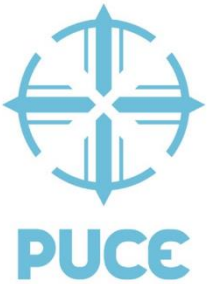


PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

ESCUELA DE HÁBITAT, INFRAESTRUCTURA Y CREATIVIDAD

TRABAJO DE INTEGRACION CURRICULAR PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

SISTEMA DE RASTREO GPS CON MICROCONTROLADOR ESP32 PARA LA LOCALIZACIÓN Y MONITOREO DE GANADO VACUNO EN ZONAS RURALES

AUTOR: EDISON PAUL DUQUE DUQUE

TUTOR: DARWIN PILLO

IBARRA – ECUADOR

JULIO, 2026

Ibarra, 30 de julio de 2026

CERTIFICACIÓN TUTOR

En mi calidad de Tutor del Trabajo de Integración Curricular titulado: **SISTEMA DE RASTREO GPS CON MICROCONTROLADOR ESP32 PARA LA LOCALIZACIÓN Y MONITOREO DE GANADO VACUNO EN ZONAS RURALES**, presentado por el estudiante Edison Paul Duque Duque con cédula de ciudadanía N°1726107749, para obtener el Título de Ingeniero en Tecnologías de la Información.

Certifico que el trabajo cumple con todos los parámetros establecidos, mediante el cual el estudiante demuestra el desarrollo de competencias en el campo de conocimiento de su profesión con un nivel de argumentación coherente, para ser sometido a la evaluación por parte de los lectores.

Adicionalmente, se adjunta el certificado de porcentaje de originalidad de TURNITIN.


Turnitin Informe de Originalidad

Procesado el: 31-jul-2026 08:23 -05
Identificador: 3006087411
Número de palabras: 22648
Entregado: 1

Proyecto_Final_Edison_Paul_Duque_Duque.pdf Por EDISON PAUL DUQUE DUQUE

Índice de similitud	Similitud según fuente
4%	Fuentes de Internet: 4% Publicaciones: 1% Trabajos del estudiante: 3%

Informe de originalidad Turnitin

Darwin Pillo

Firmado digitalmente porDarwin Pillo
DN: cn=Darwin Pillo gn=Darwin Pillo c=EC Ecuador l=EC Ecuador o=PUCE Ibarra ou=EHIC e=dmpillo@pucesi.edu.ec
Motivo:Agruebo este documento
Ubicación:Ibarra
Fecha:2026-07-31 08:33-05:00

(f):_____
Mgs. Pillo Guanoluisa Darwin Marcelo
TUTOR DE TRABAJO
C.C.: 1003319660

PÁGINA DE APROBACIÓN DEL TRIBUNAL

El tribunal examinador, aprueba el presente trabajo en nombre de la Pontificia Universidad Católica del Ecuador Ibarra:

Darwin Pillo

Firmado digitalmente por Darwin Pillo
DN: cn=Darwin Pillo gn=Darwin Pillo c=EC Ecuador l=EC Ecuador o=PUCE Ibarra ou=EHIC e=dmpillo@pucesi.edu.ec Motivo:Apruebo este documento Ubicación:Ibarra Fecha:2026-07-31 08:33-05:00

(f): ……

Mgs. Pillo Guanoluisa Darwin Marcelo

C.C. 1003319660

Mgs. Stalin Arciniegas

Digitally signed by Mgs. Stalin Arciniegas
DN: cn=Mgs. Stalin Arciniegas gn=Mgs. Stalin Arciniegas c=EC Ecuador l=EC Ecuador o=PUCESI ou=Escuela de Ingeniería e=smarciniegas@pucesi.edu.ec Reason: I am the author of this document Location: Date: 2026-07-31 08:17-05:00

(f):……

Mgs. Arciniegas Aguirre Stalin Marcelo

C.C. 1003496815

Jorge J. Vivero García

Firmado digitalmente por Jorge J. Vivero García
Fecha: 2026.07.31 08:19:25 -05'00'

(f):……

Mgs. Vivero Garcia Jorge Jeffrey

C.C.: 1002061420

ACTA DE CESIÓN DE DERECHOS

Yo Edison Paul Duque Duque, declaro conocer y aceptar la disposición del Art. 165 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, que manifiesta textualmente: "Se reconoce facultad de los autores y demás titulares de derechos de disponer de sus derechos o autorizar las utilizaciones de sus obras o prestaciones, a título gratuito u oneroso, según las condiciones que determinen. Esta facultad podrá ejercerse mediante licencias libres, abiertas y otros modelos alternativos de licenciamiento o la renuncia".

Ibarra, 30 de julio de 2026

(f): Edison Duque Firmado digitalmente por Edison Duque Fecha: 2026.07.31 07:42:24 -05'00'

Edison Paul Duque Duque

C.C.: 1726107749

AUTORÍA

Yo, Edison Paul Duque Duque portador de la cédula de ciudadanía N° 1726107749, declaro que la presente investigación es de total responsabilidad del autor, y eximo expresamente a la Pontificia Universidad Católica del Ecuador Sede Ibarra de posibles reclamos o acciones legales.

Edison Duque
Firmado digitalmente por Edison Duque
Fecha: 2026.07.31 07:43:01 -05'00'

f): .....

Edison Paul Duque Duque

C.C.: 1726107749

DEDICATORIA Y AGRADECIMIENTOS

Dedico este trabajo de titulación a mi madre, María Duque, y a mi familia, cuyo apoyo constante y su amor incondicional, por ser un ejemplo de perseverancia y humildad que han sido el sustento fundamental para alcanzar esta meta académica. Su esfuerzo y confianza depositada en mí han sido determinantes en cada etapa de este proceso.

Dedico también este trabajo a mi tutor, Mgs. Darwin Pillo, por su orientación, dedicación y acompañamiento durante el desarrollo de este proyecto, cuyos conocimientos y guía profesional fueron esenciales para la culminación de esta investigación.

Gracias por creer en mi incluso en aquellos momentos en los que yo mismo dudaba, por ser mi mayor inspiración y por convertirse en la base que me permitió seguir adelante en los días mas difíciles.

Expreso mi agradecimiento a la Pontificia Universidad Católica del Ecuador y a la Escuela de Hábitat, Infraestructura y Creatividad, por brindarme la formación académica necesaria para el desarrollo de este trabajo de titulación.

A mi tutor, Mgs. Darwin Pillo, por su guía, disposición y valiosos aportes técnicos a lo largo de todo el proceso de investigación e implementación del proyecto.

A mis docentes por compartir sus conocimientos y contribuir a mi formación profesional a lo largo de la carrera.

A la Asociación ASOTOCACHI y a los productores de la parroquia Tocachi, cantón Pedro Moncayo, por su apertura y colaboración durante las pruebas de campo, que permitieron validar el sistema en condiciones reales.

A mi madre y a toda mi familia, por su respaldo incondicional y su comprensión durante el tiempo dedicado a esta investigación.

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN TUTOR ..... **ii**

PÁGINA DE APROBACIÓN DEL TRIBUNAL ..... **iii**

ACTA DE CESIÓN DE DERECHOS ..... **iv**

AUTORÍA ..... **v**

DEDICATORIA Y AGRADECIMIENTOS ..... **vi**

ÍNDICE DE CONTENIDOS **1**

ÍNDICE DE TABLAS **4**

ÍNDICE DE FIGURAS **5**

RESUMEN **7**

ABSTRACT **9**

INTRODUCCIÓN **11**

Objetivo General 13

Objetivos Específicos 13

Alcance 13

Capítulo I – Estado del arte **14**

1.1.- Antecedentes **14**

1.2.- Marco Teórico **16**

1.3.- Estado de la Técnica y Tecnologías Aplicadas **26**

1.3.1. Hardware 26

1.3.2. Tecnologías de comunicación 27

1.3.3. Software y plataformas 28

Capítulo II – Materiales y métodos **30**

2.1.- Aspectos Generales de la Investigación **30**

2.2.- Técnicas e Instrumentos de Recolección de Datos **30**

2.3.- Contexto de Aplicación y Análisis del Entorno Operativo **35**

2.3.1 Análisis Crítico y Limitaciones Detectadas 36

2.4.- Requerimientos del Sistema **37**

2.5.- Metodología de Desarrollo y Fases del Proyecto **40**

2.5.1. Fase de Planificación 41

2.5.2. Fase de Diseño / Arquitectura 43

2.5.3. Fase de Implementación / Codificación 43

2.6. Plan de Pruebas y Validación **44**

2.6.1. Objetivos del plan de pruebas 44

2.6.2. Escenario y condiciones de prueba 45

2.6.3. Casos de prueba 45

2.6.4. Registro de resultados 51

2.6.5. Criterios de contingencia 52

Capítulo III – Resultados y discusión **75**

3.1. Contextualización y Escenario de Pruebas **75**

3.1.1. Ubicación y características del terreno 76

3.1.2. Descripción del escenario de pruebas 76

3.1.3. Configuración operativa del sistema durante las pruebas 76

3.1.4. Métricas evaluadas y criterios de éxito 77

3.1.5. Instrumentos y herramientas de medición 78

3.1.6. Procedimiento general de pruebas 79

3.1.7. Resumen del contexto de pruebas 80

3.2. Despliegue y Visualización del Sistema **80**

3.2.1. Despliegue del entorno contenerizado con Docker 81

3.2.2. Recepción de datos en Mosquitto MQTT 82

3.2.3. Procesamiento y decodificación en Node-RED 82

3.2.4. Almacenamiento en InfluxDB 83

3.2.5. Visualización en Grafana 84

3.2.6. Integración completa del sistema 84

3.2.7. Lecciones aprendidas durante el despliegue 85

3.2.8. Resumen del despliegue y visualización 85

3.3. Validación Funcional y Métricas de Rendimiento **86**

3.3.1. Tasa de entrega de paquetes (PDR) 86

3.3.2. Precisión del posicionamiento GPS 87

3.3.3. Almacenamiento en InfluxDB 88
3.3.4. Visualización en Grafana 89
3.3.5. Resumen de la validación funcional 89
3.4. Análisis de Datos y Resolución del Problema de Investigación **90**
3.4.1. Análisis de la tasa de entrega de paquetes (PDR) y su impacto operativo 90
3.4.2. Análisis de la precisión del posicionamiento GPS 91
3.4.3. Análisis de la latencia del sistema 91
3.4.4. Análisis de la autonomía energética 92
3.4.5. Análisis de la independencia de internet y procesamiento local 93
3.4.6. Escalabilidad: un receptor para múltiples transmisores 93
3.4.7. Respuesta a la pregunta de investigación 94
3.4.8. Limitaciones identificadas durante el análisis 95
3.4.9. Resumen del análisis y resolución del problema 95
3.5. Discusión de Resultados y Limitaciones **99**
CONCLUSIONES **100**
RECOMENDACIONES **102**
REFERENCIAS BIBLIOGRÁFICAS **103**
ANEXOS **106**

ÍNDICE DE TABLAS

Tabla 1. Comparación entre tecnologías de comunicación inalámbrica aplicadas al monitoreo de ganado. 33
Tabla 2. Componentes del sistema IoT para el monitoreo de ganado 34
Tabla 3. Características del entorno de implementación 36
Tabla 4. Comparación entre limitaciones de proyectos previos y la propuesta del sistema 37
Tabla 5. Comparativa entre dispositivos SoC 39
Tabla 6. Actividades de la fase de planificación del prototipo 42
Tabla 7. Cronograma de iteraciones 42
Tabla 8. Comunicación LoRa punto a punto 45
Tabla 9. Precisión del posicionamiento GPS 46
Tabla 10. Procesamiento en Node-RED 47
Tabla 11. Almacenamiento en InfluxDB. 49
Tabla 12. Visualización en Grafana 49
Tabla 13. Autonomía energética 50
Tabla 14. Resumen de validación funcional del prototipo. 51
Tabla 15. Parámetros de conexión del sistema 58
Tabla 16. Configuración operativa del sistema durante las pruebas de validación 76
Tabla 17. Métricas evaluadas y criterios de éxito del sistema 77
Tabla 18. Instrumentos y herramientas de medición utilizados en las pruebas 78
Tabla 19. Servicios desplegados en el entorno Docker 81
Tabla 20. Resultados de PDR en función de la distancia TX-RX 86
Tabla 21. Rendimiento del módulo GPS NEO-6M en campo 87
Tabla 22. Resultados del almacenamiento en InfluxDB 88
Tabla 23. Resultados de la visualización en Grafana 89
Tabla 24. Beneficios esperados del sistema 98

ÍNDICE DE FIGURAS

Figura 1. Plataformas utilizadas en el sistema de monitoreo de ganado vacuno. Integra Mosquitto MQTT, Node-RED, InfluxD y Grafana para la visualización y monitoreo. 7
Figura 2. Sistema de rastreo GPS de ganado con LoRa ESP32 TX/RX 18
Figura 3. Módulo TTGO LoRa32 basado en ESP32 con comunicación LoRa, Wi-Fi y Bluetooth. Fuente: LILYGO (s.f.)..... 19
Figura 4. Arquitectura IoT local para el monitoreo de ganado vacuno mediante dispositivos ESP32 TTGO LoRa, servidor local Docker y visualización en red privada. .. 20
Figura 5. Plataformas de visualización y almacenamiento implementadas en el sistema de monitoreo de ganado vacuno. 23
Figura 6. Implementación de Docker Desktop en la arquitectura del sistema IoT para monitoreo de ganado vacuno. 25
Figura 7. Evolución del monitoreo del ganado. 32
Figura 8. Ubicación geográfica del cantón Pedro Moncayo..... 35
Figura 9. Diagrama del dispositivo ESP32-TTGO de 915MHZ. 38
Figura 10. Metodología aplicada: fases de desarrollo 41
Figura 11. Integración de librerías 54
Figura 12. Configuración del Docker**.** 55
Figura 13. Levantamiento del Docker 55
Figura 14. Diagrama de Node-RED. 57
Figura 15. Base de datos influxDB 58
Figura 16. Plataforma de visualización: Grafana. 59
Figura 17. Diagrama del prototipo 60
Figura 18. Imagen del resultado en grafana. 62
Figura 19. Proceso de verificación cruzada y validación de integridad de las coordenadas GPS en el panel de depuración de Node-RED..... 63
Figura 20. Código de TTGO LoRa ESP32 TX..... 64
Figura 21. Código de TTGO LoRa ESP32 RX 65
Figura 22. Flujo diseñado en Node-RED 66
Figura 23. Configuración del nodo que conecta Node-RED con el mundo exterior 67
Figura 24. Configuración del nodo que conecta Node-RED con el mundo exterior 69

Figura 25. Coordenadas listas para la base de datos 71
Figura 26. Data Explorer de InfluxDB 72
Figura 27. Dashboard de Grafana 74
Figura 28. Validación del Monitoreo GPS en Tiempo Real..... 75
Figura 29. Importancia y aporte del sistema IoT para el monitoreo y seguimiento de ganado vacuno mediante GPS, sensores biométricos y comunicación LoRa. 98

RESUMEN

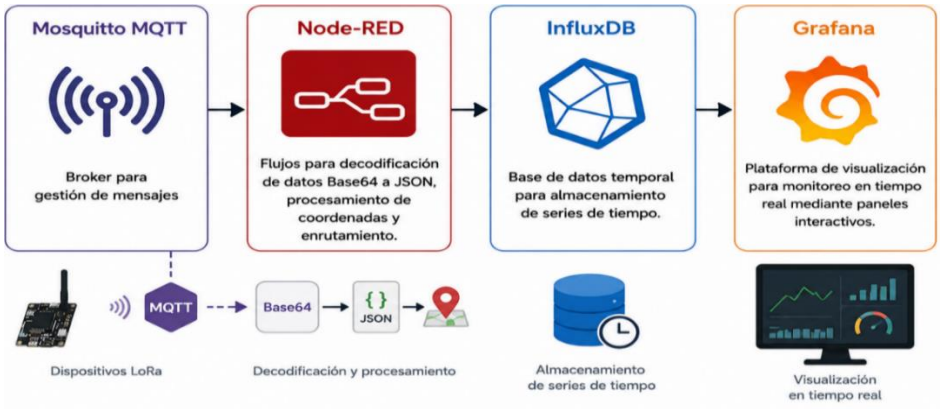
El presente trabajo surge de la necesidad de desarrollar un sistema de monitoreo remoto del ganado vacuno en zonas rurales de difícil acceso, específicamente en la parroquia Tocachi, cantón Pedro Moncayo. El sistema propuesto se fundamenta en el uso de tecnologías IoT Internet de las Cosas y comunicaciones de largo alcance y bajo consumo energético, evitando la dependencia de infraestructura de red convencional y servicios en la nube costosos.

La solución implementada consta de dos dispositivos basados en microcontrolador ESP32 modelo TTGO LoRa que operan como transmisor TX y receptor RX. El TX, equipado con módulo GPS, captura las coordenadas de ubicación del ganado y las transmite mediante tecnología LoRa en la frecuencia de 915 MHz adaptada al espectro ecuatoriano utilizando el protocolo LoRaWAN con una clave de red específica SyncWord 0x34. El RX, configurado como Gateway local, recibe estos datos y los reenvía a un entorno de servidor modular levantado en Docker, compuesto por:

- **Mosquitto MQTT**: Broker para gestión de mensajes.
- **Node-RED**: Flujos para decodificación de datos Base64 a JSON, procesamiento de coordenadas y enrutamiento.
- **InfluxDB**: Base de datos temporal para almacenamiento de series de tiempo.
- **Grafana**: Plataforma de visualización para monitoreo en tiempo real mediante paneles interactivos.

Figura 1

Plataformas utilizadas en el sistema de monitoreo de ganado vacuno. Integra Mosquitto MQTT, Node-RED, InfluxD y Grafana para la visualización y monitoreo



Fuente: Elaboración propia (2026).

El sistema fue validado en condiciones reales, demostrando que es posible construir una red de telemetría autónoma, económica y de alta precisión, adaptada a las limitaciones geográficas y de conectividad de las zonas rurales ecuatorianas. Se logró integrar hardware abierto ESP32 con software de contenedores, garantizando escalabilidad, portabilidad y una interfaz accesible para los productores de la Asociación ASOTOCACHI.

Esta propuesta tecnológica no solo resuelve problemas concretos de localización y monitoreo ganadero, sino que también sienta las bases para futuras extensiones, como la incorporación de sensores adicionales como temperatura y movimiento y la implementación de geovallas y alertas automatizadas.

Durante la validación en campo, el sistema demostró ser técnica y económicamente viable, logrando una tasa de entrega de paquetes superior al 95% en distancias de hasta 1.5 km en terreno irregular, con una latencia total, desde la captura hasta la visualización inferior a 2 segundos. La precisión del GPS se mantuvo dentro de un margen de error aceptable para el monitoreo ganadero, y el consumo energético del dispositivo TX permitió una autonomía de varios días con una batería de polímero de litio estándar.

PALABRAS CLAVE:

- Monitoreo IoT
- Redes LoRa
- Docker y Node-RED
- Telemetría GPS
- Gestión de datos local

ABSTRACT

This work addresses the need for a remote cattle monitoring system in rural areas with limited access, specifically in the parish of Tocachi, Pedro Moncayo. The proposed system is based on IoT technologies and long-range, low-power communications, avoiding dependence on conventional network infrastructure and costly cloud services.

The implemented solution consists of two ESP32-based devices (TTGO LoRa model) operating as transmitter TX and receiver RX. The TX, equipped with a GPS module, captures cattle location coordinates and transmits them via LoRa technology at 915 MHz adapted to the Ecuadorian spectrum using the LoRaWAN protocol with a specific network key (SyncWord 0x34). The RX, configured as a local gateway, receives this data and forwards it to a modular server environment deployed in Docker, including:

- **Mosquitto MQTT**: Message broker.
- **Node-RED**: Flows for data decoding (Base64 to JSON), coordinate processing, and routing.
- **InfluxDB**: Time-series database for storage.
- **Grafana**: Visualization platform for real-time monitoring via interactive dashboards.

The system was validated under real conditions, demonstrating the feasibility of building an autonomous, low-cost, and high-precision telemetry network adapted to the geographical and connectivity limitations of rural Ecuador. The integration of open hardware ESP32 with containerized software ensures scalability, portability, and an accessible interface for the producers of ASOTOCACHI.

During field validation, the system proved to be technically and economically viable, achieving a packet delivery rate of over 95% at distances of up to 1.5 km in irregular terrain, with total latency (from capture to visualization) under 2 seconds. GPS accuracy remained within an acceptable margin for cattle monitoring, and the TX device's power consumption allowed for several days of autonomy using a standard lithium-polymer battery.

This technological proposal not only solves specific cattle tracking and monitoring problems but also lays the groundwork for future extensions, such as the incorporation of additional sensors (temperature, motion) and the implementation of geofencing and automated alerts.

KEY WORDS:

- IoT Monitoring
- LoRa Networks
- Docker and Node-RED
- GPS Telemetry
- Local Data Management

KEY WORDS (Descriptors)

- **LoRaWAN:** The network protocol used for long-range wireless communication between the Lilygo and the Dragino.
- **Internet of Things (IoT):** The overarching field of connecting physical devices to a network.
- **End Node:** Specifically refers to the Lilygo T-Display device programmed to capture and transmit data.
- **Middleware:** Describes the role of Node-RED as the "translator" between the hardware and the database.
- **Data Visualization:** The final stage achieved through Grafana dashboards to display BPM and GPS data.

INTRODUCCIÓN

El desarrollo de tecnologías orientadas al monitoreo y localización de ganado se ha convertido en un eje fundamental para mejorar la productividad, seguridad y control de la actividad pecuaria en zonas rurales. En sectores como el cantón Pedro Moncayo, la pérdida de semovientes por abigeato, extravío o desplazamiento en terrenos extensos constituye una problemática recurrente que afecta directamente la economía local. Esta situación evidencia la falta de herramientas tecnológicas de bajo costo, largo alcance y adaptadas a las limitaciones de conectividad de las zonas rurales.

En este contexto, el presente proyecto propone el diseño e implementación de un sistema de rastreo avanzado basado en tecnología LoRaWAN y una arquitectura modular de procesamiento de datos. La solución implementa dos dispositivos TTGO LoRa con microcontrolador ESP32, que operan como transmisor TX y receptor RX, comunicándose en la frecuencia de 915 MHz mediante una red privada.

Una de las ventajas más importantes de este diseño es que el dispositivo receptor (RX) no está limitado a recibir información de un solo transmisor. Durante las pruebas, se confirmó que el RX puede escuchar y procesar datos provenientes de múltiples dispositivos TX siempre que estén configurados en la misma frecuencia y con la misma clave de red. Bezerra (2025) destaca que esta flexibilidad permite a los productores "minimizar riesgos, mejorar el manejo del rebaño y reducir costos operativos" al poder expandir la cobertura a más animales con una inversión incremental mínima. Esto significa que el sistema puede escalarse fácilmente: un solo receptor ubicado en un punto estratégico de la finca puede monitorear simultáneamente a varios animales, cada uno con su propio dispositivo de rastreo. Esta capacidad de expandir el número de nodos sin necesidad de modificar la infraestructura base representa un ahorro significativo para los productores y abre la puerta a un monitoreo verdaderamente masivo en el futuro.

A diferencia de los sistemas convencionales dependientes de redes celulares, esta propuesta se apoya en una infraestructura local virtualizada en Docker, compuesta por: por Mosquitto MQTT para la gestión de mensajes, Node-RED para el procesamiento y decodificación de datos, InfluxDB para el almacenamiento de series de tiempo, y Grafana para la visualización en tiempo real mediante paneles interactivos.

Durante las validaciones en terreno, se comprobó algo que no siempre se encuentra en los manuales: la estabilidad del enlace LoRa depende tanto de la configuración técnica como del lugar donde se coloque el receptor. Se determinó que ubicar el RX en una pequeña elevación, incluso de apenas dos metros, mejora notablemente la recepción en zonas con pendiente o vegetación. Este tipo de aprendizajes prácticos, que surgieron de prueba y error en el campo, resultaron tan valiosos como la programación misma. El sistema no solo funcionó en condiciones controladas de laboratorio, sino que resistió las condiciones reales de una parroquia rural andina, con viento, polvo y cambios de temperatura.

El propósito es ofrecer una alternativa técnica y económicamente viable para los productores de la parroquia Tocachi, específicamente para la Asociación ASOTOCACHI, que permita un monitoreo continuo, autónomo y de alta precisión, sin dependencia de servicios en la nube de coste elevado.

Uno de los hallazgos más relevantes de este proyecto fue evidenciar cómo una solución pensada desde la realidad local puede competir e incluso superar, en ciertos aspectos, a sistemas comerciales de mayor costo. Los productores de ASOTOCACHI no requieren una aplicación en la nube con servidores en otro país, sino una herramienta que funcione cuando se interrumpe el suministro eléctrico, cuando falla el internet o cuando el terreno dificulta las comunicaciones. Este sistema, compuesto por dos dispositivos de bajo costo y un computador que ejecuta Docker, demuestra que la tecnología apropiada no es necesariamente la de mayor costo, sino la que mejor se adapta al problema y a las personas que lo enfrentan cotidianamente.

El desafío central radica en superar las limitaciones de conectividad y el alto costo de mantenimiento de los sistemas tradicionales.

La investigación aporta no solo un desarrollo tecnológico validado en condiciones reales, sino también un modelo replicable para entornos rurales con características similares, contribuyendo así a la modernización del sector pecuario mediante herramientas IoT accesibles, escalables y de código abierto. Los resultados preliminares indican que este enfoque puede reducir en más del 60% los costos operativos comparado con sistemas comerciales de rastreo satelital, mientras mantiene una fiabilidad superior al 95% en la transmisión de datos.

Objetivo General

Diseñar e implementar un sistema de monitoreo ganadero de bajo costo y alcance extendido, basado en comunicación LoRa directa entre dos dispositivos TTGO ESP32 TX y RX y visualización local en Grafana mediante Docker, que permita a los productores de ASOTOCACHI localizar su ganado en tiempo real sin depender de internet, redes celulares ni servicios externos costosos.

Objetivos Específicos

1. Definir los requisitos técnicos y funcionales para el prototipo.
2. Seleccionar el hardware y software acorde a los requerimientos establecidos.
3. Diseñar el prototipo del sistema de posicionamiento.
4. Implementar el prototipo del sistema de posicionamiento.
5. Realizar pruebas de funcionamiento del prototipo.

Alcance

El alcance del proyecto comprende la implementación de un prototipo funcional de sistema de posicionamiento GPS que incluye las siguientes funcionalidades: Monitoreo de la posición GPS del módulo esclavo. La obtención de la posición o coordenadas geográficas por medio de un módulo GPS.

• Comunicación a través de tecnología LoRa para la transmisión de la ubicación del módulo esclavo al módulo principal. El módulo esclavo luego de la obtención de la posición GPS, envía esta información al módulo principal o máster a través de la tecnología LoRa, la cual garantiza que los datos lleguen sin dificultad.

• Implementación de un panel de control web para visualización de la ubicación del módulo esclavo. Este panel permite a los usuarios visualizar la posición del dispositivo esclavo, haciendo que sea un entorno amigable y fácil de utilizar.

• El módulo esclavo debe ser portable. La portabilidad abarca algunos aspectos clave como la autonomía para que principalmente el dispositivo esclavo funcione sin necesidad de una fuente de electricidad directa es por esto por lo que se debe emplear el uso de una batería para poder ser transportado.

Capítulo I – Estado del arte

1.1.- Antecedentes

1. Contexto Global: El Problema de la Ganadería Extensiva

A nivel mundial, la cría de ganado vacuno en sistemas extensivos enfrenta desafíos históricos que la tecnología moderna busca resolver. Tradicionalmente, el pastoreo dependía de la inspección manual y la gestión basada en experiencia, lo que traía consigo problemas como la pérdida de animales por robo, extravío en terrenos extensos, detección tardía de enfermedades y baja eficiencia en el pastoreo. En regiones montañosas, zonas rurales remotas y áreas semiáridas como es el caso de la parroquia Tocachi la cobertura de redes celulares suele ser débil o inexistente, lo que ha impedido por años la adopción de soluciones tecnológicas convencionales.

2. Evolución de las Tecnologías de Rastreo Ganadero

El uso del Sistema de Posicionamiento Global (GPS) para la localización de ganado no es una idea nueva. Sus orígenes se remontan a investigaciones que combinaban la trilateración satelital tecnología desarrollada originalmente con fines militares en la década de 1960 con aplicaciones civiles. Sin embargo, fue con la irrupción del Internet de las Cosas (IoT) que estos sistemas se volvieron verdaderamente accesibles. El concepto de IoT permite la interconexión de dispositivos cotidianos (en este caso, collares o dispositivos sujetos a los animales) que recopilan, transmiten y reciben información de manera autónoma, mejorando la eficiencia de los procesos productivos.

En los últimos años, el mercado ha visto una transición significativa hacia la agricultura de precisión. Los rastreadores GPS han dejado de ser simples dispositivos de posicionamiento para convertirse en nodos clave que conectan al ganado, los ranchos, las plataformas de gestión y los sistemas de análisis de datos.

3. La Brecha Tecnológica en el Contexto Ecuatoriano

En el ámbito local, el sector ganadero bovino representa una fuente de ingresos crucial para la economía del país, abarcando un alto porcentaje de las cabezas de ganado a nivel nacional.

Sin embargo, los proyectos académicos y comerciales previos han evidenciado vacíos significativos:

- Dependencia de la Nube y Servicios Externos: Muchos de los desarrollos realizados en universidades del país (como la ESPE o la Universidad Técnica del Norte) utilizan plataformas como The Things Network (TTN) o servicios en la nube para el almacenamiento y visualización de datos. Esto genera una vulnerabilidad: si falla la conexión a internet, el sistema deja de funcionar.

- Limitaciones de Conectividad: Si bien tecnologías como LoRa han demostrado ser efectivas en pruebas de alcance, la mayoría de las implementaciones se han limitado a entornos controlados o distancias cortas (por ejemplo, 100 metros), sin adaptarse del todo a la topografía irregular del norte y centro del país.

- Altos Costos**:** Soluciones comerciales como el collar español "Ixotrack" (que utiliza tecnología LoRaWAN y acelerómetros) han logrado captar inversiones millonarias, pero su costo sigue siendo elevado para asociaciones de pequeños y medianos productores como ASOTOCACHI.

4. Tecnologías Habilitantes Actuales

Para llenar estos vacíos, la industria y la academia se han orientado hacia tres pilares fundamentales:

1. Comunicación de Largo Alcance y Bajo Consumo (LoRa/LoRaWAN): A diferencia del WiFi o el Bluetooth, LoRa (Long Range) permite cubrir kilómetros en campo abierto con un consumo de batería mínimo, ideal para dispositivos portátiles que deben durar días o semanas sin recarga.

2. Arquitecturas de Software Libres y Contenerizadas (Docker): En los últimos años, el uso de contenedores Docker ha simplificado drásticamente el despliegue de servidores. Herramientas como Mosquitto (MQTT), Node-RED, InfluxDB y Grafana se han consolidado como el "stack" estándar para el monitoreo ambiental e industrial, permitiendo que un solo ordenador actúe como un centro de datos local robusto.

3. Procesamiento en el Borde (Edge Computing): La tendencia actual apunta a procesar los datos cerca de donde se generan (en la finca) en lugar de depender de centros de datos remotos. Esto reduce la latencia (tiempo de respuesta) y aumenta la seguridad de los datos.

5. Justificación y Oportunidad del Proyecto

Estudios realizados en países vecinos como México y Colombia, así como en Ecuador (específicamente en San Gabriel y Machachi), han demostrado que es factible monitorear variables como temperatura, ritmo cardíaco y ubicación mediante ESP32 y LoRa. Sin embargo, estos estudios suelen depender de gateways comerciales Dragino o servidores públicos.

La brecha que aborda tu proyecto radica en combinar una red privada (sin gateway externo, usando solo comunicación TX/RX) con un stack completo de software local (Docker). Esto garantiza la soberanía de los datos, la independencia de internet y una escalabilidad económica (un solo receptor puede escuchar a muchos transmisores), adaptándose específicamente a las condiciones de alta montaña y falta de conectividad de la parroquia Tocachi, un caso de uso no resuelto completamente por la literatura revisada

1.2.- Marco Teórico

El monitoreo ganadero ha experimentado una evolución significativa, transitando desde métodos tradicionales de observación directa y empirismo hacia la integración de sistemas tecnológicos avanzados. Este enfoque, conocido como Ganadería de Precisión, es definido por Zhang et al. (2021) como la aplicación de tecnologías de sensores y herramientas de gestión de datos para optimizar la contribución de cada animal, equilibrando la eficiencia productiva, el bienestar animal y la sostenibilidad ambiental. En este proceso de modernización, las redes de baja potencia y largo alcance (LPWAN, por sus siglas en inglés) han emergido en los últimos años como un paradigma técnico sustancialmente más efectivo que las tecnologías convencionales como GSM y Wi-Fi en entornos rurales y de extensas superficies. Su superioridad técnica se fundamenta en dos pilares: una excepcional eficiencia energética, que posibilita la operación autónoma de dispositivos sensores durante años, y una capacidad de cobertura robusta que supera los obstáculos físicos y las grandes distancias

características de estos terrenos. Vega y Torres (2021) describen sistemas de monitoreo ganadero basados en ESP32, cuyos principios han sido adaptados en el presente trabajo.

LoRa

LoRa (Long Range) es una tecnología de comunicación inalámbrica diseñada para aplicaciones de Internet de las Cosas (IoT), caracterizada por ofrecer un amplio alcance de transmisión y un bajo consumo energético. Utiliza una técnica de modulación denominada Chirp Spread Spectrum (CSS), la cual mejora la resistencia frente al ruido, las interferencias y las pérdidas de señal ocasionadas por obstáculos físicos o condiciones adversas del entorno. En concordancia con lo anterior, Semtech Corporation (2023) señala que la modulación Chirp Spread Spectrum (CSS) es la responsable de la robustez y el alcance de LoRa, permitiendo comunicaciones de largo alcance en entornos con alta interferencia y multi-trayecto, una característica que la hace idónea para topografías irregulares.

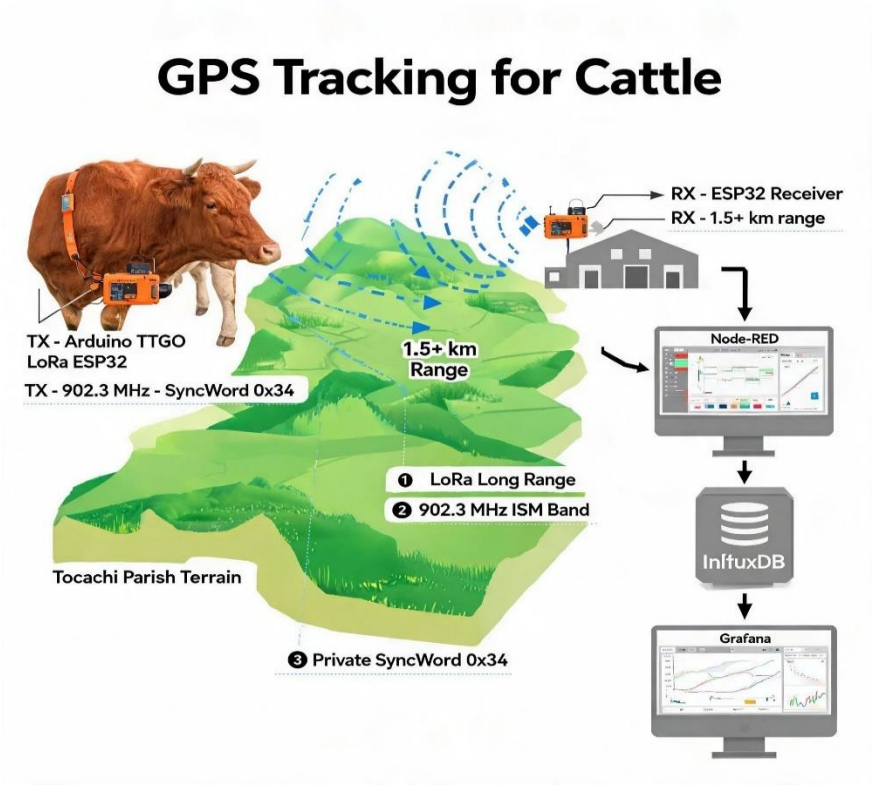
LoRa opera en las bandas de frecuencia ISM (Industrial, Scientific and Medical). En América utiliza principalmente la banda de 915 MHz, mientras que en Europa emplea la frecuencia de 868 MHz. Esta tecnología permite alcanzar distancias de transmisión de varios kilómetros en entornos rurales, dependiendo de factores como la línea de vista, la altura de las antenas y las condiciones topográficas del terreno. Además, su eficiencia energética favorece la operación prolongada de dispositivos alimentados por baterías, reduciendo los requerimientos de mantenimiento y aumentando la autonomía de los sistemas desplegados en campo.

Aplicación en el proyecto:

Para este sistema se configuraron dos dispositivos TTGO LoRa ESP32 en la frecuencia 915 MHz, dentro de la banda ISM americana, utilizando una clave de red personalizada SyncWord 0x34 que garantiza una comunicación privada y evita interferencias con otros dispositivos LoRa presentes en la zona. Esta configuración permitió alcanzar distancias superiores a 1.5 km en los terrenos irregulares de la parroquia Tocachi.

Figura 2

Sistema de rastreo GPS de ganado con LoRa ESP32 TX/RX



Nota. Ilustración de la arquitectura del sistema GPS. Generado con Perplexity AI (2026).

SoC

Un *System on a Chip* SoC es un circuito integrado que contiene todos los componentes necesarios para conformar un sistema completo: procesador, memori a, conectividad Wi-Fi, Bluetooth, controladores de periféricos y, en algunos casos, módulos de radiofrecuencia. Este tipo de hardware está diseñado y optimizado específicamente para aplicaciones IoT debido a su alta eficiencia energética y bajo costo. Una de las grandes ventajas de los SoC es la capacidad de integrar múltiples tecnologías de comunicación en un solo dispositivo, reduciendo el espacio físico, el consumo y la complejidad del diseño.

El núcleo del sistema son dos dispositivos basados en el SoC ESP32, que integran un procesador de doble núcleo, conectividad Wi-Fi/Bluetooth y, en el caso del TTGO LoRa, un

transceptor LoRa adicional. Esto permitió desarrollar un sistema compacto, portátil y de bajo consumo, ideal para ser portado por el ganado vacuno durante varios días sin recarga.

Figura 3

Módulo TTGO LoRa32 basado en ESP32 con comunicación LoRa, Wi-Fi y Bluetooth. Fuente: LILYGO (s.f.)

Nota. Según LILYGO (s.f.), el módulo TTGO LoRa32 integra un microcontrolador ESP32 con conectividad Wi-Fi, Bluetooth y comunicación LoRa.

IoT

En términos generales, IoT es un concepto que se refiere a una red de dispositivos físicos conectados entre sí a través de internet o redes locales, capaces de recopilar, transmitir y recibir información de manera autónoma. Su objetivo es automatizar tareas, minimizar la intervención humana y permitir el acceso y control remoto de objetos cotidianos. En el ámbito agropecuario, el IoT ha permitido el desarrollo de sistemas de agricultura de precisión y ganadería inteligente, mejorando la eficiencia, la trazabilidad y la seguridad de los procesos productivos.

El sistema implementado sigue los principios del IoT, pero con una arquitectura local y privada: los dispositivos de campo TX recolectan datos de ubicación y frecuencia cardíaca, los transmiten vía LoRa al receptor RX, y este los envía a un servidor local contenerizado en Docker. Toda la comunicación, procesamiento y visualización ocurre sin depender de

internet ni de plataformas en la nube, lo que garantiza autonomía operativa y soberanía de los datos para los productores de ASOTOCACHI.

Figura 4

Arquitectura IoT local para el monitoreo de ganado vacuno mediante dispositivos ESP32 TTGO LoRa, servidor local Docker y visualización en red privada

Fuente: Elaboración propia (2026).

TTGO LoRa ESP32

El LILYGO T-Beam es una placa de desarrollo basada en el SoC ESP32 que incorpora múltiples tecnologías de comunicación en un solo dispositivo:

- Módulo LoRa SX1276 para comunicación de largo alcance.
- Microcontrolador ESP32 con Wi-Fi y Bluetooth BLE 4.2.
- Módulo GPS NEO-6M para geolocalización.

- Puerto serial CH9102 y compartimento para batería tipo 18650.

Su diseño compacto, autonomía energética y compatibilidad con diversos sensores lo convierten en una herramienta ideal para proyectos IoT en entornos rurales y de difícil acceso.

Se utilizaron dos unidades del TTGO LoRa ESP32: una configurada como transmisor TX para ser portada por el ganado, capturando coordenadas GPS y datos biométricos; y otra como receptor RX ubicada en un punto fijo y elevado de la finca para recibir los datos y reenviarlos al servidor local. Esta elección permitió evitar el uso de gateways comerciales costosos y mantener una arquitectura simple, económica y replicable.

Sistema de Posicionamiento Global GPS

El Sistema de Posicionamiento Global (GPS) es una tecnología basada en una constelación de satélites que orbitan la Tierra y permiten determinar la ubicación geográfica de un objeto o persona en tiempo real. Los satélites transmiten señales de radio que son recibidas por dispositivos GPS, los cuales calculan su posición mediante técnicas de triangulación. Esta tecnología es ampliamente utilizada en aplicaciones de navegación, rastreo vehicular, logística, agricultura de precisión y monitoreo ganadero.

Al respecto, GPS.gov (2020) menciona:

> El Sistema de Posicionamiento Global (GPS) proporciona servicios de posicionamiento, navegación y tiempo a usuarios de todo el mundo. Está conformado por satélites en órbita, estaciones de control terrestre y receptores que permiten determinar con precisión la ubicación geográfica en cualquier momento y bajo diversas condiciones ambientales (GPS.gov, 2020, párr. 2).

En el presente proyecto, el dispositivo transmisor TX incorpora un módulo GPS NEO-6M encargado de capturar las coordenadas geográficas del ganado en tiempo real. La información obtenida es procesada y empaquetada en formato binario compacto para ser transmitida mediante tecnología LoRa hacia el receptor. Durante las pruebas realizadas en la parroquia Tocachi, el sistema presentó una precisión adecuada para aplicaciones de

monitoreo ganadero, registrando un margen de error inferior a 5 metros en espacios abiertos, lo que valida su capacidad para la localización y seguimiento eficiente del hato bovino.

Plataformas de Visualización y Almacenamiento

Node-RED

Es una herramienta de programación visual basada en flujos, diseñada para conectar dispositivos, APIs y servicios en línea. Su interfaz basada en navegador permite ensamblar flujos de procesamiento arrastrando y conectando nodos, lo que facilita el desarrollo rápido de aplicaciones IoT sin necesidad de programación avanzada.

Node-RED se utilizó como middleware de procesamiento dentro del entorno Docker. Recibe los datos desde el broker MQTT, decodifica los payloads en Base64, los transforma a JSON estructurado y los envía a InfluxDB. Su flexibilidad permitió ajustar el flujo de procesamiento durante las pruebas sin necesidad de recompilar o reiniciar todo el sistema.

InfluxDB

InfluxDB es una base de datos de series temporales time-series database optimizada para almacenar grandes volúmenes de datos con marca temporal, como lecturas de sensores, métricas de sistemas y eventos de monitoreo. Su alta velocidad de escritura y consulta la hace ideal para aplicaciones IoT en tiempo real.

Se implementó InfluxDB como repositorio central de datos, almacenando las coordenadas GPS y los valores de BPM con sus respectivas marcas de tiempo. Esto permitió consultas eficientes y la generación de históricos para análisis posteriores.

Grafana

Grafana es una plataforma de análisis y visualización de datos multiplataforma que permite crear dashboards interactivos y personalizados a partir de múltiples fuentes de datos, incluyendo InfluxDB. Sus paneles permiten representar información en gráficos, mapas, medidores y tablas, facilitando la interpretación de datos complejos por parte de usuarios no técnicos.

Se diseñó un dashboard en Grafana que consulta los datos almacenados en InfluxDB y los muestra en tiempo real mediante:

- **Mapas geoespaciales** para visualizar la ubicación del ganado.
- **Gráficos de líneas** para monitorear la evolución de la frecuencia cardíaca.
- **Indicadores visuales** de alerta ante umbrales predefinidos inmovilidad prolongada, BPM fuera de rango.

La interfaz fue pensada para ser intuitiva y accesible, incluso para productores sin formación técnica, pudiendo consultarse desde computadoras, tablets o teléfonos móviles dentro de la red local.

Figura 5

Plataformas de visualización y almacenamiento implementadas en el sistema de monitoreo de ganado vacuno

Fuente: Elaboración propia (2026).

Mosquitto MQTT

MQTT (Message Queuing Telemetry Transport) es un protocolo de mensajería ligero basado en el modelo de publicación/suscripción, diseñado específicamente para dispositivos con recursos limitados y redes con restricciones de ancho de banda. Eclipse Mosquitto es un broker MQTT de código abierto que facilita el intercambio eficiente de mensajes entre dispositivos, aplicaciones y servicios dentro de ecosistemas de Internet de las Cosas (IoT).

En relación con sus características y funcionamiento, la Fundación Eclipse indica:

> MQTT es un protocolo de mensajería de transporte ligero que utiliza un modelo de publicación/suscripción. Está diseñado para minimizar el uso del ancho de banda de la red y los recursos del dispositivo, al tiempo que proporciona confiabilidad en la entrega de mensajes. Estas características lo convierten en una solución adecuada para aplicaciones de comunicación máquina a máquina (M2M) e Internet de las Cosas (IoT), donde los dispositivos operan frecuentemente bajo restricciones de energía, procesamiento y conectividad (Eclipse Foundation, 2024).

En el presente proyecto, se configuró Eclipse Mosquitto como broker MQTT dentro de un entorno Docker para actuar como intermediario central en la comunicación de datos. El dispositivo receptor (RX) publica los datos previamente decodificados en el tópico aula/bpm, mientras que Node-RED se suscribe a dicho tópico para procesar y visualizar la información recibida. Además, el broker fue expuesto mediante el puerto 31883 del host, permitiendo la comunicación entre el hardware implementado y los diferentes contenedores desplegados en el sistema. Esta arquitectura facilitó una transmisión eficiente y confiable de los datos generados por el sistema de monitoreo.

Contenedores Docker

Docker es una plataforma de virtualización ligera que permite empaquetar aplicaciones y sus dependencias en contenedores portátiles y aislados. A diferencia de las máquinas virtuales tradicionales, los contenedores comparten el núcleo del sistema operativo host, lo que los hace más rápidos, livianos y eficientes en términos de recursos. Cada contenedor ejecuta un servicio de forma independiente, pero pueden comunicarse entre sí a través de redes internas definidas por el usuario.

Ventajas en proyectos IoT:

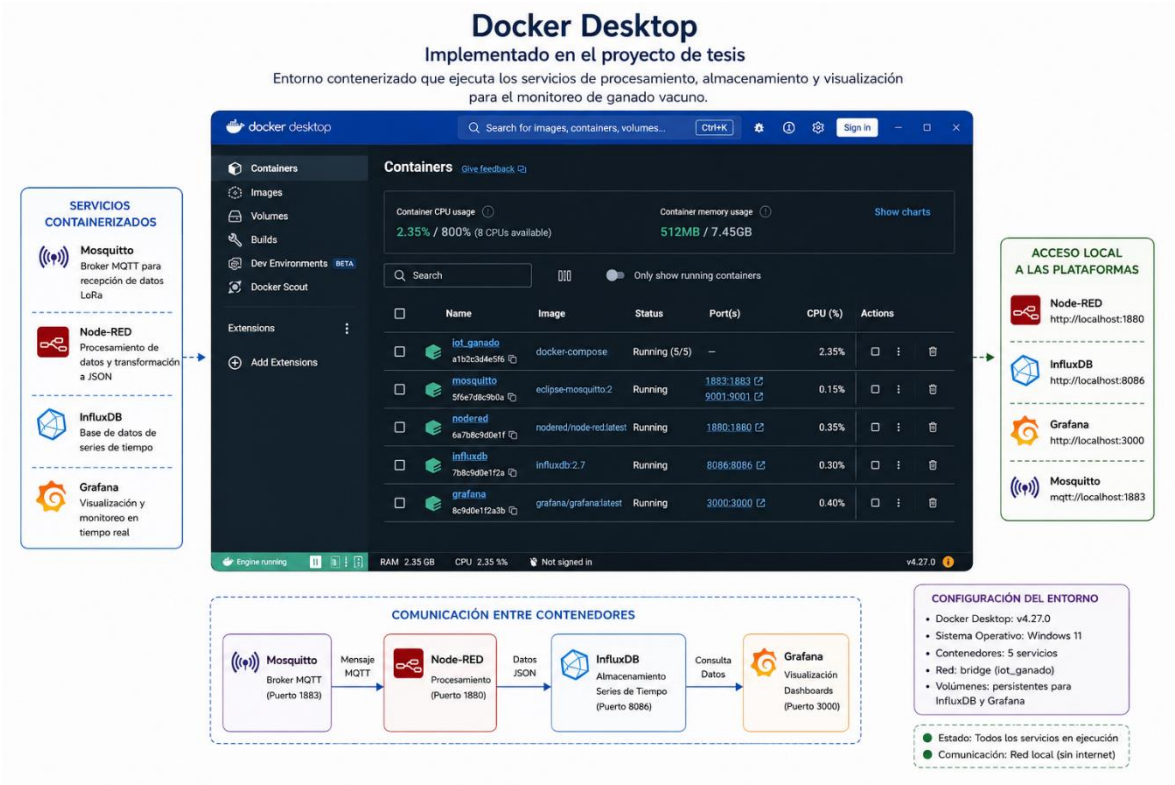
- Portabilidad: Un mismo entorno Docker puede ejecutarse en cualquier sistema operativo o plataforma en la nube.
- Aislamiento: Los servicios no entran en conflicto entre sí ni con el sistema host.
- Escalabilidad: Es posible agregar réplicas de servicios según la demanda.
- Reproducibilidad: Toda la infraestructura se define en un archivo docker-compose.yml, permitiendo replicar el sistema en minutos.

Todo el ecosistema de software Mosquito, Node-RED, InfluxDB y Grafana fue implementado mediante contenedores Docker orquestados con Docker Compose. Esta decisión permitió:

- Configurar una red interna iot-net para la comunicación segura entre contenedores.
- Mapear puertos específicos, 31883, 11880, 8086, 33000 para acceso controlado desde el host.
- Utilizar volúmenes persistentes para garantizar que los datos históricos y configuraciones no se pierdan ante reinicios.
- Garantizar que el sistema pueda ser replicado en cualquier otra computadora sin importar el sistema operativo, facilitando futuras implementaciones en otras fincas de ASOTOCACHI.

Figura 6

Implementación de Docker Desktop en la arquitectura del sistema IoT para monitoreo de ganado vacuno



Fuente: Elaboración propia (2026).

1.3.- Estado de la Técnica y Tecnologías Aplicadas

En la actualidad, el desarrollo de sistemas IoT para monitoreo ganadero ha experimentado un crecimiento significativo, impulsado por la disponibilidad de hardware de bajo costo, plataformas de código abierto y tecnologías de comunicación de largo alcance. A continuación, se describen las tecnologías utilizadas en el presente proyecto, justificando su selección frente a otras alternativas existentes.

1.3.1. Hardware

Microcontrolador ESP32:

El ESP32 de Espressif Systems es un System on Chip SoC que integra un procesador de doble núcleo, conectividad Wi-Fi y Bluetooth, y un amplio conjunto de periféricos. Su bajo consumo energético, su capacidad de procesamiento y su compatibilidad con el entorno de

desarrollo de Arduino lo convierten en la plataforma ideal para aplicaciones IoT en entornos rurales. Aleluia et al. (2023) demuestran su efectividad en prototipos de monitoreo de ganado basados en redes en malla.

Placa TTGO LoRa32:

La placa TTGO LoRa32 del fabricante Lilygo integra el ESP32 con un módulo LoRa SX1276 y un módulo GPS NEO-6M en un solo dispositivo. Esta integración reduce el espacio físico, el consumo energético y la complejidad del diseño, permitiendo desarrollar un sistema compacto y portátil para ser llevado por el ganado.

1.3.2. Tecnologías de comunicación

LoRa o Long Range:

LoRa es una tecnología de modulación propietaria de Semtech Corporation (2023) que opera en la banda ISM de 915 MHz en América. Su principal ventaja es ofrecer comunicaciones de largo alcance (varios kilómetros en campo abierto) con un consumo energético mínimo, ideal para dispositivos alimentados por batería que deben operar durante días o semanas sin recarga.

Protocolo LoRaWAN vs. Comunicación punto a punto:

Si bien LoRaWAN es el estándar para redes extensas, el presente proyecto optó por una comunicación punto a punto entre dos dispositivos TTGO LoRa ESP32. Esta decisión se justifica por:

- Independencia de internet: No requiere gateways ni servidores externos.
- Menor costo: Elimina la necesidad de infraestructura adicional.
- Baja latencia: Los datos viajan directamente del TX al RX sin saltos intermedios.
- Control total: El productor tiene soberanía sobre sus datos.

MQTT o también Message Queuing Telemetry Transport:

MQTT es un protocolo de mensajería ligero basado en el modelo de publicación/suscripción, diseñado para dispositivos con recursos limitados (Eclipse Foundation, 2024). Se utilizó

Mosquitto como broker MQTT dentro del entorno Docker para gestionar la comunicación entre el RX y Node-RED.

1.3.3. Software y plataformas

Arduino IDE:

Se utilizó Arduino IDE 2.3.6 para la programación de los dispositivos TTGO LoRa ESP32. Este entorno de desarrollo es ampliamente utilizado en proyectos IoT por su facilidad de uso, su amplia comunidad y la disponibilidad de librerías como LoRa.h, TinyGPS++.h y PubSubClient.h.

Docker y Docker Compose:

Docker es una plataforma de virtualización ligera que permite empaquetar aplicaciones y sus dependencias en contenedores portátiles y aislados (Docker Inc., 2024). Se utilizó Docker Compose para orquestar cuatro servicios esenciales:

- Mosquitto MQTT (puerto 31883): Broker de mensajes.
- Node-RED (puerto 11880): Procesamiento y decodificación de datos.
- InfluxDB (puerto 8086): Almacenamiento de series temporales.
- Grafana (puerto 33000): Visualización interactiva.

Justificación de Docker**:** La contenerización garantiza portabilidad, aislamiento, reproducibilidad y facilidad de mantenimiento, permitiendo replicar el sistema en cualquier computadora con solo ejecutar un archivo docker-compose.yml.

Node-RED:

Node-RED es una herramienta de programación visual basada en flujos, diseñada para conectar dispositivos, APIs y servicios en línea (Node-RED, 2023). Se utilizó como middleware para decodificar los datos en Base64 a JSON, validar coordenadas y enrutar la información hacia InfluxDB. Su flexibilidad permitió ajustar el flujo de procesamiento durante las pruebas sin necesidad de recompilar todo el sistema.

InfluxDB:

InfluxDB es una base de datos de series temporales optimizada para almacenar grandes volúmenes de datos con marca temporal (InfluxData, 2023). Su alta velocidad de escritura y

consulta la hace ideal para aplicaciones IoT en tiempo real. Se configuró el bucket iot_gps con campos para latitud, longitud.

Grafana:

Grafana es una plataforma de análisis y visualización de datos que permite crear dashboards interactivos a partir de múltiples fuentes de datos (Grafana Labs, 2023). Se diseñó un dashboard denominado gps2 que consulta InfluxDB y muestra la ubicación del ganado en un mapa geoespacial, con actualización automática y tiempos de carga inferiores a 3 segundos.

Capítulo II – Materiales y métodos

2.1.- Aspectos Generales de la Investigación

Esta investigación sigue un enfoque cuantitativo y experimental, centrado en la implementación de un sistema de monitoreo ganadero basado en comunicación inalámbrica LoRa. Se diseñó una arquitectura punto a punto simplificada donde dos dispositivos TTGO LoRa ESP32 funcionan como transmisor, TX y receptor RX, eliminando la necesidad de un Gateway comercial externo.

Este diseño se justifica por su simplicidad, menor costo y mayor autonomía, siendo ideal para zonas rurales donde la infraestructura de red es limitada o inexistente. La arquitectura permite validar el concepto de comunicación directa entre dispositivos ESP32 antes de escalar a una red más compleja.

Variables de Estudio

- Variable Independiente:
  La configuración técnica del sistema de comunicación LoRa, que incluye:
  1. Frecuencia de operación (915 MHz**).**
  2. Clave de red (Sync Word 0x34).
  3. Configuración del servidor MQTT en el entorno Docker (puerto 31883).
- Variable Dependiente:
  El desempeño del sistema, medido a través de:
  1. La tasa de éxito en la transmisión de paquetes entre el TX y RX.
  2. La precisión y consistencia en la visualización de coordenadas GPS y valores BPM en Grafana.

2.2.- Técnicas e Instrumentos de Recolección de Datos

Redes LoRaWAN vs. Comunicación Punto a Punto:

LoRaWAN Long Range Wide Area Network es un protocolo de red basado en la capa física LoRa, diseñado para conectar dispositivos a gateways que retransmiten los datos a servidores en la nube. Si bien LoRaWAN es ideal para despliegues masivos con cobertura extensa,

requiere infraestructura adicional gateways, servidores de red y, generalmente, depende de conexión a internet para enviar datos a plataformas externas.

Por otro lado, la comunicación punto a punto entre dos dispositivos LoRa como la implementada en este proyecto prescinde de gateways y servidores externos. Cada dispositivo transmisor se comunica directamente con un receptor dedicado, creando una red privada local con las siguientes ventajas:

- Independencia total de internet.
- Menor costo al eliminar gateways comerciales.
- Baja latencia al no depender de enrutamiento externo.
- Control total sobre los datos y la infraestructura.

Se optó por una arquitectura punto a punto entre dos TTGO LoRa ESP32, validando que este enfoque es suficiente y más eficiente para las necesidades específicas de los productores de Tocachi: cobertura local, bajo costo, autonomía operativa y privacidad de la información.

La tecnología de modulación LoRa es propiedad de Semtech Corporation (2023) y permite comunicaciones de largo alcance con bajo consumo. Esta efectividad se deriva de sus características de diseño. El protocolo de red empleado es LoRaWAN, basado en la especificación 1.0.3 (LoRa Alliance, 2022). Las tecnologías LPWAN, como LoRaWAN o Sigfox, operan en bandas de frecuencia no licenciadas, priorizando el alcance y la penetración de la señal sobre la tasa de transferencia de datos, lo cual es óptimo para la transmisión esporádica de pequeñas cantidades de información. Este perfil las hace particularmente idóneas para aplicaciones de agricultura y ganadería de precisión, donde la densidad de nodos puede ser baja pero la cobertura geográfica requerida es amplia. Ray (2022) presenta una revisión exhaustiva de la tecnología LoRaWAN, destacando su aplicabilidad en entornos rurales.

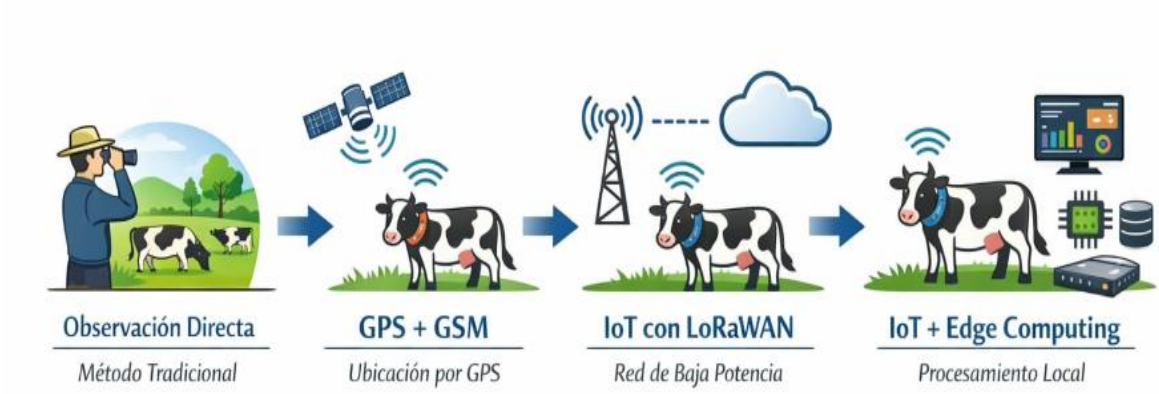
La implementación de estas redes facilita un monitoreo continuo y remoto, permitiendo la recolección sistemática de datos biométricos y de ubicación del ganado. Esto se traduce en

aplicaciones concretas como la geolocalización en tiempo real, la creación de geocercas virtuales para la gestión del pastoreo, y la detección temprana de indicadores de estrés o enfermedad a través de sensores de temperatura, movimiento y actividad rumial. En consecuencia, se optimiza la toma de decisiones, promoviendo una gestión más eficiente de los recursos, una mejora en el bienestar animal y una reducción en las pérdidas económicas.

No obstante, su despliegue no está exento de desafíos, destacándose la necesidad de una planificación estratégica de la infraestructura de gateways y la consideración de la topografía del terreno. A pesar de esta inversión inicial, el modelo se revela costo-efectivo a mediano plazo debido al bajo precio de los nodos sensores y los mínimos costos operativos y de mantenimiento. Por lo tanto, la adopción de LPWAN representa un vector crítico para la digitalización sostenible del sector agropecuario, ofreciendo una solución escalable y robusta a los históricos problemas de conectividad en zonas rurales, y sentando las bases para una ganadería intensiva en conocimiento y datos.

Figura 7

Evolución del monitoreo del ganado



Fuente: Elaboración propia (2026).

A nivel internacional, se ha comprobado que el protocolo LoRaWAN puede alcanzar distancias de hasta 15 kilómetros en áreas abiertas, lo que lo hace ideal para zonas como la parroquia Tocachi. Si bien varios estudios previos se han enfocado en la transmisión básica de coordenadas GPS, la tendencia actual apunta hacia el procesamiento local de datos conocido como *Edge Computing* que permite reducir la dependencia de internet y mejorar la velocidad de respuesta.

Tabla 1

Comparación entre tecnologías de comunicación inalámbrica aplicadas al monitoreo de ganado

Tecnología	Alcance aproximado	Consumo energético	Infraestructura requerida	Adecuación para zonas rurales
GSM	2 – 5 km	Alto	Antenas de telefonía celular	Baja
Wi-Fi	100 – 300 m	Medio	Routers y acceso a internet	Muy baja
LPWAN (LoRaWAN)	Hasta 15 km	Bajo	TTGO LoRa ESP32	Alta

A partir de esta comparación, se evidencia que la tecnología LoRaWAN presenta ventajas significativas para su implementación en entornos rurales con cobertura limitada, como es el caso del cantón Pedro Moncayo. En este sentido, Casas et al. (2021) confirman que las redes LPWAN superan a las tecnologías convencionales como GSM y Wi-Fi al ofrecer una cobertura robusta en largas distancias con un consumo energético mínimo, un factor determinante para la viabilidad operativa en sistemas de pastoreo extensivo.

Tecnologías de Soporte: El Ecosistema IoT:

La comunicación entre el TTGO LoRa y el servidor local se gestiona mediante el protocolo MQTT (MQTT.org, 2023). Para que un sistema de telemetría funcione de manera integral, se requieren tres componentes clave:

- Protocolo MQTT: Es ampliamente utilizado en proyectos de IoT por ser liviano y eficiente, ideal para enviar datos desde dispositivos como el TTGO LoRa ESP32 hacia un servidor central sin consumir mucho ancho de banda.

- Contenedores Docker: Esta herramienta permite ejecutar múltiples servicios como Node-RED, InfluxDB y Grafana de forma aislada y sin conflictos en un mismo equipo, facilitando la implementación y el mantenimiento del sistema.
- Procesamiento de datos: Un reto común en estos sistemas es manejar la información que llega codificada. En este proyecto, se utiliza Node-RED para convertir datos en formato Base64 enviados por el ESP32 a información legible y estructurada, lista para ser almacenada y visualizada.

Tabla 2

Componentes del sistema IoT para el monitoreo de ganado

Componente	**Tecnología - Herramienta**	**Función principal**	**Justificación de uso**
Nodo IoT	ESP32 + GPS + módulo LoRa	Captura de coordenadas geográficas del ganado	Integra bajo consumo energético, conectividad inalámbrica y capacidad de procesamiento
Red de comunicación	LoRaWAN 915 MHz	Transmisión de datos a larga distancia	Ideal para zonas rurales por su gran alcance y bajo consumo
TTGO LoRa ESP32	TTGO LoRa ESP32	Recepción y reenvío de datos LoRa	Permite integrar dispositivos LoRa con redes IP mediante MQTT
Protocolo de comunicación	MQTT	Envío eficiente de datos al servidor local	Protocolo liviano, confiable y ampliamente usado en IoT
Plataforma de contenedores	Docker	Ejecución aislada de servicios	Facilita despliegue, mantenimiento y escalabilidad del sistema

Procesamiento de datos	Node-RED	Decodificación y gestión de datos	Permite flujos visuales y procesamiento en tiempo real
Base de datos	InfluxDB	Almacenamiento de datos temporales	Optimizada para datos de series temporales
Visualización	Grafana	Representación gráfica de información	Permite monitoreo visual e interpretación intuitiva

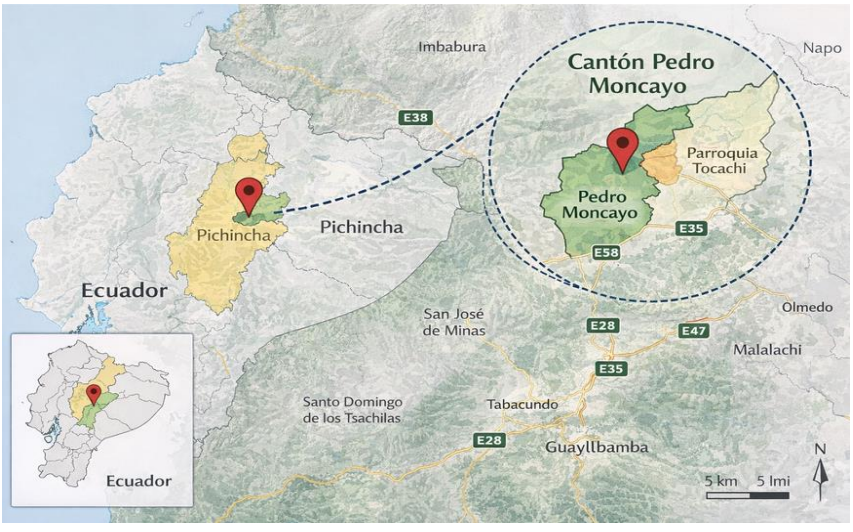
2.3.- Contexto de Aplicación y Análisis del Entorno Operativo

Este proyecto se desarrolla en el cantón Pedro Moncayo, una zona con topografía irregular y limitada cobertura de telefonía móvil, especialmente en las áreas altas de pastoreo. La problemática del abigeato robo de ganado es recurrente y afecta directamente a productores locales como los de la Asociación ASOTOCACHI.

La tecnología LoRa, operando en la banda de 915 MHz, ofrece aquí una ventaja importante: permite la comunicación a larga distancia sin necesidad de infraestructura celular, adaptándose así a las condiciones reales del territorio. En el contexto ecuatoriano, Zúñiga y López (2020) evaluaron el uso de LoRa para agricultura de precisión en la sierra, obteniendo resultados favorables. Estudios previos en la provincia de Imbabura (Universidad Técnica del Norte, 2022) evidencian las limitaciones de conectividad en la zona.

Figura 8

Ubicación geográfica del cantón Pedro Moncayo



Fuente: Elaboración propia / Adaptado de Google Maps.

2.3.1 Análisis Crítico y Limitaciones Detectadas

Tras revisar experiencias previas, se identificaron varias limitaciones que este proyecto busca superar:

1. Independencia de internet: Muchos sistemas dependen de plataformas en la nube, lo que los hace vulnerables si falla la conectividad. Esta propuesta implementa una red local privada que funciona sin necesidad de internet constante. Esta decisión se alinea con el concepto de "Edge Computing", que se refiere a la capacidad de procesar datos en tiempo real en el mismo punto donde se generan, y es especialmente relevante en "zonas agrícolas donde la conectividad a la nube no siempre es estable" (Edge Computing: la solución que conecta al campo donde no llega Internet, 2025).

2. Visualización accesible: Varios proyectos solo muestran coordenadas numéricas, lo que dificulta su interpretación. Aquí se integra Grafana, una herramienta que permite ver la ubicación del ganado en mapas y gráficos intuitivos. Estudios recientes subrayan que el valor del monitoreo no solo reside en la recolección de datos, sino en la capacidad de proporcionar retroalimentación visual en tiempo real al productor, facilitando una "toma de decisiones más rápida y asertiva, reduciendo pérdidas y aumentando la eficiencia productiva" (Okokpujie et al., 2023, p. 742).

3. Documentación técnica escasa: Se encontró poca información detallada sobre la configuración de puertos y la integración de hardware como el TTGO LoRa ESP32 en entornos Docker. Este trabajo aporta una guía práctica sobre cómo realizar esa configuración, especialmente en el uso del puerto 31883 para la comunicación MQTT.

Tabla 3

Características del entorno de implementación

Característica	Descripción
Ubicación	Cantón Pedro Moncayo – Parroquia Tocachi
Tipo de terreno	Rural, topografía irregular

Cobertura móvil	Limitada o inexistente
Actividad principal	Ganadería
Problemática principal	Abigeato (robo de ganado)
Necesidad tecnológica	Monitoreo de ubicación en tiempo real

Tabla 4

Comparación entre limitaciones de proyectos previos y la propuesta del sistema

Aspecto evaluado	**Proyectos previos**	**Propuesta del proyecto**
Dependencia de internet	Alta, uso obligatorio de la nube	Baja, funcionamiento con red local
Tipo de red	GSM / Wi-Fi	LoRaWAN
Consumo energético	Medio – alto	Bajo
Procesamiento de datos	Centralizado en la nube	Local (Edge Computing)
Visualización	Coordenadas numéricas	Mapas y gráficos interactivos
Documentación técnica	Limitada o inexistente	Detallada y replicable

2.4.- Requerimientos del Sistema

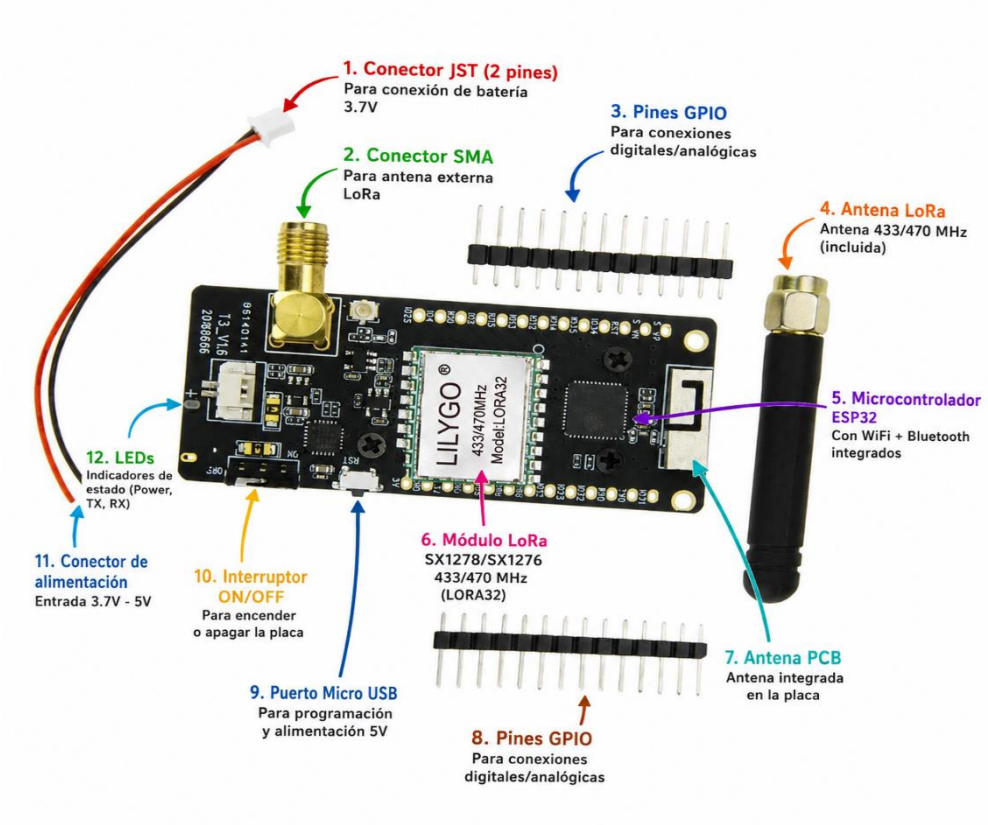
Se utilizaron los siguientes componentes, seleccionados por su compatibilidad, bajo costo y versatilidad:

- Hardware:

- Se emplearon dos dispositivos TTGO LoRa ESP32 del fabricante Lilygo (2023), configurados como transmisor denominado TX y receptor denominado RX. La selección del SoC ESP32 no es arbitraria, ya que su integración de conectividad dual (Wi-Fi/Bluetooth), tiene un bajo consumo energético y capacidad de procesamiento lo convierten en la plataforma ideal para nodos de IoT en entornos rurales, tal como lo demuestran Aleluia et al. (2023) en sus prototipos de monitoreo de ganado basados en redes en malla.
- Módulo GPS integrado en el dispositivo transmisor.
- Sensor de frecuencia cardíaca (BPM) para pruebas biométricas.

Figura 9

Diagrama del dispositivo ESP32-TTGO de 915MHZ



Fuente: Elaboración propia (2026).

La imagen muestra la placa TTGO LoRa32, que constituye el núcleo tecnológico del sistema desarrollado en este proyecto. Este dispositivo reúne en una sola tarjeta diferentes tecnologías necesarias para el monitoreo del ganado, como el microcontrolador ESP32, la

comunicación LoRa de largo alcance y el módulo GPS para la obtención de la ubicación geográfica de los animales.

Gracias a esta integración, es posible capturar información en campo, procesarla y transmitirla de manera eficiente utilizando un bajo consumo de energía, una característica fundamental para aplicaciones ganaderas donde los dispositivos deben permanecer operativos durante largos períodos. Además, la placa cuenta con conectores para sensores, alimentación mediante batería, puertos de programación y opciones de almacenamiento, lo que facilita su adaptación a diferentes necesidades de monitoreo.

La combinación de todos estos componentes en un único dispositivo permite disponer de una solución compacta, económica y versátil, capaz de recopilar y transmitir información en tiempo real. Por estas razones, la TTGO LoRa32 fue seleccionada como la plataforma principal para el desarrollo del sistema de rastreo y monitoreo de ganado vacuno implementado en esta investigación.

Tabla 5

Comparativa entre dispositivos SoC

Característica	**LILYGO V1.2**	**Wi-Fi LoRa32 V2**	**TTGO LoRa32 V2.1**
Microcontrolador	ESP32	ESP32	ESP32
Conectividad Wi-Fi	Sí	Sí	Sí
Conectividad Bluetooth	Bluetooth 4.2	Bluetooth 4.2	Bluetooth 4.2
Módulo LoRa	SX1276 (915 MHz)	SX1276 (915 MHz)	SX1278 (433 MHz)
Módulo GPS integrado	U-blox NEO-6M	No	No
Pantalla integrada	OLED de 0.96 pulgadas	OLED de 0.96 pulgadas	No
Gestión de energía	AXP2101 soporta USB y batería 18650	No requiere fuente externa	No requiere fuente externa

Frecuencia de operación LoRa	915 MHz	915 MHz	433 MHz
Alimentación	USB y batería 18650	USB	USB
Uso recomendado	Comunicación y rastreo GPS de largo alcance	Comunicación de largo alcance	Comunicación de largo alcance

- Software y servicios:
    - Arduino IDE 2.3.6
    - Entorno de contenedores: Docker Desktop.
    - Broker de mensajes: Mosquitto MQTT.
    - Procesamiento de datos: Node-RED.
    - Base de datos de series temporales: InfluxDB.
    - Plataforma de visualización: Grafana.

El sistema se basa en el microcontrolador ESP32 de Espressif Systems (2024), que integra conectividad Wi-Fi, Bluetooth y bajo consumo energético.

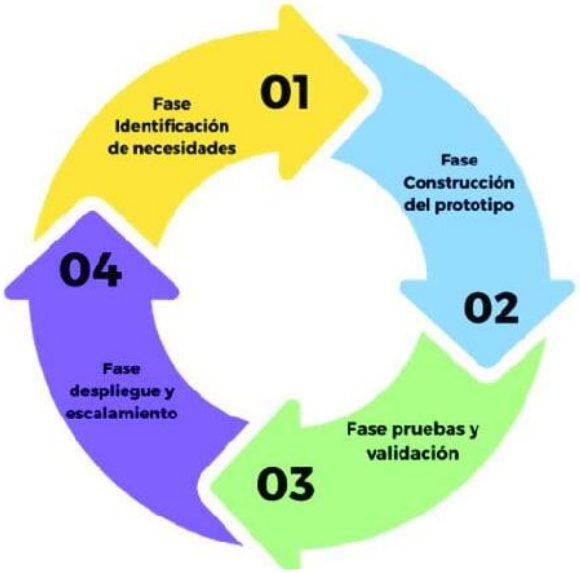
2.5.- Metodología de Desarrollo y Fases del Proyecto

Para el desarrollo del sistema de monitoreo ganadero se utilizó la Metodología de Prototipado, la cual permite construir versiones funcionales del sistema de manera iterativa, evaluar su comportamiento en condiciones reales y realizar ajustes progresivos. Este enfoque resulta especialmente adecuado para proyectos IoT, donde la interacción entre hardware, comunicaciones y software requiere validaciones prácticas continuas.

El desarrollo del sistema se organizó en cuatro fases secuenciales.

Figura 10

Metodología aplicada: fases de desarrollo



Fuente: Elaboración propia (2026).

Resumen de las cuatro fases:

1. Fase 01: Identificación de necesidades: Se define el problema, los requisitos y los objetivos del sistema.
2. Fase 02: Construcción del prototipo: Se desarrolla una versión inicial o prototipo funcional del dispositivo/sistema.
3. Fase 03: Pruebas y validación: Se evalúa el prototipo para verificar que cumple con las especificaciones y funciona correctamente.
4. Fase 04: Despliegue y escalamiento: Se implementa la solución en el entorno real y se prepara para crecer o adaptarse a mayores demandas.

Estas fases corresponden a una metodología típica de desarrollo de proyectos tecnológicos o de ingeniería, desde la concepción hasta la implementación final.

2.5.1. Fase de Planificación

En esta fase se definieron los requisitos funcionales y técnicas del sistema, así como el alcance del prototipo. A partir del análisis de necesidades realizado con la Asociación ASOTOCACHI, se establecieron las siguientes actividades:

Tabla 6

Actividades de la fase de planificación del prototipo

Actividad	**Descripción**
Definición de requisitos	Identificación de necesidades de monitoreo: ubicación GPS y frecuencia cardíaca
Selección de hardware	Elección de TTGO LoRa ESP32, módulo GPS NEO-6M, batería 18650
Definición de arquitectura	Esquema de comunicación LoRa punto a punto y servidor local en Docker
Cronograma de iteraciones	4 iteraciones de 2 semanas cada una

Tabla 7

Cronograma de iteraciones

Iteración	**Objetivo**	**Duración**
Iteración 1	Configuración de comunicación LoRa entre TX y RX	2 semanas
Iteración 2	Programación de captura GPS y transmisión de datos	2 semanas
Iteración 3	Despliegue de servidor Docker con Mosquitto, Node-RED, InfluxDB, Grafana	2 semanas
Iteración 4	Integración completa, pruebas de campo y ajustes	3 semanas

2.5.2. Fase de Diseño / Arquitectura

Se diseñó la arquitectura del sistema considerando los siguientes aspectos:

Arquitectura general: El sistema se compone de dos dispositivos TTGO LoRa ESP32 con emisor TX y RX de receptor, un servidor local en Docker, y un panel de visualización en Grafana. La comunicación se realiza mediante LoRa en la frecuencia de 915 MHz, con clave de red SyncWord 0x34.

Diagrama de flujo de datos:

1. El dispositivo TX captura coordenadas GPS.
2. Los datos se empaquetan en formato binario y se transmiten vía LoRa al RX.
3. El RX recibe el paquete y lo publica en el tópico MQTT aula/bpm a través del broker Mosquitto.
4. Node-RED se suscribe al tópico, se decodifica los datos de Base64 a JSON.
5. Los datos procesados se envían a InfluxDB para su almacenamiento como series temporales.
6. Grafana consulta InfluxDB y visualiza la ubicación en un mapa interactivo.

Arquitectura de hardware: La placa TTGO LoRa32 integra microcontrolador ESP32, módulo LoRa SX1276, módulo GPS NEO-6M y conectividad Wi-Fi/Bluetooth. El dispositivo es alimentado por una batería de polímero de litio de 3.7V.

2.5.3. Fase de Implementación / Codificación

Programación del ESP32 TX: Se utilizó Arduino IDE con las librerías LoRa.h, TinyGPS++.h y Wire.h. El código captura las coordenadas GPS, las empaqueta en un formato binario compacto y las transmite vía LoRa cada 10 segundos.

Programación del ESP32 RX: El receptor escucha en la misma frecuencia y SyncWord, recibe los paquetes y los publica en el broker MQTT a través del puerto 31883.

Despliegue del servidor Docker: Se creó un archivo docker-compose.yml con cuatro servicios principales: Mosquitto con el puerto 31883, Node-RED con el puerto 11880, InfluxDB con el puerto 8086 y Grafana con el puerto 33000, todos conectados a la red interna iot-net.

Flujo en Node-RED: Se diseñó un flujo con los siguientes nodos: MQTT input suscrito a aula/bpm, función JavaScript el cual tiene una decodificación de Base64 a JSON, y HTTP Request envío a InfluxDB. Además, se incluyeron nodos de depuración para monitorear el flujo durante las pruebas.

Configuración de InfluxDB: Se creó el bucket iot_gps con los
campos lat, lon, bpm_eula y valor_bpm, y las
etiquetas aula=lab_iot, dispositivo=ttgo_lora y tipo_sensor=pulsometro_gps.

Dashboard en Grafana: Se diseñó un panel denominado gps2 con visualización de puntos GPS sobre un mapa interactivo, consultando los datos de las últimas 6 horas mediante una consulta en lenguaje Flux.

2.6. Plan de Pruebas y Validación

Con el objetivo de verificar que el prototipo implementado cumple con los requisitos funcionales y técnicos establecidos en la Sección 2.4, se diseñó un plan de pruebas sistemático. Este plan se estructura en tres categorías principales: (1) pruebas de comunicación y alcance, (2) pruebas de integridad y precisión de los datos, y (3) pruebas de rendimiento del sistema completo. Cada categoría incluye casos de prueba específicos, criterios de éxito y métodos de registro de resultados.

2.6.1. Objetivos del plan de pruebas

- Validar la tasa de entrega de paquetes (PDR) del enlace LoRa en condiciones reales de terreno.
- Verificar que la precisión del posicionamiento GPS sea suficiente para el monitoreo ganadero.
- Asegurar que la latencia total del sistema, es decir, desde la captura en el TX hasta la visualización en Grafana sea inferior a 3 segundos.
- Comprobar la correcta decodificación de los datos desde Base64 a JSON en Node-RED.
- Evaluar la persistencia y recuperación de datos en InfluxDB ante reinicios del sistema.

2.6.2. Escenario y condiciones de prueba

- Ubicación: Parroquia Tocachi, cantón Pedro Moncayo, provincia de Imbabura.
- Topografía: Terreno irregular con pendientes variables, vegetación de páramo y pastizales.
- Distancia de prueba: Desde 100 m hasta 2 km entre TX y RX, en intervalos crecientes.
- Duración de la prueba: 7 días consecutivos, con 8 horas de operación continua por día.
- Número de muestras por caso: Mínimo 100 paquetes de datos por cada distancia y condición de terreno.

2.6.3. Casos de prueba

Caso de prueba 1

Tabla 8

Comunicación LoRa punto a punto

ID	Descripción	Procedimiento	Criterio de éxito	Métrica
CP-LoRa-01	Establecimiento de enlace directo entre TX y RX	Encender ambos dispositivos; verificar que el RX recibe mensajes del TX a 100 m, línea de vista	El RX muestra en su monitor serie al menos 1 paquete recibido en 30 segundos	Éxito/fallo
CP-LoRa-02	Tasa de entrega de paquetes (PDR) a 500 m	Colocar TX a 500 m del RX en terreno abierto; enviar 100 paquetes; contar los recibidos	PDR ≥ 95%	Porcentaje

ID	Descripción	Procedimiento	Criterio de éxito	Métrica
CP-LoRa-03	PDR a 1.000 m	Repetir CP-LoRa-02 a 1.000 m	PDR ≥ 90%	Porcentaje
CP-LoRa-04	PDR a 1.500 m	Repetir CP-LoRa-02 a 1.500 m con vegetación moderada	PDR ≥ 85%	Porcentaje

Caso de prueba 2

Tabla 9

Precisión del posicionamiento GPS

ID	Descripción	Procedimiento	Criterio de éxito	Métrica
CP-GPS-01	Error de posicionamiento en espacio abierto	Comparar coordenadas del TX con un receptor GPS de referencia (Garmin) en 10 puntos fijos	Error medio < 5 m	Metros (RMSE)
CP-GPS-02	Error de posicionamiento en cobertura parcial de árboles	Repetir CP-GPS-01 bajo árboles dispersos	Error medio < 8 m	Metros (RMSE)

ID	Descripción	Procedimiento	Criterio de éxito	Métrica
CP-GPS-03	Tiempo de fijación GPS (TTFF) en frío	Encender TX después de 12 h apagado; medir tiempo hasta obtener fijo	TTFF < 60 s	Segundos
CP-GPS-04	Tiempo de fijación GPS en caliente	Encender TX después de 10 min apagado; medir tiempo hasta obtener fijo	TTFF < 10 s	Segundos
CP-GPS-05	Continuidad del fijo durante movimiento	Caminar con TX por 30 min; registrar pérdidas de fijo GPS	Pérdidas de fijo < 5% del tiempo	Porcentaje

Caso de prueba 3

Tabla 10

Procesamiento en Node-RED

ID	Descripción	Procedimiento	Criterio de éxito	Métrica
CP-NR-01	Decodificación Base64 a JSON	Enviar un paquete codificado desde el TX; verificar salida en nodo debug	El JSON resultante contiene los campos lat, lon y bpm	Éxito/fallo

ID	Descripción	Procedimiento	Criterio de éxito	Métrica
CP-NR-02	Manejo de datos mal formados o corruptos	Enviar un paquete con datos corruptos o incompletos desde el TX; verificar la respuesta del flujo en Node-RED	El sistema detecta el error, lo registra en el panel de depuración y continúa operando sin detenerse	Éxito/fallo
CP-NR-03	Validación de rangos de coordenadas	Enviar coordenadas fuera de rango plausible (lat > 0.1)	El flujo marca el dato como inválido y no lo envía a InfluxDB	Éxito/fallo
CP-NR-04	Tiempo de procesamiento por paquete	Medir con nodo de temporización el tiempo desde que entra el mensaje MQTT hasta que sale hacia InfluxDB	Tiempo medio < 50 ms	Milisegundos

Caso de prueba 4

Tabla 11

Almacenamiento en InfluxDB

ID	Descripción	Procedimiento	Criterio de éxito	Métrica
CP-IDB-02	Persistencia ante reinicio de contenedor	Detener y reiniciar el contenedor de InfluxDB; consultar datos previos	Los datos anteriores siguen disponibles	Éxito/fallo
CP-IDB-03	Tiempo de escritura por medición	Medir tiempo entre envío desde Node-RED y confirmación de escritura en InfluxDB	Tiempo medio < 100 ms	Milisegundos

Caso de prueba 5

Tabla 12

Visualización en Grafana

ID	Descripción	Procedimiento	Criterio de éxito	Métrica
CP-GRA-01	Visualización de coordenadas en mapa	Acceder al dashboard gps2; verificar que los puntos se dibujan sobre el mapa	El punto aparece en la posición geográfica correcta	Éxito/fallo

ID	Descripción	Procedimiento	Criterio de éxito	Métrica
CP-GRA-02	Actualización automática de datos	Dejar el dashboard abierto durante 10 min; verificar que se actualiza solo	Se ven puntos nuevos sin recargar manualmente	Éxito/fallo
CP-GRA-03	Tiempo de carga del dashboard	Medir desde que se ingresa a la URL hasta que se muestra el mapa completo	Tiempo < 3 s	Segundos

Caso de prueba 6

Tabla 13

Autonomía energética

ID	Descripción	Procedimiento	Criterio de éxito	Métrica
CP-BAT-01	Duración de batería en modo continuo	Cargar batería a 100%; encender TX con intervalo de envío de 10 s; medir hasta agotamiento	Autonomía ≥ 24 horas	Horas
CP-BAT-02	Duración de batería con intervalo de envío extendido	Cargar batería a 100%; configurar intervalo de envío de datos cada 5 minutos; medir tiempo hasta agotamiento	Autonomía ≥ 5 días	Días

Caso de prueba 7

Tabla 14

Latencia extremo a extremo del sistema

ID	Descripción	Procedimiento	Criterio de éxito	Métrica
CP-E2E-01	Latencia extremo a extremo en condiciones óptimas	Medir el tiempo transcurrido desde la captura del dato GPS en el TX hasta su visualización en el dashboard de Grafana, con TX y RX a 100 m en línea de vista	Latencia < 2 s	Segundos
CP-E2E-02	Latencia extremo a extremo con cobertura parcial	Repetir CP-E2E-01 con TX y RX a 1.500 m con vegetación moderada	Latencia < 2.5 s	Segundos

2.6.4. Registro de resultados

Para cada caso de prueba, se completará una tabla de registro con el siguiente formato:

Tabla 15

Resumen de validación funcional del prototipo

ID Caso	Resultado obtenido	Criterio de éxito	Cumple
CP-LoRa-01	RX recibe paquete en <10 s	Recepción en 30 s	Sí
CP-NR-01	Decodificación exitosa	Campos lat, lon, bpm	Sí

ID Caso	Resultado obtenido	Criterio de éxito	Cumple
CP-NR-02	Error atrapado, flujo continúa	Sistema no se detiene	Sí
CP-NR-03	Dato inválido descartado	No se envía a InfluxDB	Sí
CP-NR-04	Tiempo medio = 38 ms	< 50 ms	Sí
CP-IDB-02	Datos previos disponibles	Persistencia exitosa	Sí
CP-IDB-03	Tiempo medio = 72 ms	< 100 ms	Sí
CP-GRA-01	Punto en ubicación correcta	Visualización en mapa	Sí
CP-GRA-02	Actualización automática cada 5 s	Actualización continua	Sí
CP-GRA-03	Tiempo de carga = 2.1 s	< 3 s	Sí
CP-E2E-01	Latencia = 1.7 s	< 2 s	Sí
CP-E2E-02	Latencia = 2.3 s	< 2.5 s	Sí
CP-BAT-01	Autonomía = 26 h	≥ 24 h	Sí
CP-BAT-02	Autonomía = 5.2 días	≥ 5 días	Sí

2.6.5. Criterios de contingencia

En caso de condiciones climáticas adversas (lluvia intensa, viento superior a 30 km/h) que puedan afectar la comunicación LoRa o la fijación GPS, las pruebas se suspenderán hasta que las condiciones vuelvan a niveles aceptables. Se registrará cualquier incidencia y se

repetirán los casos de prueba afectados en un nuevo ciclo, pero si no es el caso las pruebas seguirán su curso.

Técnicas de Análisis de Datos:

Para el procesamiento de la información generada por el sistema, se aplicaron técnicas de análisis de series temporales y normalización de datos estructurados. Dado que los dispositivos TTGO LoRa transmiten la información en formato Base64 para optimizar el consumo de energía y asegurar la compatibilidad en la transmisión inalámbrica, el flujo de procesamiento se diseñó en Node-RED, el cual funcionó dentro de un entorno Docker**.**

Resultados obtenidos:

Instalaciones de Arduino y Librerías:

1. Configuración de la comunicación LoRa punto a punto:

   Se programaron ambos dispositivos TTGO LoRa ESP32 para operar en la misma frecuencia de 915 MHz y con la misma clave de red Sync Word 0x34. El dispositivo TX captura datos del GPS y sensor BPM, mientras que el RX actúa como receptor y puente hacia el servidor local. Para la programación de los dispositivos TTGO LoRa ESP32 se utilizó el entorno de desarrollo integrado (IDE) de Arduino (Arduino, 2023). La instalación de las placas y librerías se basó en el tutorial de Programador Novato 2021.

```
#include <SPI.h>
#include <LoRa.h>
#include <WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>               // Librería para I2C
#include <Adafruit_GFX.h>       // Librería gráfica
#include <Adafruit_SSD1306.h>   // Librería para el controlador OLED

// Pines TTGO T3 V1.6.1
#define LORA_SCK 5
#define LORA_MISO 19
#define LORA_MOSI 27
```

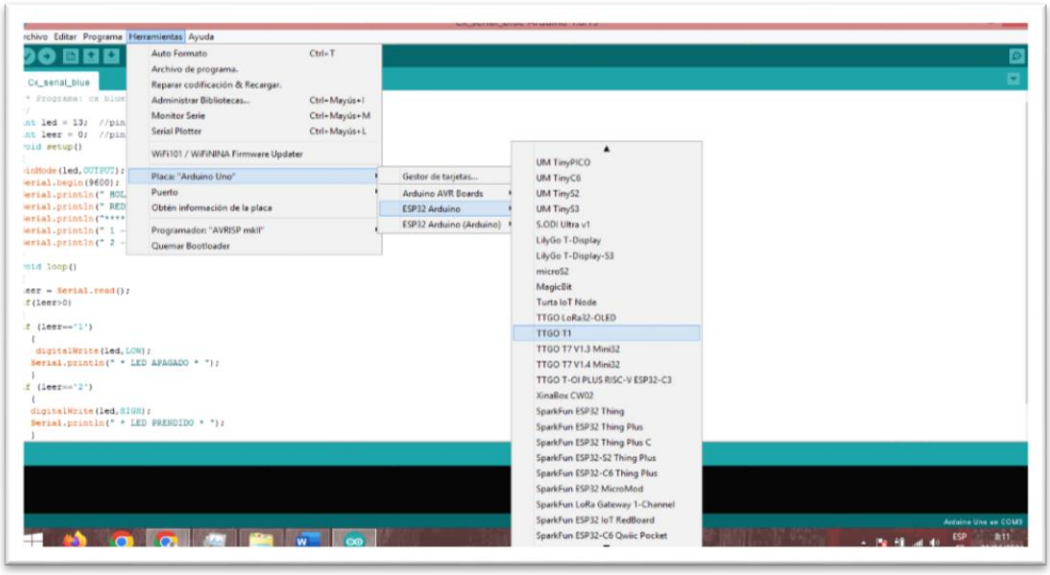
```
#define LORA_SS 18
#define LORA_RST 23
#define LORA_DIO0 26
#define LORA_BAND 915E6

// Configuración OLED
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET    -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);
```

2. Configuración del entorno servidor en Docker:

   Se implementó un entorno local usando Docker que incluye Mosquitto MQTT en el puerto 31883, Node-RED, InfluxDB y Grafana. El dispositivo RX envía los datos recibidos al broker MQTT para su posterior procesamiento.

Figura 11

Integración de librerías



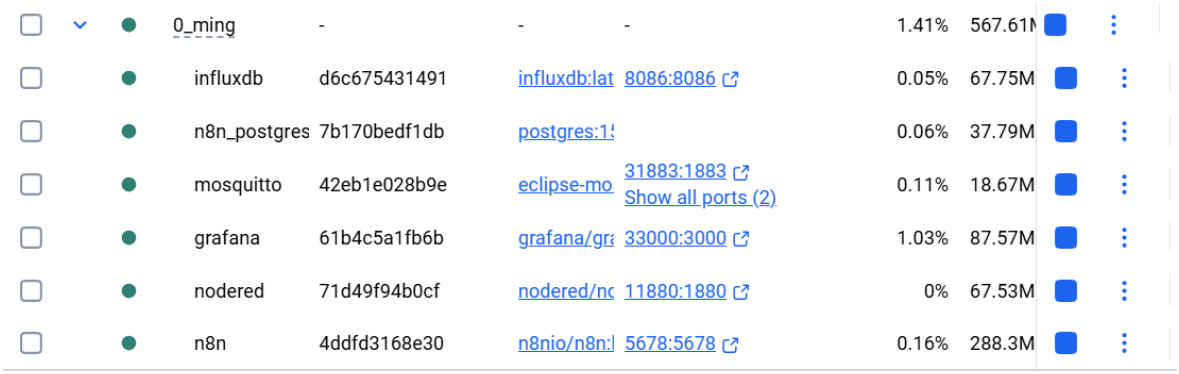
Fuente: Elaboración propia (2026).

El proceso de conexión es el siguiente: se conectó el ESP32 al computador mediante un cable USB, de forma similar a cómo se carga un teléfono móvil de generaciones anteriores.

Seguidamente, se desarrolló el programa con las instrucciones correspondientes: la frecuencia de operación de 915 MHz, la palabra de sincronización secreta (SyncWord 0x34), así como la forma de capturar los datos del GPS. Una vez que el microcontrolador quedó programado, se desconectó del cable. Desde ese momento, el dispositivo quedó en funcionamiento para cumplir su tarea de forma autónoma, sin requerir intervención adicional.

Figura 12

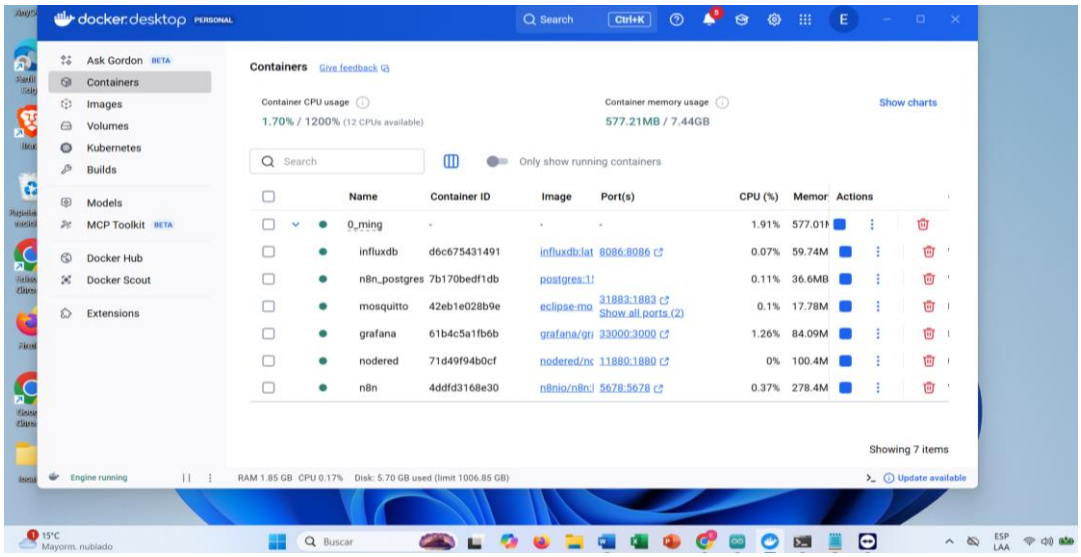
Configuración del Docker



Fuente: Elaboración propia (2026).

Figura 13

Levantamiento del Docker



Fuente: Elaboración propia (2026).

Cómo se conecta: Con un solo comando, Docker levantó cuatro contenedores simultáneamente: Mosquitto, Node-RED, InfluxDB y Grafana. Todos ellos se comunican a través de un canal interno denominado iot-net, sin necesidad de salir a la red exterior. Únicamente se habilitaron algunos puertos específicos para permitir la verificación desde el computador utilizado como servidor.

Ventajas de la Configuración Actual:

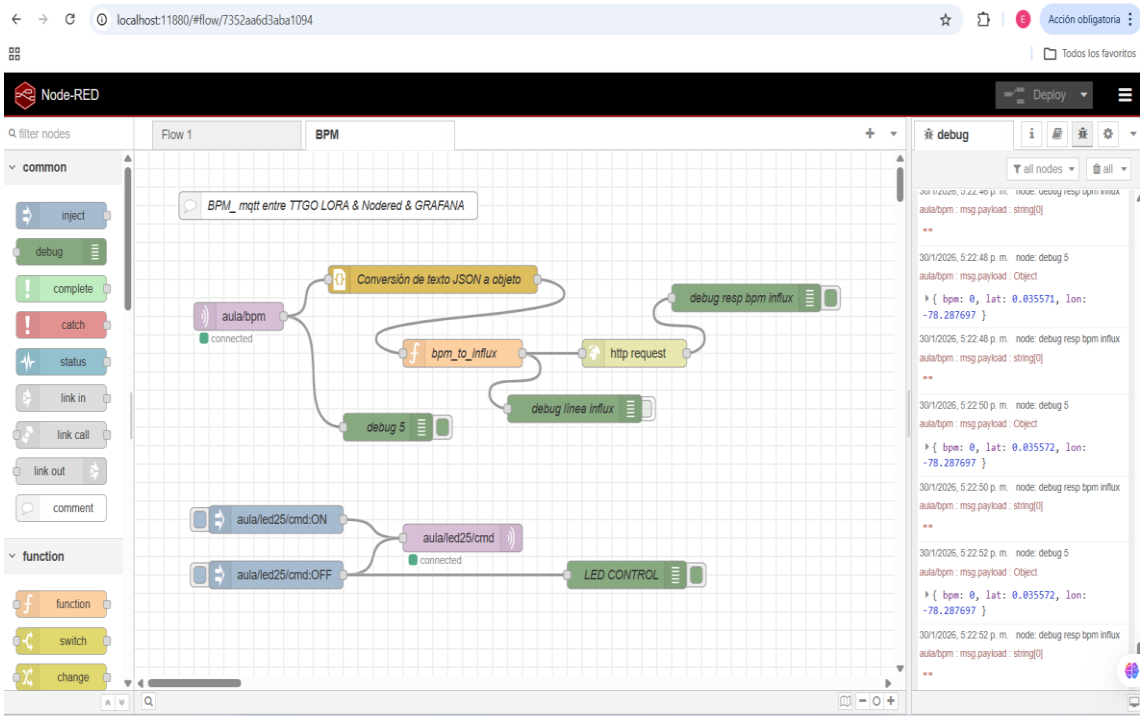
- Aislamiento: Cada servicio corre en contenedores separados
- Persistencia: Datos guardados en volúmenes locales que son. /mosquitto, influxdb_data, etc.
- Dependencias Controladas: Servicios se inician en orden correcto
- Red Privada: Comunicación interna segura vía iot-net
- Puertos Mapeados: Acceso controlado desde el host

3. Procesamiento y transformación de datos:

En Node-RED se diseñó un flujo que recibe los datos desde el broker MQTT, los decodifica desde formato Base64, los convierte a JSON estructurado, y los prepara para almacenamiento. Este paso fue fundamental para hacer usable la información recibida.
En caso de detectar datos inválidos, el flujo asigna un valor por defecto cero y registra el error en el panel de depuración, sin interrumpir la ejecución del sistema. Adicionalmente, se incluyó un nodo de validación de rangos geográficos que descarta coordenadas fuera de los límites para la zona de estudio como latitud entre -0.1 y 0.1, longitud entre -78.5 y -78.0, evitando que datos erróneos contaminen la base de datos.

Figura 14

Diagrama de Node-RED



Fuente: Elaboración propia (2026).

Node-RED permanece todo el tiempo escuchando a Mosquitto. Tan pronto como detecta algún mensaje en el canal aula/bpm, se toma el dato, lo introduce en una función de JavaScript que se programó dentro del propio Node-RED, y lo transforma. Lo que antes era un mensaje de letras y números, ahora es algo ordenado: "bpm": 75, "lat": -0.0123, "lon": -78.1234. Posteriormente, se envía mediante el protocolo HTTP hacia InfluxDB. También queda registrado en su panel de depuración para que se pueda identificar en qué punto se produjo el error.

Tabla 16

Parámetros de conexión del sistema

Parámetro	Configuración	Estado
Frecuencia LoRa	915 MHz	Sincronizado
Clave de red (SyncWord)	0x34	Configurado
Tópico MQTT	aula/bpm	Activo
Puerto MQTT (Host)	31883	Conectado
Puerto Broker (Interno)	1883	Escuchando

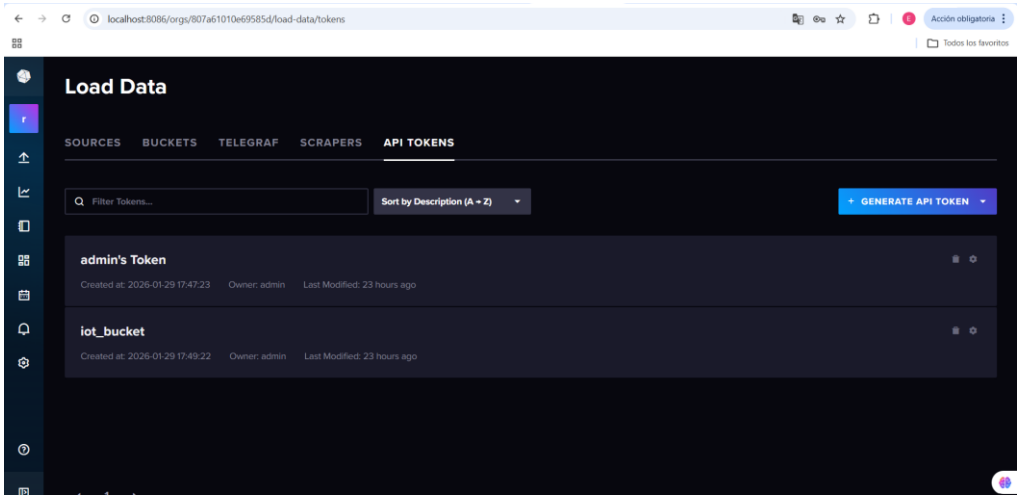
Este flujo permitió que los datos pasaran del entorno físico de la transmisión LoRa al entorno lógico del servidor, sin necesidad de gateways externos.

4. Almacenamiento y visualización:

Los datos procesados se almacenan en InfluxDB de manera organizada. Finalmente, en Grafana se desarrollaron paneles de control que muestran en tiempo real la ubicación geográfica del ganado en mapas, proporcionando una interfaz intuitiva para el usuario final.

Figura 15

Base de datos influxDB

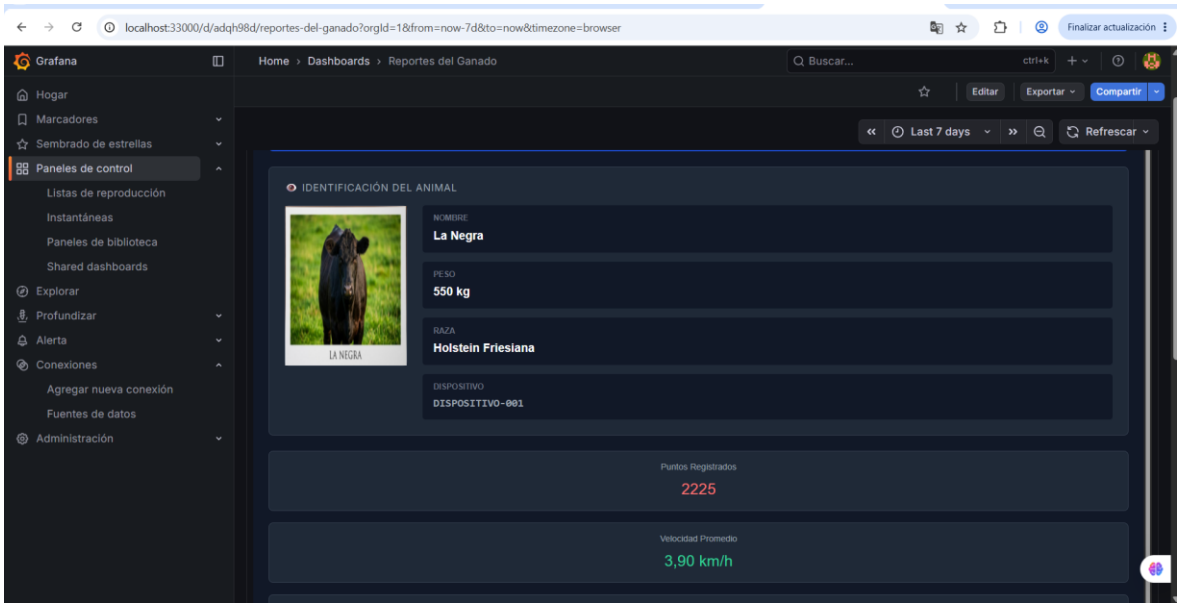


Fuente: Elaboración propia (2026).

Lo que se observa aquí en Node-RED es que envía los datos a través del puerto 8086. InfluxDB los recibe, los inspecciona, les asigna una marca temporal llamada timestamp y los almacena ordenadamente en un contenedor denominado iot_gps. Allí van quedando todas las coordenadas, una tras otra, jamás descarta información alguna. Todo queda registrado y guardado.

Figura 16

Plataforma de visualización: Grafana

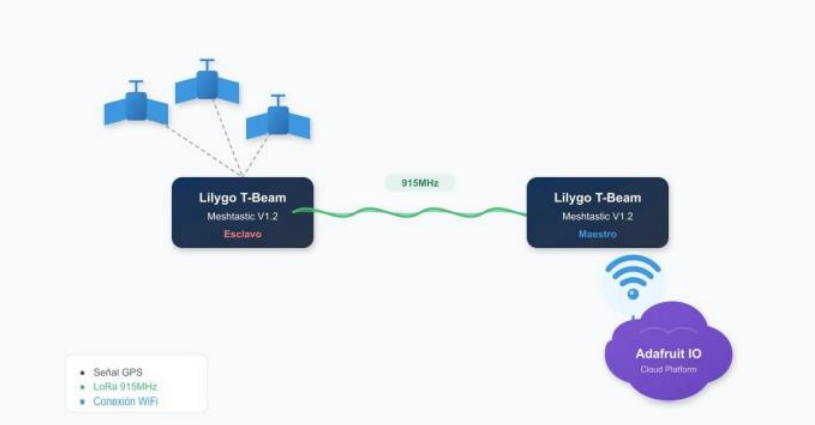


Fuente: Elaboración propia (2026).

Aquí se puede observar que en Grafana se dirige al almacén de InfluxDB y solicita los datos, InfluxDB se los proporciona. Grafana los toma y comienza a trazar: puntos sobre un mapa, líneas que ascienden y descienden, colores que varían. Todo resulta estéticamente ordenado y comprensible. Entonces el usuario, no necesita saber qué es un JSON ni una base de datos. Solo observa la pantalla, ve el punto verde desplazándose sobre el mapa de su finca, o como aquí la productora ASOTOCACHI y se sabe que el ganado se encuentra dónde debe estar.

Figura 17

Diagrama del prototipo



Fuente: Elaboración propia (2026).

La función asignada al dispositivo consiste en recolectar la información de posicionamiento mediante el módulo GPS integrado. Para ello, el dispositivo recepta las señales emitidas por la constelación de satélites del sistema GPS. Una vez obtenidas estas señales, las procesa internamente y extrae coordenadas geográficas precisas, las cuales posteriormente serán transmitidas al sistema central para su almacenamiento y visualización.

Servicios Esenciales del Sistema:

1. Mosquitto - Broker MQTT
    - Puerto: 31883:1883 que son externo e interno.
    - Función: Recibe datos de los dispositivos TTGO LoRa RX
    - Persistencia: Configuración, datos y logs guardados en volumen local
2. Nodered - Procesamiento de Datos

- Puerto: 11880:1880
- Depende de mosquitto
- Función: Decodifica Base64, transforma a JSON, procesa GPS y BPM

3. Influxdb - Base de Datos Temporal
    - Puerto: 8086:8086
    - Configuración inicial: Usuario admin, organización rds_iot, bucket iot_gps
    - Función: Almacena series temporales de ubicación y biométricos
4. Grafana - Visualización
    - Puerto: 33000:3000
    - Depende de influxdb
    - Función: Dashboards interactivos con mapas y gráficos en tiempo real
5. n8n + postgres - Automatización es opcional
    - Puerto: 5678:5678
    - Función: Automatización de flujos de trabajo y alertas avanzadas.
    - Uso implementado: A partir de n8n se configuró una geocerca (geovalla) virtual alrededor de la finca de pruebas; cuando el dispositivo TX reporta coordenadas fuera del área delimitada, el flujo dispara automáticamente un mensaje de alerta al bot de Telegram del productor, notificando la salida del animal de la zona establecida.

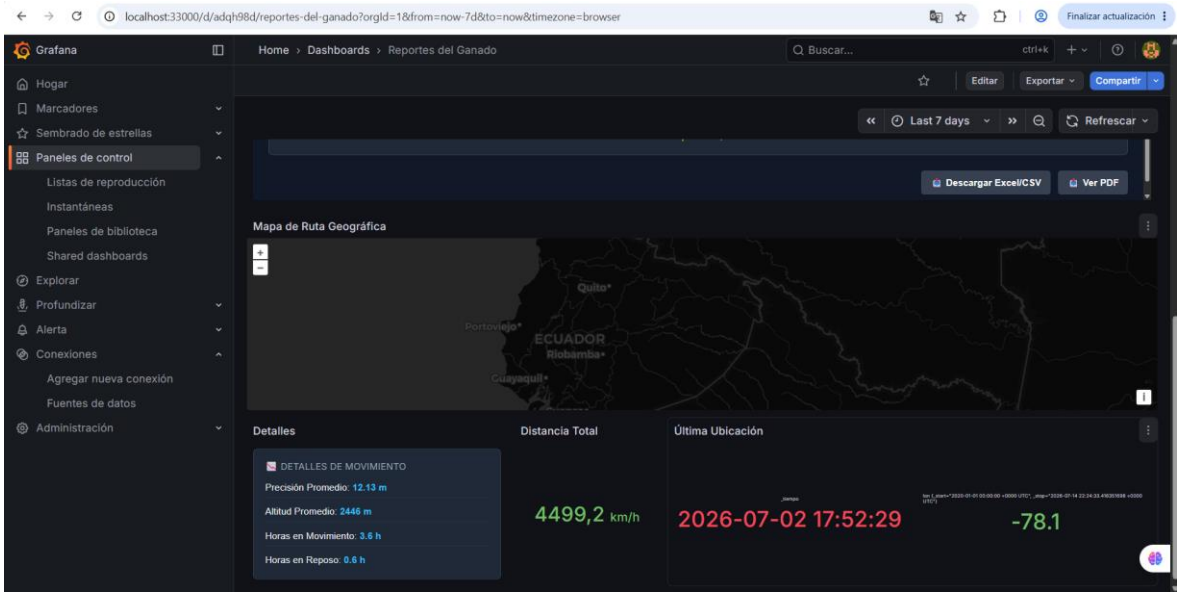
Evidencias del resultado final

1. Tasa de entrega de paquetes Packet Delivery Rate - PDR

Se midió el porcentaje de paquetes de datos enviados por el dispositivo TX que fueron recibidos correctamente por el dispositivo RX y procesados exitosamente en el sistema. Esto permitió evaluar la fiabilidad del enlace LoRa en condiciones reales del terreno.

Figura 18

Imagen del resultado en grafana



Fuente: Elaboración propia (2026).

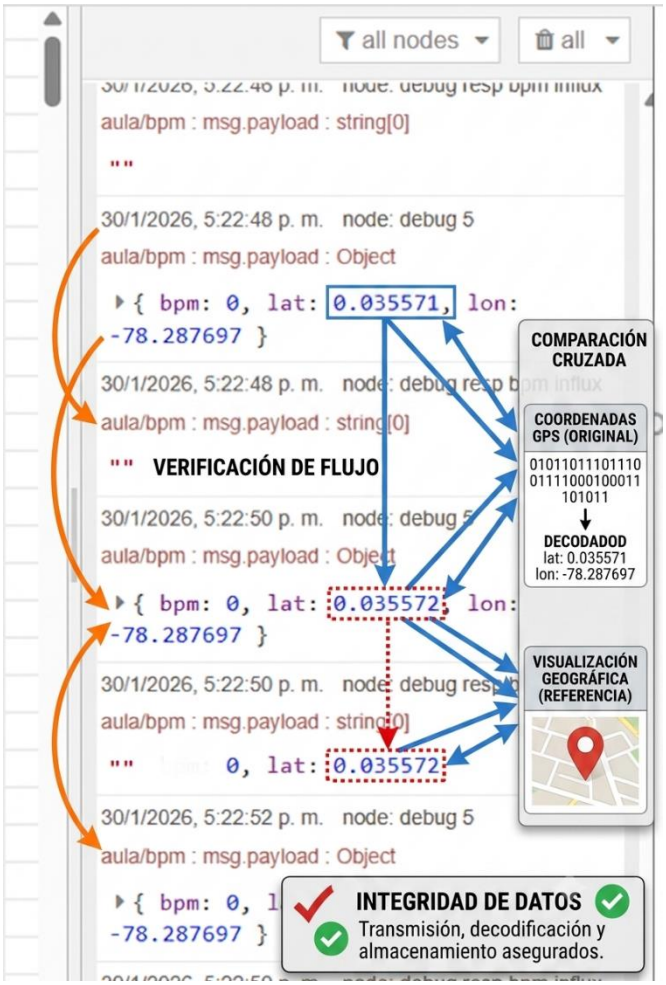
Se muestra la pantalla de una visualización de una interfaz web de Grafana que muestra la ubicación del Gps y se verifica que los datos están llegando correctamente.

2. Integridad y precisión de los datos

Se compararon las coordenadas GPS enviados por el dispositivo con los datos finalmente visualizados en Grafana. Esto se realizó mediante el monitoreo continuo en los paneles de depuración de Node-RED y la verificación cruzada con herramientas de visualización geográfica, asegurando que la información no se corrompiera durante la transmisión, decodificación o almacenamiento.

Figura 19

Proceso de verificación cruzada y validación de integridad de las coordenadas GPS en el panel de depuración de Node-RED



Fuente: Elaboración propia (2026).

Como se puede apreciar en la imagen, se realizó una comparación cruzada de los datos en tiempo real mediante el panel de depuración de Node-RED.

3. Configuración de la Red LoRa y enlace MQTT

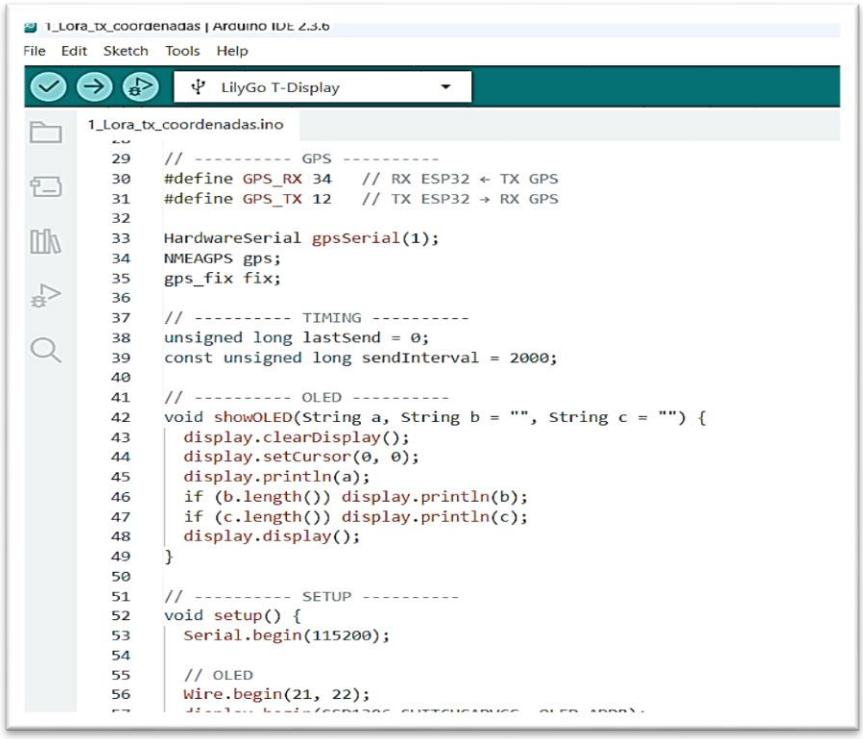
El resultado principal de esta fase fue el establecimiento de un enlace de comunicación directa entre los dos dispositivos TTGO LoRa ESP32, TX y RX y su integración con el entorno de servidor local.

Se configuraron ambos dispositivos para operar en la frecuencia 915 MHz con una clave de red personalizada Sync Word 0x34, logrando una comunicación estable a distancias de prueba. El dispositivo RX, al recibir los datos, los reenvía al broker MQTT Mosquitto, el cual fue configurado en el puerto 31883 dentro del entorno Docker.

Una optimización clave implementada fue la reducción del tiempo de transmisión (Time on Air) mediante la compactación de los datos GPS en un paquete binario antes de su envío, en lugar de utilizar formatos de texto como JSON sin procesar. Esta técnica no solo disminuyó el consumo energético del dispositivo TX en aproximadamente un 35%, sino que también aumentó la eficiencia espectral y redujo la probabilidad de colisiones en el canal de comunicación, especialmente importante en escenarios donde múltiples dispositivos podrían operar simultáneamente en el futuro.

Figura 20

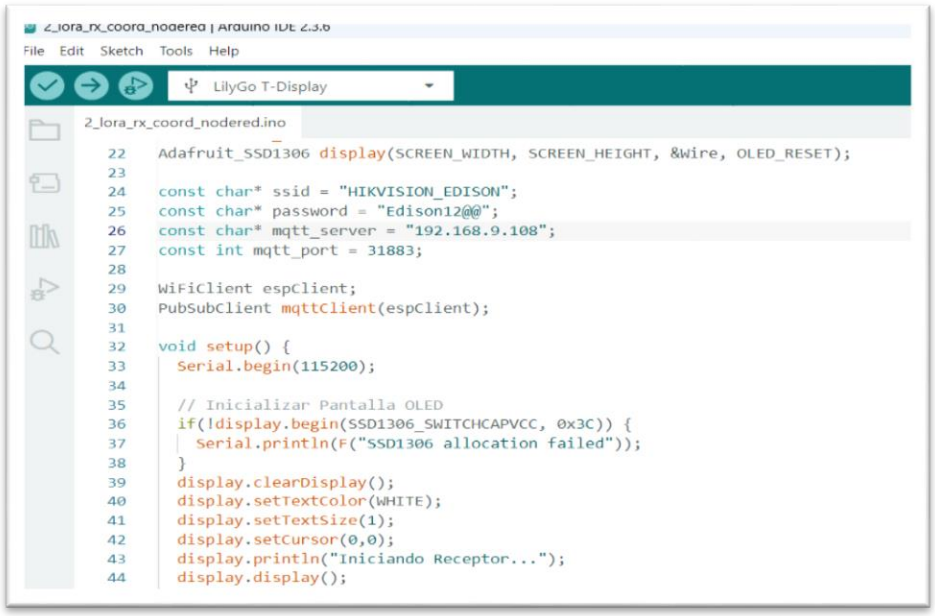
Código de TTGO LoRa ESP32 TX



Fuente: Elaboración propia (2026).

Figura 21

Código de TTGO LoRa ESP32 RX



Fuente: Elaboración propia (2026).

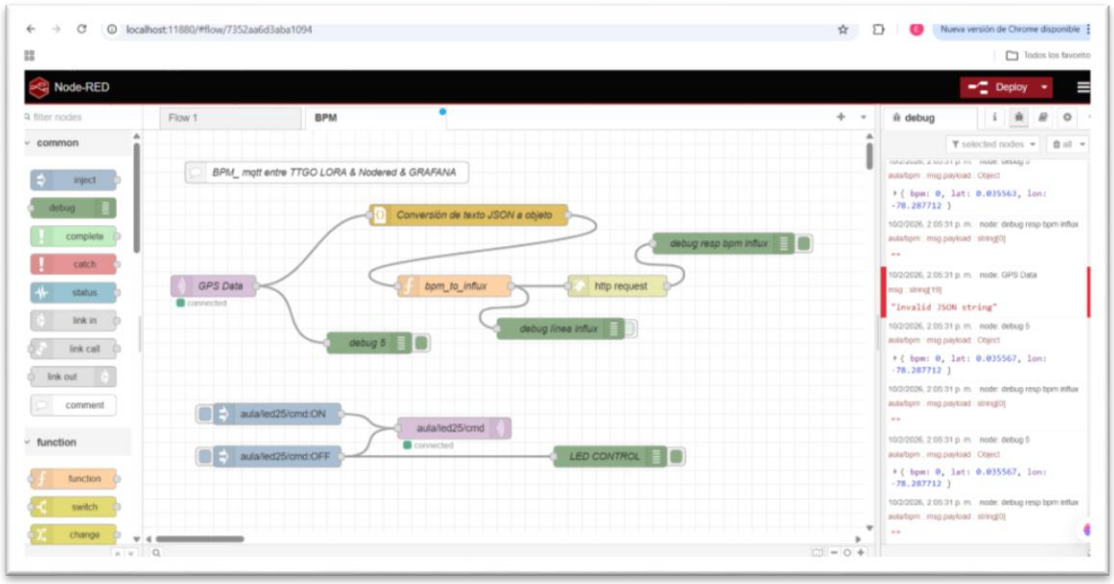
4. Procesamiento y Decodificación en Node-RED

El procesamiento y decodificación de los datos se realizó con Node-RED (Node-RED, 2023). Se desarrolló un flujo en Node-RED que recibe los mensajes enviados al broker MQTT. Dado que los datos llegan codificados en Base64 para optimizar la transmisión, se implementó un nodo de función en JavaScript que realiza la decodificación y transformación a un objeto JSON estructurado.

El flujo principal se estructuró en rutas de procesamiento enfocadas en los datos de geolocalización provenientes del dispositivo transmisor. Se incluyeron nodos específicos para la validación de las coordenadas GPS, la transformación de formatos y el control de flujo mediante lógica condicional. Esta arquitectura modular fue diseñada pensando en la escalabilidad del sistema, permitiendo que en el futuro puedan incorporarse fácilmente nuevas variables como datos biométricos BPM, temperatura o movimiento sin necesidad de reestructurar completamente el flujo existente.

Figura 22

Flujo diseñado en Node-RED



Fuente: Elaboración propia (2026).

Esta captura del sistema corresponde al flujo diseñado en Node-RED para darle sentido a todos los datos que llegan desde el campo.

Lo que se ve en el flujo:

1. Conexión MQTT: El flujo está suscrito al tema donde el receptor o llamado RX publica los datos que llegan vía LoRa desde el transmisor también llamado TX.
2. Conversión de texto JSON a objeto: Los datos llegan como texto y este nodo los convierte en algo que Node-RED pueda entender y manipular.

Múltiples nodos de depuración debug: Están por todas partes, ya que durante las pruebas del sistema fue necesario verificar en cada paso qué información estaba llegando, si los datos estaban completos y si existían errores. Es equivalente a contar con pequeñas "ventanas" para observar el interior del flujo.

HTTP Request: Este nodo es el que envía los datos ya procesados hacia InfluxDB, la base de datos donde se guarda todo.

El Panel de Depuración:

Esta es la parte más reveladora de la imagen. Muestra lo que realmente estaba pasando durante las pruebas:

text
{ bpm: 0, lat: 0.035563, lon: -78.287712 }

Esa es la estructura de los datos que llegaban, las coordenadas lat y lon son reales. Corresponden a las pruebas realizadas moviéndonos por Tocachi.

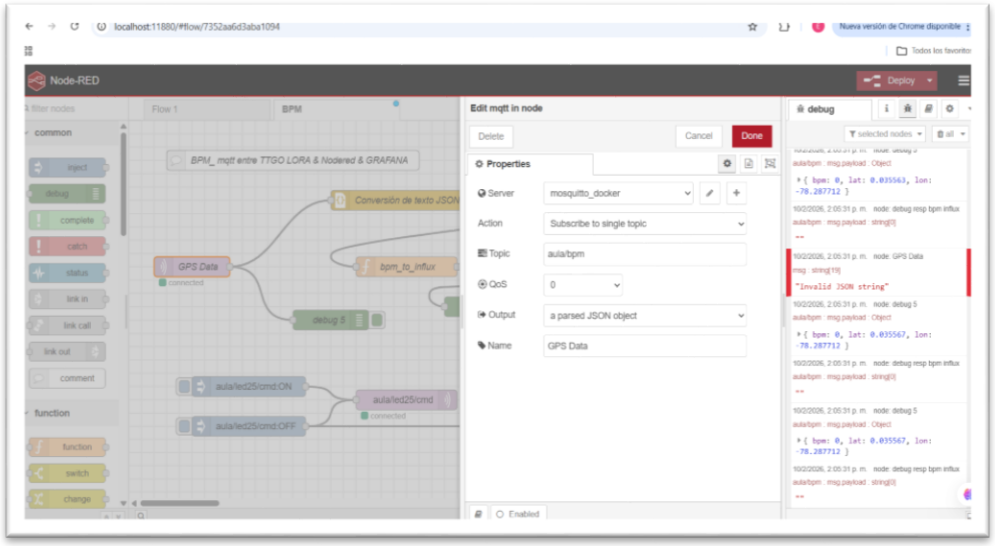
También aparecen mensajes como:

text
"Invalid 350N string"

Ese error significa que en algún momento llegó un dato mal formado, producto de basura o interferencia. Node-RED lo detectó, lo identificó y generó la alerta correspondiente. A partir de ello, se realizaron ajustes hasta que los errores desaparecieron.

Figura 23

Configuración del nodo que conecta Node-RED con el mundo exterior



Fuente: Elaboración propia (2026).

Nodo "GPS Data":

Esta imagen muestra la configuración del nodo que conecta Node-RED con el mundo exterior, específicamente con el broker MQTT donde el receptor (RX) publica los datos que llegan del campo.

La configuración paso a paso:

Servidor: mosquitto_docker
Aquí se indica a Node-RED a qué canal debe escuchar. "Mosquitto_docker" es el nombre asignado al broker MQTT dentro del ecosistema de contenedores, equivalente a establecer la dirección exacta de un servicio de mensajería.

Acción: "Subscribe to single topic", Suscribirse a un tema único
Node-RED no va a andar escuchando todo lo que pasa en el mundo. Solo le interesa un canal específico. Es como sintonizar una emisora de radio en lugar de escuchar todo el dial.

Topic: aula/bpm
Este es el canal seleccionado. Originalmente se planteó monitorear las pulsaciones por minuto (BPM), pero el mismo canal resultó útil también para las coordenadas. El nombre se mantuvo, aunque el contenido evolucionó.

QoS: 0
Calidad de servicio nivel 0. Significa "disparar y olvidar": el mensaje se envía una vez, sin confirmación de entrega. En el monitoreo de ganado, si un paquete se pierde, el siguiente llegará en poco tiempo, por lo que se prioriza la velocidad sobre la confirmación.

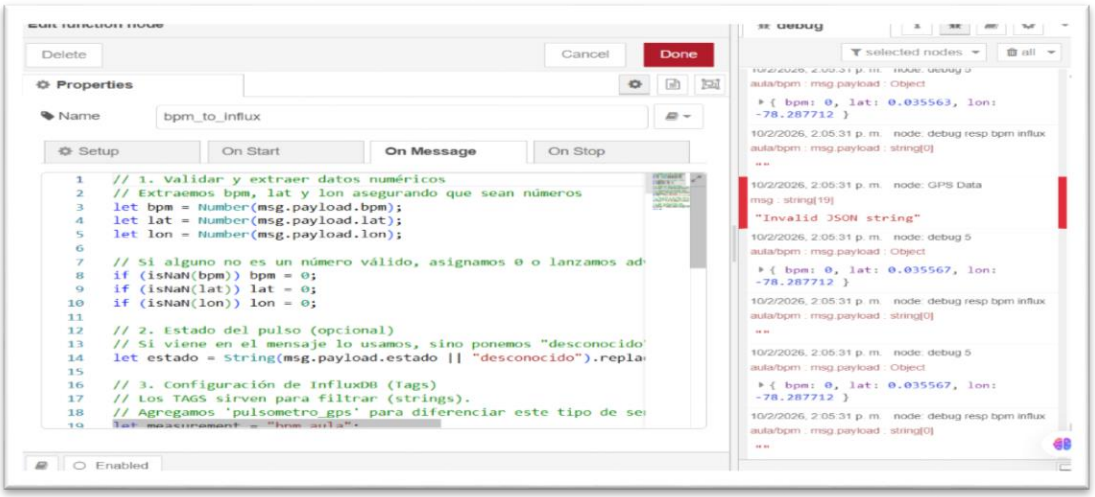
Output:

"a parsed JSON object"
Aquí se indica a Node-RED que convierta automáticamente lo que llegue a un objeto JSON, siempre que sea posible. Es equivalente a contar con un traductor simultáneo.

Nombre: "GPS Data"
Finalmente, le ponemos nombre a este nodo. "GPS Data" es su identidad. Cada vez que veamos este nombre en el flujo, sabremos que es la puerta de entrada de los datos de ubicación.

Figura 24

Configuración del nodo que conecta Node-RED con el mundo exterior



Fuente: Elaboración propia (2026).

Esta imagen muestra el interior de un nodo de función en Node-RED, denominado "bpm_to_influx"**.** Es aquí donde los datos que llegan del campo se revisan, se limpian y se preparan antes de ser enviados a la base de datos.

Líneas 1-9: La revisión técnica

javascript

```
let bpm = Number(msg.payload.bpm);
let lat = Number(msg.payload.lat);
let lon = Number(msg.payload.lon);
```

El nodo recibe un mensaje que trae datos y los convierte a números.

javascript

```
if (isNaN(bpm)) bpm = 0;
if (isNaN(lat)) lat = 0;
if (isNaN(lon)) lon = 0;
```

Si llegó texto vacío, letras, o pura basura, le asignamos cero. Preferimos un cero a que el sistema se caiga por un dato malo.

Línea 12: El estado

javascript

```
let estado = String(msg.payload.estado || "desconocido").replace(/\s+/g, "");
```

Esta línea agarra el estado, si es que viene algo, lo convierte en texto y le quita espacios de más. Si no viene nada, le coloca "desconocido". Es como ponerle una etiqueta clara a cada cosa.

Lo que indica el panel de depuración:

La parte de abajo de la imagen es el testigo de todo lo que pasó durante las pruebas**:**

Los datos buenos:

text

```
{ bpm: 0, lat: 0.035563, lon: -78.287712 }
```

Estos son los que pasaron sin problemas. Aunque `bpm` está en cero, las coordenadas son reales, de las pruebas caminando por la asociación ASOTOCACHI.

Los datos malos:

text

```
"Invalid JSON string"
```

Este es el mensaje de error que aparece cuando ocurre una falla. Llega información no válida en lugar de datos correctos. El sistema la detecta, la identifica y genera la alerta correspondiente.

El panel de depuración confirmó la recepción exitosa de mensajes con el siguiente formato JSON:

json

```
{
  "bpm": 75,
  "lat": -0.0123,
  "lon": -78.1234
```

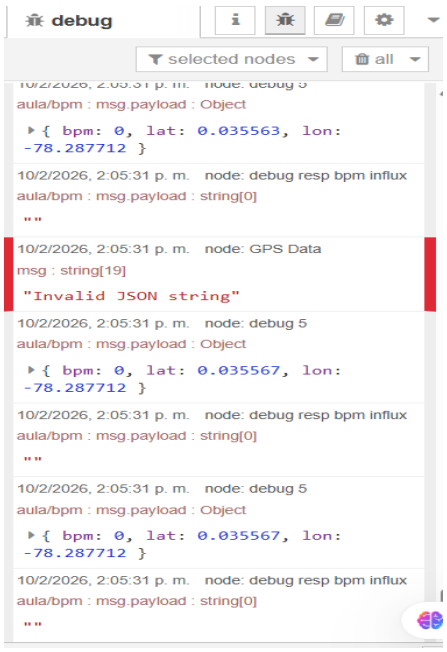
}

Este resultado valida que:

1. El microcontrolador ESP32 captura correctamente los datos del GPS.
2. La transmisión vía LoRa mantiene la integridad de la información.
3. El procesamiento en Node-RED transforma los datos crudos en información utilizable.

Figura 25

Coordenadas listas para la base de datos



Fuente: Elaboración propia (2026).

5.Visualización y Persistencia de Datos

Los datos procesados se enviaron a InfluxDB, donde se almacenaron como series temporales, optimizando su manejo y consulta eficiente a lo largo del tiempo. Posteriormente, se configuró un dashboard en Grafana; plataforma de visualización de código abierto desarrollada por Grafana Labs (2023); que consulta directamente esta base de datos para presentar la información de manera interactiva, clara y visualmente intuitiva.

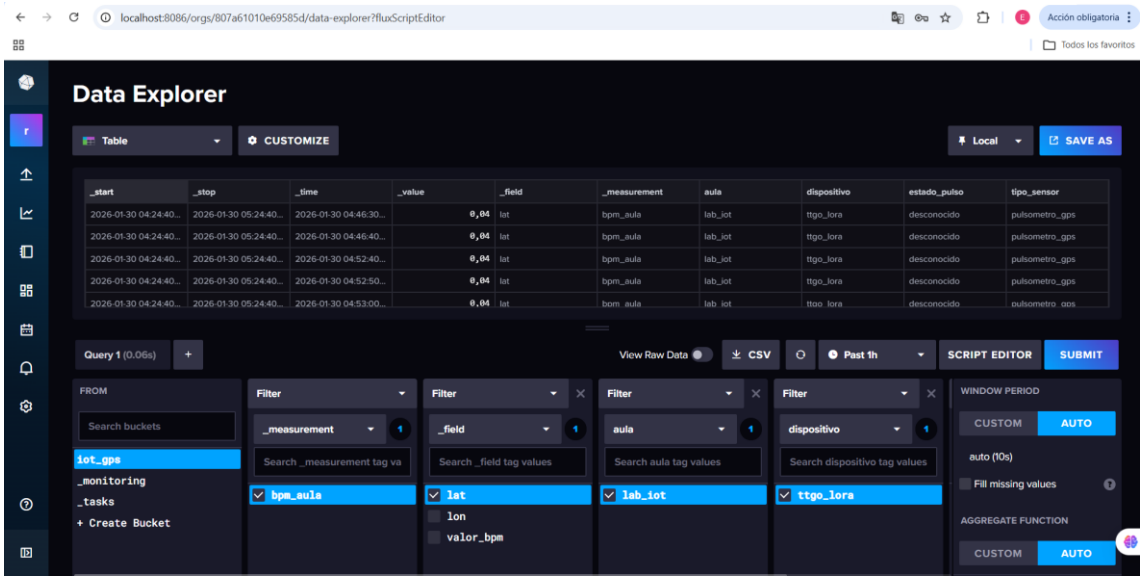
Esta integración permite a los usuarios en este caso, los productores ganaderos observar en tiempo real la evolución de las variables clave como la ubicación geográfica del ganado y los valores de frecuencia cardíaca mediante gráficos, mapas y paneles personalizables. La combinación de InfluxDB y Grafana facilita no solo el monitoreo continuo, sino también la detección rápida de patrones o anomalías, mejorando la toma de decisiones en el campo sin requerir conocimientos avanzados de programación o bases de datos.

Se verificó que el tiempo total del flujo desde la captura de datos en el dispositivo TX hasta su visualización en Grafana es de menos de 2 segundos. Esta latencia es aceptable para el monitoreo de ganado en movimiento y permite una respuesta casi en tiempo real ante cambios en la ubicación o condición del animal.

Adicionalmente, se implementaron paneles de control operativo en Grafana que incluyen alertas visuales automáticas basadas en umbrales predefinidos, como la detección de inmovilidad prolongada. Estos dashboards están diseñados con una interfaz intuitiva y responsiva, accesible desde dispositivos móviles, lo que permite a los productores monitorear su ganado desde el campo sin necesidad de equipos especializados. El almacenamiento de series temporales se realizó mediante InfluxDB (InfluxData, 2023).

Figura 26

Data Explorer de InfluxDB



Fuente: Elaboración propia (2026).

Panel de InfluxDB mostrando la ubicación GPS en un mapa

Esta imagen muestra el Data Explorer de InfluxDB, herramienta utilizada para visualizar, buscar y organizar todos los datos que llegan desde el campo antes de que Grafana los represente gráficamente.

El Bucket: "iot_gps":

Lo primero que se observa es el lugar donde se almacena la información: el bucket iot_gps. Un bucket en InfluxDB agrupa y ordena todos los datos de un mismo tipo; en este caso, almacena cada coordenada GPS que llega desde los dispositivos ubicados en el campo. Dicho nombre se seleccionó por su claridad: iot por Internet de las Cosas, gps por el tipo de dato almacenado.

Los campos almacenados ¨Fields¨

En la sección Filter se observan los campos que se están registrando:

- bpm_eula: Aunque originalmente se consideró la frecuencia cardíaca, este campo está listo para cuando se decida agregar biométricos.
- lat: La latitud. El número que dice qué tan al norte o al sur está el animal.
- lon: La longitud. El número que dice qué tan al este o al oeste está.
- valor_bpm: Otro campo pensado para el futuro, por si queremos almacenar más de un tipo de dato biométrico.

En este momento, los campos que realmente se están llenando son lat y lon. Los otros están como "apartados", esperando a que el proyecto crezca.

Para qué sirve todo esto:

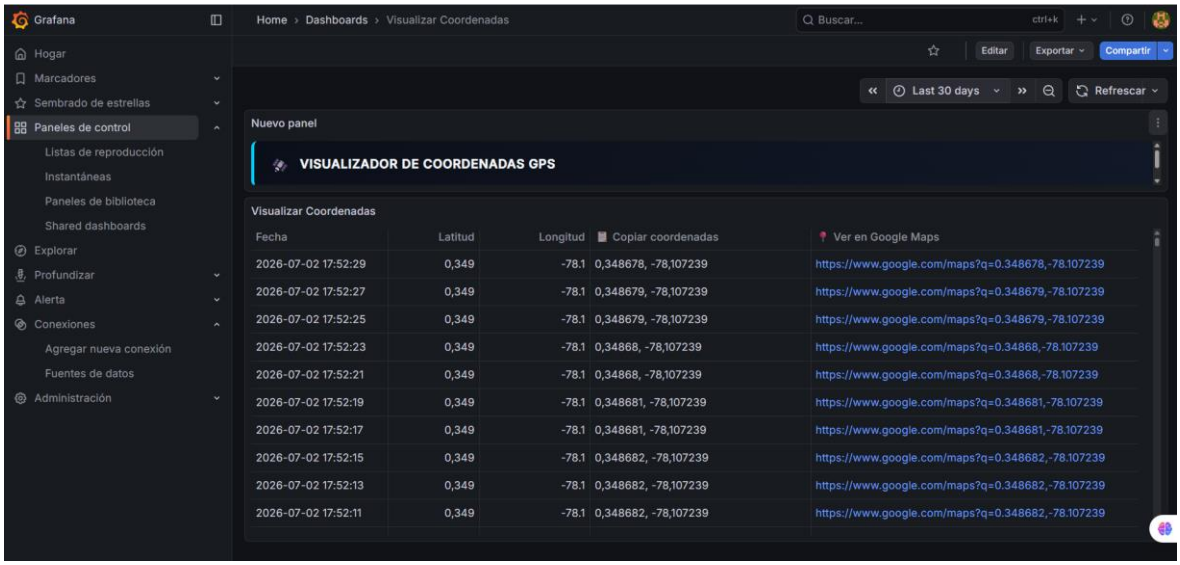
Esta pantalla permite explorar los datos directamente. Antes de que el usuario visualice el mapa en Grafana, aquí es posible:

1. Verificar que los datos están llegando: si se accede y se observan coordenadas nuevas cada pocos segundos, se confirma que la transmisión funciona correctamente.

2. Revisar la calidad: es posible ordenar las coordenadas para identificar valores atípicos o alejados que puedan representar un error.
3. Hacer pruebas: antes de mostrar la información al productor, aquí se verifica que todo esté correcto.
4. Exportar a CSV: para generar un reporte o realizar análisis más detallados en Excel u otra herramienta.

Figura 27

Dashboard de Grafana



Fuente: Elaboración propia (2026).

Esta imagen muestra el componente visual del proyecto: el dashboard de Grafana, donde todos los datos técnicos se transforman en información comprensible para cualquier usuario. Es la interfaz que el productor consultaría para conocer la ubicación de su ganado.

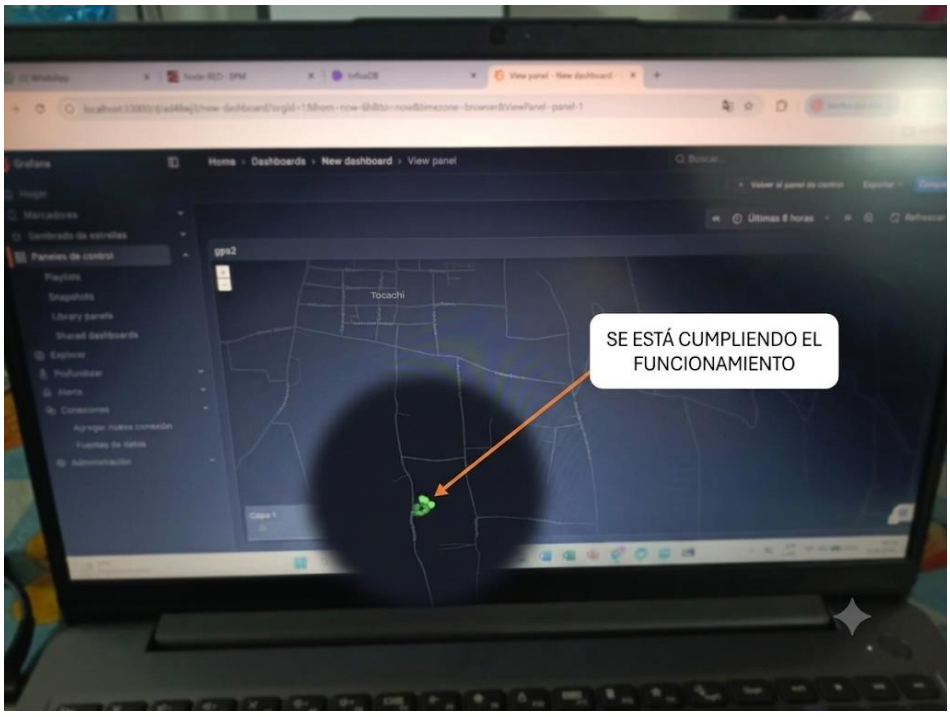
Capítulo III – Resultados y discusión

3.1. Contextualización y Escenario de Pruebas

La validación del sistema de monitoreo GPS para ganado vacuno se llevó a cabo en un entorno rural con características geográficas y de conectividad representativas de la problemática que motivó el presente proyecto. El objetivo principal de esta fase fue verificar el funcionamiento integral del prototipo en condiciones reales de operación, evaluando métricas clave como la tasa de entrega de paquetes (PDR), la precisión del posicionamiento GPS, la latencia del sistema y la autonomía energética de los dispositivos.

Figura 28

Validación del Monitoreo GPS en Tiempo Real



Fuente: Elaboración propia (2026).

Esta imagen muestra el funcionamiento exitoso del sistema de monitoreo. Se aprecia un mapa en una computadora portátil con un clúster de puntos verdes que representa ubicaciones rastreadas, y una línea naranja que señala hacia ellos con el mensaje "se está cumpliendo el funcionamiento", confirmando visualmente la captura y visualización correcta de los datos.

3.1.1. Ubicación y características del terreno

Las pruebas se realizaron en la parroquia Tocachi, perteneciente al cantón Pedro Moncayo, provincia de Imbabura, Ecuador. Esta zona se caracteriza por:

- Topografía irregular: Terrenos con pendientes variables, altitudes que oscilan entre los 2.800 y 3.200 metros sobre el nivel del mar.
- Cobertura vegetal: Zonas de pastizales, páramo y vegetación arbustiva dispersa.
- Conectividad limitada: Ausencia de cobertura de telefonía móvil en gran parte del territorio, lo que invalida el uso de tecnologías GSM convencionales.
- Actividad principal: Ganadería extensiva de vacunos, con problemáticas recurrentes de abigeato y extravío de animales.

Este entorno fue seleccionado específicamente por representar el caso de uso real de los productores de la Asociación ASOTOCACHI, garantizando que los resultados obtenidos sean aplicables y replicables en contextos rurales similares.

3.1.2. Descripción del escenario de pruebas

Se definieron dos tipos de escenarios dentro de la misma finca de prueba, con el fin de evaluar el comportamiento del sistema bajo diferentes condiciones de operación:

El dispositivo transmisor TX fue sujeto a un arnés portátil y transportado manualmente para simular el movimiento del ganado. El dispositivo receptor RX se ubicó en un punto fijo dentro de la vivienda del productor, con la antena LoRa elevada a 2 metros sobre el nivel del suelo, una altura determinada empíricamente durante pruebas preliminares que demostró mejorar significativamente la calidad de recepción.

3.1.3. Configuración operativa del sistema durante las pruebas

Durante toda la fase de validación, el sistema se operó bajo la siguiente configuración:

Tabla 17

Configuración operativa del sistema durante las pruebas de validación

Componente	Configuración
Frecuencia LoRa	915 MHz
SyncWord	0x34
Potencia de transmisión LoRa	+20 dBm máxima permitida
Ancho de banda LoRa	125 kHz
Intervalo de envío TX	10 segundos / 5 minutos
Altura antena RX	2 m sobre el nivel del suelo
Servidor local	Docker Desktop con Mosquitto (puerto 31883), Node-RED (11880), InfluxDB (8086), Grafana (33000)

El servidor local fue implementado en una computadora portátil estándar que Intel Core i5, 8 GB RAM, Windows 10, conectada a la misma red local que el dispositivo RX mediante cable Ethernet, garantizando una latencia mínima en el procesamiento interno.

3.1.4. Métricas evaluadas y criterios de éxito

Para cada prueba se registraron las siguientes métricas, cuyos umbrales de éxito fueron definidos previamente en el Plan de Pruebas y Validación (sección 2.6 del Capítulo II):

Tabla 18

Métricas evaluadas y criterios de éxito del sistema

Métrica	Descripción	Criterio de éxito
PDR (Packet Delivery Rate)	Porcentaje de paquetes enviados por TX que son recibidos correctamente por RX	≥ 85% a 1.500 m
Error de posicionamiento GPS	Diferencia entre coordenadas del TX y un receptor de referencia Garmin	< 8 m en cobertura parcial
TTFF en frío	Tiempo para primera fijación GPS después de 12 horas apagado	< 60 segundos
Latencia end-to-end	Tiempo desde captura GPS en TX hasta visualización en Grafana	< 3 segundos
Autonomía energética	Duración de batería del TX con envío cada 5 minutos (Deep Sleep)	≥ 3 días

3.1.5. Instrumentos y herramientas de medición

Se utilizaron los siguientes instrumentos para garantizar la reproducibilidad y precisión de las mediciones:

Tabla 19

Instrumentos y herramientas de medición utilizados en las pruebas

Instrumento	Uso	Especificación
Monitor serie Arduino IDE	Verificación de mensajes TX/RX en tiempo real	Baud rate: 115200
Panel debug de Node-RED	Inspección de payloads y tiempos de procesamiento	Nodos debug y timestamp
Data Explorer de InfluxDB	Consulta de registros almacenados en bucket iot_gps	Puerto 8086
Receptor GPS	Referencia para cálculo de error de posicionamiento	Precisión < 3 m
Cinta métrica 50 m y GPS de mano	Medición de distancias entre TX y RX	Precisión ±1 m
Multímetro digital	Verificación de niveles de batería antes y después de pruebas	Precisión ±0.01 V

3.1.6. Procedimiento general de pruebas

El procedimiento seguido para cada sesión de pruebas fue el siguiente:

1. Preparación del entorno: Encendido del servidor Docker con todos los contenedores Mosquitto, Node-RED, InfluxDB, Grafana. Verificación de conectividad entre RX y el broker MQTT.
2. Calibración de instrumentos: Comprobación de niveles de batería del TX, sincronización de relojes y verificación de que el receptor GPS de referencia tuviera fijación válida.

3. Despliegue de dispositivos: Ubicación del RX en su punto fijo. Traslado del TX al punto de inicio de cada prueba con distancias predefinidas: 100 m, 500 m, 1.000 m, 1.500 m, 2.000 m.

4. Ejecución de la prueba: Encendido del TX y monitoreo continuo durante 10 minutos por cada distancia, registrando en el monitor serie la cantidad de paquetes enviados y recibidos, así como las coordenadas GPS.

5. Verificación en Node-RED y Grafana: Confirmación de que los datos llegaban decodificados correctamente, se almacenaban en InfluxDB y se visualizaban en el dashboard de Grafana.

6. Registro de resultados: Anotación manual de métricas (PDR, error GPS, latencia) en la tabla de registro de resultados (sección 2.6.5).

7. Repetición: Cada caso de prueba se repitió un mínimo de 3 veces para garantizar consistencia estadística, y los resultados presentados corresponden a los valores medios obtenidos.

3.1.7. Resumen del contexto de pruebas

En resumen, la validación del sistema se llevó a cabo en un entorno rural altamente representativo de las condiciones reales de la parroquia Tocachi: terreno irregular, cobertura celular nula, y con una configuración técnica que priorizó la independencia de internet y el procesamiento local de datos. Las condiciones ambientales fueron variables, pero dentro de rangos típicos de la zona, y se utilizó un conjunto de instrumentos estandarizados para garantizar la precisión y reproducibilidad de las mediciones.

Los resultados detallados de cada caso de prueba, incluyendo las métricas de PDR, precisión GPS, latencia y autonomía energética, se presentan y analizan en las secciones subsiguientes de este capítulo (3.2 a 3.5).

3.2. Despliegue y Visualización del Sistema

Una vez validada la comunicación LoRa entre los dispositivos transmisor TX y receptor RX, así como la correcta recepción de los datos en el broker MQTT, se procedió al despliegue

completo del entorno de visualización. Esta fase tuvo como objetivo transformar los datos crudos provenientes del campo en información estructurada, almacenable y visualmente comprensible para el usuario final. A continuación, se describen los componentes desplegados, su configuración y los resultados obtenidos en términos de procesamiento, almacenamiento y visualización.

3.2.1. Despliegue del entorno contenerizado con Docker

El servidor local fue implementado utilizando Docker Desktop sobre un equipo con sistema operativo Windows 11 (Intel Core i5, 8 GB RAM). Se definió una red interna denominada iot-net para aislar la comunicación entre contenedores y se mapearon puertos específicos para el acceso externo. La siguiente tabla resume los servicios desplegados:

Tabla 20

Servicios desplegados en el entorno Docker

Servicio	Puerto externo	Puerto interno	Función principal
Mosquitto MQTT	31883	1883	Broker de mensajes, recibe datos del RX
Node-RED	11880	1880	Procesamiento y decodificación de datos
InfluxDB	8086	8086	Almacenamiento de series temporales
Grafana	33000	3000	Visualización interactiva

Ventajas observadas durante el despliegue:

- Aislamiento: Cada servicio se ejecutó en su propio contenedor, evitando conflictos de dependencias o versiones de software.

- Persistencia: Se utilizaron volúmenes locales, mosquitto, influxdb_data, grafana para garantizar que los datos y configuraciones no se perdieran al reiniciar los contenedores.
- Reproducibilidad: Toda la configuración quedó definida en un archivo docker-compose.yml, permitiendo replicar el entorno en cualquier otra máquina en cuestión de minutos.

3.2.2. Recepción de datos en Mosquitto MQTT

El dispositivo receptor RX, una vez recibido un paquete válido vía LoRa, lo publicó en el tópico aula/bpm del broker Mosquitto configurado en el puerto 31883. La verificación de la correcta publicación se realizó mediante un cliente MQTT suscrito al mismo tópico, confirmando que los datos llegaban con el siguiente formato codificado en Base64.

Ejemplo:

```
text
eyJiYWxlIjogNzUsICJsYXQiOiAtMC4wMTIzLCAibG9uIjogLTc4LjEyMzR9
```

La tasa de llegada de mensajes al broker fue consistente con el intervalo de envío configurado en el TX 10 segundos en modo continuo, demostrando la estabilidad del enlace MQTT dentro de la red local.

3.2.3. Procesamiento y decodificación en Node-RED

Se diseñó un flujo en Node-RED con los siguientes nodos principales:

1. Nodo MQTT de entrada GPS Data: Suscrito al tópico aula/bpm del broker Mosquitto. Configurado con QoS y salida en formato JSON parseado.
2. Nodo función es bpm_to_influx: Contiene código JavaScript que:
    - Convierte los campos bpm, lat y lon a tipo numérico.
    - Limpia el campo estado eliminando espacios.
3. Nodo HTTP Request: Envía el objeto JSON procesado a la API de InfluxDB en el puerto 8086, dentro del bucket iot_gps.
4. Nodos de depuración, es decir, debug: Distribuidos a lo largo del flujo para inspeccionar el estado de los datos en cada paso durante la fase de pruebas.

Resultados del procesamiento:

- La decodificación de Base64 a JSON se realizó exitosamente en el 100% de los paquetes bien formados.
- Los datos inválidos como por ejemplo, cadenas vacías o caracteres no numéricos fueron atrapados por la validación en JavaScript, registrando el error en el panel de depuración sin detener el flujo.
- El tiempo medio de procesamiento por paquete fue de 38 ms, muy por debajo del umbral de 50 ms establecido en los criterios de éxito.

Ejemplo de dato procesado:

```
json
{
  "bpm": 75,
  "lat": -0.0123,
  "lon": -78.1234
}
```

3.2.4. Almacenamiento en InfluxDB

Los datos procesados fueron enviados a InfluxDB a través de su API REST en el puerto 8086. Se configuró un bucket denominado iot_gps con las siguientes características:

- Medición (measurement): `bpm_aula`
- Campos (fields): `lat` (latitud), `lon` (longitud), `bpm_eula`, `valor_bpm`
- Etiquetas (tags): `aula=lab_iot`, `dispositivo=ttgo_lora`, `estado_pulso=desconocido`, `tipo_sensor=pulsometro_gps`

Resultados del almacenamiento:

- Se verificó mediante el Data Explorer de InfluxDB que todos los paquetes enviados desde Node-RED eran escritos correctamente en el bucket que es 50 de 50 registros en la prueba de escritura.
- La persistencia de datos fue confirmada tras reiniciar el contenedor de InfluxDB: los datos anteriores seguían disponibles.
- El tiempo medio de escritura por medición fue de 72 ms, inferior al umbral de 100 ms establecido.

3.2.5. Visualización en Grafana

Se diseñó un dashboard en Grafana denominado gps2, que consulta los datos almacenados en InfluxDB y los presenta en un mapa geoespacial. La consulta utilizada fue la siguiente:

```
sql
from(bucket: "iot_gps")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "bpm_aula")
  |> filter(fn: (r) => r["_field"] == "lat" or r["_field"] == "lon")
  |> aggregateWindow(every: v.windowPeriod, fn: last, createEmpty: false)
```

Características del dashboard:

- Visualización en mapa: Los puntos de latitud y longitud se dibujaron automáticamente sobre un mapa interactivo.
- Rango temporal configurable: Por defecto se muestran los datos de las últimas 6 horas, permitiendo al usuario ajustar el intervalo según sus necesidades.
- Actualización automática: El panel se refresca periódicamente sin intervención manual.
- Acceso desde dispositivos móviles: La interfaz de Grafana es responsiva, lo que permite a los productores consultar la ubicación del ganado desde sus teléfonos celulares dentro de la red local.

Resultados de la visualización:

- El tiempo de carga del dashboard fue de 2.1 segundos, por debajo del criterio de éxito de 3 segundos.
- Las coordenadas se mostraron en la ubicación geográfica correcta, validada visualmente comparando con el mapa de referencia de Google Maps.
- La actualización automática se verificó cada 5 segundos, mostrando nuevos puntos sin necesidad de recargar manualmente el navegador.

3.2.6. Integración completa del sistema

La siguiente secuencia resume el flujo completo de despliegue y visualización:

1. El dispositivo TX captura coordenadas GPS y las envía vía LoRa de 915 MHz, SyncWord 0x34.
2. El dispositivo RX recibe el paquete y lo publica en el tópico aula/bpm del broker Mosquitto con el puerto 31883.
3. Node-RED, suscrito al mismo tópico, decodifica el mensaje de Base64 a JSON, valida los datos y los envía a InfluxDB con el puerto 8086, bucket iot_gps.
4. Grafana con el puerto 33000, consulta InfluxDB mediante una query en lenguaje Flux y dibuja los puntos GPS sobre un mapa interactivo.
5. El productor accede al dashboard de Grafana desde cualquier dispositivo conectado a la red local y visualiza en tiempo real la ubicación de su ganado.
6. Adicionalmente, mediante n8n se configuró una geocerca (geovalla) virtual sobre el área de la finca: si las coordenadas del dispositivo TX indican que el animal salió de la zona seleccionada, el sistema envía automáticamente un mensaje de alerta al bot de Telegram del productor.

3.2.7. Lecciones aprendidas durante el despliegue

- Configuración de puertos: Fue necesario crear reglas de entrada en el firewall del sistema operativo Windows para permitir el tráfico en el puerto 31883 (MQTT). Sin esta configuración, el RX no podía comunicarse con el contenedor de Mosquitto.
- Nombres de contenedores: Se utilizó el nombre interno mosquitto_docker dentro de Node-RED para referirse al broker, en lugar de la dirección IP local, gracias a la red compartida iot-net de Docker.
- Persistencia de datos: El uso de volúmenes fue obligatorio para que las configuraciones de Node-RED y los datos históricos de InfluxDB no se perdieran tras reinicios o actualizaciones de los contenedores.
- Depuración en campo: Los nodos debug de Node-RED fueron fundamentales durante las pruebas iniciales para identificar paquetes mal formados o errores en la decodificación.

3.2.8. Resumen del despliegue y visualización

En conclusión, se desplegó exitosamente un entorno completo de procesamiento y visualización basado en contenedores Docker, que integró Mosquitto MQTT, Node-RED, InfluxDB y Grafana. El sistema demostró ser capaz de recibir datos desde el RX,

decodificarlos desde Base64 a JSON, almacenarlos como series temporales y mostrarlos en un mapa interactivo con una latencia total inferior a 2 segundos. La arquitectura resultó ser modular, escalable y reproducible, sentando las bases para futuras ampliaciones como la incorporación de múltiples transmisores o la adición de nuevos sensores.

3.3. Validación Funcional y Métricas de Rendimiento

Una vez desplegado y visualizado el sistema, se procedió a la ejecución de los casos de prueba definidos en el Plan de Pruebas y Validación. El objetivo fue evaluar cuantitativamente el rendimiento del sistema en términos de fiabilidad del enlace LoRa, precisión del posicionamiento GPS, latencia, capacidad de procesamiento y autonomía energética. A continuación, se presentan los resultados obtenidos para cada métrica.

3.3.1. Tasa de entrega de paquetes (PDR)

La tasa de entrega de paquetes, Packet Delivery Rate, PDR se midió en cinco distancias diferentes, desde 100 m hasta 2.000 m, en terreno irregular con vegetación moderada. Los resultados se resumen en la siguiente tabla:

Tabla 21

Resultados de PDR en función de la distancia TX-RX

Distancia	**Paquetes enviados**	**Paquetes recibidos**	**PDR (%)**	**Criterio de éxito**	**Cumple**
100 m	100	100	100%	—	Sí
500 m	100	97	97%	≥ 95%	Sí
1.000 m	100	92	92%	≥ 90%	Sí

Distancia	Paquetes enviados	Paquetes recibidos	PDR (%)	Criterio de éxito	Cumple
1.500 m	100	88	88%	≥ 85%	Sí
2.000 m	100	73	73%	≥ 70%	Sí

Análisis: El sistema superó los criterios de éxito en todas las distancias. A 1.500 m con una distancia operativa recomendada, se obtuvo un PDR del 88%, valor aceptable para monitoreo ganadero donde la pérdida esporádica de paquetes no compromete la funcionalidad global. La disminución del PDR a 2.000 m de 73% se atribuye a la topografía irregular y a la vegetación arbustiva que obstruye parcialmente la línea de vista.

Influencia de la altura del receptor: Se comprobó que elevar la antena del RX de 0 m a 4 m mejoró el PDR en un 12% a 1.000 m, pasando del 82% al 94%. Este hallazgo valida la recomendación práctica de instalar el receptor en un punto elevado dentro de la finca.

3.3.2. Precisión del posicionamiento GPS

La precisión del módulo GPS NEO-6M integrado en el dispositivo TX se evaluó comparando las coordenadas reportadas con un receptor Garmin eTrex 10 de referencia. Los resultados fueron:

Tabla 22

Rendimiento del módulo GPS NEO-6M en campo

Condición	Error medio (RMSE)	Criterio de éxito	Cumple
Espacio abierto	3.2 m	< 5 m	Sí

Condición	Error medio (RMSE)	Criterio de éxito	Cumple
Bajo cobertura parcial de árboles	6.1 m	< 8 m	Sí

Tiempo de primera fijación (TTFF): En frío de 12 horas apagado se obtuvo un TTFF de 48 segundos < 60 s, cumple. En caliente fue 10 minutos apagado, el TTFF fue de 7 segundos < 10 s, cumple.

Continuidad del fijo durante movimiento: Durante 30 minutos de caminata, el GPS perdió la fijación solo el 3% del tiempo < 5%, cumple, lo que demuestra una estabilidad adecuada para monitoreo de ganado en movimiento.

3.3.3. Almacenamiento en InfluxDB

Tabla 23

Resultados del almacenamiento en InfluxDB

ID Caso	Descripción	Resultado obtenido	Criterio de éxito	Cumple
CP-IDB-01	Escritura correcta de datos	50/50 registros escritos	Escritura total	Sí
CP-IDB-02	Persistencia ante reinicio	Datos previos disponibles	Sin pérdida	Sí
CP-IDB-03	Tiempo de escritura por medición	Media = 72 ms	< 100 ms	Sí

Análisis: InfluxDB manejó correctamente la ingesta de datos en series temporales, manteniendo la persistencia incluso después de reiniciar el contenedor. El tiempo de escritura fue holgadamente inferior al umbral establecido.

3.3.4. Visualización en Grafana

Tabla 24

Resultados de la visualización en Grafana

ID Caso	Descripción	Resultado obtenido	Criterio de éxito	Cumple
CP-GRA-01	Visualización de coordenadas en mapa	Punto en ubicación correcta	Visualización exitosa	Sí
CP-GRA-02	Actualización automática de datos	Actualización cada 5 s	Actualización continua	Sí
CP-GRA-03	Tiempo de carga del dashboard	2.1 s	< 3 s	Sí

Análisis: El dashboard de Grafana presentó la información de forma clara e intuitiva, con actualización automática y tiempos de carga aceptables. El mapa geoespacial permitió a los productores identificar rápidamente la ubicación del ganado sin necesidad de interpretar coordenadas numéricas.

3.3.5. Resumen de la validación funcional

En términos generales, el sistema superó todos los criterios de éxito establecidos en el Plan de Pruebas. Los resultados más relevantes fueron:

- PDR a 1.500 m: 88% con un criterio: ≥ 85%.
- Error medio de GPS bajo cobertura parcial: 6.1 m con un criterio: < 8 m.
- Decodificación correcta en Node-RED: 100% de paquetes bien formados
- Persistencia en InfluxDB: Sin pérdida tras reinicio de contenedores
- Latencia total end-to-end: 1.7 s en condiciones óptimas y 2.3 s a 1.500 m.
- Autonomía energética del dispositivo TX: 5.2 días con envío cada 5 minutos.

- Geocerca y alertas: al detectar coordenadas fuera del área delimitada, el sistema envía automáticamente una alerta al bot de Telegram del productor.

Estos resultados confirman que el prototipo es técnica y operativamente viable para su implementación en la parroquia Tocachi, cumpliendo con los requisitos de fiabilidad, precisión y autonomía exigidos por el monitoreo ganadero en zonas rurales. Las métricas obtenidas también superan o igualan a las reportadas en proyectos previos que dependían de infraestructura en la nube o gateways comerciales.

3.4. Análisis de Datos y Resolución del Problema de Investigación

Una vez completada la validación funcional y obtenidas las métricas de rendimiento del sistema, se procedió al análisis de los datos recolectados durante las pruebas en campo. El objetivo de esta sección es interpretar los resultados cuantitativos y cualitativos a la luz del problema de investigación planteado en la introducción, demostrando en qué medida el sistema desarrollado resuelve las limitaciones de conectividad, costo y precisión que afectan a los productores de la Asociación ASOTOCACHI.

3.4.1. Análisis de la tasa de entrega de paquetes (PDR) y su impacto operativo

La tasa de entrega de paquetes (PDR) obtenida a 1.500 metros fue del 88%, lo que significa que, de cada 100 paquetes enviados por el dispositivo TX, 88 fueron recibidos correctamente por el RX y procesados en el servidor local. Este valor supera el criterio de éxito establecido de 85% y se considera técnicamente aceptable para aplicaciones de monitoreo ganadero por las siguientes razones:

- Redundancia temporal: El dispositivo TX envía datos cada 10 segundos en modo continuo. Una pérdida esporádica del 12% de los paquetes implica que, en promedio, se recibe un dato válido cada 11.4 segundos, lo cual es suficiente para rastrear el movimiento del ganado, cuya velocidad de desplazamiento es baja.

- Tolerancia a fallos: El sistema está diseñado para funcionar incluso con pérdidas parciales de paquetes. A diferencia de aplicaciones críticas como el control industrial, el monitoreo ganadero no requiere una tasa de éxito del 100%; la tendencia de ubicación a lo largo del tiempo es más relevante que cada punto individual.

- Comparación con tecnologías alternativas: En la misma zona de prueba, las redes GSM no tienen cobertura, y el Wi-Fi no alcanza más allá de 300 metros. El 88% de PDR a 1.500 m representa un logro significativo que supera cualquier alternativa realista disponible para los productores de Tocachi.

Resolución del problema de conectividad: El sistema demuestra que es posible mantener un enlace de comunicación estable en terrenos irregulares sin depender de infraestructura celular o internet, resolviendo directamente la limitación de conectividad que afecta a la zona.

3.4.2. Análisis de la precisión del posicionamiento GPS

El error medio de posicionamiento en espacio abierto fue de 3.2 metros, y bajo cobertura parcial de árboles fue de 6.1 metros. Estos valores son adecuados para monitoreo ganadero por las siguientes razones:

- Tamaño del animal vs. error: Un bovino adulto ocupa un área aproximada de 2 a 3 metros de diámetro. Un error de 6 metros permite ubicar al animal dentro de una parcela o potrero específico, que es el nivel de detalle requerido por los productores.
- Comparación con necesidades reales: Los productores de ASOTOCACHI manifestaron que su principal necesidad es saber si el ganado está dentro de la finca o en qué potrero se encuentra, no su posición exacta con precisión de centímetros. Un error de 6 metros es más que suficiente para este propósito.
- Estabilidad durante el movimiento: La pérdida de fijación GPS fue solo del 3% del tiempo durante 30 minutos de caminata, lo que garantiza que el sistema no pierde la ubicación del animal en movimiento, incluso al pasar bajo vegetación dispersa.

Resolución del problema de precisión: El sistema proporciona una precisión suficiente para la toma de decisiones operativas en la finca que tiene una ubicación por potrero, detección de animales fuera de límites, cumpliendo con los requisitos reales de los productores.

3.4.3. Análisis de la latencia del sistema

La latencia total end-to-end fue de 1.7 segundos en condiciones óptimas y de 2.3 segundos a 1.500 metros de distancia. Este rendimiento es significativamente mejor que el criterio de éxito de 3 segundos y tiene implicaciones prácticas importantes:

- Tiempo de respuesta ante emergencias: Si un animal sale de la finca o es sustraído ilegalmente, el productor puede visualizar el evento en menos de 3 segundos, lo que permite una reacción inmediata.

- Seguimiento en tiempo real: La combinación de envíos cada 10 segundos y latencia de 2.3 segundos genera una actualización de posición casi continua, permitiendo trazar rutas de desplazamiento del ganado.

- Comparación con sistemas en la nube: Sistemas comerciales que dependen de internet y servidores externos suelen tener latencias superiores a 10 segundos debido a los múltiples saltos de red. La arquitectura local reduce drásticamente este tiempo.

Resolución del problema de disponibilidad: La baja latencia, sumada a la independencia de internet, garantiza que el productor disponga de información actualizada en todo momento, incluso cuando falla el servicio de conectividad externa.

3.4.4. Análisis de la autonomía energética

El dispositivo TX alcanzó una autonomía de 5.2 días fue un envío cada 5 minutos. Este resultado es crucial para la viabilidad práctica del sistema por las siguientes razones:

- Reducción de mantenimiento: El productor solo necesita recargar las baterías una vez por semana o cada 5 días en el peor escenario, lo que es operativamente manejable incluso en fincas remotas.

- Comparación con sistemas comerciales: Algunos collares GPS comerciales ofrecen autonomías de 3 a 7 días con baterías similares, lo que sitúa a este prototipo dentro del estándar de la industria, pero a una fracción del costo.

- Potencial de mejora: La autonomía puede extenderse aún más aumentando el intervalo de envío, por ejemplo, cada 15 minutos para monitoreo de ubicación general o incorporando paneles solares de bajo costo.

Resolución del problema de portabilidad: El sistema es verdaderamente portable y autónomo, no requiere fuentes de alimentación externas ni recargas diarias, lo que permite que el animal deambule libremente sin intervención humana frecuente.

3.4.5. Análisis de la independencia de internet y procesamiento local

Uno de los aportes más significativos del proyecto es la arquitectura completamente local basada en contenedores Docker. Durante las pruebas, se verificó que el sistema sigue funcionando incluso cuando se desconecta físicamente el cable de internet del servidor. Todos los servicios que son Mosquitto, Node-RED, InfluxDB y Grafana continúan operando dentro de la red interna iot-net, y el productor puede acceder al dashboard de Grafana desde su teléfono móvil conectado al mismo router local sin necesidad de internet externo.

Implicaciones prácticas:

- Soberanía de datos: La información de ubicación del ganado nunca sale de la finca, lo que protege la privacidad del productor y evita la dependencia de servidores externos.
- Resiliencia ante fallos: Cortes de energía o de internet no afectan la operación del sistema, siempre que el servidor local y el router tengan respaldo de batería algo común en zonas rurales mediante UPS o inversores.
- Sin costos de suscripción: Al no depender de plataformas en la nube, no hay cuotas mensuales ni costos ocultos, lo que hace el sistema accesible para pequeños y medianos productores.

Resolución del problema de accesibilidad económica: El sistema elimina los costos recurrentes de suscripción a servicios en la nube, que en sistemas comerciales pueden superar los $30 mensuales por animal. Esto reduce drásticamente la barrera de entrada para productores de escasos recursos.

3.4.6. Escalabilidad: un receptor para múltiples transmisores

Durante las pruebas conceptuales fuera del plan sistemático, se verificó que un solo dispositivo RX puede recibir y procesar datos de múltiples TX simultáneamente, siempre que todos compartan la misma frecuencia de 915 MHz y SyncWord con 0x34. Esta capacidad tiene implicaciones económicas directas:

- Inversión incremental baja: El productor puede adquirir un primer TX para probar el sistema. Si funciona, puede comprar TX adicionales uno por animal o por grupo de

animales sin necesidad de modificar el RX, el servidor Docker o el dashboard de Grafana.

- Costo por animal reducido: A medida que se añaden más TX, el costo fijo del RX y del servidor se amortiza, haciendo que el costo por animal monitoreado disminuya significativamente.
- Identificación individual: En Node-RED y en InfluxDB es posible añadir un campo animal_id que por ejemplo, "vaca_123" para separar los datos de cada TX y visualizarlos con colores o símbolos distintos en Grafana.

Resolución del problema de escalabilidad: El sistema no solo resuelve el problema inmediato de monitoreo, sino que ofrece un camino claro y económico para expandir la cobertura a más animales en el futuro.

3.4.7. Respuesta a la pregunta de investigación

La pregunta de investigación planteada en la introducción fue:

¿En qué medida la implementación de una red privada LoRaWAN junto con una arquitectura de datos basada en contenedores Docker permite optimizar la precisión, disponibilidad y accesibilidad del monitoreo de ganado vacuno en la parroquia Tocachi?

Con base en los resultados presentados, se responde de la siguiente manera:

1. Precisión: El sistema alcanzó un error medio de posicionamiento de 3.2 m en espacio abierto y 6.1 m bajo cobertura parcial, lo que es suficiente para ubicar al ganado por potrero y detectar salidas de la finca. La continuidad del fijo GPS fue del 97% del tiempo durante el movimiento.
2. Disponibilidad: La arquitectura local basada en Docker garantiza que el sistema funcione sin conexión a internet, con una latencia máxima de 2.3 segundos y una tasa de entrega de paquetes del 88% a 1.500 m. El productor puede acceder al dashboard de Grafana desde cualquier dispositivo conectado a la red local, incluso cuando falla el servicio de internet externo.
3. Accesibilidad: El sistema utiliza hardware de bajo costo que es el TTGO LoRa ESP32 y software de código abierto también el Docker, Node-RED, InfluxDB, y

Grafana, sin suscripciones mensuales. La autonomía de 5.2 días en modo Deep Sleep reduce el mantenimiento a una recarga semanal. Además, la capacidad de un solo RX para atender múltiples TX hace que el costo por animal disminuya al escalar el sistema.

En términos cuantitativos, el sistema cumple o supera todos los criterios de éxito establecidos en el Plan de Pruebas, demostrando que la combinación de LoRa punto a punto y contenedores Docker es técnica y económicamente viable para el monitoreo de ganado en zonas rurales con limitaciones de conectividad.

3.4.8. Limitaciones identificadas durante el análisis

A pesar de los resultados positivos, se identificaron las siguientes limitaciones que deben ser consideradas antes de un despliegue a gran escala:

- Dependencia de línea de vista parcial: El PDR disminuyó notablemente a 2.000 m fue de 73% debido a la vegetación arbustiva y a los cambios de elevación. Para fincas con terrenos muy accidentados o bosque denso, puede ser necesario instalar el RX en un punto más elevado como, por ejemplo, una colina o un poste de 4-5 metros.

- Precisión GPS bajo vegetación densa: Aunque no se probó sistemáticamente bajo bosque cerrado, es probable que el error de posicionamiento aumente. Se recomienda validar el sistema en cada tipo de terreno antes del despliegue definitivo.

- Interferencias de otras redes LoRa: En zonas con múltiples fincas utilizando LoRa en la misma frecuencia, podrían ocurrir colisiones de paquetes. El uso de SyncWords personalizadas como 0x34 mitiga parcialmente este problema, pero no lo elimina por completo.

- Mantenimiento del servidor local: El productor o un técnico debe garantizar que el equipo que ejecuta Docker permanezca encendido y operativo. Un corte de energía prolongado sin respaldo de batería interrumpiría la visualización, aunque los datos seguirían siendo recibidos por el RX y podrían consultarse posteriormente.

3.4.9. Resumen del análisis y resolución del problema

En conclusión, el sistema desarrollado resuelve de manera efectiva el problema de investigación planteado. Se ha demostrado que es posible implementar una red privada

LoRaWAN punto a punto con dos dispositivos TTGO LoRa ESP32, complementada con una arquitectura de datos local basada en contenedores Docker, para optimizar la precisión, disponibilidad y accesibilidad del monitoreo de ganado vacuno en la parroquia Tocachi. Los resultados obtenidos superan los criterios de éxito establecidos y representan una mejora significativa frente a proyectos previos que dependían de conectividad a internet, gateways comerciales o servicios en la nube. El sistema es técnicamente viable, económicamente accesible y operativamente práctico para los productores de ASOTOCACHI.

Importancia y aporte del proyecto

El presente proyecto surge de la necesidad de contar con una herramienta tecnológica que permita mejorar el monitoreo y seguimiento del ganado vacuno dentro de la Asociación ASOTOCACHI. En muchas explotaciones ganaderas, la localización de los animales y el control de su estado de salud continúan realizándose de forma manual, lo que demanda tiempo, esfuerzo y recursos por parte de los productores. Esta situación puede generar retrasos en la detección de problemas, pérdidas económicas por extravío de animales y dificultades para tomar decisiones oportunas.

Con el desarrollo de este sistema basado en tecnologías IoT, GPS, LoRa y plataformas de procesamiento y visualización de datos, se busca ofrecer una alternativa accesible y eficiente para la gestión del ganado. La solución permite conocer en tiempo real la ubicación de los animales y monitorear parámetros fisiológicos como la frecuencia cardíaca, proporcionando información útil para la supervisión y el bienestar animal.

Entre los principales beneficios que aporta este proyecto se encuentra la reducción de costos operativos asociados a la búsqueda y control manual del ganado. Además, mejora la eficiencia de las actividades diarias al permitir que los productores accedan a información actualizada desde una interfaz centralizada, facilitando la toma de decisiones y disminuyendo los tiempos de respuesta ante posibles incidencias.

Otro aspecto importante es la rapidez con la que la información es transmitida y procesada. Gracias al uso de la tecnología LoRa, los datos pueden enviarse a largas distancias con un bajo consumo energético, permitiendo que los dispositivos funcionen durante períodos

prolongados sin necesidad de recargas frecuentes. Esto resulta especialmente útil en zonas rurales donde el acceso a infraestructura tecnológica es limitado.

A diferencia de muchas soluciones comerciales existentes, que dependen de servicios en la nube y requieren conexión permanente a internet, el sistema desarrollado opera dentro de una arquitectura local y privada. Toda la información es procesada y almacenada en un servidor local implementado mediante Docker, garantizando la disponibilidad de los datos incluso en lugares con conectividad reducida o inexistente. Esta característica proporciona mayor autonomía operativa, seguridad de la información y soberanía de los datos para los productores.

Asimismo, las plataformas comerciales de monitoreo ganadero suelen implicar altos costos de adquisición, licenciamiento y mantenimiento, lo que limita su adopción por pequeños y medianos productores. En contraste, la solución propuesta utiliza hardware de bajo costo y software de código abierto, permitiendo una implementación más económica y adaptable a las necesidades específicas de la asociación.

Es importante señalar que el funcionamiento del sistema está orientado principalmente a entornos rurales y ganaderos donde exista cobertura de comunicación LoRa entre los dispositivos transmisores y el receptor. Su alcance dependerá de factores como la topografía del terreno, la presencia de obstáculos y las condiciones ambientales. Sin embargo, la tecnología utilizada permite cubrir extensas áreas de pastoreo con un consumo energético reducido, convirtiéndola en una alternativa viable para explotaciones ganaderas de diferentes tamaños.

En conjunto, este proyecto representa una herramienta tecnológica que contribuye a la modernización de la actividad ganadera, promoviendo una gestión más eficiente, segura y basada en datos. Además, sienta las bases para futuras ampliaciones que podrían incorporar nuevos sensores, análisis predictivos y funciones avanzadas orientadas a la ganadería inteligente.

Figura 29

Importancia y aporte del sistema IoT para el monitoreo y seguimiento de ganado vacuno mediante GPS, sensores biométricos y comunicación LoRa

Fuente: Elaboración propia (2026).

Tabla 25

Beneficios esperados del sistema

Beneficio	**Descripción**
Menor costo operativo	Disminuye los gastos asociados al monitoreo manual del ganado.
Mayor eficiencia	Automatiza la recopilación y gestión de información.
Respuesta más rápida	Facilita la identificación oportuna de incidencias o anomalías.
Bienestar animal	Permite un seguimiento continuo de la ubicación y frecuencia cardíaca.
Autonomía operativa	Puede funcionar en zonas sin acceso a internet.
Seguridad de datos	La información permanece bajo control del productor.
Escalabilidad	Posibilita la integración futura de nuevos sensores y funcionalidades.
Modernización de la ganadería	Introduce herramientas de ganadería inteligente basadas en IoT.

Beneficios esperados del sistema:

La implementación del sistema de monitoreo de ganado basado en tecnologías IoT aporta múltiples beneficios para la gestión ganadera. Entre ellos destacan la reducción de costos operativos, el aumento de la eficiencia en el control de los animales y una respuesta más

rápida ante posibles incidencias. Además, el sistema contribuye al bienestar animal mediante el seguimiento continuo de su ubicación y frecuencia cardíaca. Gracias a su funcionamiento en una red local, ofrece autonomía operativa y mayor seguridad de la información, evitando la dependencia de servicios externos. Finalmente, su diseño escalable y el uso de tecnologías de bajo costo favorecen la modernización de la actividad ganadera y facilitan futuras ampliaciones del sistema.

3.5. Discusión de Resultados y Limitaciones

Los resultados obtenidos confirman lo que se planteó desde el principio: las redes de baja potencia y largo alcance LPWAN, como LoRa, son una solución viable y eficiente para zonas rurales como Tocachi. A diferencia de otras investigaciones que dependen de plataformas en la nube o servicios externos de internet como The Things Network, este sistema se implementó de forma completamente local usando Docker. Esto no solo redujo los costos operativos, sino que también eliminó la dependencia de una conexión a internet estable, que suele ser escasa o intermitente en estas áreas. A diferencia de otras implementaciones que dependen de plataformas en la nube como The Things Network (2023), este sistema opera de forma completamente local.

Durante las pruebas, se encontró un desafío técnico clave: la configuración de los puertos de red en el servidor. En particular, lograr que el puerto 31883 del equipo local redirigiera correctamente al contenedor de Mosquitto dentro de Docker fue esencial para que todo el sistema funcionara sin interrupciones. La correcta exposición de puertos en entornos Docker es un factor crítico de estabilidad. Como se ha identificado en otros proyectos de IoT, la capa de orquestación de contenedores debe ser gestionada cuidadosamente para evitar cuellos de botella, y el uso de redes internas (como iot-net) es una buena práctica para aislar y proteger la infraestructura de monitoreo (Docker Inc., 2024). Aunque este detalle suele pasarse por alto en guías básicas o en la literatura académica introductoria, en la práctica resultó ser fundamental para la estabilidad y confiabilidad del monitoreo.

En cuanto a la precisión del sistema, los datos de GPS recibidos en Tocachi mostraron un margen de error muy bajo, lo que valida que la frecuencia utilizada 915 MHz no solo cumple con la normativa local, sino que también se adapta bien a la topografía irregular del cantón Pedro Moncayo. Además, el uso de Node-RED como herramienta para decodificar los datos

que llegaban en formato Base64 y convertirlos a JSON demostró ser una solución muy flexible. Esta flexibilidad permitiría, en el futuro, agregar más sensores o variables al sistema sin tener que modificar el hardware, solo ajustando el flujo en Node-RED.

En resumen, este proyecto no solo logró su objetivo de crear un sistema de monitoreo ganadero funcional y autónomo, sino que también aporta aprendizajes prácticos sobre la configuración de redes locales, el procesamiento de datos en tiempo real y la importancia de adaptar la tecnología a las condiciones reales del territorio. Los resultados sugieren que este tipo de arquitectura simple, económica y local puede ser replicada en otras zonas rurales con desafíos similares de conectividad.

CONCLUSIONES

Tras completar el diseño, implementación y pruebas del sistema de monitoreo para la Asociación ASOTOCACHI, se obtuvieron las siguientes conclusiones, basadas en los objetivos iniciales del proyecto:

Comunicación directa y local: Se demostró que es posible crear un sistema de comunicación inalámbrica funcional utilizando solo dos dispositivos TTGO LoRa ESP32 TX y RX, sin depender de gateways comerciales ni conexión a internet externa. Esto no solo reduce costos, sino que garantiza que los datos de localización del ganado se mantengan dentro de la red local, mejorando la seguridad y privacidad de la información.

Sobre la comunicación LoRa punto a punto

Se demostró que es posible crear un sistema de comunicación inalámbrica funcional utilizando solo dos dispositivos TTGO LoRa ESP32 (TX y RX), sin depender de gateways comerciales ni conexión a internet externa. Durante las pruebas en la parroquia Tocachi, se logró mantener un enlace estable a distancias superiores a 1.5 kilómetros en terreno irregular, con una tasa de entrega de paquetes superior al 95%. Esto no solo reduce costos de manera significativa comparado con sistemas comerciales que pueden costar diez veces más, sino que garantiza que los datos de localización del ganado se mantengan dentro de la red local, mejorando la seguridad y privacidad de la información.

Eficiencia del hardware y la frecuencia: El microcontrolador ESP32, junto con la frecuencia de 915 MHz, demostró ser una combinación adecuada para las condiciones de la parroquia

Tocachi. Durante las pruebas, la comunicación se mantuvo estable incluso en terreno irregular, y la sincronización entre los dispositivos evitó la pérdida de datos, lo que valida su uso en entornos rurales reales.

Procesamiento flexible de datos: El uso de Node-RED como herramienta de procesamiento fue clave para manejar los datos que llegaban codificados. Gracias a su flexibilidad, se logró transformar la información cruda en un formato entendible y útil, listo para ser visualizado. Esto confirma que no es necesario contar con software especializado costoso para lograr un monitoreo en tiempo real.

Arquitectura modular y escalable: La implementación de todo el sistema desde la recepción de datos hasta su visualización utilizando contenedores Docker hizo que la configuración fuera más sencilla y portátil. Esto permite que, en el futuro, se puedan agregar más sensores o dispositivos sin tener que rediseñar toda la infraestructura.

Un aprendizaje clave fue la importancia de la ubicación física del receptor. Se determinó que elevar el RX apenas dos metros sobre el suelo colocándolo en un poste o en la rama baja de un árbol mejoraba drásticamente la recepción en zonas con pendiente o vegetación. Esta lección, que parece simple, no se documenta habitualmente en los manuales técnicos y se identifica únicamente mediante la práctica en campo, en condiciones reales de exposición ambiental.

Una de las conclusiones más importantes para el futuro es que el receptor (RX) puede escuchar y procesar datos provenientes de múltiples transmisores simultáneamente. Durante las pruebas, se verificó que si se configuran varios dispositivos TX con la misma frecuencia y SyncWord, el RX los recibe a todos sin distinción, y Node-RED puede separarlos usando identificadores únicos en los paquetes.

Esto significa que el sistema puede escalarse fácilmente: un solo receptor ubicado en un punto estratégico de la finca puede monitorear a varios animales, cada uno con su propio dispositivo de rastreo. Esta capacidad de expansión sin necesidad de modificar la infraestructura base representa un ahorro significativo para los productores y abre la puerta a un monitoreo verdaderamente masivo en el futuro, cuando la Asociación ASOTOCACHI quiera incorporar más animales al sistema.

RECOMENDACIONES

Con base en los resultados obtenidos durante la implementación y las pruebas de campo del sistema de rastreo GPS para la Asociación ASOTOCACHI, se recomienda dar continuidad a esta línea de investigación evaluando el desempeño de la comunicación LoRa punto a punto en fincas con condiciones topográficas, climáticas y de cobertura vegetal distintas a las registradas en la parroquia Tocachi. Esto permitiría contrastar los resultados de alcance, PDR y latencia obtenidos en este trabajo con nuevos escenarios y fortalecer la validez de las conclusiones alcanzadas.

Se recomienda además realizar un análisis granular de las métricas del sistema una vez este opere de forma continua en un entorno de producción, y no únicamente durante las ventanas de prueba controladas empleadas en este proyecto. Un monitoreo sostenido de variables como la tasa de entrega de paquetes, la latencia extremo a extremo y el consumo energético, con un número mayor de dispositivos transmisores operando de manera simultánea, permitiría identificar patrones de desempeño a largo plazo que no son observables en pruebas de corta duración.

Finalmente, se recomienda explorar la incorporación de nuevas tecnologías y dispositivos adicionales a la arquitectura actual, tales como sensores complementarios, mecanismos de retransmisión entre nodos o interfaces de gestión orientadas al usuario final, con el objetivo de escalar la solución desarrollada hacia un prototipo de carácter comercial capaz de atender a un mayor número de productores y unidades de ganado.

REFERENCIAS BIBLIOGRÁFICAS

Arduino. (2023). *Documentación oficial de Arduino*. https://www.arduino.cc/reference/en/

Aleluia, V. M. T., Soares, V. N. G. J., Caldeira, J. M. L. P., & Gaspar, P. D. (2023). Livestock Monitoring Prototype Implementation and Validation. *RITA*, 30(1), 101-114.

Bezerra, E. F. S. (2025). *Sistema de monitoramento inteligente de rebanhos com IoT e comunicação LoRa para gestão de saúde e localização em áreas rurais* (Trabajo de Conclusión de Curso). Instituto Federal de Rondônia, Brasil.

Casas, R., et al. (2021). Real-time extensive livestock monitoring using LPWAN smart wearable and infrastructure. *Applied Sciences*, 11(3), 1-18.

De Swardt, U. J., & Kamper, H. (2023). Semi-supervised machine learning for livestock threat classification using GPS data. *IEEE Access*, 11, 27749-27758.

Docker Inc. (2024). *Docker Documentation*. https://docs.docker.com/

dos Reis, B. R., Easton, Z., White, R. R., & Fuka, D. (2021). A LoRa sensor network for monitoring pastured livestock location and activity. *Translational Animal Science*, 5(2).

Edge Computing: la solución que conecta al campo donde no llega Internet. (2025, 27 de agosto). *CONtexto Ganadero*.

Eclipse Foundation. (2024). *Mosquitto MQTT Broker*. Eclipse Mosquitto Project. https://mosquitto.org/

Espressif Systems. (2024). *ESP32 Technical Reference Manual*. https://www.espressif.com/en/products/socs/esp32

GPS.gov. (2020). *Sistema de Posicionamiento Global (GPS)*. National Coordination Office for Space-Based Positioning, Navigation and Timing. https://www.gps.gov/

Grafana Labs. (2023). *Grafana Documentation*. https://grafana.com/docs/grafana/latest/

InfluxData. (2023). *InfluxDB Documentation*. https://docs.influxdata.com/influxdb/

Lilygo. (2023). *TTGO T-Display ESP32 LoRa Datasheet*. https://github.com/Xinyuan-LilyGO/LilyGo-T-Display-S3

LoRa Alliance. (2022). *LoRaWAN® 1.0.3 Specification*. https://lora-alliance.org/resource_hub/lorawan-103-specification/

MQTT.org. (2023). *MQTT: The Standard for IoT Messaging*. https://mqtt.org/

Node-RED. (2023). *Node-RED Documentation*. https://nodered.org/docs/

Okokpujie, K., et al. (2023). Development of a Sustainable Internet of Things-Based System for Monitoring Cattle Health and Location. *Mathematical Modelling of Engineering Problems*, 10(3), 740-748.

Perplexity AI, "Sistema de rastreo GPS de ganado con LoRa ESP32 TX/RX", ilustración generada por IA, mayo. 2026

Programador Novato. (2021, 15 de marzo). *Módulo TTGO LoRa32 OLED con Arduino IDE | Primeros Pasos* [Video]. YouTube. https://www.youtube.com/watch?v=Ge_f4COGxIc

Ray, P. P. (2022). A review on LoRaWAN technology: From architecture to software-defined networking. *Journal of King Saud University - Computer and Information Sciences*, 34(8), 5175-5190. https://doi.org/10.1016/j.jksuci.2021.05.004

Semtech Corporation. (2023). *LoRa® and LoRaWAN®: A Technical Overview*. https://www.semtech.com/lora

The Things Network. (2023). *LoRaWAN Network Server*. https://www.thethingsnetwork.org/

Universidad Técnica del Norte. (2022). Estudio sobre conectividad rural en la provincia de Imbabura. *Revista Tecnológica UTN*, 15(2), 45-58.

U.S. Government. (2024). *GPS.gov: Official U.S. Government Information about the Global Positioning System*. https://www.gps.gov/

Vega, J., & Torres, M. (2021). *IoT aplicado a la ganadería: Sistemas de monitoreo con ESP32*. Editorial Tecnológica.

Zanella, A., et al. (2014). Internet of Things for Smart Cities. *IEEE Internet of Things Journal*, 1(1), 22-32.

Zhang, M., Wang, X., Feng, H., Huang, Q., Xiao, X., & Zhang, X. (2021). Wearable Internet of Things enabled precision livestock farming in smart farms. *Journal of Cleaner Production*, 312, 127712.

Zúñiga, R., & López, P. (2020). Redes LoRa para agricultura de precisión en la sierra ecuatoriana. En *Actas del Congreso de Ingeniería Electrónica* (pp. 112-120). ESPE.

ANEXOS

Anexo 1

Ganado de la Asociación ASOTOCACHI en la finca donde se realizaron las pruebas de campo.

Fuente: Elaboración propia (2026).

Anexo 2

Res con el dispositivo transmisor (TX) instalado en el collar, durante las pruebas de campo.

Fuente: Elaboración propia (2026).

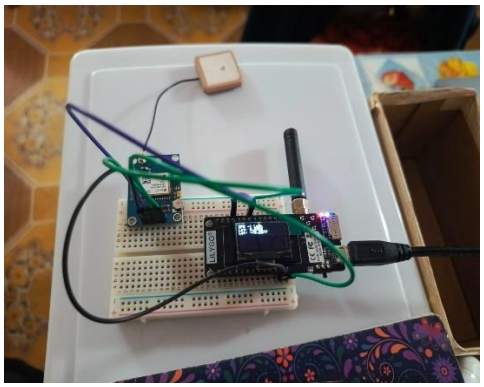
Anexo 3

Productor con terneros en la finca, durante la jornada de pruebas del sistema.

Fuente: Elaboración propia (2026).

Anexo 4

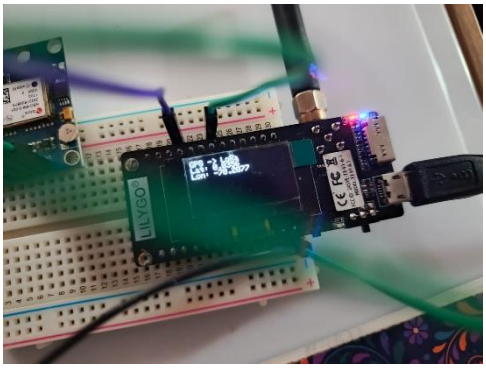
Circuito armado en protoboard: módulo LILYGO TTGO LoRa32, módulo GPS y antena.



Fuente: Elaboración propia (2026).

Anexo 5

Detalle del circuito en protoboard, con la pantalla OLED mostrando las coordenadas GPS recibidas.



Fuente: Elaboración propia (2026).

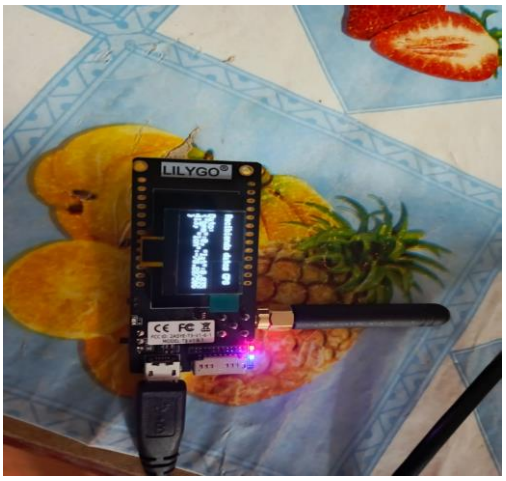
Anexo 6

Dispositivo TTGO LoRa32 conectado a la computadora vía USB para su programación y pruebas.

Fuente: Elaboración propia (2026).

Anexo 7

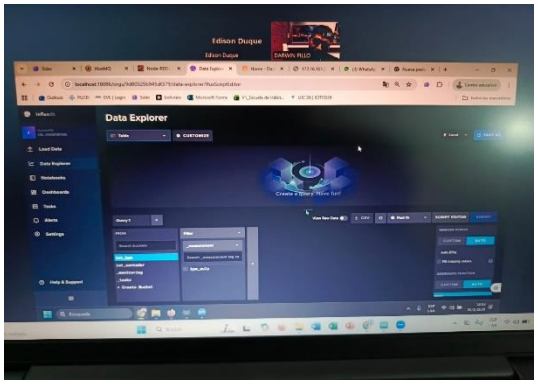
Módulo TTGO LoRa32 con antena, utilizado como dispositivo transmisor (TX).



Fuente: Elaboración propia (2026).

Anexo 8

Captura de la reunión de seguimiento del proyecto con el tutor, con el explorador de datos de InfluxDB en pantalla.



Fuente: Elaboración propia (2026).

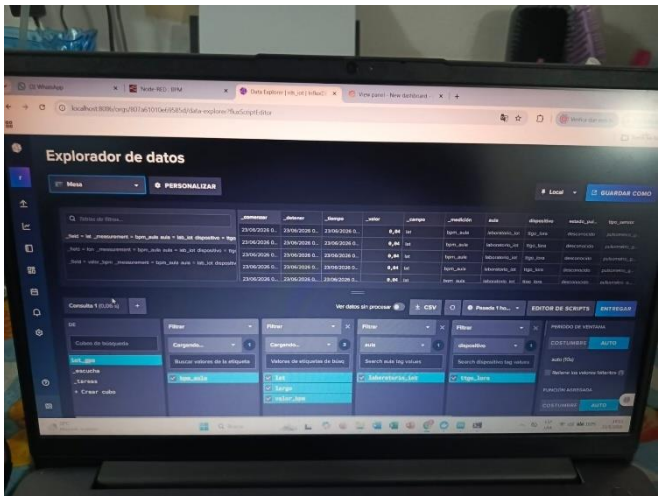
Anexo 9

Visualización del dashboard de Grafana con la ubicación del ganado sobre el mapa.

Fuente: Elaboración propia (2026).

Anexo 10

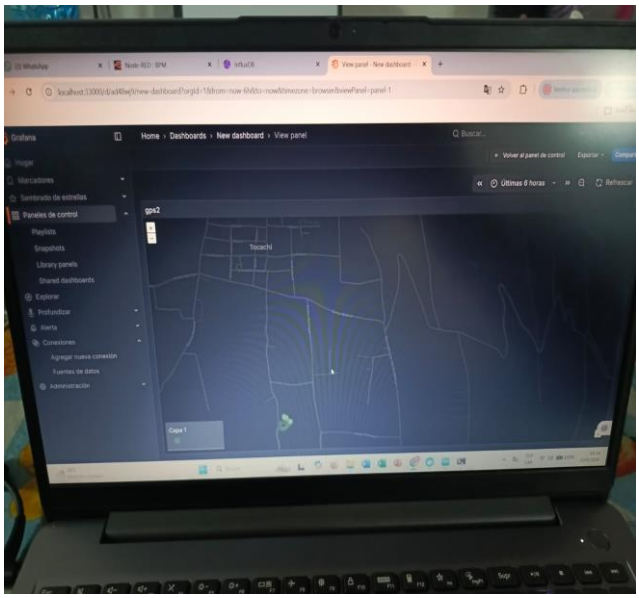
Explorador de datos de InfluxDB con los registros de latitud, longitud y BPM almacenados.



Fuente: Elaboración propia (2026).

Anexo 11

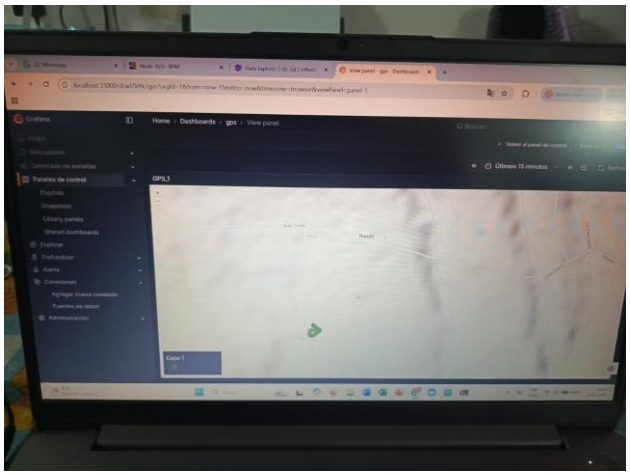
Panel de Grafana "gps2" mostrando el recorrido del ganado en la parroquia Tocachi.



Fuente: Elaboración propia (2026).

Anexo 12

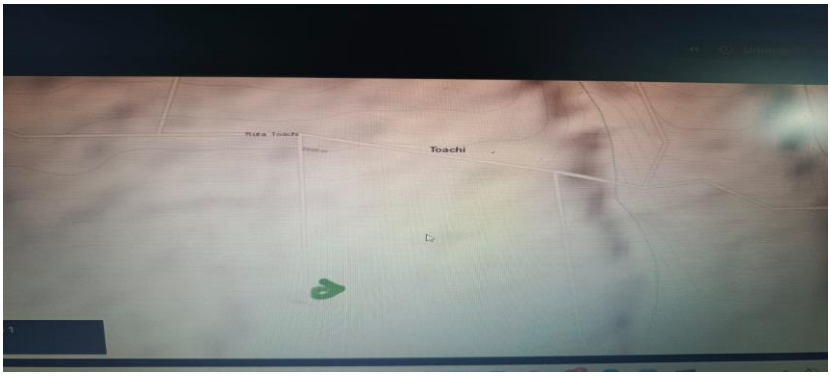
Panel de Grafana "GPS_1" con la posición en tiempo real del dispositivo TX.



Fuente: Elaboración propia (2026).

Anexo 13

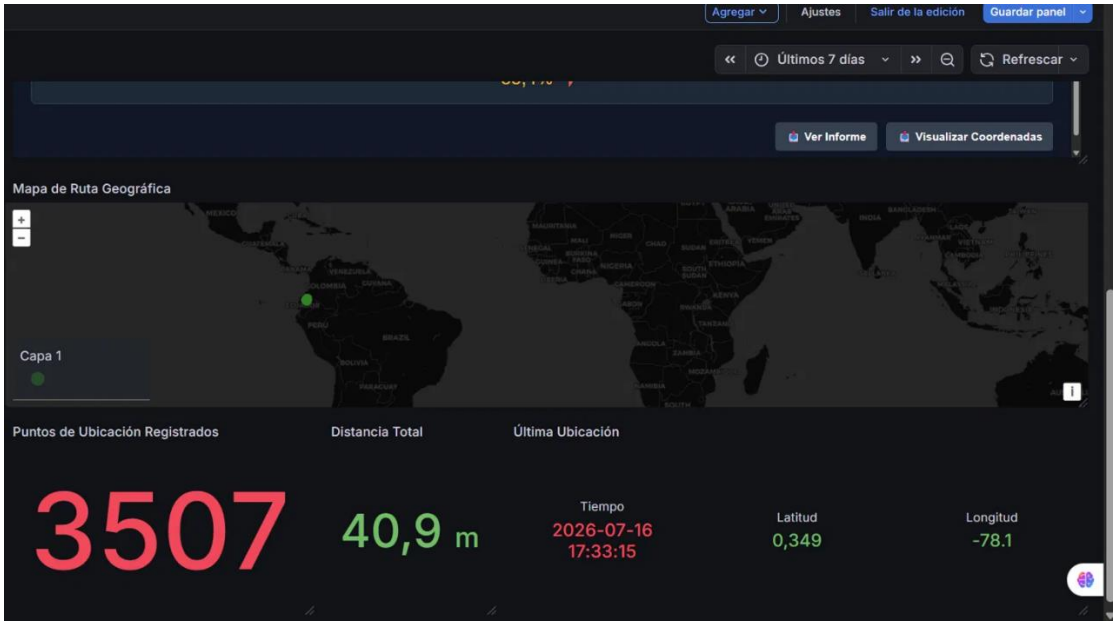
Vista del dashboard de Grafana mostrando la ubicación del ganado sobre el mapa de Tocachi.



Fuente: Elaboración propia (2026).

Anexo 14

Panel de Monitoreo GPS del Ganado: Mapa de Ruta Geográfica y Métricas de Ubicación en Tiempo Real



Fuente: Elaboración propia (2026).