se desarrolla un script PHP denominado insert_dht.php, cuya función principal es interpretar el contenido recibido desde el webhook

configurado en TTN y almacenarlo en la base de datos del sistema, en la Fig. 4.11 se puede observar el proceso para el mismo.

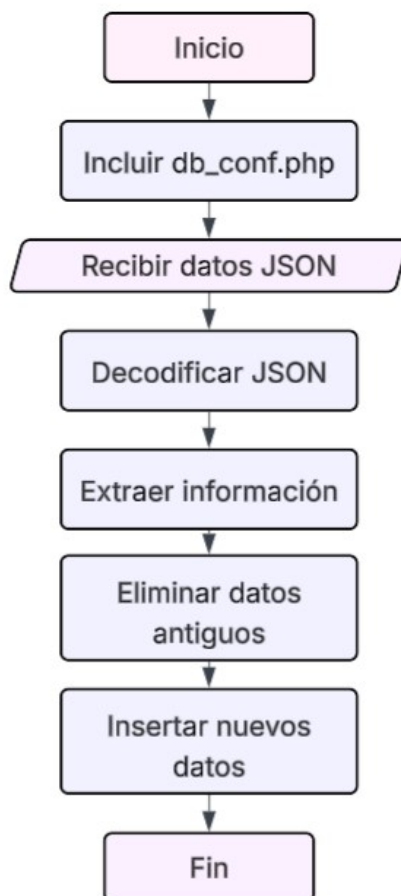


Fig. 4.11 Diagrama de flujo del proceso de recepción y almacenamiento de datos en MySQL.

4.1.7. Computador local

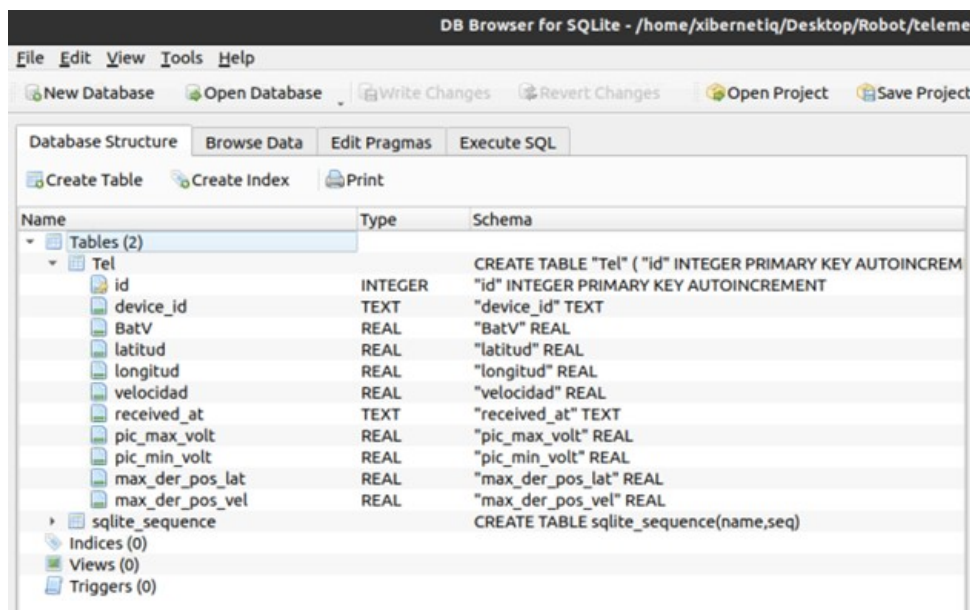
Plataforma de almacenamiento y visualización

Una vez que los datos son recibidos y procesados por el servidor puente, estos son almacenados en la base de datos MySQL. Posteriormente, un script desarrollado en Python procesa los datos para almacenarlos simultáneamente en SQLite y sincronizarlos con Firebase Realtime Database, permitiendo su acceso remoto y visualización en tiempo real desde el computador local.

SQ Lite – Almacenamiento local

El sistema de telemetría implementado incorpora una aplicación desarrollada en Python que permite la gestión y almacenamiento local de los datos recibidos desde la red LoRaWAN, integrando el servidor de red con una base de datos SQLite, el propósito principal de esta aplicación es recibir la información transmitida por el robot móvil, procesarla y almacenarla de forma persistente.

El programa inicia estableciendo una conexión con una base de datos SQLite local, dentro de la base de datos se define una tabla denominada Tel, diseñada para almacenar las variables de telemetría transmitidas por el robot, la estructura de esta tabla incluye un identificador único incremental (id), el identificador del dispositivo (device_id), el voltaje de la batería (BatV), las coordenadas geográficas de latitud y longitud (latitud y longitud), la velocidad promedio (velocidad) y una marca de tiempo (received_at), garantizando así un registro completo y ordenado de cada transmisión recibida, como se observa en la Fig. 4.12 en la cual se presenta la estructura de la base de datos SQLite utilizada en el sistema de telemetría, donde se observa la definición de la tabla Tel y los campos empleados para el almacenamiento de las variables operativas del robot móvil.



The screenshot shows the 'DB Browser for SQLite' interface. The 'Database Structure' tab is active, displaying the schema for a database. Under 'Tables (2)', the 'Tel' table is expanded, showing its columns and their data types. The 'CREATE TABLE' statement for 'Tel' is visible in the right pane.

Name	Type	Schema
id	INTEGER	"id" INTEGER PRIMARY KEY AUTOINCREMENT
device_id	TEXT	"device_id" TEXT
BatV	REAL	"BatV" REAL
latitud	REAL	"latitud" REAL
longitud	REAL	"longitud" REAL
velocidad	REAL	"velocidad" REAL
received_at	TEXT	"received_at" TEXT
pic_max_volt	REAL	"pic_max_volt" REAL
pic_min_volt	REAL	"pic_min_volt" REAL
max_der_pos_lat	REAL	"max_der_pos_lat" REAL
max_der_pos_vel	REAL	"max_der_pos_vel" REAL
sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)

Fig. 4.12 Estructura de la base de datos SQLite utilizada para el almacenamiento de datos de telemetría del robot móvil.

La aplicación se encarga de recibir los datos decodificados provenientes del servidor The Things Network, los cuales son procesados para asegurar su correcta interpretación antes de

ser almacenados, el almacenamiento local mediante SQLite asegura la permanencia de la información incluso ante pérdidas de conectividad a internet, proporcionando una solución robusta para el registro de datos de telemetría, de igual manera, esta base de datos facilita la consulta y visualización de las variables medidas, permitiendo su integración con herramientas de análisis, visualización gráfica y nodos desarrollados en ROS.

Sincronización con Firebase

Para complementar el almacenamiento local y centralizado del sistema, se implementó una base de datos en tiempo real utilizando Firebase Realtime Database, la cual permite la visualización remota y actualización instantánea de las variables de telemetría del robot móvil de alto torque.

La utilización de una base de datos en la nube responde a la necesidad de contar con acceso remoto a la información desde cualquier ubicación, facilitando el monitoreo continuo del sistema sin depender exclusivamente del almacenamiento local en MySQL y SQLite.

La configuración de la base de datos en tiempo real en Firebase Realtime Database se lleva a cabo mediante la creación de un nodo estructurado específicamente para almacenar las variables principales empleadas en el monitoreo del sistema, el cual se actualiza de manera automática cada vez que los dispositivos finales realizan una nueva transmisión de datos. La Fig. 4.13 presenta la estructura implementada para esta configuración.

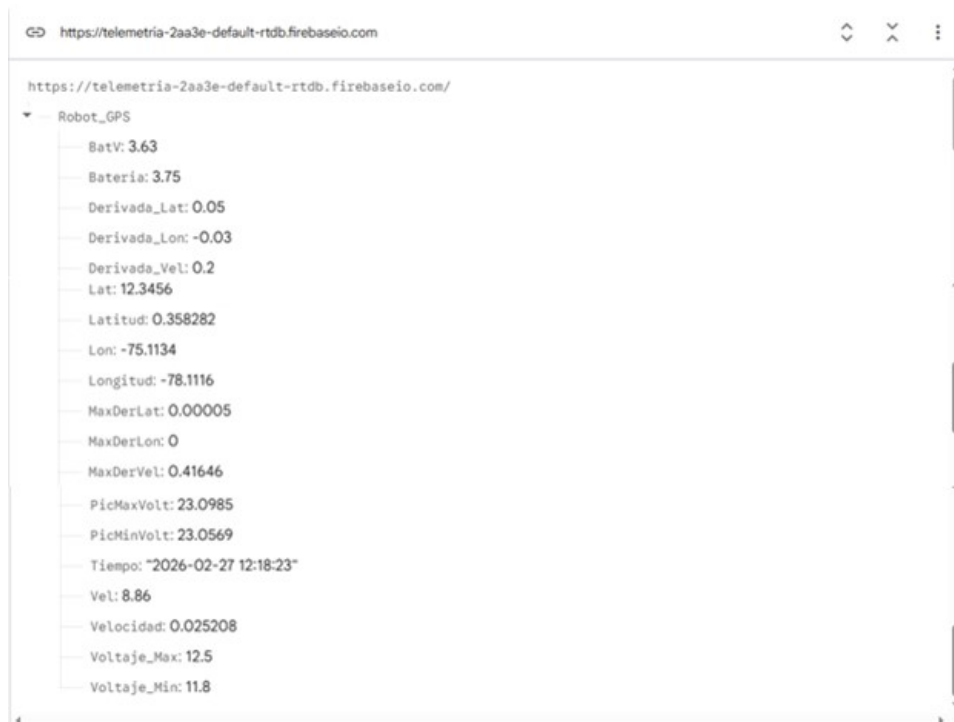


Fig. 4.13 Estructura de las variables en Firebase.

Con el objetivo de garantizar la protección de la información almacenada en Firebase Real-time Database, el acceso externo a los datos se encuentra restringido, para habilitar actualizaciones controladas y seguras, se configuran permisos específicos que regulan las operaciones de lectura y escritura de las variables, estas autorizaciones se definen mediante reglas de seguridad establecidas en la configuración de la base de datos en tiempo real, tal como se muestra en la Fig. 4.14

```
1 {
2   "rules": {
3     ".read": true,
4     ".write": true
5   }
6 }
```

Fig. 4.14 Reglas para lectura y escritura de variables en Firebase.

La arquitectura implementada establece que, una vez almacenados los datos en la base de datos MySQL del servidor, un script desarrollado en Python actúa como puente de comunicación entre el servidor local y Firebase. En primer lugar, el script accede a la base de datos MySQL para consultar los registros más recientes generados por los sensores, seleccionando únicamente los registros más recientes para asegurar la actualización en tiempo real. Posteriormente, los registros obtenidos son procesados y almacenados en una base de datos local SQLite, lo que permite mantener un respaldo de la información en el computador local y facilitar su gestión dentro del sistema. Finalmente, estos datos son sincronizados con Firebase Realtime Database, donde quedan disponibles para su visualización y análisis en tiempo real desde aplicaciones de monitoreo. El flujo completo de este proceso se ilustra en la Fig. 4.15.



Fig. 4.15 Flujo de datos desde MySQL a Firebase mediante un script de Python.

Interfaz gráfica para monitoreo en tiempo real

El sistema de monitoreo fue desarrollado en Python e integra tres niveles de almacenamiento y visualización. En primer lugar, el script principal establece conexión con la base de datos MySQL, desde donde consulta el registro más reciente generado por el dispositivo identificado como rs485-lb-robot, cuando se detecta un nuevo dato, este es almacenado localmente en una

base de datos SQLite, lo que permite mantener un respaldo interno en la máquina virtual, posteriormente, el mismo registro es enviado a Firebase Realtime Database mediante una solicitud de actualización al nodo correspondiente, permitiendo así la sincronización en tiempo real con la nube.

La interfaz gráfica es implementada utilizando Tkinter para la estructura visual y Matplotlib para la representación de las variables, la aplicación permite visualizar individualmente parámetros como latitud, longitud, velocidad y variables derivadas como picos máximos del voltaje, picos mínimos del voltaje, máxima derivada positiva de latitud y máxima derivada positiva de velocidad, además de ofrecer una opción para mostrar todas las gráficas simultáneamente como indica la Fig. 4.16



Fig. 4.16 Interfaz gráfica del sistema de telemetría.

Para optimizar el rendimiento y mantener claridad en la visualización, el sistema limita la cantidad de registros mostrados a los datos más recientes almacenados en SQLite, garantizando así un monitoreo dinámico y actualizado del robot móvil. En la Fig. 4.17 se muestra el diagrama de flujo del monitoreo y actualización de datos que se realizó.

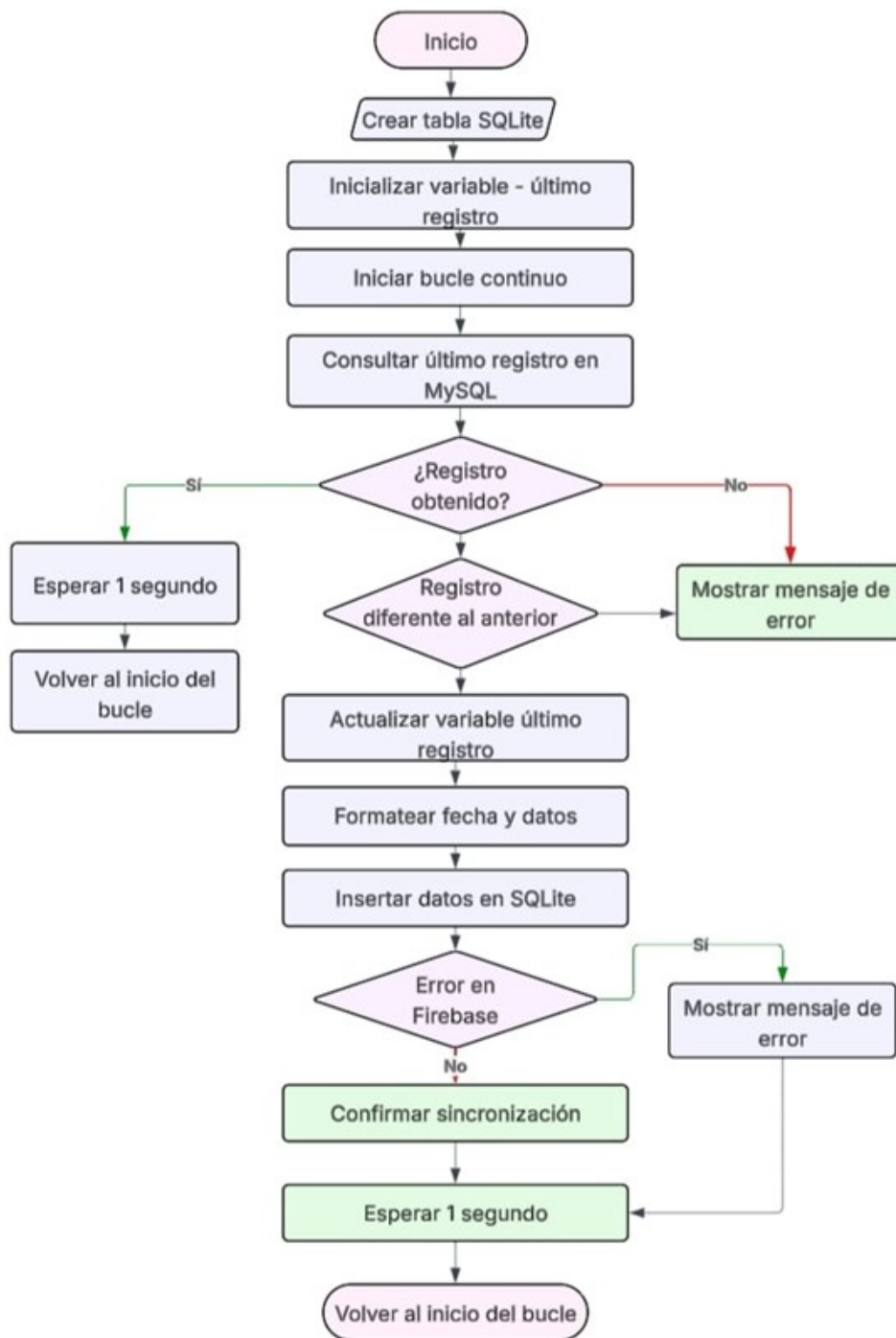
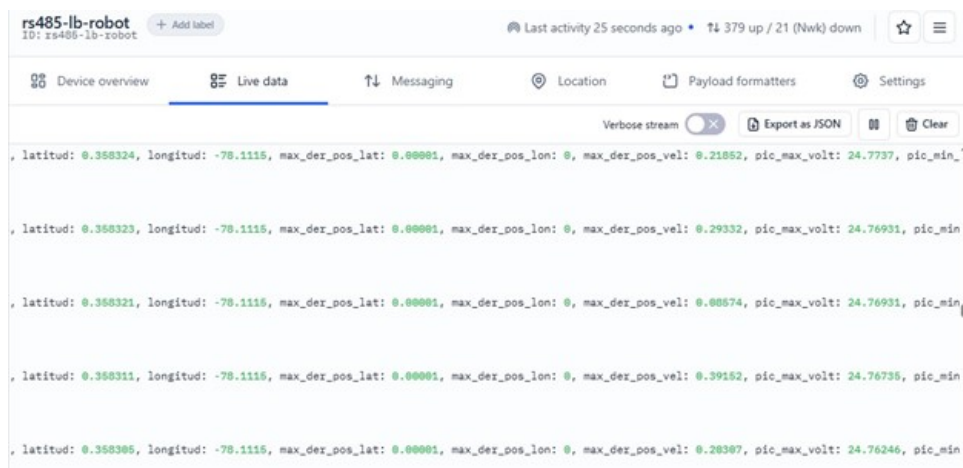


Fig. 4.17 Diagrama de flujo del monitoreo y actualización de datos.

4.2. Pruebas de funcionamiento

Para garantizar el correcto funcionamiento del sistema de telemetría, se realiza una verificación integral de cada uno de los módulos que lo componen, así como de su interacción dentro de la arquitectura general, este proceso inició con la comprobación de la adquisición de datos desde el módulo GPS, seguido por la validación de su transmisión mediante la red LoRaWAN hacia The Things Network.

Con el fin de verificar la correcta recepción de los datos, se empleó la herramienta Live Data de la plataforma The Things Network, la cual permitió observar en tiempo real la llegada periódica de los paquetes transmitidos por los nodos finales, esta verificación permitió confirmar la integridad de los datos, la correcta decodificación del payload y el cumplimiento del intervalo de transmisión configurado, la Fig. 4.18 presenta la salida de la consola Live Data mencionada anteriormente.



```
rs485-lb-robot + Add label Last activity 25 seconds ago • 379 up / 21 (Nwk) down ☆ ☰
Device overview Live data Messaging Location Payload formatters Settings
Verbose stream Export as JSON Clear
, latitud: 0.358324, longitud: -78.1115, max_der_pos_lat: 0.00001, max_der_pos_lon: 0, max_der_pos_vel: 0.21852, pic_max_volt: 24.7737, pic_min_
, latitud: 0.358323, longitud: -78.1115, max_der_pos_lat: 0.00001, max_der_pos_lon: 0, max_der_pos_vel: 0.29332, pic_max_volt: 24.76931, pic_min
, latitud: 0.358321, longitud: -78.1115, max_der_pos_lat: 0.00001, max_der_pos_lon: 0, max_der_pos_vel: 0.08574, pic_max_volt: 24.76931, pic_min
, latitud: 0.358311, longitud: -78.1115, max_der_pos_lat: 0.00001, max_der_pos_lon: 0, max_der_pos_vel: 0.39152, pic_max_volt: 24.76735, pic_min
, latitud: 0.358305, longitud: -78.1115, max_der_pos_lat: 0.00001, max_der_pos_lon: 0, max_der_pos_vel: 0.28307, pic_max_volt: 24.76246, pic_min
```

Fig. 4.18 Live data TTN.

Adicionalmente, se valida el correcto desempeño del script puente encargado de transferir los datos desde MySQL almacenarlos en SQLite y sincronizarlos hacia Firebase como indica la Fig. 4.19, durante su ejecución, se monitorean los mensajes de salida en consola, confirmando la lectura, procesamiento y actualización exitosa de los registros en la base de datos en la nube.

```
xibernetiq@xibernetiq-U55: ~/Desktop/Robot/telemetria
xibernetiq@xibernetiq-U55:~/Desktop/Robot/telemetria$ python3 envio.py
Insertado en SQLite: BatV=3.75, Lat=12.3456, Lon=-75.1134
Sincronizado con Firebase: {'BatV': 3.75, 'Longitud': -75.1134, 'MaxDerLat': 0.1, 'MaxDerVel': 0.5, 'Latitud': 12.3456, 'Velocidad': 8.86, 'MaxDerLon': -0.88, 'PicMinVolt': 21.8, 'PicMaxVolt': 22.5, 'Tiempo': '2026-02-10 14:58:40'}
```

Fig. 4.19 Ejecución del script puente y verificación de transferencia de datos.

Finalmente, se realiza un análisis del rendimiento general del sistema, considerando la latencia de transmisión desde el nodo final hasta la visualización en la interfaz gráfica, este análisis contempla el tiempo de envío, procesamiento, almacenamiento y actualización en tiempo real, permitiendo evaluar la eficiencia y confiabilidad del sistema bajo condiciones normales de operación. Las gráficas generadas reflejan el comportamiento dinámico de las variables monitoreadas y validan la capacidad del sistema para operar en tiempo real y se observan en la Fig. 4.20.

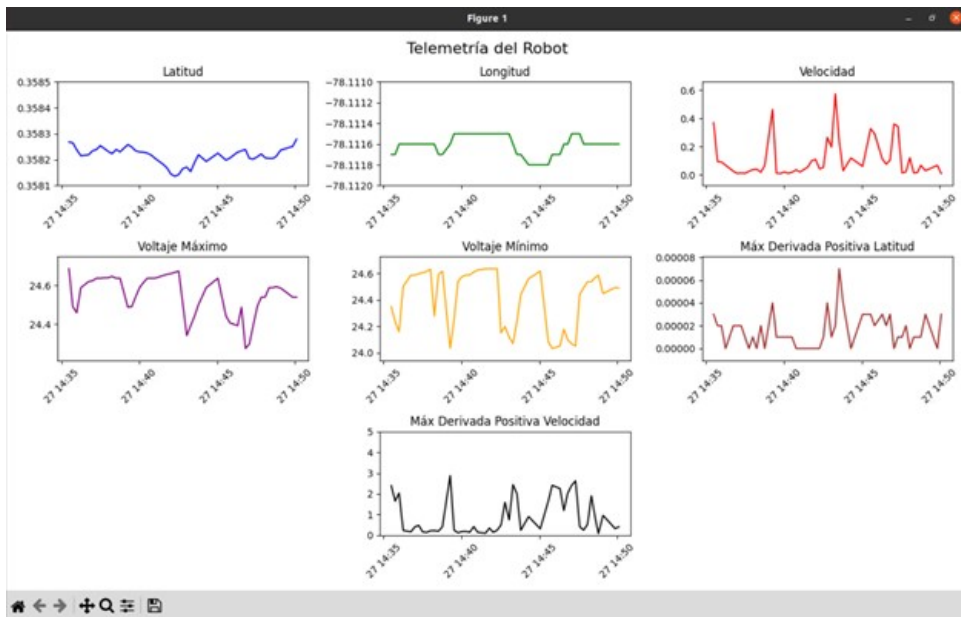


Fig. 4.20 Variables monitoreadas.

La gráfica de latitud muestra la variación de la posición geográfica del robot en el eje norte-sur durante el intervalo de tiempo analizado. Se observa que los valores presentan ligeras fluctuaciones alrededor de un punto medio, lo cual indica pequeños desplazamientos del robot dentro del área de prueba.

La gráfica de longitud representa el desplazamiento del robot en el eje este–oeste. Los valores registrados muestran cambios graduales a lo largo del tiempo, reflejando el movimiento del robot durante la prueba experimental, las variaciones observadas corresponden a los desplazamientos realizados en el entorno de prueba, confirmando que el módulo GPS proporciona información consistente para el monitoreo de la trayectoria.

La gráfica de velocidad presenta el comportamiento dinámico del robot durante su desplazamiento. Se observan variaciones y picos en los valores de velocidad, los cuales corresponden a cambios en el movimiento del robot, como aceleraciones o desaceleraciones durante la prueba.

La gráfica de voltaje máximo muestra el valor máximo del voltaje registrado en el sistema de alimentación del robot dentro de cada intervalo de medición. Los valores se mantienen cercanos a un nivel estable alrededor de los 24 V, con pequeñas variaciones que pueden asociarse a cambios en la carga del sistema durante el funcionamiento del robot.

La gráfica de voltaje mínimo representa los valores más bajos del voltaje registrados en el sistema durante los intervalos de medición. Se observan ligeras caídas de voltaje en determinados momentos, lo cual puede estar relacionado con demandas instantáneas de energía del robot, no obstante, los valores se mantienen dentro de un rango de operación aceptable, lo que confirma la estabilidad del sistema de alimentación.

La gráfica de la máxima derivada positiva de latitud representa la mayor tasa de cambio de la posición en la dirección norte–sur dentro de una ventana temporal determinada. Los picos observados en la gráfica indican momentos en los que el robot experimenta cambios más rápidos en su posición geográfica, este análisis permite identificar variaciones significativas en el desplazamiento del robot y evaluar su comportamiento dinámico durante el movimiento.

La gráfica de la máxima derivada positiva de velocidad muestra los cambios más pronunciados en la velocidad del robot durante el intervalo de análisis, lo que corresponde a eventos de aceleración. Los picos observados reflejan momentos en los que el robot incrementa su velocidad de forma más significativa, esta variable permite analizar el comportamiento dinámico del sistema y detectar variaciones importantes en el desplazamiento del robot.

4.2.1. Pruebas de funcionamiento en campo

Como parte del proceso de validación del sistema de telemetría, se realiza una prueba de funcionamiento en condiciones reales, desplazando el robot en un entorno abierto con el objetivo de verificar la adquisición, transmisión y visualización de los datos en tiempo real como se observa en la Fig. 4.21.

Durante esta prueba, el módulo GPS registra las coordenadas geográficas y la velocidad de desplazamiento del robot, mientras que el sistema de comunicación transmite la información a través de la red LoRaWAN hacia la plataforma The Things Network. Posteriormente, los datos son procesados por el servidor, almacenados en la base de datos local y sincronizados con Firebase para su visualización en la interfaz gráfica desarrollada.

Se comprueba que las variables de latitud, longitud y velocidad presentan variaciones coherentes con el movimiento físico del robot, confirmando la correspondencia entre el desplazamiento real y la información registrada en el sistema. Asimismo, se verifica la estabilidad de la comunicación durante el recorrido y la correcta actualización de los datos en la nube.

Esta prueba permite evaluar la latencia del sistema, la continuidad de transmisión y la confiabilidad general de la arquitectura implementada bajo condiciones normales de operación, validando su desempeño en un escenario real y proporcionando la correcta visualización gráfica en tiempo real de las variables monitoreadas como se indica en la Fig. 4.21



Fig. 4.21 Robot en pruebas de campo.

4.3. Evaluación del rendimiento del sistema de comunicación LoRaWAN

El rendimiento del enlace de comunicación LoRaWAN es evaluado durante pruebas de campo realizadas en un entorno abierto, con una distancia aproximada de 100 metros entre el nodo final instalado en el robot móvil y la puerta de enlace.

Las pruebas son ejecutadas enviando un total de 100 paquetes consecutivos, con un intervalo de transmisión programado de 20 segundos. El tiempo de llegada de cada paquete fue registrado en la plataforma de monitoreo, permitiendo determinar la latencia promedio y los indicadores de confiabilidad del sistema.

Durante las pruebas, el tiempo promedio requerido desde la emisión del paquete hasta su recepción en el servidor remoto fue de aproximadamente 1 segundo. Considerando el intervalo de transmisión configurado, el tiempo total de ciclo desde la adquisición de datos hasta su visualización remota fue de 21 segundos y sus parámetros son mostrados en la Tabla 4.8.

Tabla 4.8: Parámetros de desempeño del enlace LoRaWAN.

Parámetro	Descripción
Paquetes enviados	100
Paquetes recibidos	96
Paquetes perdidos	4
Intervalo de transmisión	20 s
Latencia promedio	1 s
Tiempo total de ciclo	21 s
Distancia nodo–Gateway	100 m
Entorno de prueba	Campo - abierto

Los resultados obtenidos evidencian un comportamiento estable del enlace de comunicación bajo las condiciones evaluadas, ya que el porcentaje de paquetes recibidos fue del 96 %, lo que indica un nivel adecuado de confiabilidad para aplicaciones de telemetría y monitoreo remoto.

La latencia promedio registrada de aproximadamente 1 segundo se encuentra dentro de los valores esperados para redes LoRaWAN, considerando el procesamiento en la red y el tiempo de encaminamiento hacia el servidor, este valor confirma que el sistema es capaz de transmitir

información de manera eficiente sin afectar la coherencia temporal de las variables monitoreadas.

Así, el desempeño valida la arquitectura implementada, demostrando que la integración entre el nodo remoto, la red LoRaWAN, el servidor de procesamiento y la interfaz gráfica garantiza continuidad de transmisión, estabilidad del enlace y correcta visualización de la información en un entorno real de operación.

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Se desarrolló un sistema de telemetría para un robot móvil de alto torque mediante la integración de tecnología LoRaWAN y el framework ROS 2, permitiendo la adquisición, transmisión y almacenamiento remoto de variables críticas del sistema.

Se evaluaron las variables críticas que requieren monitoreo remoto, identificando como prioritarias el voltaje de batería, la latitud, la longitud y la velocidad del robot. El análisis de estas variables permitió determinar su importancia en la supervisión del estado energético, la localización geográfica y el comportamiento dinámico del sistema.

Se diseñó una arquitectura de comunicación basada en LoRaWAN compatible con el entorno ROS 2, estructurando la transmisión de datos mediante nodos y tópicos que garantizan modularidad y organización del sistema. La integración entre el nodo embarcado, el gateway y el sistema de recepción permitió establecer un flujo de información estable y de bajo consumo energético.

La validación del rendimiento del sistema evidenció que la transmisión de datos se realizó de manera consistente y en un entorno controlado, permitiendo el análisis de picos máximos y mínimos de voltaje, así como el cálculo de derivadas discretas de latitud y velocidad. Estos resultados confirmaron que el sistema es capaz de detectar variaciones significativas en el estado energético y en la dinámica de movimiento del robot.

5.2. Recomendaciones

Se recomienda incorporar técnicas de filtrado para mejorar la precisión en el cálculo de derivadas de posición y velocidad, especialmente debido a la variabilidad inherente de los datos GPS.

Es conveniente implementar un sistema de estimación del estado de carga de la batería que permita una evaluación más precisa de la autonomía del robot, complementando el análisis basado únicamente en el voltaje instantáneo.

Se sugiere realizar pruebas adicionales en escenarios con mayores distancias y presencia de interferencias electromagnéticas, con el objetivo de caracterizar el comportamiento del enlace LoRaWAN bajo condiciones más exigentes.

5.3. Trabajo a futuro

Como trabajo a futuro, se propone integrar sensores adicionales que permitan ampliar el análisis dinámico del robot, como unidades de medición inercial o sensores de corriente para monitoreo energético avanzado.

Asimismo, se plantea optimizar la arquitectura de inicialización del sistema de telemetría para permitir la activación autónoma de los nodos sin necesidad de conexión física, integrando un módulo de comunicación Bluetooth que facilite la configuración y puesta en marcha del robot mediante una aplicación móvil.

Finalmente, se considera pertinente desarrollar una plataforma de visualización web que permita el monitoreo remoto en tiempo real y la gestión centralizada de múltiples robots dentro de una misma red, ampliando así el alcance del sistema hacia aplicaciones de flotas autónomas.

Bibliografía

- [1] W. B. M. F. Junior; Montalvo, “Libro-robots-vf3,” 2023.
- [2] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, “Robot operating system 2: Design, architecture, and uses in the wild,” *Science Robotics*, vol. 7, 5 2022.
- [3] B. V. Martínez, P. G. López, L. H. González, J. R. Hernández, and J. D. M. Elizalde, “Sistemas de telemetría y teleoperación en tiempo real usando vps y api fetch. caso de estudio: variables eléctricas de una casa habitación,” *Ciencia Latina Revista Científica Multidisciplinar*, vol. 6, pp. 4967–4987, 11 2022.
- [4] M. Karatzia, P. Kolios, and G. Ellinas, “Implementing mission-critical uav swarm coordination through the integration of lora and ros frameworks,” in *2023 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, pp. 1–7, IEEE, 9 2023.
- [5] A. T. Bou, R. M. Prades, L. B. Tambutti, P. . Sanz, and J. Martí, *Desarrollo de una interfaz para el prototipado y validación de un robot móvil autónomo de uso hospitalario*, pp. 156–164. Servizo de Publicacións da UDC, 9 2022.
- [6] F. J. F. Zermeño and E. G. C. Franco, “Aplicaciones, enfoques y tendencias del internet de las cosas (iot): Revisión sistemática de la literatura,” vol. 13, p. 568, 2021.
- [7] J. A. G. Vasquez, M. E. C. Arroyo, M. M. H. Maguiña, M. K. L. Asto, L. M. S. Quincho, and A. F. V. Cabrera, “Internet of things technologies applied in the supply chain. a systematic review,” *Gestión de Operaciones Industriales*, vol. 2, pp. 8–26, 7 2023.
- [8] V. Hernandez-Rodriguez, D. Kairuz-Cabrera, A. Martinez-Laguardia, P. Merino-Laso, and O. S. RESUMEN, “Estación meteorológica iot basada en ttgo t-beam y comunicación lora title: Iot weather station based on ttgo t-beam and lora communication,” *RIELAC*, vol. 44, p. 2302, 2023.

- [9] M. M. Pareja, “Gestión de robots autónomos y colaborativos mediante ros y computación en la nube,” tech. rep., 9 2021.
- [10] E. Mediavilla, “Robot móvil de alto torque: Desarrollo electrónico,” tech. rep., 3 2025.
- [11] J. Quiñones, “Diseño e implementación de un robot móvil autónomo para reducir contagios de covid-19 en el servicio urbano de delivery de comida con carga máxima de hasta 10 kg,” 2024.
- [12] K. A. F. Tamayo and A. D. T. Narea, “Diseño e implementación de un prototipo de robot móvil diferencial aspersor de alcohol, con tecnología iot y de telemetría,” tech. rep., 9 2022.
- [13] P. S. H. Rojas, “Robot móvil de alto torque: Visión artificial,” tech. rep., 2 2025.
- [14] M. Turrión, “Diseño e implementación de un sistema de telemetría controlado por dispositivos embebidos,” tech. rep., 7 2022.
- [15] K. R. A. Ube and F. P. G. Zambrano, “Diseño e implementación de robot móvil multitarea con visión artificial y programación en lenguaje python para la universidad politécnica salesiana,” tech. rep., 4 2022.
- [16] I. B. L. Illescas and S. P. M. Niola, “Diseño de un robot scara para el banco mecatrónico de la universidad politécnica salesiana sede - cuenca,” tech. rep., 2 2025.
- [17] Ángel Gil, “Robótica móvil: Qué es y sus aplicaciones,” 7 2022.
- [18] B. J. Smith, K. M. Hart, F. J. Mazzotti, M. Basille, and C. M. Romagosa, “Evaluating gps biologging technology for studying spatial ecology of large constricting snakes,” *Animal Biotelemetry*, vol. 6, p. 1, 12 2018.
- [19] SITRACK, “Todo sobre la telemetría,” 2019.
- [20] B. Digital, “Conceptos técnicos básicos que te ayudarán a entender lora y lorawan (low power wide area network) en pocos minutos,” 6 2022.
- [21] I. consulting, “La tecnología lora y su importancia en iot,” 2 2019.
- [22] P. Baltuille, “Lorawan y su aportación a las tecnologías iiot,” 6 2023.
- [23] Dragino, “Rs485-lb/lb – waterproof rs485/uart to lorawan converter,” 2025.

- [24] M. e. Paguayo, “Señales tx y rx,” 2025.
- [25] Robotnik, “Ros 2 (robot operating system): visión general y puntos clave del software de robótica,” 4 2025.
- [26] G. P. Garrido, Álvaro Castro González, J. C. C. Montoya, M. Ángeles Malfaz Vázquez, and M. S. Sánchez-Caballero, “Una propuesta de integración de una red de sensores iot en el robot mini para monitorización del usuario en su domicilio,” *Jornadas de Automática*, 7 2024.
- [27] Fiveable, “5.3 gps localization – intro to autonomous robots,” 8 2024.
- [28] Gabri, “¿qué es la latitud y longitud?,” 2 2022.
- [29] M. Hamid, A. Adom, N. Rahim, and M. Rahiman, “Navigation of mobile robot using global positioning system (gps) and obstacle avoidance system with commanded loop daisy chaining application method,” in *2009 5th International Colloquium on Signal Processing & Its Applications*, pp. 176–181, IEEE, 3 2009.
- [30] K. Liu, K. Li, Q. Peng, and C. Zhang, “A brief review on key technologies in the battery management system of electric vehicles,” *Frontiers of Mechanical Engineering*, vol. 14, pp. 47–64, 3 2019.
- [31] Dynamox, “Los valores pico, pico a pico y rms en el análisis de vibraciones.,” 2 2026.
- [32] J. de Pedro, “Análisis dimensional conforme a la dinámica y complejidad de las señales biológicas,” 2020.
- [33] A. Webmaster, “The things network (ttn), una red iot global para la sociedad,” 2025.
- [34] T. T. N. G. Team, “The things network, “the things network”,” 2025.
- [35] J. Aldave, “Los sistemas operativos linux y windows en las telecomunicaciones,” tech. rep., 1 2021.
- [36] A. Ramirez and I. Romero, “Red de sensores con conexión industrial y almacenamiento en la nube,” tech. rep., 3 2025.
- [37] W. Cantos, “Sistema de microcontroladores programables con tecnología arduino para brindar seguridad en el departamento de coordinación de tecnologías del campus los Ángeles en la universidad estatal del sur de manabí,” tech. rep., 2025.

- [38] BysMax, “Esp32 vs arduino diferencias y característica,” 2026.
- [39] M. Sanca, “Tipos de investigaciÓn científica,” tech. rep., 2011.
- [40] T. Grajales, “Tipos de investigacion por tevni grajales g,” tech. rep., 3 2000.
- [41] Components, “Módulo gps neo-6mv2,” 2026.
- [42] L. H. S. Subramaniam, “Desarrolle robots con ia, sistemas de visiÓn inteligente y más con el kit para desarrolladores nvidia jetson orin nano,” 2025.
- [43] J. A. G. Cabezas, “Robot móvil de alto torque: Desarrollo mecÁnico,” tech. rep., 2025.
- [44] M.Qiu, “Rs485-lb/lb: Convertidor rs485/uart a lorawan resistente al agua,” 2026.
- [45] R. Ltd, “Vnc connect – secure remote access software.,” 2026.
- [46] php.net, “Install and configure php-fpm.,” 2026.

Anexos

Anexo A.

Gráficas del sistema de telemetría.

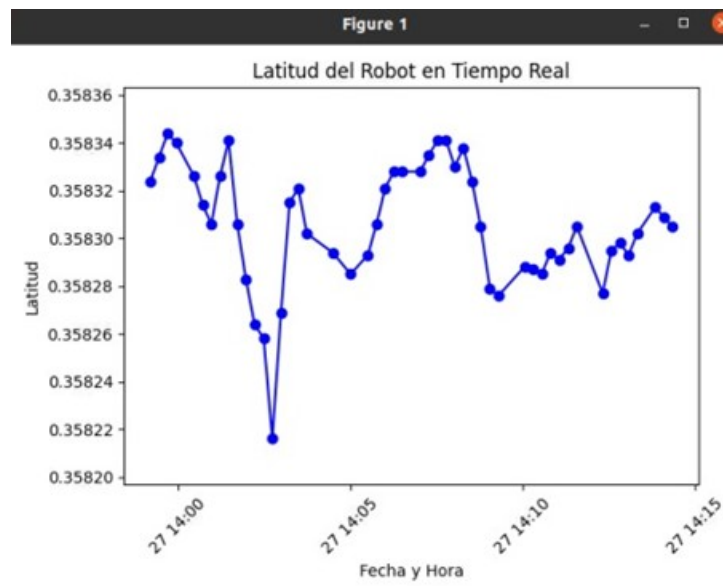


Fig. A.1. Gráfica de latitud.

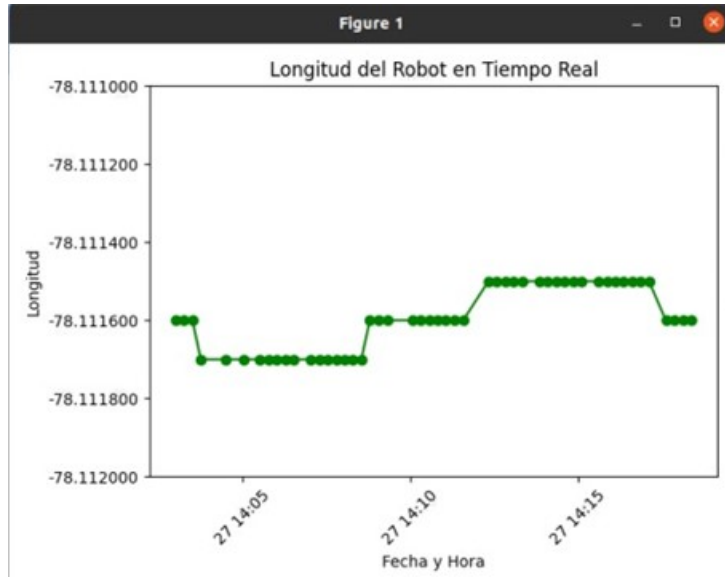


Fig. A.2. Gráfica de longitud.

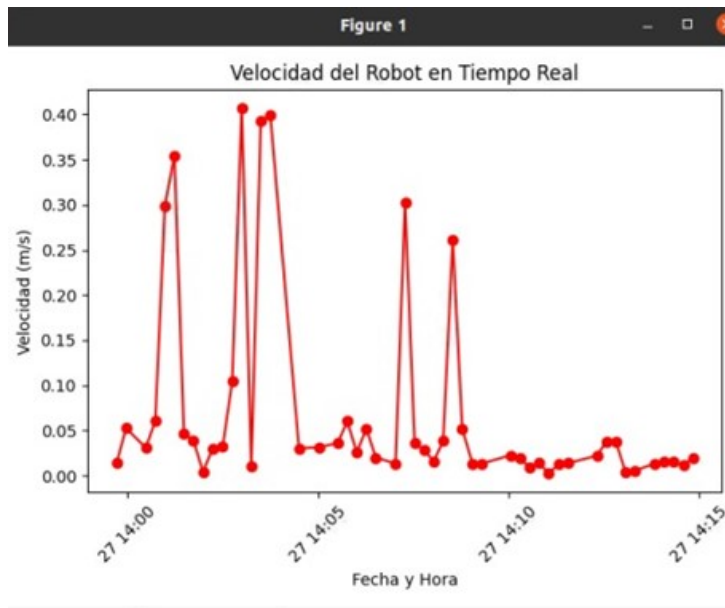


Fig. A.3. Gráfica de velocidad.

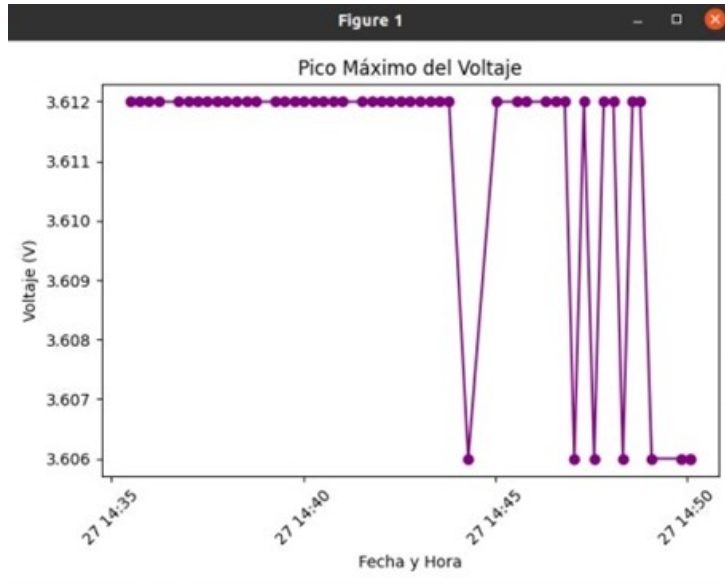


Fig. A.4. Gráfica del pico máximo del voltaje.

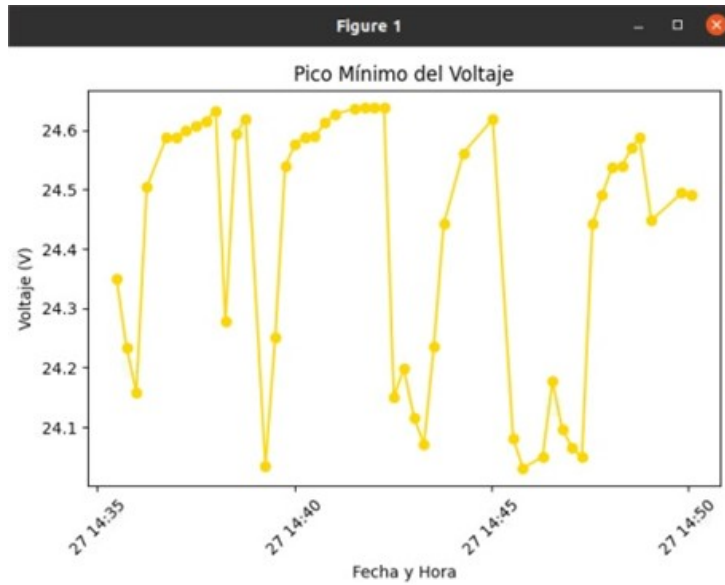


Fig. A.5. Gráfica del pico mínimo del voltaje.

Anexo B. Códigos y scripts disponibles en:

<https://github.com/alsochandi-byte/TelemetryRobot>