



UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

FACSISTEL

INGENIERÍA EN TELECOMUNICACIONES

TRABAJO DE INTEGRACIÓN CURRICULAR

**Monitoreo de ganado en zonas rurales mediante comunicación
LoRa y geolocalización para aplicaciones de seguridad.**

Fiorella Verónica Pillasagua Yagual

Dirigido por:

Ing. Carlos Andrade, M.Sc.

La Libertad - 2026



UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

Ing. Ronald Rovira Jurado, Ph.D.

DIRECTOR DE LA CARRERA

Ing. Daniel Jaramillo Chamba, M.Sc.

DOCENTE ESPECIALISTA

Ing. Carlos Efraín Andrade, M.Sc.

DOCENTE TUTOR

Ing. Luis Amaya Farfán, Mgtr.

DOCENTE GUIA UIC

Ing. Corina Gonzabay De la A, Mgtr.

SECRETARIA



UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por la cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por **Fiorella Verónica Pillasagua Yagual**, como requerimiento para la obtención de título de Ingeniero en Telecomunicaciones.

La Libertad, a los 04 días del mes de agosto del año 2026.

Ing. Carlos Efraín Andrade, M.Sc.

DOCENTE TUTOR



UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Fiorella Verónica Pillasagua Yagual** con C.I.: 2400124976 quien presentó el trabajo de titulación, **Monitoreo de ganado en zonas rurales mediante comunicación LoRa y geolocalización para aplicaciones de seguridad**, previo a la obtención del título de Ingeniera en Telecomunicaciones, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del trabajo de titulación referido.

La Libertad, a los 04 días del mes de agosto del año 2026.

Fiorella Pillasagua

Fiorella Verónica Pillasagua Yagual

AUTOR



UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado Monitoreo de ganado en zonas rurales mediante comunicación LoRa y geolocalización para aplicaciones de seguridad, presentado por la estudiante Fiorella Verónica Pillasagua Yagual fue enviado al sistema antiplagio, presentando un porcentaje de similitud correspondiente al 4%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.



Certificado de análisis
Compilatio Magister+ | UPSE-ECU

Trabajo de Integración Curricular_PILLASAGUA 1

ID : 2d61fa2c747d84e9d6099d8d26eae1e768834838



4%

Textos sospechosos

Nombre del fichero : Trabajo de Integración Curricular_PILLASAGUA 1.txt
Tamaño del archivo original : 4.37 MB
Número de palabras : 27.912

Depositante : AMAYA FARÍÑO LUIS MIGUEL
Fecha de depósito : 20 de agosto de 2026
Tipo de carga : interface
Fecha de fin de análisis : 20 de agosto de 2026

Ing. Carlos Efraín Andrade, M.Sc.

DOCENTE TUTOR



UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

AUTORIZACIÓN

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales de artículo profesional de alto nivel con fines de difusión pública, además apruebo la reproducción de este artículo académico dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

La Libertad, a los 04 días del mes de agosto del año 2026.

Fiorella Pillasagua

Fiorella Verónica Pillasagua Yagual

AUTOR

AGRADECIMIENTO

Quiero expresar mi más sincero agradecimiento al Ing. Carlos Andrade por su guía, dedicación y apoyo durante el desarrollo de este proyecto de titulación. Sus conocimientos, orientación y constante disposición para resolver inquietudes fueron fundamentales para culminar satisfactoriamente este trabajo.

De igual manera, agradezco al Ing. Luis Amaya por sus valiosas recomendaciones y acertados consejos, los cuales contribuyeron al fortalecimiento técnico de este proyecto. Asimismo, extendiendo mi agradecimiento al Ing. Daniel Jaramillo por sus observaciones y aportes que permitieron mejorar diferentes fases del desarrollo e implementación del sistema.

Expreso mi gratitud a la Universidad Estatal Península de Santa Elena por abrirme sus puertas y brindarme la formación académica, los conocimientos y las herramientas necesarias para mi crecimiento profesional. Finalmente, a todos quienes de una u otra manera contribuyeron a la realización de este proyecto, les expreso mi más profundo agradecimiento.

Fiorella Verónica Pillasagua Yagual

DEDICATORIA

Dedico este trabajo, en primer lugar, a Dios por darme la fortaleza, la sabiduría y la perseverancia necesarias para superar cada dificultad presentada durante este camino.

A mi querida familia Pillasagua Yagual, por ser mi mayor motivación durante toda esta etapa académica. Gracias por confiar en mí, por su apoyo incondicional, sus palabras de aliento y por impulsarme a seguir adelante incluso en los momentos más difíciles.

De manera especial, dedico este logro a Dilan Villacís, por estar presente a lo largo de este proceso, brindándome su apoyo, ayuda y valiosos consejos. Gracias por motivarme a continuar y por acompañarme durante esta importante etapa de mi vida.

Este trabajo representa el esfuerzo, la dedicación y la constancia invertidos durante estos años, y con profundo cariño lo comparto con quienes hicieron posible que este sueño se convirtiera en realidad.

Fiorella Verónica Pillasagua Yagual

RESUMEN

En las zonas rurales, el monitoreo del ganado significa un desafío, debido a las grandes extensiones de terreno donde los animales permanecen en pastoreo dificultan conocer la ubicación en todo momento del ganado y es más probable que se presenten situaciones de extravío. En este contexto, mi trabajo de investigación tuvo como objetivo la implementación de un dispositivo; para desarrollar el prototipo, se diseñó un collar equipado con un módulo GPS NEO-6M y una placa Heltec Wifi LoRa 32 V4. La información del posicionamiento fue enviada mediante comunicación LoRa hacia un dispositivo receptor, el cual retransmitió los datos de la ubicación a las plataformas de Blynk y Telegram mediante conectividad WiFi. También se implementó el modo Deep Sleep como estrategia para reducir el consumo energético y aumentar la autonomía del sistema.

Con las pruebas experimentales se demostró que el módulo GPS obtuvo coordenadas válidas detectando entre 10 y 12 satélites con tiempos de adquisición entre 19 y 23 segundos. El enlace LoRa mantuvo la comunicación estable hasta aproximadamente 1 km, alcanzando una efectividad del 96,93% y una pérdida de datos del 3,07%. Además, se evidenció una diferencia de 1 metro entre las distancias reportadas por el sistema y la que indica Google Maps. Por último, sobre el desempeño energético, la implementación del modo Deep Sleep maximizó la autonomía de la batería de una estimación teórica de 10,9 horas a 28 horas de funcionamiento continuo comprobado en pruebas experimentales.

Los resultados que se obtuvieron demuestran la viabilidad y confiabilidad del prototipo como una alternativa tecnológica para el monitoreo y localización del ganado en zonas rurales, lo cual facilita las labores de supervisión por parte del propietario.

ABSTRACT

In rural areas, livestock monitoring represents a challenge due to the large grazing areas where animals remain, making it difficult to determine their location at all times and increasing the likelihood of animals becoming lost. In this context, the objective of this research was to implement a monitoring device. To develop the prototype, a smart collar was designed using a NEO-6M GPS module and a Heltec WiFi LoRa 32 V4 board. Positioning information was transmitted through LoRa communication to a receiver device, which then forwarded the location data to the Blynk and Telegram platforms via Wi-Fi connectivity. In addition, the Deep Sleep mode was implemented as an energy-saving strategy to reduce power consumption and extend the system's battery life.

Experimental tests demonstrated that the GPS module obtained valid coordinates by detecting between 10 and 12 satellites, with acquisition times ranging from 19 to 23 seconds. The LoRa communication link maintained a stable connection over a distance of approximately 1 km, achieving a reception rate of 96.93% with a data loss rate of 3.07%. Furthermore, the difference between the distance reported by the system and that measured using Google Maps was only 1 meter. Regarding energy performance, the implementation of the Deep Sleep mode increased the battery life from a theoretical estimate of 10.9 hours to 28 hours of continuous operation, as verified through experimental testing.

The obtained results demonstrate the feasibility and reliability of the proposed prototype as a technological alternative for livestock monitoring and location in rural areas, facilitating supervision tasks by the livestock owner.

ÍNDICE DE CONTENIDO

TRIBUNAL DE SUSTENTACIÓN	II
CERTIFICACIÓN	III
DECLARACIÓN DE RESPONSABILIDAD	IV
CERTIFICACIÓN DE ANTIPLAGIO	V
AUTORIZACIÓN	VI
AGRADECIMIENTO	VII
DEDICATORIA	VIII
RESUMEN	IX
ABSTRACT	X
ÍNDICE DE CONTENIDO	XI
ÍNDICE DE FIGURAS	XVI
ÍNDICE DE TABLAS	XVIII
ÍNDICE DE ECUACIONES	XIX
ÍNDICE DE ANEXOS	XIX
ÍNDICE DE ABREVIATURAS	XX
INTRODUCCIÓN	24
CAPÍTULO I: ANTEPROYECTO	26
1.1. Antecedentes	26
1.2. Planteamiento del problema	26
1.3. Descripción del proyecto	27
1.4. Objetivos	28
1.4.1. Objetivo general	28
1.4.2. Objetivos específicos	28
1.5. Justificación	28

1.6.	Alcance	29
1.7.	Metodología	30
1.8.	Resultados esperados	31
CAPÍTULO II: MARCO TEÓRICO		32
2.1.	Conceptos fundamentales del sistema de monitoreo	32
2.1.1.	Internet de las cosas (IoT)	32
2.1.2.	Sistema de posicionamiento global (GPS).....	32
2.1.3.	Tecnología LoRa.....	32
2.1.4.	Protocolo MQTT	33
2.1.5.	Plataforma Blynk	33
2.2.	Topología del sistema LoRa (transmisor-receptor).....	33
2.3.	Frecuencias y configuración de parámetros LoRa (SF, BW, CR)	37
2.3.1.	Spreading factor (SF).....	38
2.3.2.	Coding rate (CR).....	41
2.3.3.	Bandwidth (BW)	43
2.4.	Implementación de la modulación CSS (Chirp Spread Spectrum).....	45
2.5.	Procesamiento de las señales en sistemas ESP32 y transmisión de datos hacia la nube mediante WiFi	51
2.5.1.	Transición de LoRa a WiFi	51
2.5.2.	Protocolos aplicados para el envío de datos en sistemas IoT	52
2.5.3.	Conversión analógica-digital en el ESP32 para procesamiento de señales.....	58
CAPÍTULO III: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA		62
3.1.	Características y selección de los componentes electrónicos.....	62
3.1.1.	Módulo LoRa.....	67
3.1.2.	Módulo GPS	69

3.1.3. Microcontrolador ESP32.....	70
3.1.4. Antena para comunicación LoRa	73
3.1.5. Batería recargable y gestión de energía	74
3.1.6. Carcasa del collar inteligente	75
3.2. Integración del sistema	77
3.2.1. Diagrama de conexión eléctrica	77
3.2.2. Asignación de pines y conexiones principales	78
3.2.3. Integración física de los componentes en el collar	79
3.3. Flujo de funcionamiento del sistema	81
3.3.1. Adquisición de datos GPS	83
3.3.2. Procesamiento de datos en el microcontrolador ESP32	84
3.3.3. Transmisión de datos mediante LoRa	84
3.3.4. Recepción y procesamiento en el nodo receptor	85
3.3.5. Envío de datos a la nube mediante WiFi y MQTT	85
3.3.6. Visualización de datos en la aplicación móvil (Blynk).....	86
3.4. Implementación de la plataforma IoT	87
3.4.1. Configuración de la plataforma Blynk	87
3.4.2. Integración con el protocolo MQTT.....	88
3.4.3. Definición de datastreams	88
3.4.4. Configuración de widgets de visualización	89
3.4.5. Visualización de datos en tiempo real	90
3.5. Validación y pruebas del sistema	90
3.5.1. Evaluación de transmisión de datos y tasa de error	91
3.5.2. Evaluación del consumo energético del dispositivo	92
3.5.3. Prueba de alcance de comunicación LoRa	93

3.6. Configuración y calibración del sistema	93
3.6.1. Parámetros LoRa aplicados en el prototipo (SF, BW, CR)	93
3.6.2. Frecuencia de actualización GPS y sincronización	95
3.6.3. Optimización energética mediante modos de suspensión.....	96
3.6.4. Calibración inicial del sistema	98
3.7. Análisis estimativo del desempeño del prototipo	100
3.7.1. Estimación del presupuesto de enlace en el entorno de aplicación	100
3.7.2. Análisis del tiempo de transmisión y tamaño de los datos	106
CAPÍTULO IV: RESULTADOS EXPERIMENTALES Y ANÁLISIS DEL DESEMPEÑO DEL SISTEMA.....	110
4.1. Verificación de la adquisición de coordenadas GPS	110
4.2. Validación de la transmisión de datos mediante LoRa.....	113
4.3. Recepción y visualización de datos en el nodo receptor	115
4.4. Verificación de la interfaz de monitoreo	117
4.4.1. Verificación del envío de alerta de seguridad	120
4.5. Verificación del funcionamiento del modo Deep Sleep.....	120
4.6. Evaluación de la precisión de la geolocalización	121
4.7. Evaluación del alcance del sistema de comunicación LoRa.....	126
4.7.1. Análisis del alcance	127
4.8. Estimación del consumo energético y autonomía del collar inteligente.....	129
4.9. Evaluación experimental de la autonomía con el modo Deep Sleep	131
4.10. Evaluación del desempeño del sistema de comunicación	134
CONCLUSIONES.....	138
RECOMENDACIONES.....	140
BIBLIOGRAFÍA.....	142

ANEXOS.....	149
--------------------	------------

ÍNDICE DE FIGURAS

Figura 1 <i>Estructura de la arquitectura</i>	34
Figura 2 <i>Arquitectura LoRa P2P</i>	34
Figura 3 <i>Nodos de la arquitectura</i>	35
Figura 4 <i>Antena helicoidal o hélice</i>	37
Figura 5 <i>Funcionamiento de RSSI</i>	39
Figura 6 <i>Bits de datos y de redundancia VS velocidad de datos/de bits</i>	42
Figura 7 <i>Ancho de banda (Hz) VS velocidad de datos/de bits</i>	43
Figura 8 <i>Ejemplo de chirp ascendente y descendente</i>	46
Figura 9 <i>Variación en la frecuencia</i>	46
Figura 10 <i>Comunicación MQTT para el envío de datos de ubicación GPS</i>	55
Figura 11 <i>Arquitectura general de un sistema IoT basado en LoRa</i>	57
Figura 12 <i>Comunicación entre ESP32, broker MQTT y Blynk para monitoreo remoto</i>	58
Figura 13 <i>Heltec WiFi LoRa 32 V4</i>	59
Figura 14 <i>Módulo Heltec Wifi LoRa 32 V4</i>	68
Figura 15 <i>SEMTECH SX1262</i>	69
Figura 16 <i>Módulo NEO-6M (GPS)</i>	70
Figura 17 <i>Microcontrolador dual-core Xtensa LX6</i>	71
Figura 18 <i>Pines del módulo Heltec WiFi LoRa 32 (V4)</i>	72
Figura 19 <i>Antena LoRa</i>	73
Figura 20 <i>Batería Lithium 3.7 V (3000 mAh)</i>	74
Figura 21 <i>Conexiones principales</i>	79
Figura 22 <i>Diseño físico del prototipo del collar inteligente para monitoreo de ganado</i>	80
Figura 23 <i>Funcionamiento del sistema</i>	83
Figura 24 <i>Parámetros para la comunicación</i>	87
Figura 25 <i>Datastream configurados</i>	89
Figura 26 <i>Arquitectura de funcionamiento Deep Sleep</i>	97
Figura 27 <i>Esquema del presupuesto de enlace en un sistema inalámbrico</i>	101
Figura 28 <i>Visualización del proceso de adquisición de señal GPS en la pantalla OLED del collar inteligente</i>	110
Figura 29 <i>Confirmación del envío de coordenadas geográficas desde el collar inteligente</i>	113

Figura 30 <i>Monitor serial del nodo receptor durante las pruebas experimentales</i>	116
Figura 31 <i>Visualización de las coordenadas geográficas y variables de monitoreo en la plataforma Blynk durante las pruebas experimentales</i>	118
Figura 32 <i>Recepción de notificaciones automáticas con información de localización mediante Telegram durante las pruebas experimentales</i>	119
Figura 33 <i>Notificación de alerta de seguridad</i>	120
Figura 34 <i>Visualización del modo Deep Sleep en la pantalla OLED del collar inteligente.....</i>	121
Figura 35 <i>Mensaje recibido en Telegram.....</i>	122
Figura 36 <i>Verificación de la ubicación reportada por el sistema mediante Google Maps</i>	123
Figura 37 <i>Medición de la distancia máxima entre el dispositivo transmisor y receptor</i>	125
Figura 38 <i>Implementación del collar inteligente en el bovino durante las pruebas experimentales de alcance</i>	127

ÍNDICE DE TABLAS

Tabla 1 <i>Valores recomendables para el correcto envío de paquetes</i>	40
Tabla 2 <i>La tasa de transferencia de datos maximiza su valor por el BW y minimiza por el aumento de SF</i>	44
Tabla 3 <i>Mejora por al aumento de SF y se deteriora por el aumento de BW</i>	44
Tabla 4 <i>Comparativa entre CSS y FSK</i>	49
Tabla 5 <i>Diferencias clave entre HTTP y MQTT</i>	55
Tabla 6 <i>Comparativa de tipos de módulos LoRa</i>	63
Tabla 7 <i>Comparativa de tipos de baterías</i>	64
Tabla 8 <i>Comparativa de tipos de módulos GPS</i>	65
Tabla 9 <i>Comparativa para la antena helicoidal</i>	66
Tabla 10 <i>Comparativa para la antena monopolo</i>	66
Tabla 11 <i>Comparación de collares GPS para ganado</i>	76
Tabla 12 <i>Asignación de pines</i>	78
Tabla 13 <i>Modos de suspensión</i>	97
Tabla 14 <i>Características principales de SX1262</i>	101
Tabla 15 <i>Valores típicos</i>	103
Tabla 16 <i>Sensibilidad</i>	105
Tabla 17 <i>Tamaño estimado de paquete</i>	107
Tabla 18 <i>Tiempo de transmisión</i>	109
Tabla 19 <i>Tiempo de adquisición de coordenadas GPS</i>	112
Tabla 20 <i>Pruebas de transmisión de datos mediante LoRa</i>	114
Tabla 21 <i>Comparación de las distancias obtenidas mediante el prototipo y Google Maps</i>	124
Tabla 22 <i>Resultados de la evaluación experimental del alcance del enlace de comunicación LoRa</i>	127
Tabla 23 <i>Consumo de corriente de los módulos electrónicos considerados en la estimación teórica</i>	129
Tabla 24 <i>Resultados de la prueba experimental de autonomía con Deep Sleep</i>	132
Tabla 25 <i>Resumen de la autonomía del prototipo con y sin Deep Sleep</i>	134
Tabla 26 <i>Resultados de la transmisión de datos durante la prueba experimental</i>	137

ÍNDICE DE ECUACIONES

Ecuación 1 <i>Velocidad de datos/de bits</i>	43
Ecuación 2 <i>Ancho de banda (BW)</i>	47
Ecuación 3 <i>Duración de tiempo</i>	48
Ecuación 4 <i>La resolución del ADC</i>	60
Ecuación 5 <i>Voltaje</i>	60
Ecuación 6 <i>Tasa de error</i>	91
Ecuación 7 <i>Autonomía de la batería</i>	92
Ecuación 8 <i>Tasa de bits</i>	94
Ecuación 9 <i>Potencia efectiva transmitida (EIRP)</i>	102
Ecuación 10 <i>Pérdidas de propagación en el espacio libre (FSPL)</i>	102
Ecuación 11 <i>Potencia recibida</i>	103
Ecuación 12 <i>Ruido del receptor</i>	104
Ecuación 13 <i>Relación señal-ruido (SNR)</i>	105
Ecuación 14 <i>Margen de enlace</i>	105
Ecuación 15 <i>Datos recibidos correctamente</i>	135
Ecuación 16 <i>Porcentaje de recepción</i>	136
Ecuación 17 <i>Tasa de error</i>	136

ÍNDICE DE ANEXOS

Anexo A <i>Código de dispositivo receptor</i>	149
Anexo B <i>Código de dispositivo transmisor (collar)</i>	152
Anexo C <i>Dispositivo transmisor (collar)</i>	154
Anexo D <i>Dispositivo receptor</i>	155
Anexo E <i>Presupuesto de proyecto</i>	155
Anexo F <i>Autorización del ingreso al centro de apoyo Manglaralto</i>	156
Anexo G <i>Aplicación para sonido de alerta</i>	157

ÍNDICE DE ABREVIATURAS

ADC (Analog to Digital Converter) Convertidor analógico-digital integrado en el ESP32, utilizado para transformar señales analógicas en valores digitales procesables por el microcontrolador.

Banda ISM (Industrial, Scientific and Medical) Banda de frecuencias de uso libre destinada a aplicaciones industriales, científicas y médicas.

Bandwidth (BW) Ancho de banda utilizado en la transmisión LoRa. Determina el rango de frecuencias empleado para transmitir la información e influye en la velocidad de transmisión y la sensibilidad del enlace.

Blynk Plataforma IoT que permite visualizar y supervisar remotamente las variables generadas por el sistema mediante una aplicación móvil o interfaz web.

Coding Rate (CR) Parámetro de configuración de LoRa que incorpora redundancia para la corrección de errores durante la transmisión.

CSS (Chirp Spread Spectrum) Técnica de modulación utilizada por la tecnología LoRa que permite comunicaciones de largo alcance con alta inmunidad al ruido y bajo consumo energético.

Deep Sleep Modo de bajo consumo del ESP32.

ESP32 Microcontrolador de 32 bits, encargado del procesamiento, control y comunicación de los dispositivos del sistema.

Frecuencia de operación Frecuencia utilizada para la transmisión de datos.

Geolocalización Proceso mediante el cual se determina la ubicación geográfica de un objeto utilizando coordenadas de latitud y longitud.

Google Maps Plataforma utilizada para visualizar la ubicación geográfica transmitida.

GPS (Global Positioning System) Sistema de posicionamiento satelital que permite determinar la ubicación de un objeto mediante el cálculo de coordenadas geográficas.

Heltec WiFi LoRa 32 V4 Tarjeta de desarrollo basada en el ESP32-S3 que integra conectividad Wi-Fi, Bluetooth, pantalla OLED y un transceptor LoRa SX1262.

IoT (Internet of Things) Interconexión de dispositivos electrónicos capaces de intercambiar información a través de Internet.

Latitud Coordenada geográfica que expresa la posición de un punto con respecto al ecuador terrestre.

LoRa (Long Range) Tecnología de comunicación inalámbrica de largo alcance y bajo consumo diseñada para aplicaciones IoT.

Longitud Coordenada geográfica que expresa la posición de un punto con respecto al meridiano de Greenwich.

LoRa P2P (Peer-to-Peer) Modo de comunicación punto a punto en el que un nodo transmisor envía información directamente a un nodo receptor sin utilizar un gateway.

Microcontrolador Circuito integrado programable encargado de ejecutar las instrucciones de control del sistema y gestionar el funcionamiento de los diferentes módulos electrónicos.

MQTT (Message Queuing Telemetry Transport) Protocolo de comunicación ligero utilizado para el intercambio de datos entre dispositivos IoT y plataformas de monitoreo.

NEO-6M Módulo receptor GPS empleado para obtener las coordenadas geográficas del collar inteligente.

Nodo receptor (RX) Dispositivo encargado de recibir la información transmitida mediante LoRa, procesarla y enviarla a las plataformas de monitoreo.

Nodo transmisor (TX) Dispositivo instalado en el collar inteligente que adquiere la ubicación mediante GPS y transmite la información al nodo receptor utilizando LoRa.

OLED (Organic Light-Emitting Diode) Pantalla de bajo consumo utilizada para visualizar el estado del sistema.

Potencia de transmisión (Tx Power) Nivel de potencia configurado para transmitir la señal LoRa.

RSSI (Received Signal Strength Indicator) Indicador de la intensidad de la señal recibida por el dispositivo receptor.

Sensibilidad del receptor Nivel mínimo de señal que un receptor puede detectar y procesar correctamente para mantener una comunicación confiable.

SNR (Signal-to-Noise Ratio) Relación entre la potencia de la señal recibida y el nivel de ruido presente en el canal de comunicación.

Spreading Factor (SF) Parámetro de configuración de LoRa que determina la duración del símbolo transmitido.

SX1262 Transceptor de radiofrecuencia utilizado para implementar la comunicación LoRa entre el collar inteligente y el dispositivo receptor.

Telegram Aplicación de mensajería utilizada para enviar notificaciones automáticas.

Tiempo de adquisición (Time to Fix) Intervalo requerido por el receptor GPS para obtener una posición válida después de ser activado.

Time on Air (ToA) Tiempo que un paquete de datos permanece ocupando el canal de comunicación durante una transmisión LoRa.

Wi-Fi Tecnología de comunicación inalámbrica.

Zona de cobertura Área geográfica dentro de la cual el sistema de comunicación mantiene una transmisión estable.

INTRODUCCIÓN

En la actualidad, el sector ganadero enfrenta importantes desafíos que se relacionan con el monitoreo constante y la seguridad del ganado, especialmente en zonas rurales. Como respuesta a esta necesidad, este proyecto propone realizar el diseño de un sistema que monitoree el ganado basándose en tecnología LoRa y geolocalización.

Esta solución se basa en implementar dos dispositivos: uno que funcione como transmisor, el cual estará instalado en un collar utilizado por el animal, y otro dispositivo receptor, que será manipulado por el dueño del ganado. El dispositivo transmisor estará conformado por un microcontrolador (ESP32), que será el encargado del procesamiento de la información, un módulo GPS que se encargará de obtener las coordenadas de la ubicación del animal y, por último, un módulo LoRa responsable de enviar las coordenadas a través de RF. El dispositivo receptor también estará basado en LoRa y ESP32. Su función será recibir la ubicación y retransmitirla vía WiFi a la plataforma de Blynk y a la aplicación de Telegram, lo que permitirá consultar la ubicación del ganado de forma remota, ya sea desde un teléfono móvil o una computadora.

En este proceso, la visualización y administración de los datos se realizará mediante la plataforma Blynk, vinculada a un broker MQTT. Para ello, el ESP32 se conectará con el broker utilizando un código programado que publicará los datos de ubicación en tópicos específicos, facilitando así la actualización de los datos en la aplicación móvil y su posible integración con servicios como Telegram. Este enfoque permite una comunicación más eficiente y flexible, especialmente en entornos con recursos de red limitados.

Para el desarrollo del sistema se considera el uso de la modulación CSS (Chirp Spread Spectrum), la cual proporciona alta sensibilidad, amplio alcance y bajo consumo energético,

características que la hacen adecuada para entornos rurales. Asimismo, el uso del protocolo MQTT resulta idóneo debido a su bajo consumo de ancho de banda, su modelo de comunicación basado en publicación/suscripción y su capacidad para mantener una transmisión de datos confiable incluso en condiciones de conectividad intermitente.

Este proyecto busca reducir las pérdidas económicas ocasionadas por el extravío del ganado bovino, ya sea por errores humanos o por el comportamiento natural de los animales. De acuerdo con los antecedentes analizados, una parte importante del sector ganadero enfrenta esta problemática, por lo que resulta necesario implementar soluciones innovadoras y estratégicas que permitan mejorar el monitoreo y la localización del ganado.

En este contexto, el proyecto busca demostrar que la integración de tecnologías como LoRa, GPS, WiFi y el protocolo MQTT constituye una alternativa eficiente para enfrentar esta necesidad. Bajo esta premisa, el desarrollo de un collar rastreador inteligente se plantea como una solución tecnológica viable para optimizar la ubicación del ganado y disminuir las pérdidas derivadas de su extravío. (Redacción ED., 2025)

CAPÍTULO I: ANTEPROYECTO

1.1. Antecedentes

Pazmiño (2021) en su informe de proyecto “Protocolo LoRa para análisis de medición con GPS y Arduino en la industria ganadera”, desarrollado en Ecuador, realizó un estudio en el cual se pudo observar el robo continuo del ganado y la necesidad de aplicar tecnología para prevenir estos casos. Su solución aplicada se basó en tecnologías como LoRaWAN y Sigfox. Aunque es importante resaltar que este proyecto únicamente fue investigativo, proponiendo una solución tecnológica, es decir, no se llevó a cabo ningún tipo de implementación física. (SANCHEZ, 2021)

Según Paz y Yagual (2023) en el proyecto “Diseño e implementación de una herramienta de rastreo y seguridad de ganado en zonas rurales vía LoRa”, diseñado en Ecuador, se empleó tecnología IoT para observar la ubicación, utilizando placas de desarrollo Lilygo y Orange Pi para llevar a cabo que el sistema registre la ubicación en tiempo real del ganado en espacios reducidos. En este proyecto, su propuesta fue implementar un collar para el ganado; sin embargo, no se evidencia que se lo haya diseñado ni puesto a prueba en el animal. (Paz Leon & Yagual Estacio, 2023)

1.2. Planteamiento del problema

En los sistemas de ganadería extensiva, el ganado permanece en bastas áreas de pastoreo donde el ganado está constantemente en busca de alimento. Por causa de la amplitud del terreno y la ausencia de herramientas dedicadas al monitoreo constante del ganado, para los productores resulta complicado conocer la ubicación del animal en tiempo real, especialmente cuando se alejan del área habitual.

Esta situación obliga a realizar recorridos de exploración, lo cual demanda tiempo, esfuerzo y recursos, ya que la localización del ganado en la mayoría de los casos depende de la búsqueda manual en áreas muy extensas. Además, el tiempo que se demoran en encontrar al ganado puede afectar sus actividades.

Entonces, tomando en cuenta esta problemática, una alternativa para mejorar el monitoreo de ganado sería el uso de tecnologías de geolocalización y comunicación inalámbrica. Integrar un sistema GPS con comunicación LoRa permite obtener y transmitir la ubicación hacia un dispositivo de monitoreo, lo cual facilita la localización y se reduce el tiempo requerido de búsqueda.

1.3. Descripción del proyecto

Con el fin de aumentar la seguridad del ganado, en este proyecto se va a establecer un sistema para el monitoreo vacuno utilizando LoRa y geolocalización. Este sistema puede aportar beneficios al sector ganadero, ya que la localización manual demanda tiempo y esfuerzo, además de riesgos como la pérdida del ganado.

El sistema propuesto consiste en la implementación de un dispositivo integrado en un collar, que posee un módulo GPS para facilitar la ubicación en tiempo real del ganado y un módulo LoRa para transmitir datos a distancias medianamente largas con un consumo energético bajo. La información de la ubicación se va a sincronizar a una plataforma IoT y se podrá visualizar ya sea en PC o en celular; en este caso se usará BLYNK, debido a que es una plataforma con servicio gratuito y será más fácil para el cliente obtenerlo, lo que ayudaría a disminuir los casos de pérdidas de ganado y el tiempo de búsqueda.

1.4. Objetivos

1.4.1. Objetivo general

Implementar un dispositivo que permita el monitoreo en tiempo real del ganado en zonas rurales, integrando elementos como módulos LoRa, posicionamiento GPS y un microcontrolador.

1.4.2. Objetivos específicos

- Establecer un sistema de comunicación basado en LoRa, que permita la transmisión de datos de localización hacia el usuario.
- Prototipar un collar inteligente que utilice tecnología LoRa y GPS, con el fin de enviar los datos de localización del ganado en tiempo real.
- Sincronizar los datos de geolocalización del ganado de modo que se pueda visualizarlos en un dispositivo móvil.
- Realizar pruebas de campo para evaluar la transferencia de datos LoRa, la precisión de la geolocalización y el rendimiento de la batería.

1.5. Justificación

Para prevenir pérdidas de cualquier índole en el sector bovino, es fundamental conocer la ubicación exacta del ganado en las zonas delimitadas. No obstante, las grandes dimensiones del terreno agravan drásticamente el manejo y monitoreo del ganado; usar métodos tradicionales como el arreo bovino o la búsqueda en potreros requiere mucho tiempo y recursos.

Se sabe que la tecnología LoRa facilita comunicaciones a distancias considerables con pocos recursos energéticos, de 5 km en áreas urbanas y hasta 10 km en condiciones de línea de vista, lo que convierte a la geolocalización por módulos LoRa en una alternativa viable para

solucionar el problema de monitoreo de ganado. La integración de esta tecnología con sistemas GPS solventa una localización eficaz del ganado; un receptor GPS correctamente estructurado tiene capacidades de orientación precisas, de hasta 3 metros de forma horizontal y 5 metros verticalmente. Lo que se traduce en un 95% de eficiencia en su ubicación todo el tiempo, lo que permite conocer con exactitud la posición real de los animales en cualquier momento. (LoRa Alliance, 2024)

Este sistema integrado entre módulos LoRa y dispositivo GPS mejora la supervisión del ganado de forma drástica, además de que reduce costos operativos y optimiza distintos recursos. Este hecho representa un factor de modernización en el sector ganadero, permitiendo el uso de herramientas inteligentes para un manejo seguro y eficiente del rebaño.

1.6. Alcance

En el presente proyecto se plantea el desarrollo de un prototipo de collar inteligente orientado al monitoreo de ganado en zonas rurales, empleando tecnología LoRa como medio de comunicación inalámbrica de largo alcance y bajo consumo energético. Para la obtención de la ubicación geográfica se integrará un módulo GPS, mientras que el procesamiento y control del sistema estará a cargo de un microcontrolador ESP32.

La propuesta busca brindar a pequeños y medianos ganaderos una alternativa tecnológica funcional y accesible, que les permita conocer la ubicación de sus animales en tiempo cercano al real, contribuyendo así a la reducción de pérdidas asociadas al extravío de ganado.

En cuanto al funcionamiento del sistema, el módulo GPS obtendrá las coordenadas geográficas del animal y las enviará mediante tecnología LoRa hacia un nodo receptor. Este

receptor deberá disponer de conectividad a internet para permitir sincronizar los datos con la plataforma en la nube Blynk. De manera complementaria, se implementará el envío de la ubicación a través de la aplicación Telegram, facilitando el acceso remoto a la información.

El desarrollo del proyecto contempla la implementación del prototipo y la realización de pruebas de funcionamiento en entornos reales, específicamente en zonas ganaderas rurales, con el fin de evaluar el desempeño del sistema en condiciones operativas. Estas pruebas permitirán validar aspectos como el alcance de comunicación, estabilidad de transmisión y consumo energético del dispositivo.

Por otra parte, es importante delimitar que este trabajo no incluye el desarrollo de una solución a nivel industrial ni la fabricación en serie del dispositivo, sino únicamente la validación de un prototipo funcional. Asimismo, no se contempla el diseño de una plataforma web propia, ya que se hará uso de servicios existentes como Blynk y Telegram para la visualización de datos. De igual manera, no se abordarán técnicas avanzadas de análisis de datos, inteligencia artificial o predicción de comportamiento del ganado, enfocándose exclusivamente en la adquisición y transmisión de la ubicación geográfica.

Finalmente, el proyecto se centra en la validación técnica del sistema de monitoreo, dejando como trabajo futuro posibles mejoras relacionadas con optimización energética, miniaturización del dispositivo y ampliación de funcionalidades.

1.7. Metodología

La presente investigación empleó una metodología de enfoque cuantitativo y experimental. El enfoque cuantitativo permitió medir y analizar variables numéricas relacionadas con el desempeño del sistema de monitoreo, tales como las coordenadas geográficas obtenidas mediante

el módulo GPS (latitud y longitud en grados decimales), la distancia de comunicación entre los nodos transmisor (TX) y receptor (RX) expresada en metros (m), y la autonomía energética del dispositivo determinada a partir de la capacidad de la batería (mAh) y el tiempo de funcionamiento (horas).

Por otra parte, el enfoque experimental consistió en el diseño, construcción e implementación de un prototipo funcional de collar inteligente basado en tecnologías LoRa y GPS. Posteriormente, el dispositivo fue sometido a pruebas en un entorno controlado ubicado en la sede Manglaralto de la Universidad Estatal Península de Santa Elena (UPSE), simulando un escenario real de monitoreo de ganado. Durante los ensayos se evaluó el desempeño del sistema mediante la transmisión y recepción de datos de geolocalización, verificando el alcance de la comunicación, la adquisición de coordenadas y el comportamiento energético del prototipo bajo condiciones reales de operación.

1.8. Resultados esperados

Como resultado final, se espera desarrollar un prototipo de collar capaz de monitorear la ubicación en tiempo real del ganado mediante la integración de un módulo GPS, un microcontrolador ESP32 y comunicación LoRa.

De igual forma, lo que se espera es que el sistema permita transmitir la información de geolocalización hacia plataformas de monitoreo accesibles para el usuario, para tratar de prevenir los extravíos en zonas rurales. Asimismo, se calcula que la implementación del modo Deep Sleep contribuya a minimizar el consumo energético y aumente la autonomía del dispositivo, favoreciendo su funcionamiento durante periodos prolongados de operación.

CAPÍTULO II: MARCO TEÓRICO

Para comprender más a fondo cómo funcionará el proyecto y cómo se ejecutará, es necesario conocer algunos fundamentos técnicos y teóricos relacionados con cada uno de los objetivos propuestos.

2.1. Conceptos fundamentales del sistema de monitoreo

2.1.1. Internet de las cosas (IoT)

El concepto de Internet de las cosas hace referencia a la interconexión de dispositivos físicos mediante redes de comunicación, permitiendo la recopilación, envío y procesamiento de datos en tiempo real. En el contexto del monitoreo de ganado, el IoT facilita la integración de sensores, módulos de comunicación y plataformas en la nube, permitiendo supervisar la ubicación y estado de los animales de manera remota y eficiente.

2.1.2. Sistema de posicionamiento global (GPS)

El GPS es una tecnología de geolocalización que permite determinar la posición exacta de un objeto en la superficie terrestre mediante el uso de satélites. En este proyecto, el módulo GPS es fundamental para obtener las coordenadas del ganado, las cuales son posteriormente procesadas y transmitidas a través del sistema de comunicación implementado.

2.1.3. Tecnología LoRa

LoRa es una tecnología de comunicación inalámbrica de largo alcance y bajo consumo energético, diseñada para aplicaciones IoT. Su principal ventaja radica en la capacidad de transmitir datos a largas distancias con un consumo mínimo de energía, lo que la hace ideal para entornos rurales donde se requiere monitoreo continuo sin acceso constante a fuentes de energía.

2.1.4. Protocolo MQTT

MQTT es un protocolo de mensajería ligero basado en el modelo publicador/suscriptor, diseñado para sistemas con recursos limitados y redes de baja capacidad. En este proyecto, se utiliza para gestionar la transmisión de datos desde el dispositivo receptor hacia la plataforma en la nube, garantizando eficiencia y bajo consumo de ancho de banda.

2.1.5. Plataforma Blynk

Blynk es una plataforma de desarrollo orientada a aplicaciones IoT que permite visualizar y controlar dispositivos de manera remota mediante interfaces gráficas en dispositivos móviles. En el sistema propuesto, Blynk se emplea para mostrar la ubicación del ganado y otros datos relevantes en tiempo real, facilitando la interacción del usuario con el sistema.

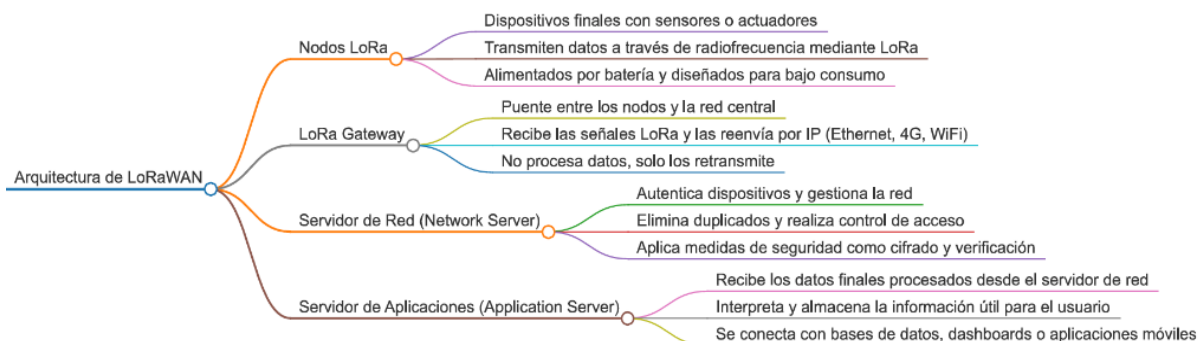
2.2. Topología del sistema LoRa (transmisor-receptor)

Es clave comprender la arquitectura de un sistema LoRa antes de conocer cómo se van comunicando los equipos. Hay que saber que LoRa es funcional en dos arquitecturas:

- LoRaWAN (Long Range Wide Area Network), protocolo que se encuentra dentro de LoRa (tecnología), permite a los dispositivos IoT conectarse a Internet estando lejos.

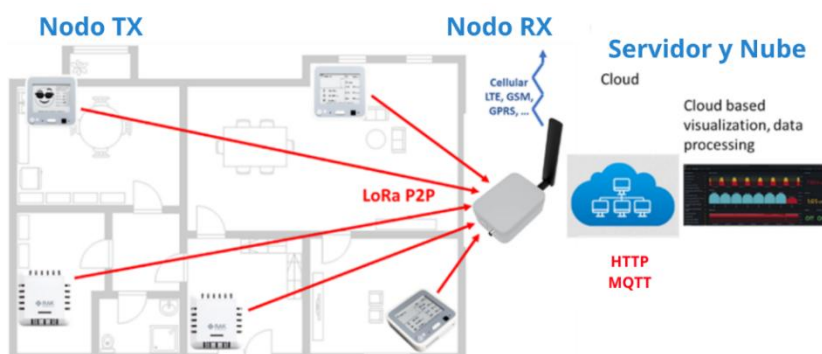
Los nodos normalmente están conectados en una topología estrella, es decir, que cada uno de los nodos se conecta a un punto de conexión central. Los nodos finales normalmente tienen una comunicación bidireccional.

Como se puede apreciar en la Figura 1, la arquitectura de LoRaWAN está compuesta de la siguiente manera:

Figura 1*Estructura de la arquitectura*

Utilizar LoRaWAN para aplicaciones extensas de IoT es perfecto, ya que tiene la capacidad de aceptar millones de mensajes que vienen de miles de dispositivos específicamente en áreas extensas. Para lograr esta eficiencia, se lo hace a través de una arquitectura de red que esté organizada y que incluya distintos tipos de operación para los equipos, teniendo en cuenta las necesidades tanto de energía como de comunicación.

- LoRa punto a punto (P2P) hace referencia a dos equipos LoRa, los cuales, tal como se observa en la Figura 2, se comunican directamente entre ellos sin necesitar un intermediario; gracias a esto, es ideal para aplicaciones localizadas.

Figura 2*Arquitectura LoRa P2P*

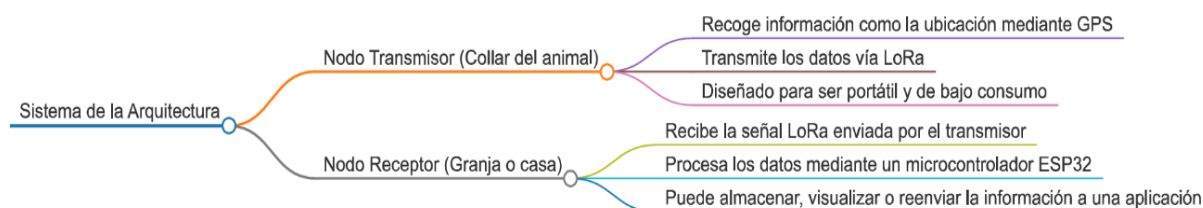
LoRa P2P es un método de comunicación directa que admite implementaciones adaptables y simples, en especial que sean favorables en ocasiones en las que los equipos necesiten alternar información por medio de distancias, ya sean cortas o medias, pero de forma directa. (Giesecke, 2024)

Como se observa en la Figura 2, cuenta con un nodo transmisor que directamente se comunica con un nodo receptor, y no es necesario tener una infraestructura como se necesita para LoRaWAN. LoRa P2P es escogido por su adaptabilidad o flexibilidad; esto permite ajustar la potencia, distancia y velocidad de los datos, para poder ajustarse a las necesidades de distintos proyectos. Es una excelente opción para cuando se necesite usar tecnología LoRa a una escala menor.

Para el proyecto se utilizará una arquitectura LoRa P2P, puesto que es perfecta para utilizarla en zonas rurales en donde no se tiene una buena cobertura y como se observa en la Figura 3, se presentan los dos nodos a utilizar, es decir, los dos puntos.

Figura 3

Nodos de la arquitectura



Esta arquitectura tiene diversas ventajas: disminuye costos y también el consumo energético, y en las áreas remotas permite una comunicación que sea confiable.

En este proyecto, el equipo receptor (nodo receptor) se encargará de recibir los paquetes que sean enviados por el equipo transmisor. De esta forma, en una base de datos se almacenarán los datos para posteriormente visualizarlos y monitorearlos; esto se realizará con la ayuda del ESP32.

Pueden presentarse casos en los que sea complicado o se podría decir que hasta imposible establecer una transmisión de datos con línea de vista, esto debido a la distancia entre el nodo transmisor y el receptor. Tomando en cuenta esto, se origina la necesidad de usar un dispositivo que no tenga dificultad al manejar largas distancias y en el trayecto que pueda enfrentar obstáculos (impiden la transmisión de los datos).

Entonces, el módulo LoRa fue el que se escogió debido a que permite distintas configuraciones variadas para diversas situaciones; por lo tanto, es el ideal para el desarrollo de este proyecto.

Para la realización seguida de la transmisión y recepción de datos, se usará un microcontrolador ESP32 programado, el cual se encargará de llevar a cabo una transmisión continua de la información generada por parte del equipo que adquiere los datos.

Ahora, el equipo transmisor (nodo transmisor) se encarga de capturar y enviar los datos que provienen de instrumentos de los que se adquieren los datos. La instalación se realizó en un collar, el cual tendrá el animal, específicamente en una caja de impresión 3D; está en su interior, tendrá toda la conexión (LoRa, ESP32, GPS y una pequeña antena que permite el correcto transporte de datos).

En el bloque de la transmisión de datos, el equipo transmisor, como ya se dijo anteriormente, cuenta con un microcontrolador ESP32; se lo utiliza para realizar la secuencia de

datos que se transmitirán y también para almacenar y empaquetar los datos que provienen del bloque de unidad sensorial. Este bloque, además, cuenta con módulo LoRa, el cual se encarga de transmitir el paquete a distancias largas que se dirigen al equipo receptor. Ahora el bloque de recepción de datos (equipo receptor) se encarga de recibir el paquete de datos que se envían por el equipo transmisor, para que se procesen y carguen hacia una base de datos de la nube.

Por otra parte, la antena helicoidal, o también conocida como antena hélice, se forma de un alambre conductor que está enrollado como un espiral. Es una evolución del monopolo vertical, en la cual el monopolo ha sido reformado para tomar la forma de un solenoide. Un solenoide es cualquier dispositivo físico capaz de crear una zona de campo magnético uniforme. Al tratarse de un monopolo enrollado en forma espiral, también cuenta con la ventaja de reducir el tamaño de la antena.

Figura 4

Antena helicoidal o hélice



Este tipo de antena, como la que se muestra en la Figura 4, es ampliamente utilizada; también se utilizan otras en UHF y VHF. (Chinchilla Ballesteros & Yepes Castillo)

2.3. Frecuencias y configuración de parámetros LoRa (SF, BW, CR)

Aquí se explicarán las configuraciones que se deben realizar a los módulos LoRa, de forma que la comunicación funcione adecuadamente.

2.3.1. Spreading factor (SF)

Los SF (Factor de propagación/dispersión) hacen referencia al número de bits que se ocupa para que un símbolo se codifique, es decir, muestra la cantidad de símbolos que se envían por cada bit de información. A su vez, la tecnología LoRa necesita máximo seis factores de propagación, los que acogen valores específicamente desde siete hasta el doce. También, una característica importante a resaltar es que con SF menor es más notoria la tasa o velocidad de datos mayor, aunque su alcance o su rango de comunicación es menor, comparándolo a cuando se le asigna SF más alto, en el que se observa un alcance mucho mayor; sin embargo, la velocidad de los datos se disminuye.

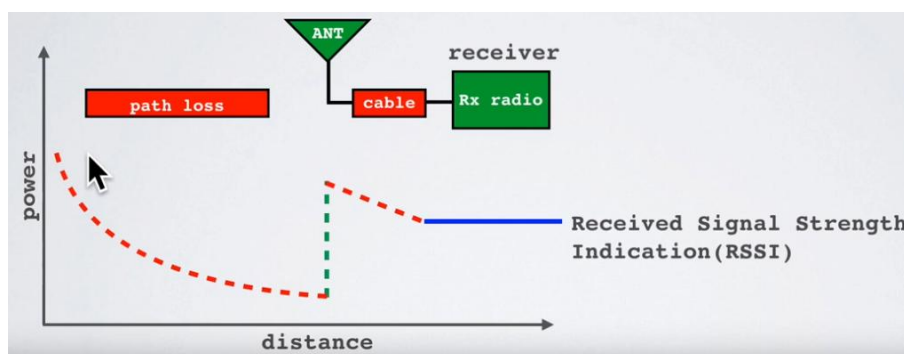
Además, al seleccionar un valor del factor de propagación específico, va a depender de la intensidad de la señal recibida promedio con la que llegan los mensajes a las puertas de enlace, que más le convenga a cada una; así que el valor en este caso del indicador de la señal que se recibe se establece en la distancia que hay entre la puerta de enlace y el dispositivo. Es importante destacar que, en el caso que estos dispositivos finales utilicen factores de propagación iguales y que operen en el mismo canal es posible que presenten interferencias de tal manera que empiecen las colisiones entre señales, lo que puede provocar “ortogonalidad imperfecta”. Esto se refiere a que las señales enviadas con diferentes SF no son del todo independientes, es decir, pueden interferir sutilmente entre sí cuando se transmiten simultáneamente y en la misma frecuencia, en especial si una señal es considerablemente más potente que otra.

Anteriormente se mencionó el RSSI (Received Signal Strength Indicator), lo que significa “Indicador de intensidad de la señal recibida”, y es al mismo tiempo una medida de que tan intensa es la señal de radio que recibe el transmisor. En la Figura 5 se visualiza cómo funciona el RSSI, el

cual, con relación a LoRa, posibilita medir la calidad del enlace que existe entre el transmisor, GPS, y el receptor.

Figura 5

Funcionamiento de RSSI



La unidad del RSSI es el dBm y su valor es negativo; algo que se debe tener presente es que cuando se acerca más a cero, habrá mejor señal. Los valores comunes de LoRa RSSI deben ser mínimo -120 dBm. (B)

Y tomar en cuenta lo siguiente:

- La señal es más fuerte cuando, $\text{RSSI} = -30\text{dBm}$.
- La señal es más débil cuando, $\text{RSSI} = -120\text{dBm}$.

La asignación de SF se puede emplear mediante tres tipos de algoritmo:

Un algoritmo autoorganizado que fue desarrollado por Perakovic y Knapcikova (2021), en el que se adapta correctamente el factor de propagación, teniendo en consideración la comunicación entre los dos nodos LoRa, la cual se recomienda que empiece con el valor mayor de SF (12) para que el envío de paquetes se realice de manera correcta, ya que en el caso de que inicie y se emplee el mínimo factor de dispersión, es decir, $\text{SF} = 7$, va a afectar al momento de transmitir

el mensaje al receptor; entonces, fallaría la adaptación del SF. También se debe mencionar que luego de que los paquetes lleguen al receptor, se obtendrán valores que serán de la relación señal-ruido, esto se puede observar en la Tabla 1.

Tabla 1

Valores recomendables para el correcto envío de paquetes

Spreading factor (SF)	SNR	Sensibilidad del receptor
7	-7.5 dB	-123 dBm
8	-10 dB	-126 dBm
9	-12.5 dB	-129 dBm
10	-15 dB	-132 dBm
11	-17.5 dB	-133 dBm
12	-20 dB	-136 dBm

Perakovic y Knapcikova (2021) mencionan el proceso que desarrolla el algoritmo tanto en el transmisor como en el receptor. En el transmisor se adapta en SF=12, tomándolo como el inicio del desarrollo para la asignación; después de eso se envían diez paquetes de datos de forma continua sin esperar algún reconocimiento.

Luego, el transmisor debe esperar un mensaje ACK (acuse de recibo) se trata de un mensaje corto en el que el equipo receptor menciona que los datos llegaron correctamente o que tuvieron algún inconveniente al momento de recibirlo, esto es para que se pueda realizar la sintonización, en este caso del SF en función de la información del mensaje recibido. En el receptor también se configura el valor inicial de SF=12. Por último, el receptor efectúa la medición de la relación que

existe entre señal-ruido promedio de los paquetes que se recibieron y se especifica un SF basándose en la medición ejecutada.

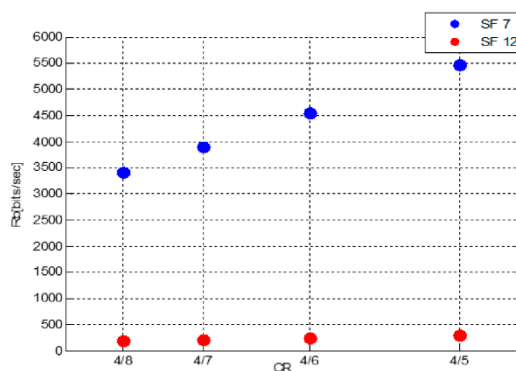
Otro algoritmo por mencionar es el ADR (algoritmo de velocidad de datos adaptativa), que tiene como base encargarse de la tasa de los datos para cada nodo final. Generalmente, se da a conocer que el ADR tiene como propósito asegurarse de que los paquetes se entreguen con una velocidad de datos excepcional y con una red escalable. Este algoritmo ADR se aplica para puntos finales que sean fijos y así mismo una tasa de datos fija. (Lucas Aguilar & Lucas Aguilar, 2023)

2.3.2. Coding rate (CR)

El CR (tasa de código) hace referencia a la asociación con la detección y corrección de fallas en la transmisión de datos, para el código de Hamming que utiliza LoRa (código de bloque). El CR puede adoptar cuatro valores distintos: $4/5$, $4/6$, $4/7$ y $4/8$ siendo estas equivalentes al H (5,4) ... H (8,4). Con los valores $4/5$ y $4/6$ solo se permite la detección de fallas, en cambio para $4/7$ y $4/8$ si es posible detectar y corregir fallas. Las proporciones mencionadas, se refieren al número de bits de datos y también de bits redundantes para detectar fallas. Un ejemplo rápido para que se entienda un poco mejor, en el caso que tengamos CR $4/5$, significa que hay 4 bits de datos y 1 bit de redundancia (error). Para los CR que tienen más bits redundantes (es decir, velocidad baja de codificación) aquí la tasa binaria se reducirá en la práctica, porque para la misma salida binaria se puede tener un número variable de bits de error (redundantes), eso quiere decir un número variable de bits de datos.

Figura 6

Bits de datos y de redundancia VS velocidad de datos/de bits



De acuerdo con lo que se muestra en la Figura 6, en caso de que los CR tengan un número mayor de bits de redundancia (error), se reducirá la velocidad binaria, como se esperaba (125kHz). (Pires & Martins, 2024)

Se puede observar una tasa de bits muy baja a un SF alto, pero hay que tener presente que un CR 4/8 comparándolo con un CR 4/5 también va a provocar una reducción notable de la tasa de bits. Un CR (con más bits redundantes) mayor, una velocidad no tan buena, pero facilita una mayor corrección de fallas.

Entonces, en un caso que se necesite una robusta corrección de errores y que sea de largo alcance, se podría escoger un SF y CR más altos, aunque esto podría disminuir la velocidad de datos. En cambio, si se prefiere una velocidad alta de los datos y los equipos cercanos unos a otros, un SF y un CR menores podrían ser los adecuados.

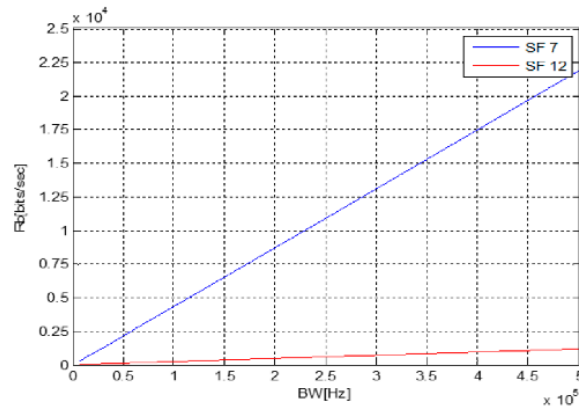
Hay que tener en cuenta que la selección de SF y CR va a depender de las necesidades específicas de la red.

2.3.3. Bandwidth (BW)

Para el BW (ancho de banda) en LoRa se puede utilizar 125 kHz, 250 kHz y 500 kHz (el BW mencionado último no se ocupa en Europa).

Figura 7

Ancho de banda (Hz) VS velocidad de datos/de bits



De acuerdo con la Figura 7 se muestran los resultados para dos Factores de propagación (SF) distintos, utilizando la ecuación 1. (Pires & Martins, 2024)

Ecuación 1

Velocidad de datos/de bits

$$R_b = \frac{BW}{2^{SF}} * SF * \frac{4}{4 + CR_{val}} [bits/s]$$

Donde:

- R_b , velocidad de datos/de bits [bits/s]
- BW , ancho de banda [Hz]
- CR , tasa de código (puede variar entre 1 y 4)

- SF, factor de propagación

Con la ecuación 1 se obtiene la velocidad binaria nominal de la señal LoRa según los valores de BW, CR y SF. También al observar la Figura 7, se nota claramente que al inicio cuando el BW aumenta, al mismo tiempo aumenta la Rb, esto quiere decir que aumenta la tasa de transferencia de datos.

Tabla 2

La tasa de transferencia de datos maximiza su valor por el BW y minimiza por el aumento de SF

Spreading factor (SF)	Ancho de banda (BW)		
	125 kHz	250 kHz	500 kHz
7	9.4 kb/s	18.8 kb/s	37.5 kb/s
12	0.293 kb/s	0.588 kb/s	1.17 kb/s

El BW no solo influye la tasa de transferencia de datos como se muestra en la Tabla 2, también afecta a la sensibilidad del receptor, ya que esta mejora gracias al aumento de SF y se deteriora por el aumento de BW como se muestra en la Tabla 3.

Tabla 3

Mejora por al aumento de SF y se deteriora por el aumento de BW

BW (kHz)	Spreading Factor (SF)					
	7	8	9	10	11	12

125	-123	-126	-129	-132	-133	-136
250	-120	-123	-125	-128	-130	-133
500	-116	-119	-122	-125	-128	-130

2.4. Implementación de la modulación CSS (Chirp Spread Spectrum)

La modulación CSS (Chirp Spread Spectrum/Señal de Espectro Ensanchado) se la conoce como “el corazón” de la modulación de largo alcance (LoRa), dado que se usa para eso y en redes que tienen área amplia. (ieeexplore, 2024)

Esta modulación CSS se la conoce por su resistencia a interferencias de señales y por el desvanecimiento multipath, lo que quiere decir que es una problemática considerable para diversas tecnologías de comunicación inalámbrica. Aquí se dispersa la señal de datos, en el dominio de la frecuencia, esto por medio de la multiplicación de un chirp, para luego modularse a la portadora de radiofrecuencia. Durante el proceso de recepción, la señal de RF se demodula y luego se desarrolla una correlación cruzada con el chirp para recuperar de esta manera la señal de datos. (Moreno, 2023).

Estas técnicas de espectro ensanchado ocupan un ancho de banda que tenga una mayor comunicación en comparación con la banda de señal principal; de esta forma se evitan inconvenientes, ya sea sombreado o desvanecimiento.

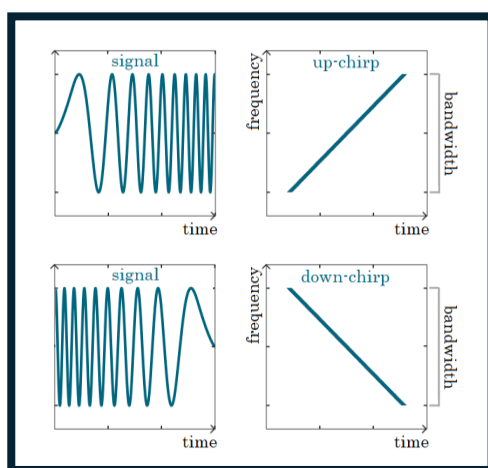
El origen del desvanecimiento puede darse por distintas razones, por ejemplo, cambios en cuanto a la intensidad de la señal, interferencias o reflejos y también el sombreado podría producirse en paredes, colinas o cualquier objeto que obstruya la señal y cause atenuación.

Al dispersar la señal, se reducirá el efecto tanto del desvanecimiento como del sombreado. Incluso en caso de que se afecte la parte del espectro, otras partes van a seguir siendo útiles.

En cuanto a la modulación de chirp (chirrido), la frecuencia de la señal que se transmite no es igual linealmente a través del tiempo. Esto ayuda a que las señales LoRa sean más robustas ante ruidos e interferencias.

Figura 8

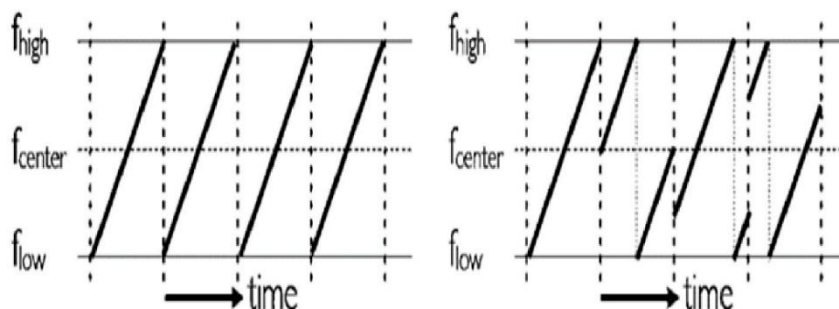
Ejemplo de chirp ascendente y descendente



Como se observa en la Figura 8, la frecuencia crece linealmente en el caso de un chirp ascendente y se reducirá en el caso de que un chirp descienda. (Tulka, 2022)

Figura 9

Variación en la frecuencia



La Figura 9 representa la SF, y se determina la cantidad de bits, estos pueden codificarse por medio de un símbolo. A la izquierda está la señal no modulada y a la derecha se encuentra la señal modulada; existe una variación tanto lineal como creciente durante el tiempo en la frecuencia. (Pires & Martins, 2024)

También, algo importante a saber es que los chirps mantienen un ancho de banda constante; esto en su duración quiere decir que, aun cuando la frecuencia cambie, seguirá siendo el mismo ancho general.

Es muy importante esta propiedad para el correcto funcionamiento de la red LoRa, esta es una de las muchas razones por las que LoRa; es capaz de proveer una comunicación a larga distancia, incluyendo un consumo energético bajo.

Existe un chirp en el extremo de la frecuencia mínima, esto cuando la incrementación llega al punto superior de la frecuencia máxima. La velocidad a la que varía la frecuencia es la pendiente de los chirps, esta se determina mediante el SF. En el caso de que se tenga un SF más bajo, se obtendrá chirps con una pendiente más pronunciada; en caso contrario, se obtendrá una pendiente más lenta.

Ecuación 2

Ancho de banda (BW)

$$BW = f_{max} - f_{min}[Hz]$$

La ecuación 2 es para el ancho de banda de este símbolo:

En la Figura 9 de la señal modulada se observan 4 símbolos representados, cada uno con un tiempo de duración que podemos representar posteriormente por T_s , y se expresa mediante la Ecuación 3:

Ecuación 3

Duración de tiempo

$$T_s = \frac{1}{R_s} = \frac{2^{SF}}{BW} [s]$$

La ecuación 3 nos enseña que el tiempo del símbolo no solo depende del BW, sino también del SF. Esta ecuación nos enseña que, en caso de que la banda sea fija, mientras más SF haya, será mayor el tiempo del símbolo.

También hay que tener en cuenta que es muy importante saber acerca de la modulación FSK (Frequency Shift Keying). Esta técnica consiste en ir cambiando la frecuencia de la señal portadora conforme a los datos binarios que se van a transmitir. Un ejemplo: se tiene un bit 0 que se representa mediante una frecuencia definida (f_0) y un bit 1 con otra frecuencia (f_1) de la misma forma. Este método se utiliza mayormente en sistemas de radios, en módems, redes de sensores o incluso en comunicaciones industriales, esto debido a la relativa sencillez de implementación y a un consumo energético bajo.

La modulación FSK facilita una eficiencia espectral buena y tasas de transmisión que sean moderadas o hasta bajas, siempre tomando en cuenta el ancho de banda que se le haya asignado y por supuesto la distancia que hay entre frecuencias de los bits; esto también tiene sus ventajas. Aunque la principal desventaja es la debilidad ante el ruido y las interferencias externas,

principalmente ante escenarios de largo alcance y en áreas rurales donde a menudo la señal es obstaculizada por diversas razones, entre ellas el terreno, grandes distancias o el clima.

A continuación, en la Tabla 4 se da a conocer la comparativa entre CSS y FSK:

Tabla 4

Comparativa entre CSS y FSK

Característica	FSK	CSS (LoRa)
Tipo de modulación	Digital	Espectro ensanchado mediante chirps
Alcance	Bajo a medio (hasta 2-3 km)	Alto (hasta 8 km, campo abierto)
Sensibilidad del receptor	-120 dBm aprox.	Hasta -137 dBm o incluso más
Resistencia al ruido/interferencias	Moderada	Alta (hasta por debajo del nivel de ruido)
Tasa de transmisión	Alta (tomando en cuenta el canal)	Baja a media (según BW y SF configurado)
Consumo de energía	Bajo	Bajo
Complejidad de implementación	Baja	Moderada (necesita DSP y sincronización)

Aplicaciones	RFID, redes industriales, sistemas de radio	IoT, redes rurales, monitoreo ambiental
--------------	--	--

Para el proyecto a realizar, la modulación CSS es una excelente opción. En las zonas rurales existen diversos problemas al transmitir datos, ya sea por distancias muy largas, áreas con poca o incluso nada de señal, mal tiempo, entre otras interferencias. Tomando en cuenta estos escenarios, la modulación FSK no es la mejor opción, ya que no ofrecería una comunicación segura ni un alcance necesario.

Al contrario de CSS, que permite a los collares inteligentes enviar la posición del animal a la estación base (dispositivo receptor) sin necesitar infraestructura adicional. Es posible que los receptores LoRa reciban señales muy bajas, inclusive cuando la señal se encuentra atenuada. De esta forma se obtiene una codificación precisa gracias a la modulación CSS, además de que tiene un consumo mínimo de energía, es decir, que permite prolongar la autonomía del collar, ya sea por días, incluso por semanas, sin necesidad de recargar.

Algo importante que se debe conocer es que, para el monitoreo del ganado, esta modulación CSS no tiene tanta desventaja, debido a que el sistema no requiere el envío de flujos grandes de información, sino que únicamente transmite pequeñas cantidades de datos, como coordenadas GPS.

En conclusión, la diferencia que existe entre CSS y FSK nos muestra que la modulación mediante chirps es la más idónea para un alcance mayor, también para un consumo energético menor y para más fiabilidad en entornos con ruido o alejados. En cambio, FSK es mejor en

aplicaciones que son de rango menor y velocidad de datos alta, y CSS es mejor para enfrentarse a desafíos; quiere decir que para proyectos sobre IoT en zonas rurales es la mejor opción.

2.5. Procesamiento de las señales en sistemas ESP32 y transmisión de datos hacia la nube mediante WiFi

2.5.1. Transición de LoRa a WiFi

Es importante conocer cómo se realiza la conversión de señal LoRa a señal WiFi. Como ya se ha mencionado en el proyecto a realizar, se utilizarán dos dispositivos: uno que será el collar del ganado (transmisor) y otro que lo tendrá el dueño del ganado (receptor).

El proceso para esta conversión de señales es el siguiente:

1. Transmisión con LoRa, es decir, señal de radio.

Primero, el ESP32 del collar, es decir, del dispositivo transmisor, toma los datos de ubicación que le da el módulo GPS. Luego los datos se envían al módulo LoRa del dispositivo transmisor, el cual ajusta la información en una señal de radio (viaja por el aire a una frecuencia de 915 MHz hasta el receptor), utilizando modulación CSS (Chirp Spread Spectrum).

2. Recepción de LoRa, se refiere a la decodificación.

El dispositivo que tiene el dueño del ganado, es decir, el dispositivo receptor también tiene un microcontrolador; en este caso igual usa ESP32. Este capta la señal LoRa con ayuda de su antena. Luego, este mismo interpreta los datos de la ubicación que llegan.

Ahora sí, después de esto, la señal de radio LoRa se ha convertido en datos digitales, los cuales estarán listos para ser enviados.

3. Envío de datos a la nube mediante WiFi usando MQTT.

Una vez procesados los datos, el ESP32 del nodo receptor utiliza su módulo WiFi integrado para conectarse a una red local. A diferencia del enfoque basado en HTTP, en este caso se emplea el protocolo MQTT, el cual funciona bajo un modelo de publicación/suscripción.

El dispositivo actúa como un cliente MQTT que publica los datos de ubicación en un tópico específico dentro de un broker MQTT. La plataforma Blynk, previamente configurada, se suscribe a dicho tópico para recibir la información en tiempo real. Este mecanismo permite una comunicación más eficiente, con menor consumo de ancho de banda y mejor desempeño en redes con recursos limitados.

Finalmente, los datos pueden ser visualizados desde dispositivos como teléfonos móviles o computadoras, facilitando el monitoreo continuo del ganado. Además, este enfoque permite evaluar la estabilidad del sistema bajo diferentes condiciones de operación, especialmente en entornos rurales donde la conectividad puede ser variable.

2.5.2. Protocolos aplicados para el envío de datos en sistemas IoT

2.5.2.1. Protocolo HTTP

El Protocolo HTTP (HyperText Transfer Protocol) es un estándar de comunicación ampliamente utilizado en redes de datos, especialmente en Internet, cuya función principal es permitir el intercambio de información entre dispositivos. Su funcionamiento se basa en el modelo cliente-servidor, donde un cliente (como un microcontrolador, aplicación o navegador) envía solicitudes a un servidor, y este responde con los datos requeridos.

HTTP trabaja mediante mensajes estructurados que incluyen métodos de solicitud, siendo los más comunes GET (para obtener información) y POST (para enviar datos). Este esquema

facilita la transmisión de datos de forma organizada, permitiendo que diferentes dispositivos puedan comunicarse entre sí sin importar su arquitectura o sistema operativo.

En aplicaciones de monitoreo, HTTP resulta útil para enviar información hacia plataformas remotas, porque permite transferir datos como coordenadas, estados o registros en tiempo real o en intervalos definidos. Además, al ser un protocolo ampliamente soportado, se integra fácilmente con diversos servicios web y sistemas de almacenamiento.

Por otro lado, HTTP es un protocolo sin estado, lo que implica que cada solicitud se procesa de manera independiente, sin depender de interacciones previas. Esta característica simplifica su implementación y reduce la complejidad en la comunicación entre dispositivos, lo que resulta adecuado para sistemas de monitoreo donde los datos se envían de forma periódica o bajo demanda.

2.5.2.2. Protocolo MQTT.

El protocolo MQTT (Message Queuing Telemetry Transport) es un mecanismo de comunicación ligero orientado al envío eficiente de datos entre dispositivos, especialmente en entornos donde la conectividad a internet es limitada, inestable o de baja calidad. Debido a estas características, es ampliamente utilizado en aplicaciones de Internet de las Cosas (IoT), tales como sensores remotos, dispositivos embebidos y sistemas de monitoreo en tiempo real.

A diferencia de otros protocolos de comunicación más robustos, MQTT opera bajo un modelo de tipo publicador/suscriptor (publish/subscribe). En este esquema, los dispositivos no establecen una conexión directa entre sí, sino que interactúan a través de un intermediario denominado broker. De esta manera, un dispositivo emisor (publicador) envía la información al

broker, y los dispositivos interesados (suscriptores) reciben los datos según los temas a los que estén suscritos, lo que simplifica la arquitectura del sistema y mejora la escalabilidad.

Entre sus principales ventajas destaca el bajo consumo de ancho de banda y energía, lo que lo convierte en una alternativa idónea para dispositivos con recursos limitados, como microcontroladores o sistemas alimentados por baterías, como un collar inteligente para monitoreo de ganado. Adicionalmente, su estructura liviana permite una transmisión de datos rápida y eficiente, incluso en condiciones de cobertura reducida.

MQTT también se caracteriza por el uso de mensajes de tamaño reducido, lo cual minimiza el consumo de recursos de red y procesamiento. Esta característica resulta especialmente relevante en aplicaciones rurales o de largo alcance, donde puede integrarse con tecnologías como LoRa para el envío de información hacia servidores o plataformas de supervisión. (OASIS, s.f.)

En cuanto a la confiabilidad, el protocolo incorpora diferentes niveles de calidad de servicio (QoS), que permiten ajustar el grado de garantía en la entrega de los mensajes: desde envíos sin confirmación (QoS 0), pasando por entrega asegurada al menos una vez (QoS 1), hasta la entrega única garantizada (QoS 2), aunque con mayor sobrecarga.

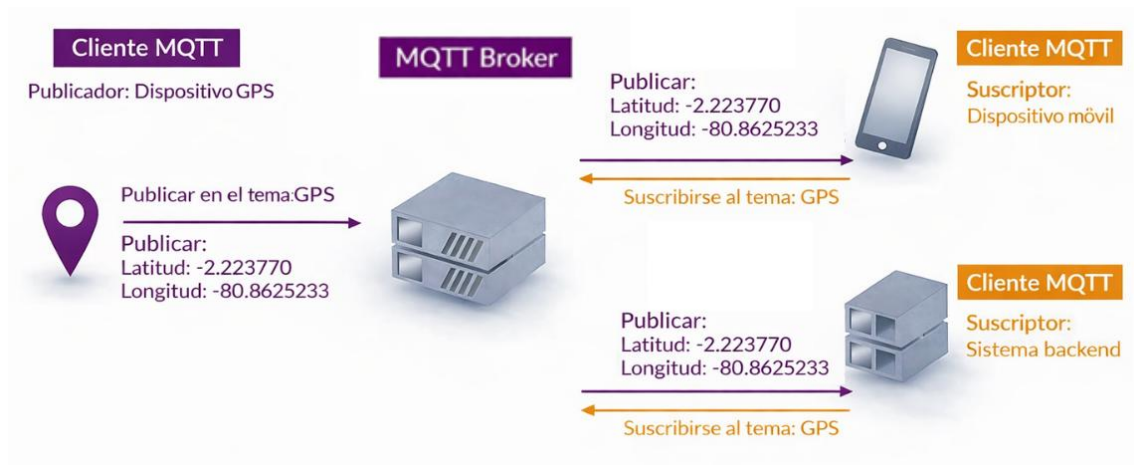
Finalmente, en términos de seguridad, MQTT puede implementar mecanismos de autenticación y cifrado mediante TLS/SSL, lo que permite proteger la integridad y confidencialidad de los datos transmitidos, aspecto fundamental en aplicaciones que manejan información sensible o crítica. (HIVEMQ, s.f.)

La Figura 10 muestra una arquitectura general de un sistema IoT, en la cual los dispositivos sensores envían información hacia un gateway que actúa como intermediario entre el entorno físico y la nube. En este tipo de esquemas, protocolos como MQTT son ampliamente utilizados debido

a su capacidad para gestionar la comunicación eficiente entre múltiples dispositivos y aplicaciones mediante un modelo de publicación y suscripción.

Figura 10

Comunicación MQTT para el envío de datos de ubicación GPS



A continuación, se presenta la Tabla 5, comparando los protocolos HTTP y MQTT, con el fin de identificar sus principales características y determinar su idoneidad dentro de aplicaciones IoT.

Tabla 5

Diferencias clave entre HTTP y MQTT

Diferencias Clave		HTTP		MQTT	
Modelo de comunicación	de Solicitud-respuesta (pregunta, otro responde)	(un dispositivo responde)		Publicar-suscribir (un dispositivo comparte, muchos reciben)	
Dirección de datos	Unidireccional			Multidireccional	
	(el cliente solicita → el servidor responde)			(un editor → muchos suscriptores)	

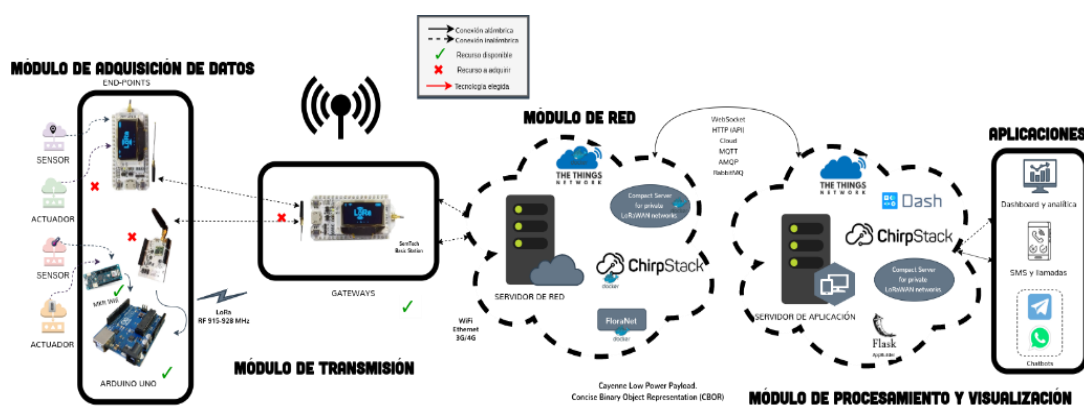
Conciencia Estado	del	El servidor no conserva el estado del cliente (trata cada solicitud como nueva)	El bróker mantiene el estado (rastrea suscripciones y conexiones de clientes)
Método conexión	de	Sin estado (nueva conexión para cada solicitud)	Conexión persistente (mantiene la línea abierta)
Tamaño mensaje	del	Encabezados de texto voluminosos (legibles para humanos, pero de gran tamaño)	Formato binario compacto (datos adicionales mínimos)
Adaptabilidad de la red		Optimizado para redes estables y de gran ancho de banda	Optimizado para redes inestables y de bajo ancho de banda
Rendimiento tiempo real	en	Depende del sondeo (las solicitudes repetidas provocan retrasos)	Envía actualizaciones instantáneas a los suscriptores
Escalabilidad		Tiene problemas con muchos dispositivos (cada uno necesita una conexión directa al servidor)	Maneja millones de dispositivos sin problemas a través de un sistema intermediario
Casos de uso		Navegación web, transacciones en línea, descargas de archivos	Automatización de hogar inteligente, monitoreo de dispositivos IoT, transmisión remota de datos de sensores
Mecanismo confiabilidad	de	Se basa principalmente en reintentos de solicitud para la transmisión de datos	Admite diferentes niveles de QoS, garantiza la entrega de mensajes, más estable en redes deficientes

Una vez descritos los protocolos HTTP y MQTT, es importante comprender cómo estos se integran dentro de una arquitectura completa de comunicación en sistemas IoT. En este tipo de soluciones, ambos protocolos pueden coexistir y ser utilizados según los requerimientos de cada etapa del sistema. (COMESTAR, 2025)

A continuación, se presenta un esquema general de una arquitectura basada en LoRaWAN, donde se observa en la Figura 11 la interacción entre los distintos módulos y el uso de dichos protocolos en el envío y procesamiento de la información.

Figura 11

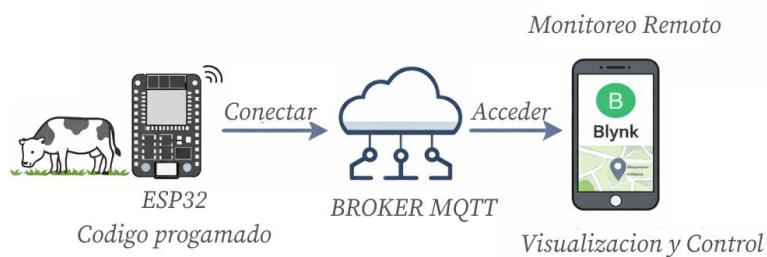
Arquitectura general de un sistema IoT basado en LoRa



En el desarrollo del sistema se optará por utilizar la plataforma Blynk como interfaz de visualización y control, integrándola mediante el protocolo MQTT. Para ello, el ESP32 se conectará a un broker MQTT a través de un código programado, el cual permitirá la publicación de los datos de ubicación en tópicos específicos. De esta manera, Blynk podrá acceder a la información en tiempo real, facilitando la supervisión remota del ganado de forma eficiente. (Rivadeneira, 2019)

Figura 12

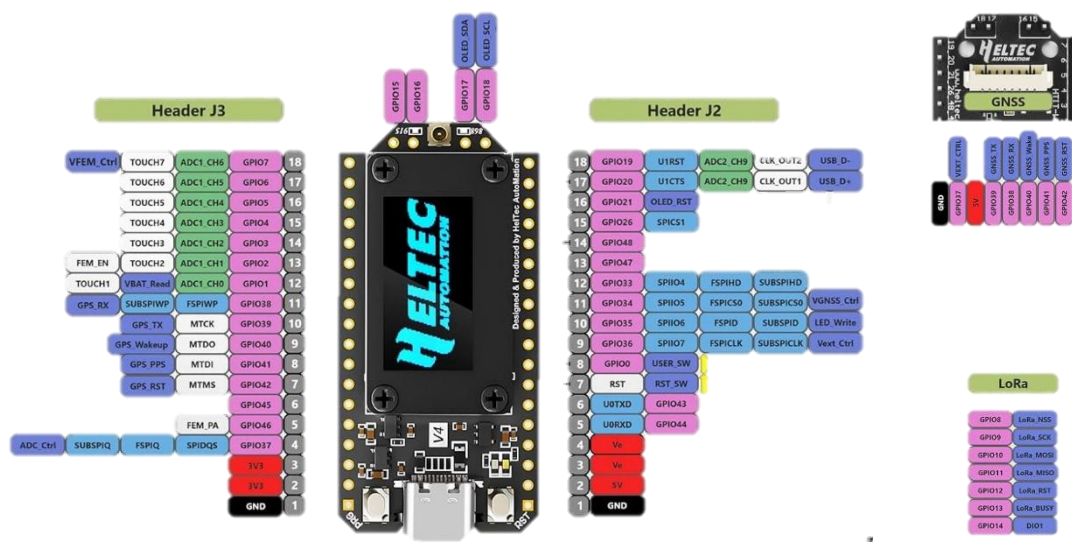
Comunicación entre ESP32, broker MQTT y Blynk para monitoreo remoto



2.5.3. Conversión analógica-digital en el ESP32 para procesamiento de señales

El microcontrolador ESP32 integrado en el módulo Heltec WiFi LoRa 32 V4 dispone de convertidores analógico-digitales (ADC) de 12 bits, lo que permite transformar señales analógicas en valores digitales dentro de un rango de 0 a 4095 niveles. Esta capacidad resulta fundamental para la lectura de sensores que entregan información en forma de voltaje.

El ESP32 cuenta con dos módulos ADC denominados ADC1 y ADC2. Sin embargo, en la práctica, no todos los pines asociados a estos convertidores se encuentran disponibles en la placa Heltec, ya que algunos no están expuestos físicamente.

Figura 13*Heltec WiFi LoRa 32 V4*

En el caso de ADC1, se dispone de varios pines accesibles y estables para la adquisición de datos analógicos, entre los que destacan GPIO32, GPIO33, GPIO34, GPIO35, GPIO36 (VP) y GPIO39 (VN). Estos pines son los más recomendados para el uso con sensores, debido a su funcionamiento confiable incluso cuando el sistema utiliza comunicación inalámbrica.

Por otro lado, el ADC2 también permite la conversión de señales analógicas mediante pines como GPIO0, GPIO2, GPIO4, GPIO12, GPIO13, GPIO14, GPIO15, GPIO25, GPIO26 y GPIO27. No obstante, su uso presenta ciertas limitaciones, puesto que comparte recursos internos con el módulo WiFi del ESP32, lo que puede ocasionar interferencias o lecturas inestables cuando se emplean ambas funciones simultáneamente. (HELTEC AUTOMATION, s.f.)

En este contexto, para garantizar una medición precisa y continua dentro del sistema propuesto, se recomienda priorizar el uso de los canales correspondientes al ADC1. (Espressif Systems (Shanghai), s.f.)

El proceso de conversión analógico-digital consta de tres pasos importantes; son los siguientes

- 1) **Muestreo:** En este paso, el convertidor analógico-digital toma de la señal analógica ciertas muestras a intervalos regulares.
- 2) **Cuantificación:** A cada una de estas muestras tomadas anteriormente, se le asigna el nivel digital que esté más cercano; claramente, debe estar dentro del rango de resolución del convertidor analógico-digital.
- 3) **Codificación:** El nivel anterior, es decir, el cuantitativo, se lo representa por medio de un número binario de 12 bits, como se lo mencionó anteriormente.

Algunas fórmulas importantes para llevar a cabo la correcta conversión analógico-digital son:

Ecuación 4

La resolución del ADC

$$Resolucion = \frac{V_{ref}}{2^n - 1}$$

La ecuación 4 muestra la resolución del ADC, que este detecta el mínimo cambio que existe en el voltaje.

Ecuación 5

Voltaje

$$V_{out} = \frac{D_{out} * V_{ref}}{2^n - 1}$$

Para calcular el voltaje que corresponde a un valor digital que el convertidor analógico-digital lo pueda leer, se utiliza la ecuación 5. (impl Rust for ESP32, s.f.)

CAPÍTULO III: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

En el contexto de la ganadería en zonas rurales, el control y seguimiento del ganado representa un desafío constante debido a la extensión del terreno y el riesgo de pérdida de los animales. Estas limitaciones dificultan la supervisión continua y afectan directamente la productividad y seguridad de los ganaderos.

Frente a esta problemática, el presente proyecto propone el desarrollo de un sistema de monitoreo basado en tecnología LoRa y geolocalización GPS, el cual permite conocer la ubicación del ganado en tiempo real, incluso en áreas con cobertura limitada de redes tradicionales. Esta solución tecnológica facilita la toma de decisiones, optimiza el manejo del ganado y reduce el tiempo y esfuerzo requeridos para su localización.

De esta manera, el proyecto no solo aporta una alternativa eficiente y de bajo consumo energético, sino que también contribuye a mejorar las condiciones de trabajo del ganadero, brindando mayor control, seguridad y confiabilidad en la gestión de sus recursos.

3.1. Características y selección de los componentes electrónicos

Antes de realizar la selección de módulos y componentes electrónicos, se realizará una comparación de módulos en donde se analizará cuál es el más adecuado para el proyecto a realizar.

- **Módulos LoRa.**

Tabla 6*Comparativa de tipos de módulos LoRa*

Módulo	Chip Base	Bandas ISM compatibles	Versatilidad / Integración	Alcance (entorno rural)	Consumo energético
Heltec Wifi LoRa 32 V4	SX1262	137 - 1020 MHz (433 / 868 / 915 MHz más usados)	Muy buena (ESP32 integrado, LoRa+WiFi)	Hasta 10 km	Bajo, con modos sleep
RAK4630	SX1276	868 / 915 MHz	Muy alta (BLE + LoRa)	Hasta 8 km	Muy bajo (chip Nordic nRF)
RFM95W	SX1276	433 / 868 / 915 MHz	Buena (requiere MCU externo)	5 - 10 km	Bajo
Dragino LoRa Hat	SX1276/ SX1278	433 / 868 MHz	Media (para Raspberry Pi)	3 - 8 km	Depende de Pi

LoRa	SX1276	850 - 930	Baja	5 - 8 km	Bajo
E32-		MHz	(requiere		
915T20D			configuración		
(EByte)			AT)		

- **Baterías.**

Tabla 7

Comparativa de tipos de baterías

Tipo de batería	Voltaje típico	Capacidad (mAh)	Peso aprox. (g)	Tamaño aprox. (mm)	Ciclos de carga
Lithium 3.7 V (3000 mAh)	3.7 V	2000-3000 mAh	35-60 g	70x40x6	300-500 ciclos
Li-Ion 18650 (celdas cilíndricas)	3.7 V	2200-3400 mAh	45-70 g	18 mm x 65 mm (cilíndrica)	500-1000 ciclos
Batería NiMH AA (4 en serie)	4.8-6 V	1000-2000 mAh	100-120 g	Depende del portapilas	500-1000 ciclos

Batería	3.2 V	1000-2000	40-70 g	Variable	1000+
LiFePO4		mAh		(cilíndrica)	ciclos
3.2 V					

- **Módulo GPS.**

Tabla 8

Comparativa de tipos de módulos GPS

Módulo	Precisión	Consumo	Tiempo de	Voltaje de	Comunicación
GPS	(horizontal)	promedio	fijación	operación	
			(TTFF)		
NEO-6M	±2.5 m	~35 mA	30-35 s (frío), 1 s (cálido)	3.3-5 V	UART (Serial)
NEO- M8N	±1.5 m	~29 mA	26 s (frío), 1 s (cálido)	3.3 V (ideal)	UART (Serial), I2C
L76K	±2 m	~25-30 mA	25 s (frío), <2 s (cálido)	3.3 V	UART (Serial), I2C

- **Antena Helicoidal.**

Tabla 9*Comparativa para la antena helicoidal*

Modelo de Antena	Frecuencia (MHz)	Ganancia (dBi)	Tamaño aproximado	Polarización
Heltec Original 915 MHz Helical Antenna	902-928	~1.2	Compacta (resorte ~3 cm)	Omnidireccional / Lineal
433/868/915 MHz Helical Antenna genérica	868-915	~1.5	2.5-4 cm de largo	Lineal
LoRaTech Mini Helical	915	~1.5	4 cm x 1 cm	Lineal (vertical)

- **Antena Monopolo**

Tabla 10*Comparativa para la antena monopolo*

Modelo	Frecuencia	Ganancia	Tamaño (aprox.)	Polarización
ANT-B14-CW-QW- SMA (Linx)	902 - 928 MHz	1.2 - 2.2 dBi	48 mm	Omnidireccional / Vertical

TTGO Antena	900 - 930	~1.8 dBi	47 mm	Omnidireccional
Monopolo SMA	MHz			
Dragino 915 MHz	900 - 930	~ 2.0 dBi	50 mm	Lineal vertical
Antenna	MHz			

3.1.1. Módulo LoRa

La figura 14 presenta el módulo Heltec Wifi LoRa 32 V4, el cual fue seleccionado debido a su integración con el microcontrolador ESP32, lo que facilita el desarrollo del firmware mediante entornos accesibles como Arduino IDE. Asimismo, este dispositivo permite establecer comunicación LoRa punto a punto (P2P), eliminando la necesidad de depender de infraestructuras LoRaWAN, lo cual resulta especialmente adecuado para aplicaciones en zonas rurales donde la cobertura de red es limitada.

Otro aspecto relevante es su capacidad de alcanzar distancias de hasta 10 km en condiciones de línea de vista, característica fundamental para el monitoreo de ganado en extensas áreas.

Finalmente, el módulo incorpora modulación CSS a nivel de hardware, lo que contribuye a garantizar una transmisión de datos más confiable y eficiente en entornos con posibles interferencias.

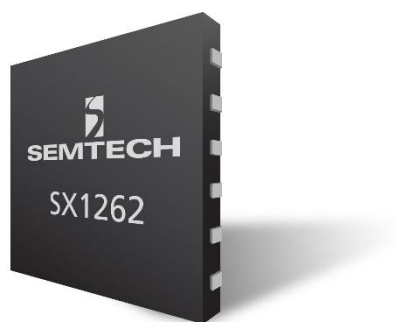
Figura 14

Módulo Heltec Wifi LoRa 32 V4



El módulo LoRa integrado en la tarjeta Heltec WiFi LoRa 32 V4 está basado en el chip Semtech SX1262, desarrollado por la empresa Semtech como una evolución de generaciones anteriores de transceptores LoRa. Este dispositivo emplea modulación por espectro ensanchado tipo CSS (Chirp Spread Spectrum), lo que le permite alcanzar comunicaciones de largo alcance con alta resistencia a interferencias, característica fundamental para aplicaciones en entornos rurales. (HELTEC AUTOMATION, s.f.)

El chip LoRa SX1262 que se observa en la Figura 15 opera en bandas ISM sub-GHz, principalmente en frecuencias como 868 MHz y 915 MHz, siendo esta última la más utilizada en América Latina. En condiciones favorables, especialmente en zonas abiertas y con línea de vista, puede alcanzar distancias de varios kilómetros, lo que lo convierte en una solución adecuada para sistemas de monitoreo de ganado. (Manuals+, s.f.)

Figura 15*SEMTECH SX1262*

Entre las principales ventajas del módulo se destacan:

- Permite establecer enlaces de comunicación de largo alcance, que pueden superar los 5 km en entornos rurales con condiciones óptimas y antenas adecuadas.
- Presenta un consumo energético optimizado, con valores aproximados de hasta 120 mA durante la transmisión y niveles considerablemente bajos en modos de reposo, lo que favorece su implementación en dispositivos alimentados por batería.
- Dispone de interfaz de comunicación SPI, facilitando su integración con el microcontrolador ESP32 incorporado en la misma placa, lo que simplifica el diseño del sistema. (SEMTECH, s.f.)

3.1.2. Módulo GPS

La figura 16 presenta el módulo GPS NEO-6M de u-blox, el cual fue seleccionado debido a su diseño compacto y a su bajo consumo energético, características fundamentales para su integración en dispositivos portátiles como el collar propuesto.

Este módulo permite obtener coordenadas geográficas con una precisión aproximada de 2.5 metros (CEP 50%), lo que resulta adecuado para aplicaciones de monitoreo de ganado en campo abierto.

Figura 16

Módulo NEO-6M (GPS)



Adicionalmente, presenta un tiempo de readquisición cercano a 1 segundo, lo que favorece una rápida recuperación de la señal en caso de interrupciones temporales. En cuanto al consumo energético, durante la operación de navegación mantiene un valor típico de alrededor de 35 mA, contribuyendo así a optimizar la autonomía del sistema. (ardumotica.cl, s.f.)

En conjunto, estas características lo convierten en una alternativa eficiente y confiable para sistemas de geolocalización en entornos rurales. (ublox, s.f.)

3.1.3. Microcontrolador ESP32

El módulo Heltec ESP32 LoRa que se seleccionó anteriormente para el desarrollo de este proyecto incluye un transceptor LoRa, una pantalla OLED y un microcontrolador ESP32. El ESP32 que incluye el módulo Heltec se muestra en la Figura 17; es un microcontrolador dual-core Xtensa LX6 de 32 bits que trabaja a una frecuencia de 240 MHz, con una RAM interna de 520 KB y una memoria Flash de 4MB lo cual es suficiente para manejar correctamente la lógica de

adquisición de los datos GPS, transmisión del LoRa y la conexión a la nube a través de WiFi.
(Heltec Automation., s.f.)

Figura 17

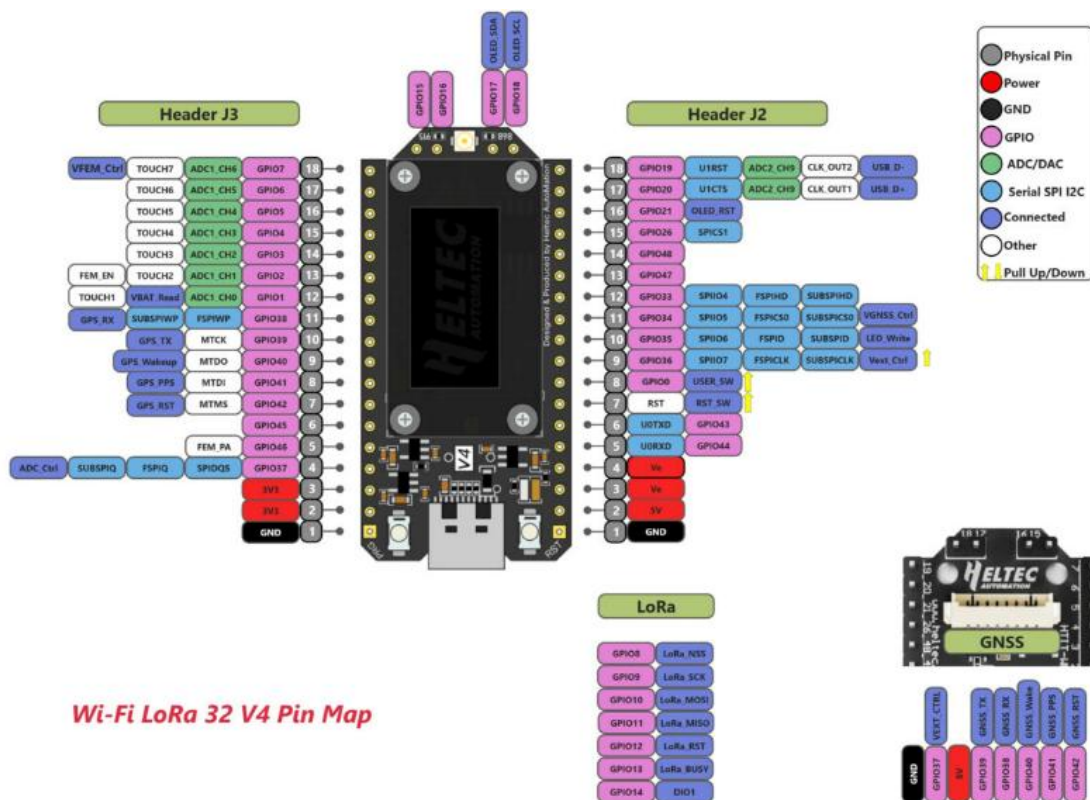
Microcontrolador dual-core Xtensa LX6



El microcontrolador dual-core Xtensa LX6 permite una conectividad WiFi para protocolos 802.11 b/g/n; además de Bluetooth clásico y BLE, cuenta con múltiples interfaces de comunicación; entre ellas destacan: UART, I2C, DAC y ADC de 12 bits, estas características brindan versatilidad al microcontrolador para trabajar con distintos sensores y periféricos. También posee un modo de bajo consumo que extiende en gran medida la capacidad del sistema energético, siendo ideal para su integración en el collar inteligente propuesto. (Thanksbuyer, s.f.)

Figura 18

Pines del módulo Heltec WiFi LoRa 32 (V4)



La ventaja principal del Heltec ESP32 LoRa es su integración con distintas tecnologías, ya que, al poseer un microcontrolador y un transceptor LoRa dentro de un mismo hardware, facilita gradualmente el diseño de conexiones y el número de componentes, además de mejorar aspectos como el consumo energético del dispositivo y la versatilidad. Este dispositivo es programable a través de diferentes entornos, como: Arduino IDE, ESP-IDF, PlatformIO, entre otros. (Heltec Automation., s.f.)

La justificación de este proyecto se basa en parámetros técnicos, como el rendimiento, la conectividad y la eficiencia energética, porque son aspectos clave para poder desarrollar el sistema de monitoreo de ganado en zonas rurales. Variables como la cobertura entre los equipos, la

funcionalidad de operar sin una red intermedia o el tiempo prolongado de duración del sistema son conceptos necesarios para el collar localizador, que son cubiertos gracias a los parámetros mencionados anteriormente (Heltec Automation., s.f.).

3.1.4. Antena para comunicación LoRa

Sus características principales incluyen un diseño compacto y un patrón de radiación omnidireccional, lo que permite una cobertura uniforme en diferentes direcciones. Presenta un buen desempeño en condiciones de línea de vista, siendo adecuada para aplicaciones donde existe visibilidad directa entre el transmisor y el receptor.

La antena ilustrada en la figura 19 está diseñada para operar en la banda ISM de 915 MHz, por lo que resulta adecuada para configuraciones LoRa punto a punto (P2P), como la implementada en el diseño del collar inteligente. Esta banda es de uso común en Ecuador para este tipo de aplicaciones. Además, es compatible con el módulo Heltec WiFi LoRa 32 V4, presentando una impedancia característica de 50 ohmios que permite un adecuado acoplamiento con el sistema.

Figura 19

Antena LoRa



La ganancia de la antena es aproximadamente de 1.2 dBi, valor que resulta suficiente para aplicaciones de corto y mediano alcance en entornos abiertos. No obstante, el rendimiento final dependerá de factores como la topografía del terreno, la presencia de obstáculos y las condiciones ambientales. (Electronicos Caldas, s.f.)

En cuanto a su instalación, la antena puede conectarse al módulo mediante un conector U.FL o interfaz RF equivalente, facilitando su integración en el sistema. (Heltec Automation., s.f.).

3.1.5. Batería recargable y gestión de energía

Como ya se ha descrito anteriormente, el módulo Heltec ESP32 LoRa cuenta con un modo de bajo consumo, pensado precisamente para trabajar constantemente sin necesidad de recurrir a fuentes externas o interrupciones, por lo que suele trabajar con baterías de 3.7 V de Lithium.

Figura 20

Batería Lithium 3.7 V (3000 mAh)



Algo importante a resaltar es que su funcionalidad puede extenderse según la frecuencia de envío de paquetes; un caso práctico sería mantener el modo sleep transmitiendo información en periodos de cada 5 o 10 minutos, que es adecuado para la aplicación de monitoreo de ganado. (UNIT Electronics, s.f.)

Ventajas:

- Tamaño compacto y peso ligero (entre 35 y 60g)
- Integración sencilla a distintos proyectos
- Gran durabilidad dependiendo de las necesidades del sistema
- Versatilidad en métodos de carga, como puerto USB o circuitos de carga.

- Circuito interno de seguridad, equilibrio entre capacidad y peso.

3.1.6. Carcasa del collar inteligente

La carcasa es un elemento muy importante para el prototipo a realizar. Esto debido a que protege correctamente los circuitos electrónicos ante condiciones ambientales y debe brindar comodidad al animal.

Los materiales más idóneos para la fabricación del collar son polímeros que sean resistentes, como por ejemplo ABS (Acrilonitrilo Butadieno Estireno), PC (Policarbonato) o también TPU (Poliuretano Termoplástico). En la figura 21 podemos observar un ejemplo de cómo debe quedar. (Tippannavar, 2023)

Algunas características de estos materiales son:

- ABS, tiene una buena rigidez, es resistente a impactos y cuenta con estabilidad térmica.
- PC, cuenta con una gran resistencia mecánica y una transmisión RF ideal cuando la carcasa cubre por completo las antenas.
- TPU, es más flexible y preciso en el caso de que se necesite una elasticidad específica para absorber vibraciones presentadas.

Ahora, en cuanto al collar, el diseño que se vaya a realizar debe ser amigable para no lastimar ni incomodar el cuello del animal; de esta manera, permitir que se ajuste cómodamente y sin riesgos de causar cualquier tipo de heridas. Lo que es recomendable utilizar es una correa ajustable de nailon o también puede ser de cuero, incluyendo cierres de seguridad. (Zhou, 2025)

Tabla 11*Comparación de collares GPS para ganado*

Modelo	Componentes incluidos	Peso total aproximado	Autonomía	Observaciones
Pellego (Followit)	GPS + módulo GSM/GPRS + batería + carcasa IP-67	175 g	6-18 meses según frecuencia	Muy liviano, resistente y cómodo para el animal.
Tellus Livestock (Followit)	GPS + GSM o satélite + batería + carcasa robusta	700 g	Hasta 2 años en modo básico	Robusto y de gran autonomía, pero pesado para animales
RUMI GPS Cow Collar	GPS + panel solar + batería + carcasa ABS IP67	810 g	Hasta 8 meses sin carga	Muy funcional, pero con peso elevado.
LPWAN Space LoRa Collar	GPS + batería Li-SOCl ₂ (2×19 000 mAh) + carcasa IP67	360 g	3-5 años en modo 1 posición/hora	Buen rendimiento en LoRa, pero batería no recargable y costosa.

Propuesta:	ESP32-LoRa +	220 - 260 g	2-3 días (deep	Muy
Heltec +	NEO-6M GPS	(estimado)	sleep)	competitivo en
NEO-6M + Li-	+ Li-Ion 3.7 V			ligereza, útil en
Ion	3000 mAh +			zonas rurales.
	helicoidal			

3.2.Integración del sistema

Mencionar la integración del sistema hace referencia a la interconexión y configuración adecuada de los distintos módulos electrónicos para el funcionamiento correcto de los mismos. En el caso del proyecto a realizar, el collar debe tener una comunicación fluida entre el módulo LoRa, el GPS y el microcontrolador ESP32, asegurando un óptimo consumo energético y que la parte física del prototipo sea adecuada.

3.2.1. Diagrama de conexión eléctrica

El sistema del collar rastreador está integrado por 3 componentes principales:

- Heltec WiFi LoRa 32 V4
- GPS NEO-6M
- Batería Li-Ion de 3.7V

Estos componentes se encuentran interconectados entre sí; el microcontrolador ESP32 es el cerebro del sistema, encargado de procesar la lógica y variables respectivas, haciendo posible que exista comunicación entre el GPS y el módulo LoRa. Por otro lado, la función de la batería es suministrar la corriente necesaria para todo el sistema, esta será conectada mediante un conector JST-PH 2.0 mm, compatible con el ESP32 Heltec.

El diagrama de conexión incluye:

- VCC del GPS al pin VIN del Heltec (regulado a 3.3V).
- GND compartido entre todos los componentes.
- TX del GPS al GPIO16 del ESP32.
- RX del GPS al GPIO17 del ESP32. (uPesy Electronics, s.f.).

3.2.2. Asignación de pines y conexiones principales

Es muy importante la asignación correcta de los pines para mejorar el funcionamiento del ESP32.

Tabla 12

Asignación de pines

Componente	ESP32 (Heltec)	Descripción
GPS (NEO-6M)	GPIO16 (RX2)	Recepción de datos provenientes del módulo GPS
	GPIO17 (TX2)	Transmisión de comandos hacia el GPS (opcional)
Antena LoRa	U. FL o RF Pad	Salida de RF a antena (helicoidal)
Batería Li-Ion 3.7V	JST (VIN/GND)	Alimentación general del sistema

microcontrolador se ubican en el interior de la carcasa, garantizando protección mecánica y estabilidad durante el desplazamiento del animal. Este diseño busca equilibrar aspectos de rendimiento, resistencia y comodidad en condiciones reales de operación.

Figura 22

Diseño físico del prototipo del collar inteligente para monitoreo de ganado



En función de estas condiciones, se toman en cuenta los siguientes aspectos para la integración de los componentes:

- Microcontrolador (HELTEC ESP32 LoRa): Su ubicación será centrada en la carcasa plástica del dispositivo, además de incluir ventilación pasiva gracias al diseño compacto que permite su integración.
- Antena para comunicación LoRa: conectada al módulo mediante conector U. FL (o interfaz RF correspondiente) y orientada verticalmente, lo que favorece una adecuada propagación de la señal en el entorno de operación.
- Batería Li-Ion 3.7V (3000 mAh): fijada con soporte impreso o en su defecto, con cinta de doble cara; incluye protección ante vibraciones y temperaturas elevadas.

- Módulo GPS: ubicado en la parte superior del collar para mejor recepción, la antena cerámica estará ubicada linealmente para sintonía espacial.
- Carcasa del collar: resistente a la intemperie y condiciones atmosféricas; dentro incluye todo el sistema de geolocalización y protección contra impactos; el material de diseño puede ser silicona o resina de amortiguación.

El diseño del collar inteligente se divide en dos partes principales:

- Unidad transmisora (TX): Esta incluye el GPS, ESP32 Heltec, la antena helicoidal y la batería de 3.7V. Esta unidad estará en todo momento con el animal.
- Unidad Receptora (RX): Similar al módulo Heltec, puede ser alimentado constantemente por unidad USB o, en su defecto por batería, cuenta con una antena helicoidal con buena ganancia. Ubicado en la casa del dueño (SparkFun Electronics, s.f.).

3.3.Flujo de funcionamiento del sistema

La Figura 23 muestra el flujo de funcionamiento del sistema implementado para el monitoreo de ganado en zonas rurales, el cual integra tecnologías de geolocalización, comunicación inalámbrica y plataformas IoT para la visualización de datos en tiempo real.

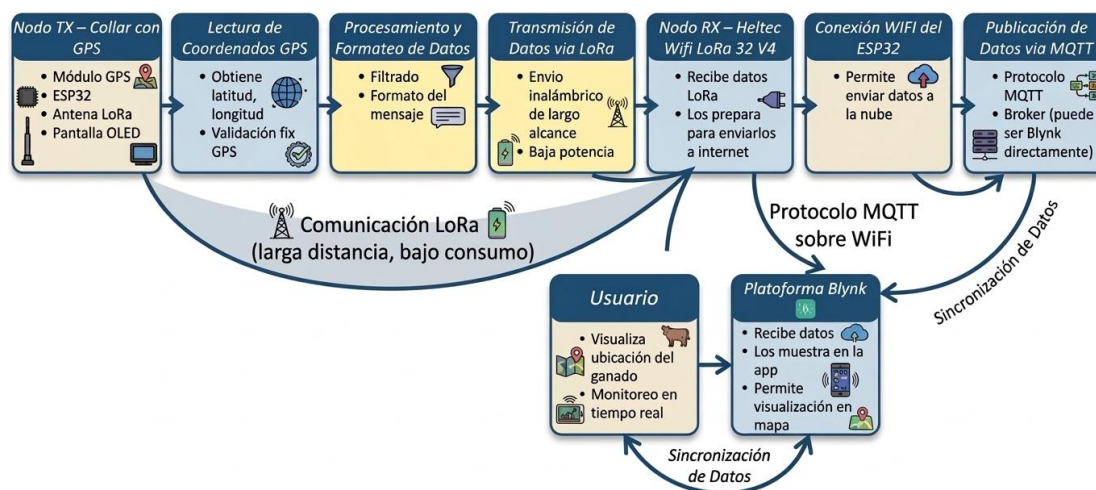
El proceso inicia en el nodo transmisor, desarrollado como un collar inteligente instalado en el animal, el cual incorpora un módulo GPS y un microcontrolador ESP32 con capacidad de comunicación LoRa. En esta etapa, el sistema realiza la adquisición de las coordenadas geográficas (latitud y longitud), verificando previamente la validez de la señal GPS mediante la obtención de un “fix”, con el fin de asegurar la precisión de la información.

Posteriormente, los datos son procesados en el microcontrolador, donde se realiza un filtrado básico y se estructuran en un formato de mensaje adecuado para su transmisión. Esta organización permite optimizar el uso del canal de comunicación y reducir el consumo energético del sistema.

A continuación, la información es transmitida mediante tecnología LoRa hacia el nodo receptor, aprovechando su capacidad de comunicación a larga distancia y bajo consumo, lo cual resulta especialmente adecuado para entornos rurales donde no existe infraestructura de red convencional.

Una vez que los datos llegan al nodo receptor, estos son captados, procesados y preparados para su envío hacia internet, permitiendo la integración con plataformas de monitoreo remoto. En este punto, el sistema realiza la transición desde una red de comunicación de largo alcance hacia una red basada en conectividad WiFi, habilitando así el acceso a servicios en la nube.

Seguidamente, los datos son enviados mediante el protocolo MQTT hacia la plataforma IoT seleccionada, donde son gestionados y organizados para su posterior visualización. Finalmente, la información es presentada al usuario a través de una aplicación móvil, permitiendo el monitoreo en tiempo real de la ubicación del ganado.

Figura 23*Funcionamiento del sistema*

A partir de este flujo general, en las siguientes subsecciones se detalla el funcionamiento específico de cada una de las etapas que conforman el sistema, lo que permite comprender de manera más clara y estructurada la interacción entre los diferentes componentes del prototipo.

3.3.1. Adquisición de datos GPS

El módulo GPS NEO-6M es el encargado de la adquisición de la información de localización, permitiendo obtener las coordenadas geográficas correspondientes a latitud y longitud. Este dispositivo se comunica con el microcontrolador ESP32 mediante una interfaz serial UART (TX/RX), facilitando la transmisión continua de datos.

Para garantizar la validez de la información, es necesario que el módulo establezca conexión con al menos cuatro satélites, proceso conocido como “fix GPS”, el cual asegura una ubicación precisa. El tiempo requerido para lograr esta fijación depende de las condiciones del entorno, siendo típicamente de entre 30 segundos y un minuto en espacios abiertos con visibilidad directa al cielo.

Una vez establecida la conexión, el módulo comienza a emitir datos bajo el protocolo NMEA, el cual define un conjunto de tramas estándar utilizadas para la transmisión de información de posicionamiento entre dispositivos electrónicos. (profwhitetower, 2025)

3.3.2. Procesamiento de datos en el microcontrolador ESP32

Una vez recibidas las tramas NMEA desde el módulo GPS, el microcontrolador ESP32 se encarga de procesar la información con el fin de extraer los datos relevantes, tales como latitud y longitud. Para esta tarea, se emplean bibliotecas especializadas como TinyGPS++, la cual permite interpretar de manera eficiente las cadenas de texto provenientes del GPS.

Durante esta etapa, el sistema aplica validaciones para asegurar la integridad de los datos, verificando que las coordenadas obtenidas sean válidas antes de su utilización. Adicionalmente, se pueden implementar mecanismos de filtrado para evitar el envío de información redundante, optimizando así el uso del canal de comunicación y el consumo energético del sistema. (mikalhart, 2022)

3.3.3. Transmisión de datos mediante LoRa

Una vez procesada la información, el ESP32 transmite los datos hacia el nodo receptor mediante el módulo LoRa integrado en la placa Heltec. Esta comunicación se realiza en la banda ISM de 915 MHz, empleando modulación Chirp Spread Spectrum (CSS), la cual proporciona alta robustez frente a interferencias y permite alcanzar largas distancias con bajo consumo energético.

Para optimizar la eficiencia de la transmisión, los datos de ubicación se estructuran en un formato de texto simple que contiene la latitud y longitud, reduciendo así el tamaño del paquete enviado. Este enfoque resulta adecuado para redes LoRa, donde la capacidad de transmisión es limitada y se prioriza la eficiencia energética. (World Sensing, 2020)

3.3.4. Recepción y procesamiento en el nodo receptor

En esta etapa, el sistema se encarga de captar la información transmitida desde el nodo emisor a través de la tecnología LoRa. El nodo receptor, implementado mediante una placa basada en ESP32 con módulo LoRa integrado, permanece en modo de escucha para detectar la llegada de paquetes de datos enviados por el collar inteligente.

Una vez que se recibe la señal, el módulo LoRa entrega la información al microcontrolador, el cual realiza la decodificación del mensaje recibido. Debido a que los datos fueron previamente estructurados en un formato simple, el proceso de interpretación resulta eficiente, permitiendo extraer directamente las coordenadas geográficas sin necesidad de operaciones complejas.

Adicionalmente, en esta fase se pueden aplicar validaciones básicas para verificar la integridad del mensaje, como la comprobación de longitud de datos o la detección de valores atípicos. Esto contribuye a evitar errores en etapas posteriores del sistema.

De esta manera, el nodo receptor actúa como un punto intermedio que transforma la comunicación de largo alcance (LoRa) en información lista para ser enviada a través de redes convencionales.

3.3.5. Envío de datos a la nube mediante WiFi y MQTT

Una vez procesados los datos en el nodo receptor, el siguiente paso consiste en su transmisión hacia una plataforma en la nube. Para ello, el microcontrolador ESP32 establece una conexión a una red WiFi disponible, lo que permite el acceso a internet y la integración con servicios IoT.

En este sistema, la comunicación se realiza mediante el protocolo MQTT, el cual está diseñado para aplicaciones de bajo consumo y transmisión eficiente de datos. A diferencia de otros

protocolos más pesados, MQTT utiliza un modelo basado en publicación/suscripción, donde el dispositivo actúa como cliente y envía la información a un servidor intermedio conocido como broker.

Los datos de ubicación son publicados en un tópico específico previamente configurado, lo que permite que la plataforma IoT suscrita a dicho tópico reciba la información de forma casi inmediata. Este enfoque reduce el consumo de ancho de banda y optimiza el uso de recursos del sistema, características especialmente importantes en entornos rurales donde la conectividad puede ser limitada.

Gracias a esta arquitectura, se logra una comunicación confiable y continua entre el sistema físico y la aplicación de monitoreo.

3.3.6. Visualización de datos en la aplicación móvil (Blynk)

La etapa final del sistema corresponde a la visualización de la información por parte del usuario. Para ello, se utiliza la plataforma Blynk como interfaz gráfica, la cual permite representar los datos recibidos desde la nube de manera intuitiva y en tiempo real.

A través de la suscripción al tópico MQTT correspondiente, la plataforma recibe las coordenadas geográficas enviadas por el nodo receptor. Estos datos son procesados internamente por Blynk y representados en diferentes elementos visuales dentro de la aplicación móvil, como etiquetas de texto o widgets de geolocalización.

Esto permite al usuario conocer la ubicación del ganado en cualquier momento, facilitando tareas de supervisión y control, especialmente en zonas rurales donde el monitoreo directo resulta complicado.

Además, la interfaz de Blynk ofrece flexibilidad para adaptar la visualización según las necesidades del sistema, pudiendo incorporar alertas, históricos o indicadores adicionales que mejoren la experiencia de uso. (BLYNK, s.f.)

3.4.Implementación de la plataforma IoT

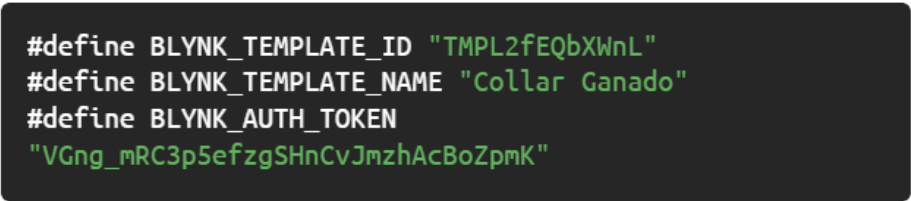
3.4.1. Configuración de la plataforma Blynk

La configuración inicial consistió en la creación de un entorno de trabajo dentro de la plataforma Blynk, utilizando su sistema basado en plantillas (templates) y dispositivos (devices). Este enfoque permite estructurar de manera organizada los datos que serán recibidos desde el nodo receptor.

Se definió un template asociado al dispositivo ESP32, donde se establecieron los parámetros necesarios para la comunicación, incluyendo el identificador único del dispositivo (Device ID) y el token de autenticación (Auth Token), el cual garantiza una conexión segura entre el hardware y la plataforma.

Figura 24

Parámetros para la comunicación

A screenshot of a code editor showing Blynk configuration macros for an ESP32. The code is written in a dark-themed editor with green and white text. It defines the template ID, name, and authentication token.

```
#define BLYNK_TEMPLATE_ID "TMPL2fEQbXWnL"  
#define BLYNK_TEMPLATE_NAME "Collar Ganado"  
#define BLYNK_AUTH_TOKEN  
"VGng_mRC3p5efzgSHnCvJmzhAcBoZpmK"
```

Adicionalmente, se configuró la conectividad inalámbrica del nodo receptor mediante una red WiFi, permitiendo al microcontrolador establecer comunicación con servicios en la nube. Esta

configuración se realizó a nivel del dispositivo ESP32, el cual actúa como un nodo intermediario entre la red LoRa y la plataforma de monitoreo.

De esta manera, el sistema implementa una arquitectura tipo gateway, donde los datos adquiridos por el nodo transmisor son recibidos vía LoRa, procesados en el nodo receptor y posteriormente enviados a la nube utilizando el protocolo MQTT.

3.4.2. Integración con el protocolo MQTT

Para el envío de datos hacia la plataforma, se empleó el protocolo MQTT debido a su eficiencia en aplicaciones IoT y su bajo consumo de ancho de banda. La integración se realizó mediante la configuración de un cliente MQTT en el ESP32, el cual se encarga de publicar los datos recibidos desde el nodo transmisor.

Se definieron tópicos (topics) específicos para la transmisión de información, tales como coordenadas geográficas (latitud y longitud). Estos datos son enviados en formato estructurado, permitiendo su correcta interpretación en la plataforma.

3.4.3. Definición de datastreams

Dentro de la plataforma Blynk, se configuraron datastreams para cada variable a monitorear. En este caso, se definieron principalmente:

- Latitud
- Longitud
- Enlace de ubicación (opcional para visualización directa en mapas)
- Distancia

Cada datastream fue configurado con un tipo de dato adecuado (por ejemplo, tipo “Cadena” para coordenadas), asegurando la precisión en la representación de la información.

Estos canales de datos permiten vincular directamente la información recibida desde el dispositivo con los elementos visuales de la aplicación.

Figura 25

Datastream configurados

☐	Pin	Nombre	Tipo de d...	Semántica
	V0	Latitud	Cadena	+ Agregar
	V1	Longitud	Cadena	+ Agregar
	V2	Google maps	Cadena	+ Agregar
	V3	Distancia	Cadena	+ Agregar

3.4.4. Configuración de widgets de visualización

Para la representación de los datos en la aplicación móvil, se utilizaron diferentes widgets disponibles en la plataforma Blynk, los cuales fueron configurados en función del tipo de información a mostrar.

Entre los principales elementos implementados se encuentran:

- Widgets tipo Label, utilizados para mostrar valores numéricos como latitud y longitud.
- Widget tipo Map, empleado para representar gráficamente la ubicación del dispositivo sobre un entorno geográfico.

Cada widget fue enlazado a su respectivo datastream, permitiendo una actualización automática conforme se reciben nuevos datos. Esta configuración facilita una visualización clara e intuitiva para el usuario final.

3.4.5. Visualización de datos en tiempo real

Una vez establecida la conexión entre el dispositivo y la plataforma, el sistema permite la visualización de datos en tiempo real. Cada vez que el nodo transmisor envía información, esta es procesada por el ESP32 receptor y publicada en la nube mediante MQTT.

La aplicación móvil actualiza automáticamente los valores mostrados, permitiendo observar cambios en la ubicación del ganado sin necesidad de recargar o reiniciar el sistema.

Este comportamiento en tiempo real resulta fundamental para aplicaciones de monitoreo, ya que permite una respuesta rápida ante posibles situaciones de riesgo, como desplazamientos fuera de la zona establecida.

Cabe destacar que, si bien en este apartado se describe el funcionamiento general del sistema en tiempo real, los resultados detallados de las pruebas, así como el análisis de desempeño y comportamiento del sistema, se presentan en el Capítulo IV.

3.5. Validación y pruebas del sistema

Con el propósito de verificar el desempeño del prototipo desarrollado, se definió un conjunto de pruebas orientadas a evaluar la confiabilidad de la transmisión de datos, el consumo energético del dispositivo, el alcance de comunicación y la relación entre la potencia de transmisión y el consumo energético. Estas pruebas permiten determinar el comportamiento del sistema en condiciones reales de operación, especialmente en entornos rurales donde será implementado.

Para garantizar la validez de los resultados, cada prueba fue diseñada bajo condiciones controladas, estableciendo variables específicas de medición, unidades de análisis y procedimientos repetibles. Los resultados obtenidos serán analizados en el Capítulo 4.

3.5.1. Evaluación de transmisión de datos y tasa de error

Esta prueba tiene como objetivo analizar la confiabilidad del enlace de comunicación LoRa, determinando la cantidad de datos transmitidos correctamente y aquellos que se pierden durante el proceso de envío.

Para ello, el dispositivo transmisor fue configurado para enviar paquetes de datos a intervalos de tiempo definidos, incorporando un identificador secuencial en cada transmisión, lo que permite llevar un control del orden de los paquetes enviados. En el nodo receptor, se registran los datos recibidos y se comparan con la secuencia esperada, de modo que la ausencia de un identificador indica la pérdida de un paquete.

A partir de esta información, se calcula la tasa de error en la transmisión, la cual se expresa como un porcentaje que relaciona los paquetes perdidos respecto al total de paquetes enviados.

Ecuación 6

Tasa de error

$$Tasa\ de\ error\ (\%) = \frac{N_{perdidos}}{N_{enviados}} * 100$$

De acuerdo con la Ecuación 6, donde $N_{perdidos}$ corresponde al número de paquetes no recibidos y $N_{enviados}$ al total de paquetes transmitidos. Esta métrica permite evaluar la eficiencia del sistema de comunicación bajo diferentes condiciones de operación, como variaciones en la distancia o la presencia de obstáculos.

3.5.2. Evaluación del consumo energético del dispositivo

El análisis del consumo energético tiene como finalidad determinar la autonomía del dispositivo, aspecto fundamental en aplicaciones de monitoreo remoto donde el acceso a fuentes de energía es limitado.

Para esta prueba, el dispositivo se somete a un ciclo de funcionamiento continuo, manteniendo activos los módulos de adquisición de datos GPS y transmisión LoRa. Se registra el tiempo total de operación desde una carga completa de la batería hasta el punto en que el sistema deja de funcionar, expresando la autonomía en horas.

Adicionalmente, se consideran diferentes intervalos de transmisión de datos, con el fin de analizar su impacto en el consumo energético del sistema. Para complementar el análisis, se puede estimar la autonomía teórica del dispositivo a partir de la relación entre la capacidad de la batería y el consumo promedio de corriente, como se muestra a continuación:

Ecuación 7

Autonomía de la batería

$$Autonomia(h) = \frac{C_{bateria}}{I_{promedio}}$$

De acuerdo con la Ecuación 7, donde $C_{bateria}$ representa la capacidad de la batería en miliamperios-hora (mAh), e $I_{promedio}$ corresponde a la corriente promedio consumida por el dispositivo en miliamperios (mA). Este análisis permite identificar configuraciones de operación más eficientes desde el punto de vista energético.

3.5.3. Prueba de alcance de comunicación LoRa

El objetivo de esta prueba es determinar la distancia máxima a la cual el sistema es capaz de mantener una comunicación estable entre el nodo transmisor y el nodo receptor.

Para su ejecución, el receptor se ubica en un punto fijo, mientras que el transmisor se desplaza progresivamente, incrementando la distancia en intervalos definidos. En cada punto de medición se realizan múltiples envíos de datos, con el fin de evaluar la estabilidad del enlace de comunicación.

Durante la prueba, se registran parámetros como la cantidad de paquetes recibidos correctamente y la distancia entre los dispositivos, así como indicadores de calidad de señal tales como el RSSI (Received Signal Strength Indicator), expresado en dBm, y la relación señal-ruido (SNR), expresada en dB. Estos parámetros permiten analizar el comportamiento del sistema en función de la distancia y las condiciones del entorno, proporcionando una visión más completa del rendimiento de la comunicación LoRa.

3.6. Configuración y calibración del sistema

Al realizar la configuración del sistema correctamente, se cumple un requisito importante para establecer que el rendimiento sea confiable durante la transmisión de datos a través del módulo LoRa y geolocalización.

3.6.1. Parámetros LoRa aplicados en el prototipo (SF, BW, CR)

Es necesario seleccionar los siguientes parámetros para el módulo LoRa; así se obtiene un balance entre alcance, robustez y a la vez consumo energético:

- Ancho de banda (BW): Se utilizará 125 kHz, lo cual es ampliamente seleccionado para aplicaciones de telemetría, esto gracias a su excelente sensibilidad ante interferencias.
- Tasa de codificación (CR): Se configurará en 4/8, ya que así se mejora la tolerancia ante errores sin aumentar tanto la latencia.
- Factor de propagación (SF): Se deberá adoptar un valor alto; de esta forma se extiende el alcance en las zonas rurales sin afectar tanto la velocidad de transmisión.
En este caso se utilizará SF12. (KWKWII, 2018)

Teniendo en cuenta estos parámetros, se hacen los siguientes cálculos de la velocidad/tasa de bits. Para esto se utiliza la ecuación 1, tal como se mencionó y explicó anteriormente.

Ecuación 8

Tasa de bits

- Para SF9.

$$R_b = \frac{BW}{2^{SF}} * SF * \frac{4}{4 + CR_{val}} [bits/s]$$

$$R_b = \frac{125000}{2^9} * 9 * \frac{4}{4 + 1} [bits/s]$$

$$R_b = \frac{125000}{512} * 9 * \frac{4}{5} [bits/s]$$

$$R_b = \frac{25000}{128} * 9 * \frac{1}{1} [bits/s]$$

$$R_b = \frac{25000}{128} * 9 [bits/s]$$

$$R_b = 1757,8 \text{ [bits/s]} \approx 1,76 \text{ [Kbits/s]}$$

- Para SF10.

$$R_b = \frac{BW}{2^{SF}} * SF * \frac{4}{4 + CR_{val}} \text{ [bits/s]}$$

$$R_b = \frac{125000}{2^{10}} * 10 * \frac{4}{4 + 1} \text{ [bits/s]}$$

$$R_b = \frac{125000}{1024} * 10 * \frac{4}{5} \text{ [bits/s]}$$

$$R_b = \frac{125000}{256} * 2 * \frac{1}{1} \text{ [bits/s]}$$

$$R_b = \frac{125000}{128} * 1 \text{ [bits/s]}$$

$$R_b = \frac{15625}{16} \text{ [bits/s]}$$

$$R_b = 976,6 \text{ [bits/s]} \approx 0,98 \text{ [Kbits/s]}$$

Se puede observar que al subir de SF9 a SF10 mejora el alcance/sensibilidad, es decir, será más robusto contra ruidos. Sí se pierde un poco de velocidad, sin embargo, no tanto como para decir que será inviable aplicarla en este proyecto, dado que lo que enviará serán coordenadas (textos cortos) que ocupan pocos bytes. (ELECTRICAL ENGINEERING, 2020)

3.6.2. Frecuencia de actualización GPS y sincronización

El módulo GPS se configurará a una frecuencia de 1 Hz, lo que equivale a una lectura cada segundo; esto es ideal para hacer el seguimiento correcto al ganado, el cual se desplaza a un ritmo

no tan acelerado. Este valor ayuda a que todo se mantenga en equilibrio entre la precisión de la geolocalización y el consumo energético.

Así mismo, se aplicará un mecanismo para sincronizar las lecturas GPS con la transmisión LoRa; de esta forma se evita el envío de redundancia y asegura que se transmita cada paquete.

3.6.3. Optimización energética mediante modos de suspensión

Para el sistema de monitoreo descrito, la autonomía del dispositivo depende de varios factores:

- Tiempo activo del microcontrolador
- Frecuencia de transmisión LoRa
- Tiempo de adquisición GPS
- Uso de WiFi
- Uso de pantalla OLED
- Periféricos activos

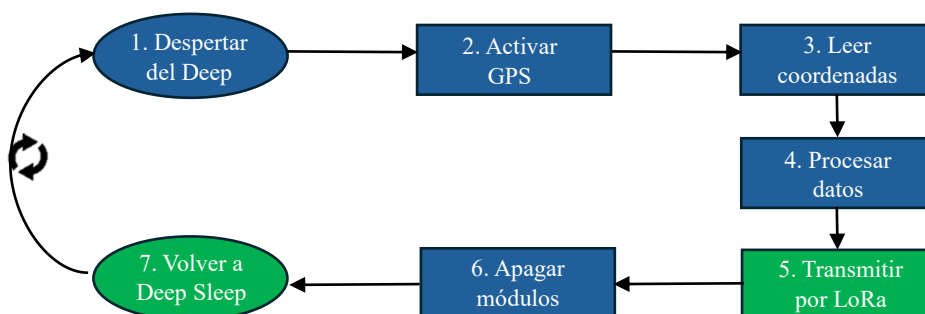
Para conseguir la mayor autonomía posible, se requiere de un diseño con un consumo energético pequeño, con tiempos de suspensión, y que únicamente funcione en su totalidad cuando sea estrictamente necesario. A raíz de esto, se exponen los modos de suspensión disponibles para el módulo Heltec seleccionado:

Tabla 13*Modos de suspensión*

Modo	Consumo aproximado	Aplicación en el collar
Active Mode	80-240 mA	Procesamiento y TX
Modem Sleep	15-20 mA	CPU activa sin WiFi/BT
Light Sleep	2-10 mA	Pausas cortas
Deep Sleep	10-800 μ A	Ideal

Como se puede observar en la Tabla 13, el módulo Heltec tiene varios modos de ahorro energético, los cuales se diseñan para disminuir el consumo energético en el caso de que el sistema se encuentre en operación inactiva.

Tomando en cuenta varios puntos sobre el consumo energético sin dejar atrás el funcionamiento del prototipo, se decidió el modo Deep Sleep, ya que este únicamente mantiene activos los circuitos más esenciales que se requieren para las interrupciones externas. De esta forma, el consumo se reduce hasta microamperios, lo que hace de él la alternativa ideal para aplicarlo en el monitoreo remoto que se alimenta de baterías.

Figura 26*Arquitectura de funcionamiento Deep Sleep*

Se implementó un modo de suspensión personalizado basado en una estrategia híbrida de despertar espaciado y transmisión inteligente. En lugar de transmitir siempre, se puede hacer lo siguiente:

1. Despertar cada 2 minutos.
2. Activar GPS.
3. Leer coordenadas.
4. Se transmite al receptor.
5. Volver a Deep Sleep.

Esta metodología evita envíos innecesarios, reduce el tiempo activo del sistema y optimiza significativamente el consumo energético promedio del collar inteligente.

Con esta estrategia híbrida, la optimización energética se logró mediante una arquitectura de activación periódica inteligente, en la cual el sistema permanece en modo Deep Sleep la mayor parte del tiempo, activando únicamente el módulo GPS para verificar cambios significativos de posición. La transmisión LoRa se realiza únicamente cuando el desplazamiento supera un umbral predefinido, reduciendo el duty cycle del sistema y extendiendo la autonomía operativa del collar en entornos rurales.

3.6.4. Calibración inicial del sistema

Antes de implementar el prototipo en condiciones reales, es necesario realizar una calibración inicial que permita comprobar que todos los componentes funcionan de manera correcta y coordinada. Este proceso no solo ayuda a detectar posibles fallos, sino que también permite hacer pequeños ajustes para mejorar el desempeño general del sistema.

En una primera etapa, se verifica el funcionamiento del módulo GPS. Para ello, se comprueba que el dispositivo logre obtener señal de los satélites y proporcione coordenadas válidas. Es recomendable realizar esta prueba en un espacio abierto, donde la señal no tenga interferencias. Además, se pueden comparar las coordenadas obtenidas con una ubicación conocida, lo que permite tener una idea de la precisión del sistema y detectar posibles desviaciones.

Una vez validado el GPS, se comprueba la comunicación mediante LoRa. En esta fase se revisa que los datos enviados desde el nodo transmisor sean recibidos correctamente en el nodo receptor. Más allá de confirmar que exista comunicación, es importante observar si la información llega completa y sin errores. También se pueden realizar pruebas a distancias cortas para asegurarse de que el sistema funciona de forma estable antes de someterlo a pruebas más exigentes.

Otro punto clave es la sincronización entre la captura de datos y su transmisión. Es decir, se debe verificar que el tiempo en el que se obtiene la información del GPS coincida adecuadamente con los intervalos de envío configurados. Una mala sincronización podría generar pérdida de información o un uso innecesario de energía.

En cuanto al consumo energético, se revisa que el dispositivo entre correctamente en los modos de bajo consumo cuando no está activo. Esto es fundamental en aplicaciones como el monitoreo de ganado, donde se busca maximizar la duración de la batería. Durante esta etapa, se observa el comportamiento del sistema en ciclos de funcionamiento, evaluando cómo responde en las fases de lectura, transmisión y reposo.

Finalmente, se realiza una prueba general del sistema, donde se analiza el funcionamiento conjunto de todos los módulos. En esta prueba se verifica que el proceso completo, desde la

obtención de la ubicación hasta la recepción de los datos, se ejecute sin inconvenientes. Esto permite tener una mayor seguridad de que el prototipo está listo para ser utilizado en campo.

3.7. Análisis estimativo del desempeño del prototipo

En esta sección se presenta un análisis orientado a estimar el comportamiento del prototipo bajo condiciones de operación similares a las que enfrentará en un entorno real. A través de distintos criterios técnicos, se busca evaluar de manera anticipada aspectos clave como el alcance de comunicación, el consumo energético y la eficiencia en la transmisión de datos.

Para ello, se abordan tres puntos principales. En primer lugar, se realiza una estimación del presupuesto de enlace, lo que permite tener una idea del rendimiento de la comunicación LoRa en el entorno de aplicación. Posteriormente, se analiza el consumo energético del sistema, considerando su impacto en la autonomía del dispositivo. Finalmente, se estudia el tiempo de transmisión y el tamaño de los datos enviados, con el fin de entender cómo estos factores influyen en la eficiencia general del sistema.

Este análisis permite tener una visión más clara del desempeño esperado del prototipo, facilitando la toma de decisiones y posibles mejoras antes de su implementación definitiva en campo.

3.7.1. Estimación del presupuesto de enlace en el entorno de aplicación

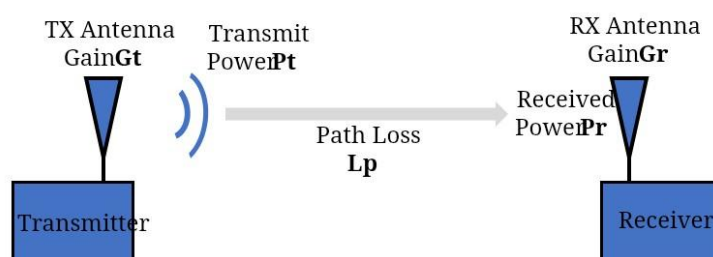
En esta sección se realiza el cálculo del presupuesto de enlace del sistema de monitoreo de ganado utilizando el módulo Heltec WiFi LoRa 32 (V4). El objetivo es analizar si la comunicación entre el collar inteligente y el receptor será confiable en un entorno rural, considerando factores como la potencia de transmisión, las pérdidas en el espacio libre, el ruido y la relación señal-ruido.

Con estos datos se determina el margen de enlace, el cual permite verificar si la señal recibida es suficiente para un funcionamiento adecuado del sistema.

De acuerdo con la Figura 27, se ilustra el modelo teórico del presupuesto de enlace considerado en este análisis, donde se identifican los principales parámetros que intervienen en la propagación de la señal desde el transmisor hasta el receptor.

Figura 27

Esquema del presupuesto de enlace en un sistema inalámbrico



El módulo integra el chip LoRa SX1262, cuyas características principales se muestran en la Tabla 14:

Tabla 14

Características principales de SX1262

Parámetro	Valor típico
Frecuencia de operación	863–928 MHz
Potencia máxima de transmisión	+25 dBm
Sensibilidad máxima	≈ -148 dBm (SF12, BW 125 kHz)

Ancho de banda típico	125 kHz
Ganancia antena incluida	≈ 2 dBi

Para este tipo de aplicaciones en América Latina, generalmente se utiliza una frecuencia de 915 MHz.

Ecuación 9

Potencia efectiva transmitida (EIRP)

$$EIRP = P_{TX} + G_{TX} - L_{cable}$$

La potencia radiada depende de la potencia del módulo más la ganancia de la antena.

Usando valores típicos:

- Potencia TX = 25 dBm
- Ganancia antena TX = 2 dBi

$$EIRP = 25 + 2 - 1$$

$$EIRP = \mathbf{26\ dBm}$$

Este resultado indica la potencia real con la que la señal es radiada al medio, considerando la antena. Es clave para determinar el alcance potencial del sistema.

Ecuación 10

Pérdidas de propagación en el espacio libre (FSPL)

$$FSPL(dB) = 20\log_{10}(d) + 20\log_{10}(f) + 32.44$$

Para un escenario más real, se va a evaluar una distancia de 2 km.

Resultado aproximado para:

- Frecuencia: 915 MHz
- Distancia: 2 km

$$FSPL = 20(0.3010) + 20(2.961) + 32.44$$

$$FSPL = 6.02 + 59.22 + 32.44$$

$$FSPL \approx \mathbf{97.68\ dB}$$

Las pérdidas por propagación en el espacio libre se estiman en aproximadamente 97.68 dB, lo que refleja la atenuación que experimenta la señal al desplazarse por el entorno. Este valor resulta clave para entender cómo disminuye la potencia antes de llegar al receptor.

Cabe mencionar que este análisis se basa en condiciones ideales de espacio libre, por lo que en un entorno real podrían presentarse pérdidas adicionales debido a obstáculos, vegetación o condiciones climáticas.

Ecuación 11

Potencia recibida

$$P_r = P_{TX} + G_{TX} + G_{RX} - FSPL - L$$

Ahora se calcula cuánta señal llega al receptor.

Tabla 15

Valores típicos

Parámetro	Valor
Potencia TX	25 dBm

Ganancia antena TX 2 dBi

Ganancia antena RX 2 dBi

Pérdidas sistema 1 dB

FSPL 97.68 dB

$$P_r = 25 + 2 + 2 - 97.68 - 1$$

$$P_r = 28 - 97.68$$

$$P_r \approx -69.68 \text{ dBm}$$

Este valor muestra cuánta señal llega efectivamente al receptor. Es fundamental, puesto que permite verificar si la señal es suficiente para ser detectada correctamente.

Ecuación 12

Ruido del receptor

$$N(\text{dBm}) = -174 + 10\log_{10}(B) + NF$$

El nivel de ruido térmico se calcula con:

Donde:

- $B = 125000 \text{ Hz}$
- $NF = 6 \text{ dB}$

$$N = -174 + 10\log_{10}(125000) + 6$$

$$N = -174 + 10(5.097) + 6$$

$$N = -174 + 50.97 + 6$$

$$N \approx -117.03 \text{ dBm}$$

El resultado aproximado de ruido sería -117 dBm. El nivel de ruido representa las interferencias internas del sistema. Su análisis es importante porque establece el límite mínimo sobre el cual la señal debe estar para poder ser interpretada.

Ecuación 13

Relación señal-ruido (SNR)

$$SNR = P_r - N$$

$$SNR = -69.68 - (-117.03)$$

$$SNR \approx 47.35 \text{ dB}$$

Esto es muy bueno para comunicación LoRa, ya que indica qué tan clara llega la señal respecto al ruido.

Ecuación 14

Margen de enlace

$$\text{Margen de enlace} = P_r - \text{Sensibilidad}$$

El chip SX1262 del módulo Heltec tiene una sensibilidad aproximada de:

Tabla 16

Sensibilidad

Spreading Factor	Sensibilidad
SF7	-124 dBm

SF12	-137 dBm
------	----------

Por ejemplo, si se escoge trabajar con SF12:

$$\text{Margen de enlace} = P_r - \text{Sensibilidad}$$

$$\text{Margen de enlace} = -69.68 - (-137)$$

$$\text{Margen de enlace} \approx \mathbf{67.32 \text{ dB}}$$

A partir de los resultados obtenidos, se puede observar que la señal recibida en el sistema es considerablemente superior al nivel mínimo requerido para una correcta comunicación. Esto se refleja en un margen de enlace amplio, indicando una alta robustez del enlace inalámbrico. En términos prácticos, este comportamiento sugiere que el sistema podría mantener su funcionamiento incluso ante la presencia de factores propios del entorno rural, como vegetación, obstáculos moderados o irregularidades del terreno. En conjunto, estos resultados permiten afirmar que la tecnología LoRa resulta adecuada para aplicaciones de monitoreo de ganado en este tipo de escenarios.

3.7.2. Análisis del tiempo de transmisión y tamaño de los datos

Al utilizar el módulo Heltec Wifi LoRa 32 V4, se tiene la ventaja de contar con el microcontrolador ESP32, el transceptor Semtech SX1262 para la comunicación LoRa y un receptor GPS. De ahí que esta combinación de componentes resulte idónea para la geolocalización y comunicaciones de largo alcance con un bajo consumo energético.

Para el diseño del collar inteligente con el módulo seleccionado, la transmisión de datos requiere de ciertas variables específicas como:

- Latitud

- Longitud
- Altitud
- Número de satélites
- Identificador del nodo
- Contador de transmisión

Normalmente, la comunicación vía GPS se muestra con el siguiente esquema:

ID, Latitud, Longitud, Altitud, SAT

01,-0.21543,-78.51234,2815,7

Al utilizar el formato ASCII, el tamaño estimado de este paquete es:

Tabla 17

Tamaño estimado de paquete

Campo	Tamaño aproximado
ID	2 bytes
Latitud	10 bytes
Longitud	11 bytes
Altitud	6 bytes
Satélites	2 bytes
Separadores	4 bytes

Dando un total aproximado de 35 bytes.

En el caso de utilizar código binario, el tamaño puede reducirse a una cantidad considerable de 16 a 20 bytes, lo que mejoraría la transmisión de datos y consumiría menos recursos al momento de entablar comunicaciones.

Tiempo de transmisión

El tiempo de transmisión en redes LoRa depende de varios parámetros:

- Spreading Factor (SF)
- Bandwidth (BW)
- Coding Rate (CR)
- Tamaño del payload

Estos parámetros influyen directamente en el tiempo que tarda un paquete en transmitirse, conocido como Time on Air (ToA), que corresponde al período durante el cual el módulo LoRa ocupa el canal para enviar la información. En términos generales, cuando se utiliza un spreading factor más alto, el tiempo de transmisión aumenta, aunque esto permite mejorar la sensibilidad del receptor y alcanzar mayores distancias de comunicación. Por otro lado, un bandwidth mayor permite que los datos se transmitan más rápido, reduciendo la duración del envío. El coding rate, al añadir información adicional para la detección y corrección de errores, también puede incrementar ligeramente el tiempo de transmisión. Finalmente, el tamaño del payload tiene una relación directa con la duración del envío, ya que, a mayor cantidad de datos transmitidos, mayor será el tiempo necesario para completar la comunicación. (Savoie Mont Blanc University , 2022)

Para el prototipo se considera una configuración típica para aplicaciones IoT de largo alcance:

$$BW = 125 \text{ kHz}$$

$$CR = 4/5$$

$$SF = 12$$

Con un payload aproximado de **35 bytes**, el tiempo de transmisión estimado es:

Tabla 18

Tiempo de transmisión

Spreading Factor	Tiempo de transmisión
SF7	~56 ms
SF9	~185 ms
SF10	~371 ms

A partir de estos valores, se observa que el tiempo de transmisión es bastante reducido, incluso al emplear factores de dispersión elevados. Esto resulta favorable debido a que permite mantener un equilibrio entre alcance y eficiencia energética.

CAPÍTULO IV: RESULTADOS EXPERIMENTALES Y ANÁLISIS DEL DESEMPEÑO DEL SISTEMA

4.1. Verificación de la adquisición de coordenadas GPS

La Figura 28 muestra la información recopilada durante la etapa inicial de funcionamiento del collar inteligente. Cuando el sistema sale del modo Deep Sleep, el microcontrolador activa el módulo GPS. Así inicia la búsqueda y sincronización con los satélites disponibles para obtener una posición válida.

La información visualizada permite verificar en tiempo real el estado del proceso de adquisición de coordenadas y muestra tres parámetros fundamentales: el estado de activación del sistema, la cantidad de satélites detectados y el tiempo transcurrido durante la búsqueda de la ubicación.

Figura 28

Visualización del proceso de adquisición de señal GPS en la pantalla OLED del collar inteligente



El indicador "Wake" confirma que el collar ha salido correctamente del estado de bajo consumo e inicia un nuevo ciclo de operación. La visualización de este estado permite verificar

que la estrategia de ahorro energético implementada funciona de manera adecuada y que el sistema reactive correctamente todos los módulos necesarios para la captura y transmisión de datos.

El número de satélites detectados constituye uno de los parámetros más relevantes para evaluar el desempeño del receptor GPS, ya que influye directamente en la precisión de las coordenadas obtenidas. Aunque es posible identificar satélites desde los primeros segundos de funcionamiento, el receptor requiere recibir información de al menos cuatro de ellos para calcular una posición tridimensional confiable. A medida que aumenta la cantidad de satélites visibles, la calidad de la señal mejora eficientemente.

Por su parte, el contador de tiempo permite determinar el intervalo necesario para que el módulo GPS obtenga una solución de posicionamiento válida después de cada activación. Este parámetro resulta especialmente importante debido a su relación directa con el consumo energético del collar, puesto que el receptor permanece activo hasta completar la adquisición de las coordenadas.

Durante las pruebas realizadas en entornos rurales, se observó que la cantidad de satélites detectados presentó variaciones entre diferentes mediciones. Este comportamiento está asociado principalmente a las condiciones del entorno, tales como la presencia de vegetación, obstáculos físicos, la orientación de la antena GPS y la visibilidad del cielo. Del mismo modo, la distribución orbital de la constelación de satélites cambia continuamente, por lo que el número de señales disponibles puede variar según la hora y la ubicación de la prueba.

La Tabla 19 presenta los resultados obtenidos durante las pruebas de adquisición de coordenadas geográficas realizadas por el módulo GPS integrado en el collar inteligente. Las pruebas se llevaron a cabo en un área rural de campo abierto, seleccionada por representar las

condiciones reales de operación del sistema de monitoreo de ganado. Durante cada medición se registró el número de satélites detectados, el tiempo necesario para obtener una ubicación válida y la verificación de las coordenadas generadas.

Tabla 19

Tiempo de adquisición de coordenadas GPS

Prueba	Satélites detectados	Tiempo de adquisición	Coordenadas válidas
1	11	21 s	SI
2	12	23 s	SI
3	10	20 s	SI
4	10	19 s	SI

Los resultados que muestran que el receptor GPS detectó entre 10 y 12 satélites durante las pruebas realizadas. Esta cantidad de satélites fue suficiente para obtener coordenadas válidas en todos los casos. La diferencia en el número de satélites detectados entre una prueba y otra se debe a que los satélites GPS se encuentran en constante movimiento alrededor de la Tierra, por lo que su disponibilidad varía con el tiempo. La cantidad de 4 pruebas fue seleccionada de forma aleatoria con el fin de verificar el correcto funcionamiento del sistema, es decir, el número no representa un criterio estadístico o un número específico de las pruebas.

Aunque el sistema GPS esté conformado por una constelación de más de 30 satélites operativos que están distribuidos alrededor de la Tierra, el receptor GPS NEO-6M no visualiza todos simultáneamente. La cantidad de satélites que se detectan depende de la posición geográfica, la hora, la geometría de la constelación y, muy importante, las condiciones del entorno. Un aspecto

fundamental es que solo se consideran aquellos satélites que tengan señal con intensidad y calidad suficiente para calcular una posición confiable.

Asimismo, se observó que el tiempo de adquisición de coordenadas se mantuvo entre 19 y 23 segundos, permitiendo obtener coordenadas válidas en todas las pruebas realizadas. La ligera variación registrada entre mediciones puede atribuirse a cambios en la visibilidad de los satélites y al proceso interno de cálculo de posición efectuado por el receptor.

En general, los resultados evidencian un comportamiento estable del módulo GPS, logrando adquirir coordenadas confiables en un tiempo adecuado para la aplicación propuesta. Esto garantiza la disponibilidad de información geográfica precisa para su posterior transmisión mediante la tecnología LoRa y su visualización en la plataforma de monitoreo.

4.2. Validación de la transmisión de datos mediante LoRa

La interfaz visual desplegada que se muestra en la Figura 29 se generó durante la fase de transmisión de datos mediante tecnología LoRa. Esta visualización permite verificar en tiempo real el estado operativo del sistema y confirmar que el proceso de adquisición y envío de la información de ubicación se ejecuta correctamente.

Figura 29

Confirmación del envío de coordenadas geográficas desde el collar inteligente



Entre los parámetros mostrados se encuentran las coordenadas geográficas de latitud y longitud, las cuales representan la posición exacta del animal dentro del área de monitoreo. La visualización de estos datos permite comprobar que el receptor GPS ha obtenido una solución válida y que la información enviada corresponde a una ubicación real y actualizada.

Adicionalmente, durante las pruebas realizadas, la visualización del mensaje “TX OK” permitió verificar que el ESP32 procesó correctamente las coordenadas obtenidas por el GPS y completó el envío del paquete de datos hacia el nodo receptor mediante tecnología LoRa.

La Tabla 20 presenta los resultados obtenidos durante las pruebas de transmisión de datos mediante tecnología LoRa. En cada prueba se verificó la recepción correcta de las coordenadas geográficas enviadas por el collar inteligente hacia el nodo receptor, registrando la latitud, longitud y distancia calculada entre la ubicación del dispositivo y el punto de referencia establecido para la prueba.

Tabla 20

Pruebas de transmisión de datos mediante LoRa

Prueba	Latitud recibida	Longitud recibida	Distancia
1	-2.219952	-80.869362	15 m
2	-2.219879	-80.869232	31 m
3	-2.219833	-80.869034	52 m
4	-2.219780	-80.868797	79 m

Los datos obtenidos evidencian que las coordenadas transmitidas fueron recibidas correctamente en todas las pruebas realizadas, permitiendo determinar la ubicación del dispositivo en diferentes posiciones dentro del área de prueba. Asimismo, se observa una variación progresiva

de la distancia registrada, pasando de 15 m a 79 m, lo que confirma la capacidad del sistema para detectar y reportar cambios en la localización del animal a medida que este se desplaza dentro de la zona monitoreada.

Las diferencias observadas entre las coordenadas recibidas son consistentes con el movimiento realizado durante las pruebas y demuestran que la información transmitida conserva la precisión necesaria para el seguimiento geográfico. Esto indica que el enlace de comunicación LoRa implementado fue capaz de transportar los datos de ubicación sin pérdidas apreciables ni alteraciones que afectaran la interpretación de la posición.

En conjunto, los resultados obtenidos permiten validar el correcto funcionamiento de la transmisión de datos mediante LoRa, garantizando que la información de ubicación generada por el módulo GPS pueda ser enviada y recibida de forma confiable para su posterior visualización y monitoreo remoto.

4.3. Recepción y visualización de datos en el nodo receptor

La información registrada en el monitor serial del nodo receptor se muestra en la Figura 30 que se realizó durante las pruebas de funcionamiento del sistema de monitoreo de ganado. En esta interfaz se visualizan los eventos asociados a la recepción, procesamiento y reenvío de los datos transmitidos por el collar inteligente mediante tecnología LoRa.

Figura 30

Monitor serial del nodo receptor durante las pruebas experimentales

```

...WiFi conectado
Blynk conectado
WiFi: 3
Blynk: 1
Paquete recibido:
-2.223815,-80.862511
https://www.google.com/maps?q=-2.223815,-80.862511
Estado Blynk antes de enviar: 1
Datos enviados a Blynk
Estado Blynk despues: 1
Ubicacion nueva -> enviando Telegram
WiFi: 3
Blynk: 1
Paquete recibido:
SLEEP,10,120
Transmisor entro en Deep Sleep

```

Inicialmente, los mensajes de conexión WiFi y autenticación con la plataforma Blynk verifican que el nodo receptor dispone de acceso a Internet y mantiene comunicación activa con los servicios de monitoreo remoto. Los indicadores "WiFi: 3" y "Blynk: 1" corresponden a variables internas de estado utilizadas para supervisar la conectividad del sistema. El valor 3 equivale al estado WL_CONNECTED del ESP32, indicando una conexión exitosa a la red inalámbrica, mientras que el valor 1 representa una sesión activa con la plataforma Blynk.

Posteriormente, el mensaje "Paquete recibido" confirma la recepción satisfactoria de la trama enviada por el collar inteligente. Las coordenadas geográficas mostradas en el monitor serial evidencian que la información obtenida por el módulo GPS fue transmitida y procesada correctamente. A partir de estos datos, el sistema genera automáticamente un enlace compatible con Google Maps y actualiza la información en la plataforma Blynk, permitiendo la visualización remota de la ubicación del animal.

Asimismo, la notificación "Ubicación nueva → enviando Telegram" indica la ejecución del mecanismo de alertas implementado, mediante el cual se informa al usuario sobre cambios en la posición del animal en tiempo real.

Finalmente, los mensajes "SLEEP,10,120" y "Transmisor entró en Deep Sleep" confirman la correcta ejecución de la estrategia de ahorro energético implementada en el collar inteligente. Una vez completadas las tareas de adquisición, transmisión y actualización de la información, el nodo transmisor ingresa al modo de bajo consumo del ESP32, reduciendo el gasto energético y contribuyendo a incrementar la autonomía del sistema durante su operación en zonas rurales. En conjunto, los resultados obtenidos demuestran que el nodo receptor es capaz de gestionar adecuadamente la recepción de datos, la sincronización con las plataformas de monitoreo y la supervisión del estado energético del sistema.

4.4. Verificación de la interfaz de monitoreo

Con el propósito de validar la visualización remota de la información generada por el sistema, durante las pruebas experimentales se verificó la actualización de los datos transmitidos por el collar inteligente en las plataformas de monitoreo implementadas.

Las coordenadas geográficas obtenidas por el módulo GPS fueron enviadas desde el nodo transmisor hacia el nodo receptor mediante comunicación LoRa. Posteriormente, el ESP32 del nodo receptor estableció una conexión Wi-Fi para transmitir la información hacia las plataformas Blynk y Telegram a través de servicios basados en Internet, permitiendo la supervisión remota de la ubicación del animal en tiempo real.

En la interfaz de Blynk que se muestra en la Figura 31 se comprobó la visualización de las coordenadas de latitud y longitud obtenidas por el módulo GPS, así como la generación automática

del enlace de acceso a Google Maps y el cálculo de la distancia entre el punto de referencia establecido y la ubicación reportada por el collar inteligente.

Figura 31

Visualización de las coordenadas geográficas y variables de monitoreo en la plataforma Blynk durante las pruebas experimentales

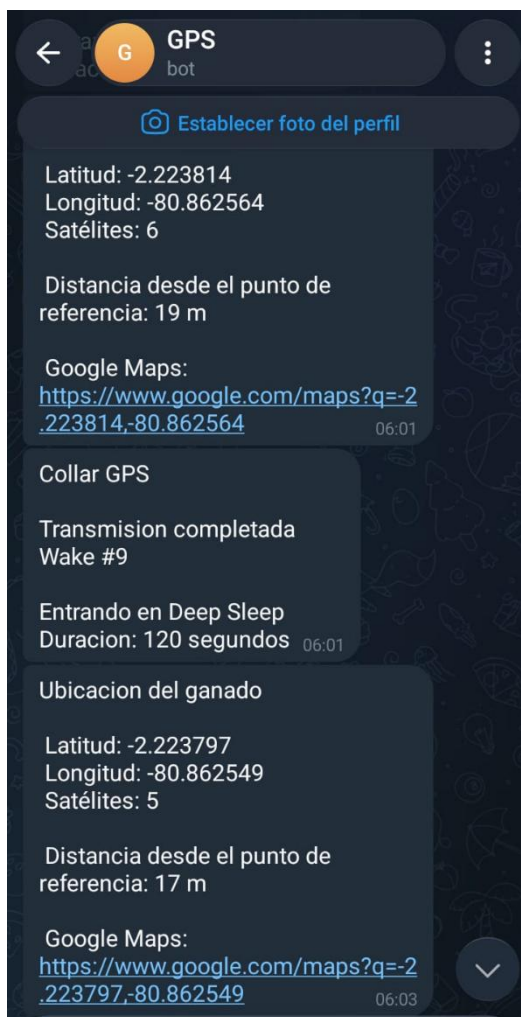


La actualización periódica de estos parámetros permitió verificar la integridad de la información transmitida y la sincronización entre el nodo receptor y la plataforma de monitoreo.

De manera complementaria, se confirmó la recepción de notificaciones automáticas mediante Telegram, como se observa en la Figura 32, las cuales incluyeron información relacionada con la ubicación del animal, el número de activaciones del sistema mediante el contador de ciclos de funcionamiento (Wake) y el tiempo configurado para el modo Deep Sleep.

Figura 32

Recepción de notificaciones automáticas con información de localización mediante Telegram durante las pruebas experimentales



Asimismo, los mensajes enviados incorporaron un enlace directo a Google Maps, facilitando la consulta inmediata de la ubicación geográfica desde dispositivos móviles con acceso a Internet.

Los resultados obtenidos evidenciaron la integración adecuada entre el enlace LoRa, la conectividad Wi-Fi y las plataformas de monitoreo implementadas, garantizando la disponibilidad y accesibilidad de la información generada durante las pruebas de campo.

4.4.1. Verificación del envío de alerta de seguridad

Se configuró un umbral máximo de distancia respecto al punto de referencia como sistema de seguridad. De acuerdo con los resultados obtenidos durante las pruebas de alcance que se visualizarán en la Tabla 22, el sistema presentó una comunicación estable hasta una distancia aproximada de 1 km. Sin embargo, una estrategia que se utilizó fue alertar cuando el collar detecta que el ganado se aleja más del límites establecido; en este caso fue de 600 metros. El dispositivo receptor genera automáticamente una notificación a través del bot de Telegram.

Como se puede observar en la Figura 33, el mensaje de alerta fue recibido indicando que el bovino ha excedido el umbral de seguridad. Posteriormente, se envía el mensaje de la ubicación del animal y la distancia a la que se encuentra.

Figura 33

Notificación de alerta de seguridad



4.5. Verificación del funcionamiento del modo Deep Sleep

Durante las pruebas experimentales, se verificó el funcionamiento del modo Deep Sleep implementado en el collar inteligente como estrategia para optimizar el consumo energético del sistema.

Una vez completado el proceso de adquisición y transmisión de las coordenadas geográficas, el dispositivo ingresó automáticamente al estado de bajo consumo, conforme al intervalo de suspensión previamente configurado. La pantalla OLED que se muestra en la Figura 34 permitió visualizar el tiempo establecido para el modo Deep Sleep, proporcionando una evidencia del cambio de estado del sistema.

Figura 34

Visualización del modo Deep Sleep en la pantalla OLED del collar inteligente



Asimismo, se comprobó que, al finalizar el periodo de suspensión, el microcontrolador ESP32 reanudó su funcionamiento de manera automática, reiniciando el ciclo de adquisición, transmisión y monitoreo sin requerir intervención externa.

Los resultados obtenidos durante las pruebas de campo evidenciaron que la estrategia de gestión energética implementada permitió reducir el tiempo de operación en estado activo, contribuyendo a disminuir el consumo energético y a incrementar la autonomía estimada del collar inteligente; esto se evidenciará en la Tabla 24.

4.6. Evaluación de la precisión de la geolocalización

Con el propósito de determinar el nivel de precisión alcanzado por el sistema de geolocalización, se realizó una serie de pruebas en las que se comparó la información generada

por el prototipo con la ubicación mostrada en Google Maps. Esta evaluación permitió verificar si las coordenadas obtenidas por el módulo GPS representaban correctamente la posición del nodo transmisor y si la distancia calculada entre el transmisor (TX) y el receptor (RX) mantenía concordancia con la separación observada durante las pruebas de campo.

Se presenta un ejemplo en la Figura 35 sobre el mensaje recibido en Telegram durante una de las pruebas experimentales. En la captura puede observarse la información correspondiente a las coordenadas geográficas registradas por el GPS, la distancia estimada entre el transmisor y el receptor y el enlace generado automáticamente para consultar la ubicación en Google Maps.

Figura 35

Mensaje recibido en Telegram



Al acceder al enlace proporcionado por el sistema, Google Maps ubicó el punto geográfico correspondiente a las coordenadas transmitidas. Como se aprecia en la Figura 36, la posición representada coincide con el lugar donde se encontraba el dispositivo durante la ejecución de la prueba. Asimismo, la distancia visual obtenida mediante las herramientas de medición del mapa presentó una diferencia mínima respecto al valor calculado automáticamente por el prototipo.

Figura 36

Verificación de la ubicación reportada por el sistema mediante Google Maps



La comparación entre ambas evidencias permitió constatar que la información de posicionamiento enviada por el prototipo resulta consistente con la ubicación real del nodo transmisor. Las pequeñas variaciones observadas entre la distancia estimada por el sistema y la medida obtenida en Google Maps se consideran normales, ya que la precisión del GPS puede verse influenciada por factores como la geometría de los satélites disponibles, las condiciones ambientales, la presencia de obstáculos físicos y las características propias del receptor utilizado. A pesar de ello, durante las pruebas realizadas, las diferencias registradas fueron reducidas y no afectaron la capacidad del sistema para identificar la ubicación del animal dentro del área de monitoreo.

Tabla 21

Comparación de las distancias obtenidas mediante el prototipo y Google Maps

Distancia enviada (Telegram)	Distancia real (Google Maps)
50 metros	50 metros
150 metros	151 metros
300 metros	301 metros
450 metros	451 metros
600 metros	601 metros
750 metros	750 metros
850 metros	851 metros
950 metros	951 metros
1000 metros	1000 metros

La Tabla 21 muestra la comparación entre la distancia reportada por el sistema a través de la aplicación de Telegram y la distancia obtenida mediante Google Maps. Los resultados evidencian una alta correspondencia entre ambos valores; se observan diferencias máximas de 1 metro en ciertas mediciones. Estas pequeñas diferencias son atribuibles al margen de error esperado del sistema de posicionamiento GPS que se está utilizando, el cual puede verse influenciado por factores como, por ejemplo, la presencia de vegetación, árboles y otros obstáculos en el entorno, los cuales pueden causar atenuación en las señales que provienen de los satélites.

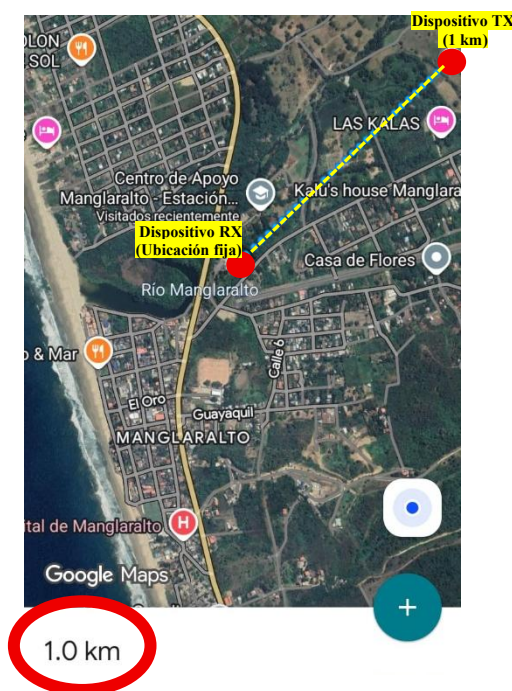
En términos generales, los resultados obtenidos demuestran que el prototipo proporciona una referencia espacial confiable para aplicaciones de seguimiento de ganado. La correspondencia entre las coordenadas transmitidas, la ubicación representada en Google Maps y la distancia

calculada entre los dispositivos constituye una evidencia de que el sistema mantiene un desempeño adecuado para las condiciones en las que fue diseñado, permitiendo localizar al animal con un nivel de precisión suficiente para apoyar las labores de vigilancia y control en zonas rurales.

A continuación, en la Figura 37 se muestra la verificación de esta distancia mediante la herramienta de medición de Google Maps, tomando como referencia la ubicación de ambos dispositivos

Figura 37

Medición de la distancia máxima entre el dispositivo transmisor y receptor



En cada prueba que se realizó, las coordenadas fueron recibidas correctamente por el dispositivo receptor y posteriormente enviadas a las plataformas de monitoreo, lo cual permitió verificar el correcto funcionamiento del enlace LoRa bajo las condiciones del entorno de prueba.

A medida que la distancia aumentaba entre ambos dispositivos, el sistema conservó la capacidad de transmitir la información de geolocalización sin interrupciones apreciables hasta el límite de cobertura alcanzado durante las pruebas.

4.7. Evaluación del alcance del sistema de comunicación LoRa

Las pruebas de alcance del sistema de comunicación LoRa se realizaron en la sede Manglaralto de la Universidad Estatal Península de Santa Elena, ubicada en un entorno completamente rural. El área de evaluación presenta vegetación natural compuesta por árboles y arbustos, elementos que representan el ambiente donde normalmente se encuentra el ganado.

Se utilizó un bloque académico de dos plantas como punto fijo de recepción. El dispositivo receptor fue instalado sobre la terraza del edificio y fijado sobre una estructura de soporte de caña, con el propósito de aumentar la altura de la antena y reducir la influencia de los obstáculos cercanos sobre la propagación de la señal. Durante esta prueba, el dispositivo transmisor fue desplazado progresivamente alrededor del área, aumentando la distancia respecto al receptor hasta alcanzar un límite de comunicación del sistema.

Antes de iniciar las pruebas de alcance, el collar inteligente fue instalado en el bovino para evaluar el funcionamiento del sistema en un entorno real. La Figura 38 evidencia la implementación del prototipo durante las pruebas.

Figura 38

Implementación del collar inteligente en el bovino durante las pruebas experimentales de alcance



4.7.1. Análisis del alcance

Una vez fijado el dispositivo receptor, se evaluó el alcance del enlace de comunicación LoRa mediante una serie de pruebas realizadas a diferentes distancias entre ambos dispositivos. Durante cada distancia se verificó la correcta recepción de las coordenadas geográficas transmitidas por el dispositivo transmisor (collar), permitiendo identificar el comportamiento del sistema conforme aumentaba la separación entre ambos dispositivos. Los resultados se presentan a continuación:

Tabla 22

Resultados de la evaluación experimental del alcance del enlace de comunicación LoRa

Punto	Distancia (m)	Coordenadas recibidas	Satélites	Estado de enlace
1	50	-1.845250, -80.743767	11	✓
2	150	-1.843999, -80.742981	12	✓
3	300	-1.843330, -80.741707	10	✓
4	450	-1.842178, -80.741501	12	✓
5	600	-1.841524, -80.740160	11	✓
6	750	-1.841320, -80.739066	11	✓
7	850	-1.841272, -80.737977	10	✓
8	950	-1.840688, -80.737446	11	✓
9	1000	-1.840567, -80.736794	11	✓
10	1010	--	--	X

Los resultados que se obtuvieron y se presentan en la Tabla 22 evidencian que el sistema mantuvo una comunicación estable durante todos los puntos evaluados hasta alcanzar una distancia aproximada de 1km.

Aunque la tecnología LoRa puede alcanzar distancias de hasta 5 km, esto depende de las condiciones del entorno y la configuración de los parámetros de comunicación establecidos. Durante las pruebas realizadas, el prototipo logró mantener una comunicación estable aproximadamente de 1 km. Esta diferencia respecto al alcance teórico puede deberse al exceso de

vegetación, árboles, y otros obstáculos en el área de la prueba, lo que puede provocar atenuación de la señal y minimizar el alcance efectivo del enlace de comunicación.

4.8. Estimación del consumo energético y autonomía del collar inteligente

Como parte de la evaluación del desempeño energético del prototipo, se realizó una estimación del consumo energético durante la operación activa del collar inteligente. Este análisis se desarrolló considerando el funcionamiento del sistema sin la implementación del modo Deep Sleep, con el propósito de establecer una referencia inicial de la autonomía que puede ofrecer el dispositivo cuando todos los módulos necesarios para la adquisición, procesamiento y transmisión de la información permanecen en funcionamiento.

Para esta estimación se tomó en consideración el consumo de corriente del ESP32, el módulo GPS NEO-6M, el transceptor LoRa durante la transmisión de datos y la pantalla OLED.

Tabla 23

Consumo de corriente de los módulos electrónicos considerados en la estimación teórica

Componente	Corriente (mA)	Condición de operación
ESP32	85	Procesamiento activo
GPS NEO-6M	40	Seguimiento de satélites
SX1262	130	Transmisión LoRa
Pantalla OLED	20	Pantalla encendida
TOTAL	275	Operación continua

En la tabla 23 se muestran las corrientes de operación consideradas para cada uno de los módulos electrónicos que integran el collar inteligente. Para la estimación del consumo energético se tomaron como referencia las especificaciones técnicas publicadas por los fabricantes del ESP32 (ESPRESSIF, s.f.), del módulo GPS NEO-6M (U-BLOX, s.f.), del transceptor SX1262 (SEMTECH, s.f.) y la pantalla OLED SSD1306 (SOLOMON SYSTECH, s.f.).

A partir de estos valores se determinó el consumo total del sistema en estado activo, cuya expresión se presenta en la Ecuación.

$$I_T = I_{ESP32} + I_{GPS\ NEO6M} + I_{tx} + I_{pantalla}$$

$$I_T = 85\ mA + 40\ mA + 130\ mA + 20\ mA$$

$$I_T = 275\ mA$$

Con el consumo de corriente obtenido, se estimó el tiempo de funcionamiento utilizando una batería Litio con una capacidad nominal de 3000 mAh. La autonomía se calculó mediante la relación entre la capacidad disponible de la batería y la corriente demandada por el sistema durante su operación activa, como se muestra en la Ecuación.

$$Autonomia = \frac{Capacidad\ total}{Corriente\ total}$$

$$Autonomia = \frac{3000\ mAh}{275\ mA}$$

$$Autonomia = \mathbf{10.9\ horas}$$

El resultado obtenido representa una estimación teórica para un escenario de operación continua, en el que el collar permanece en estado activo durante todo su ciclo de funcionamiento, es decir, realizando la adquisición de la ubicación mediante el GPS, el procesamiento de la

información en el ESP32, la transmisión continua de los datos mediante el módulo LoRa y la visualización de la información en la pantalla OLED. Bajo estas condiciones, para efectos de la estimación se asumió un consumo promedio constante de los módulos considerados durante la operación continua del sistema.

La autonomía obtenida corresponde a una estimación teórica bajo una condición de operación continua, en la que el collar permanece activo durante todo el ciclo de adquisición de la ubicación, procesamiento de la información y transmisión de los datos. Este cálculo proporciona una referencia inicial del comportamiento energético del prototipo sin considerar la implementación de estrategias orientadas a la reducción del consumo de energía. En consecuencia, el valor obtenido representa el tiempo aproximado de funcionamiento que puede esperarse cuando el sistema opera de manera seguida utilizando la capacidad nominal de la batería.

Si bien esta estimación permite establecer un punto de partida para analizar la autonomía del collar inteligente, durante el desarrollo del prototipo se implementó posteriormente el modo Deep Sleep con el propósito de reducir el consumo energético entre cada ciclo de funcionamiento. Con el fin de evaluar el efecto de esta estrategia sobre la duración de la batería, se realizó una prueba experimental bajo las condiciones reales de operación del sistema.

4.9. Evaluación experimental de la autonomía con el modo Deep Sleep

Con el propósito de evaluar el efecto del modo Deep Sleep sobre la autonomía del prototipo, se realizó una prueba experimental utilizando una batería Litio con una capacidad de 3000 mAh. A diferencia de la estimación teórica presentada anteriormente, en esta evaluación el collar operó bajo las condiciones reales de funcionamiento implementadas en el sistema, donde el

dispositivo permanece en estado de bajo consumo entre cada ciclo de adquisición y transmisión de la información.

En esta configuración, el ESP32 ingresa al modo Deep Sleep durante un intervalo de 2 minutos; se va reduciendo significativamente el consumo energético mientras el sistema permanece inactivo. Esta prueba se realizó de manera continua hasta que se agotó la batería, registrando el tiempo total de funcionamiento.

Tabla 24

Resultados de la prueba experimental de autonomía con Deep Sleep

Tiempo transcurrido	Hora	Capacidad restante	Batería	Estado del sistema
0 h	Viernes 14h24	3000 mAh	100 %	Inicio de la prueba
4 h	Viernes 18h24	2571 mAh	85.7 %	Funcionamiento normal
8 h	Viernes 22h24	2143 mAh	71.4 %	GPS, LoRa y Deep Sleep operando correctamente
12 h	Sábado 02h24	1714 mAh	57.1 %	Operación estable del sistema
16 h	Sábado 06h24	1286 mAh	42.9 %	Comunicación y adquisición de datos sin incidencias
20 h	Sábado 10h24	857 mAh	28.6 %	Funcionamiento continuo del prototipo
24 h	Sábado 14h24	429 mAh	14.3 %	Se mantiene la transmisión de datos y el funcionamiento del GPS

26 h	Sábado 16h24	214 mAh	7.1 %	Se presentan los primeros indicios de disminución de voltaje de alimentación
28 h	Sábado 17h30	≈0 mAh	0 %	Finalización de la prueba por agotamiento de la batería

La Tabla 24 presenta el comportamiento de la batería durante la prueba experimental de autonomía realizada con el modo Deep Sleep. La evaluación inició un viernes a las 14h24 y finalizó el siguiente sábado a las 17h30, registrando un tiempo total de funcionamiento aproximado de 28 horas.

Los resultados obtenidos muestran que la incorporación del modo Deep Sleep permitió reducir el consumo promedio de energía del sistema, incrementando considerablemente la autonomía de la batería respecto a la estimación teórica obtenida.

Durante la mayor parte de la prueba, el sistema mantuvo un funcionamiento estable, conservando la comunicación entre ambos dispositivos (transmisor y receptor), así como el constante envío de la ubicación hacia las plataformas de monitoreo. Únicamente durante la etapa final de la descarga se observaron comportamientos asociados a la disminución del voltaje de alimentación, tales como la pérdida de comunicación del dispositivo receptor y la dificultad del módulo GPS del dispositivo transmisor para mantener la detección de los satélites, indicando que la batería había alcanzado su límite de descarga útil.

La autonomía experimental obtenida demuestra que la implementación del modo Deep Sleep disminuye el consumo energético, permitiendo extender el tiempo de funcionamiento continuo. Este resultado confirma la efectividad de la estrategia de ahorro de energía implementada

y valida su aplicación en sistemas de monitoreo de ganado que requieren largos periodos de operación.

La Tabla 25 muestra un resumen de la autonomía que se obtuvo por el prototipo bajo dos condiciones de funcionamiento: operación continua, es decir, sin aplicar Deep Sleep, y operación con la estrategia para el ahorro de energía aplicando Deep Sleep. Como se puede observar, la implementación de este modo permitió incrementar la duración de la batería de 10.9 horas a 28 horas.

Tabla 25

Resumen de la autonomía del prototipo con y sin Deep Sleep

Duración sin Deep Sleep	10.9 horas (654 minutos)
(Calculado)	
Duración con Deep Sleep	28 horas
(Práctico)	

4.10. Evaluación del desempeño del sistema de comunicación

Con el propósito de evaluar la confiabilidad del sistema de comunicación implementado en el prototipo, se analizó el comportamiento de la transmisión de datos durante la prueba experimental de autonomía. Esta evaluación permitió verificar la capacidad del sistema para mantener el intercambio de información entre el dispositivo transmisor y el dispositivo receptor durante el periodo de funcionamiento continuo, así como la entrega de coordenadas geográficas hacia la plataforma de Telegram.

Durante la prueba se registró el número total de transmisiones realizadas por el dispositivo transmisor y se comparó con la cantidad de mensajes recibidos y procesados correctamente por el dispositivo receptor. A partir de esta información fue posible determinar la tasa de recepción y el porcentaje de pérdida de datos, parámetros que permiten evaluar el desempeño del enlace de comunicación.

Con base en los resultados que se obtuvieron durante la prueba experimental, se realizó el cálculo del número de mensajes recibidos correctamente, así como del porcentaje de recepción y de pérdida de datos. Estos indicadores permiten cuantificar el desempeño del sistema de comunicación y evaluar la confiabilidad de la transmisión de la información entre ambos dispositivos durante el periodo de funcionamiento continuo. Las ecuaciones empleadas para obtener estos parámetros se presentan a continuación:

Ecuación 15

Datos recibidos correctamente

$$Datos\ recibidos = Datos\ transmitidos - Datos\ perdidos$$

$$Datos\ recibidos = 553 - 17$$

$$Datos\ recibidos = 536$$

La Ecuación 15 muestra el total de datos recibidos (ubicaciones) por el dispositivo receptor, porque durante la prueba experimental se registraron un total de 553 transmisiones realizadas por el dispositivo transmisor; sin embargo, de ese total, 17 no fueron recibidas por el dispositivo receptor.

Ahora, tomando en cuenta los valores de los datos transmitidos y los datos recibidos, se reemplazan los valores en la Ecuación 16, en la cual se muestra el porcentaje de recepción, es decir, el porcentaje de funcionamiento y fiabilidad en cuanto a la recepción de datos.

Ecuación 16

Porcentaje de recepción

$$Recepcion(\%) = \frac{Datos\ recibidos}{Datos\ transmitidos} \times 100$$

$$Recepcion(\%) = \frac{536}{553} \times 100$$

$$Recepcion(\%) = 96.93\%$$

Por último, en la Ecuación 17 se muestra el porcentaje de pérdida, el cual se determinó considerando la cantidad de transmisiones que no fueron recibidas correctamente durante la prueba.

Ecuación 17

Tasa de error

$$Tasa\ de\ error(\%) = \frac{Datos\ perdidos}{Datos\ transmitidos} \times 100$$

$$Tasa\ de\ error(\%) = \frac{17}{553} \times 100$$

$$Tasa\ de\ error(\%) = 3.07\%$$

Los resultados obtenidos mediante las ecuaciones expresadas anteriormente permitieron determinar los indicadores utilizados para evaluar el desempeño del sistema de comunicación durante la prueba experimental. En la Tabla 26 se muestra de forma resumida:

Tabla 26*Resultados de la transmisión de datos durante la prueba experimental*

Parámetros	Valor
Duración de la batería	28 horas
Datos esperados	553
Datos recibidos	536
Datos no recibidos	17
Tasa de recepción	96.93%
Tasa de pérdida	3.07%

En la Tabla 26 se observa que, durante la prueba, el dispositivo transmisor realizó un total de 553 transmisiones; sin embargo, solo 536 fueron recibidas y procesadas por el dispositivo receptor, es decir, 17 no se recibieron. Esto resulta con un porcentaje de recepción del 96.93% y un porcentaje de pérdida de datos del 3.07%, lo cual evidencia un funcionamiento estable del enlace de comunicación durante la mayor parte del periodo de evaluación.

Las pérdidas que se registraron fueron eventuales, es decir, no comprometieron el funcionamiento general del sistema, ya que posteriormente la comunicación se restableció de forma automática y continua con el envío de coordenadas geográficas.

CONCLUSIONES

Se implementó favorablemente un sistema de comunicación para la transmisión de datos geográficos del ganado, basado en tecnología LoRa, de esta forma se logró una comunicación estable hasta una distancia aproximadamente de 1 km en un entorno completamente rural con presencia de vegetación. Durante las pruebas de campo, el enlace recibió correctamente 536 datos de ubicación de 553 que fueron transmitidos en total, teniendo una efectividad del 96,93% y una pérdida del 3,07%; estos resultados evidencian que su funcionamiento es confiable para aplicaciones de monitoreo en zonas rurales.

El prototipo del collar integró de manera adecuada el módulo GPS, un módulo LoRa el cual incluye un microcontrolador ESP32, lo cual permitió obtener las coordenadas geográficas válidas en las pruebas realizadas. El módulo GPS detectó entre 10 y 12 satélites, alcanzando tiempos de adquisición entre 19 y 23 segundos, esto aseguró la obtención de coordenadas válidas para su transmisión mediante el sistema.

Se logró sincronizar la información de geolocalización con las plataformas de monitoreo implementadas, permitiendo recibir notificaciones automáticas a través de la aplicación de Telegram, visualizar la ubicación del ganado mediante Blynk y acceder al enlace directo de Google Maps. Además, también se mostró evidencia de que la comparación entre la distancia calculada por el prototipo y la que mide Google Maps presentó una diferencia de 1 metro; esto demostró que el sistema es confiable para tareas de seguimiento y vigilancia del ganado.

Las pruebas de autonomía de la batería evidenciaron que la implementación del modo Deep Sleep mejoró notablemente el rendimiento energético del collar inteligente. En condiciones de operación continua, la autonomía teórica estimada fue de 10,9 horas (654 minutos), tomando en

cuenta que se utilizó una batería Lithium de 3000 mAh, sin embargo, al implementar la estrategia de ahorro de energía, la autonomía de la prueba realizada registró un tiempo de funcionamiento de 28 horas. Estos resultados confirman la efectividad de la estrategia implementada para extender el tiempo de funcionamiento del sistema en aplicaciones de monitoreo.

RECOMENDACIONES

Durante el ensamblaje del dispositivo transmisor (collar), se recomienda que las conexiones eléctricas se las realice cuidadosamente y, además, verificar que los conductores se encuentren correctamente aislados para evitar cortocircuitos, lo cual puede afectar el funcionamiento de los módulos electrónicos o provocar daños en los mismos.

Debido a que, durante las pruebas, el módulo GPS presentó dificultades aproximadamente a las 9 horas de funcionamiento continuo para mantener la comunicación con los satélites. Se recomienda presionar el botón RST del dispositivo, esto permitió que el GPS restablezca la detección de satélites y continúe funcionando con normalidad. Otra alternativa sería implementar un reinicio automático, tal vez cada 8 horas, en el programa del microcontrolador.

Para aplicaciones que requieran una buena autonomía de la batería, es recomendable mantener la implementación del modo Deep Sleep y ajustar el tiempo tanto de suspensión como de activación del dispositivo de acuerdo con las necesidades de monitoreo. Los resultados alcanzados demostraron que esta implementación sí reduce notoriamente el consumo energético y prolonga el tiempo de operación del collar.

Se recomienda incorporar mecanismos de protección eléctrica para futuras versiones del collar, pueden ser sistemas de protección contra descargas electrostáticas o sobretensiones, esto con el objetivo de minimizar el riesgo de daños en los componentes electrónicos que se utilizan y, además, mejora la confiabilidad del dispositivo durante su operación en entornos rurales.

Para futuras versiones del prototipo, se recomienda que la carcasa sea diseñada y validada para alcanzar un grado de protección IP65 o superior, con el fin de mejorar la resistencia al ingreso

de polvo y agua, permitiendo un funcionamiento más confiable en ambientes rurales donde el ganado está expuesto a lluvia, tierra y humedad.

BIBLIOGRAFÍA

ardumotica.cl. (s.f.). *ardumotica.cl*. Obtenido de <https://ardumotica.cl/producto/modulo-gps-ublox-neo-6m/>

AutoCore. (s.f.). *Antena LoRa 915Mhz 2dBi con Pigtail*. Obtenido de <https://www.autocorerobotica.com.br/antena-lora-915mhz-2dbi-com-pigtail>

B, E. (s.f.). *Lora*. Obtenido de Lora: <https://lora.readthedocs.io/en/latest/>

BLYNK. (s.f.). *Componentes de la plataforma Blynk IoT*. (BLYNK) Recuperado el 2025, de <https://docs.blynk.io/en/>

Cabalceta, D. L. (22 de 10 de 2025). *32 mil unidades de ganado se perdieron por enfermedades*. Obtenido de Diario EXTRA: <https://www.diarioextra.com/noticia/32-mil-unidades-de-ganado-se-perdieron-por-enfermedades/>

Chinchilla Ballesteros, F. M., & Yepes Castillo , J. A. (s.f.). *repositorio.uts*. Obtenido de <http://repositorio.uts.edu.co:8080/xmlui/bitstream/handle/123456789/1067/8RDC%2095%20final.pdf?sequence=1&isAllowed=y>

COMESTAR. (24 de Julio de 2025). *MQTT vs HTTP: Diferencias clave, rendimiento en IoT y cómo elegir*. Obtenido de <https://www.come-star.com/es/blog/mqtt-vs-http-key-differences-performance-in-iot-and-how-to-choose/>

ELECTRICAL ENGINEERING. (Marzo de 2020). *Understanding the relationship between LoRa chips, chirps, symbols and bits*. Recuperado el 2025, de

<https://electronics.stackexchange.com/questions/278192/understanding-the-relationship-between-lora-chirps-symbols-and-bits>

Electronicos Caldas. (s.f.). *SW915-TH12*. Obtenido de

<https://www.electronicoscaldas.com/en/antenas/1347-antena-helicoidal-de-915-mhz-sw915-th12.html>

ElectronicWings. (s.f.). *ESP32 Wi-Fi Basics Getting Started*. Obtenido de

<https://www.electronicwings.com/esp32/esp32-wi-fi-basics-getting-started>

ESPRESSIF. (s.f.). *ESP32 Series Datasheet v5.2*. Obtenido de ESPRESSIF:

https://documentation.espressif.com/esp32_datasheet_en.pdf?

ESPRESSIF. (s.f.). *ESPRESSIF*. Obtenido de

https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

Espressif Systems (Shanghai). (s.f.). *Analog to Digital Converter (ADC)*. Obtenido de

<https://docs.espressif.com/projects/esp-idf/en/v4.4/esp32/api-reference/peripherals/adc.html>

ESPRESSIF SYSTEMS. (2020). *ESP32 Technical Reference Manual 5.4*. Obtenido de

https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

GABRI. (24 de mayo de 2018). *La precisión del GPS ¿cómo funciona y cuáles son sus errores?*

Obtenido de ArcGeek: <https://acolita.com/la-precision-del-gps-como-funciona-y-cuales-son-sus-errores/>

Giesecke, B. (3 de Abril de 2024). *Decodificación de la tecnología LoRa: comprensión de las diferencias entre LoRaWAN®, LoRa P2P y LoRa Mesh*. Obtenido de RAK:

<https://news.rakwireless.com/decoding-lora-technology-understanding-the-differences-between-lorawan-r-lora-p2p-and-lora-mesh-2/>

HELTEC AUTOMATION. (s.f.). *HELTEC AUTOMATION*. Obtenido de uPesy:

<https://heltec.org/project/wifi-lora-32-v4/>

Heltec Automation. (s.f.). *WiFi LoRa 32(V3), ESP32S3 + SX1262 LoRa Node, Meshtastic and LoRaWAN Compatible*. Obtenido de <https://heltec.org/project/wifi-lora-32-v3/>

HIVEMQ. (s.f.). *FUNDAMENTOS DE MQTT*. Obtenido de <https://www.hivemq.com/mqtt/>

Huang, Y. (21 de Octubre de 2019). *¿Qué es la tecnología LoRa y cómo funciona? – Una guía detallada*. Obtenido de MOKO SMART: <https://www.mokosmart.com/es/lora-technology/>

ieeexplore. (25 de Julio de 2024). *A Tutorial on Chirp Spread Spectrum Modulation for LoRaWAN: Basics and Key Advances*. Obtenido de ieeexplore:

<https://ieeexplore.ieee.org/document/10609524>

impl Rust for ESP32. (s.f.). *ADC (Analog to Digital Converter)*. Obtenido de impl Rust for ESP32: <https://esp32.implrust.com/core-concepts/adc/index.html?>

KWKWII. (14 de agosto de 2018). *Factor de ensanchamiento, ancho de banda, velocidad de codificación y velocidad de bits en LoRa (inglés)*. (Josef Matondang) Recuperado el 2025, de <https://josefmttd.com/2018/08/14/spreading-factor-bandwidth-coding-rate-and-bit-rate-in-lora-english/>

La Hora. (12 de Noviembre de 2024). *La organización de ganaderos ha contrarrestado el abigeato en Tulcán*. Obtenido de La Hora: <https://www.lahora.com.ec/imbabura-carchi/organizacion-ganaderos-tulcan-permite-contrarrestar-abigeato/>

LoRa Alliance. (2024). *LoRaWAN®*. Obtenido de <https://lora-alliance.org/wp-content/uploads/2021/11/LoRaWAN-Link-Layer-Specification-v1.0.4.pdf>

Lucas Aguilar, E. A., & Lucas Aguilar, J. E. (2023). *dspace.espol*. Obtenido de dspace.espol: <https://www.dspace.espol.edu.ec/bitstream/123456789/58222/1/T-113572%20Lucas%20-%20Lucas%20.pdf>

Manuals+. (s.f.). *Manual de instrucciones de los módulos LoRa WAVESHARE SX1262*. Obtenido de <https://manuals.plus/es/waveshare/sx1262-lora-modules-manual>

mikalhart. (20 de Septiembre de 2022). *TinyGPSPlus*. (GitHub) Recuperado el Agosto de 2025, de <https://github.com/mikalhart/TinyGPSPlus>

MOKO LORA. (17 de Agosto de 2021). *Comprensión completa de LoRa y LoRaWAN*. Obtenido de MOKO LORA: <https://www.mokolora.com/es/full-understanding-of-lora-and-lorawan/>

Moreno, J. A. (11 de Julio de 2023). *Sesión 3: LoRa/LoRaWAN*. Obtenido de hackmd.io: <https://hackmd.io/@LF0FqZwDTDWM-MMK06IEWA/S1ebk4ZPn>

OASIS. (s.f.). *OASIS*. Obtenido de <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>

Paz Leon, P. J., & Yagual Estacio, D. J. (2023). *DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA DE RASTREO Y SEGURIDAD DE GANADDO EN ZONAS RURALES VIA LORA*. Guayaquil.

Pires, L., & Martins, J. (05 de Noviembre de 2024). *Prerpints.org*. Obtenido de Preprints.org:

<https://www.preprints.org/manuscript/202411.0254/v1>

profwhitetower. (Enero de 2025). *[ESP/ENG] Módulo GPS NEO-6M para Localización*

Geográfica NEO-6M GPS Module for Geographic Location . (HIVE BLOG) Recuperado

el 14 de Agosto de 2025, de [https://hive.blog/stem-espanol/@profwhitetower/esp-eng-](https://hive.blog/stem-espanol/@profwhitetower/esp-eng-modulo-gps-neo-6m-para-localizacion-geografica-neo-6m-gps-module-for-geographic-location)

[modulo-gps-neo-6m-para-localizacion-geografica-neo-6m-gps-module-for-geographic-](https://hive.blog/stem-espanol/@profwhitetower/esp-eng-modulo-gps-neo-6m-para-localizacion-geografica-neo-6m-gps-module-for-geographic-location)

[location](https://hive.blog/stem-espanol/@profwhitetower/esp-eng-modulo-gps-neo-6m-para-localizacion-geografica-neo-6m-gps-module-for-geographic-location)

Rivadeneira, B. M. (06 de 12 de 2019). *Prototipo de red LoRaWAN para solución IoT*. Obtenido

de <https://hackmd.io/vUoy9APrS3OCS6Nkyrvpkg#Desarrollo>

SANCHEZ, C. A. (2021). *PROTOCOLO LORA PARA ANALISIS DE MEDICION CON GPS Y*

ARDUINO EN LA INDUSTRIA GANADERA DEL ECUADOR. GUAYAQUIL.

Savoie Mont Blanc University . (2022). *lora-alliance.org*. Obtenido de [https://lora-](https://lora-alliance.org/wp-content/uploads/2024/01/Book-LoRa-LoRaWAN-and-Internet-of-Things.pdf)

[alliance.org/wp-content/uploads/2024/01/Book-LoRa-LoRaWAN-and-Internet-of-](https://lora-alliance.org/wp-content/uploads/2024/01/Book-LoRa-LoRaWAN-and-Internet-of-Things.pdf)

[Things.pdf](https://lora-alliance.org/wp-content/uploads/2024/01/Book-LoRa-LoRaWAN-and-Internet-of-Things.pdf)

Securitylab.lab. (04 de 04 de 2025). *securitylab.lab*. Obtenido de securitylab.lab:

<https://www.securitylab.lat/news/575017.php>

SEMTECH. (s.f.). *SEMTECH*. Obtenido de

https://www.mouser.com/datasheet/2/761/DS_SX1261-2_V1.1-1307803.pdf

SEMTECH. (s.f.). *SX1262*. Obtenido de SEMTECH:

<https://www.semtech.com/products/wireless-rf/lora-connect/sx1262?>

- SOLOMON SYSTECH. (s.f.). *SSD1306*. Obtenido de SOLOMON SYSTECH: <https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf?>
- SparkFun Electronics. (s.f.). *NEO-6M GPS Breakout Hookup Guide*. Obtenido de <https://learn.sparkfun.com/tutorials>
- tamiranashenyungu. (s.f de s.f). Obtenido de <https://es.scribd.com/document/895837084/Battery-Sizing-and-Power-Supply-Sizing-for-LoRa>
- TE Connectivity. (s.f.). *Antennas*. Obtenido de <https://www.te.com/en/products/brands/linux-technologies.html>
- Thanksbuyer. (s.f.). *WIFI Bluetooth Module Dual Core Xtensa 32bit LX6 CPU Network Port 520K RAM ESP32S for IOT*. Obtenido de <https://www.thanksbuyer.com/wifi-bluetooth-module-dual-core-xtensa-32bit-lx6-cpu-network-port-520k-ram-esp32s-for-iot-52406>
- Tippannavar, S. S. (Marzo de 2023). *ResearchGate GmbH*. Obtenido de https://www.researchgate.net/publication/369689572_Smart_Collar_for_Cattle_Tracking_and_Health_Monitoring_using_IOT_on_ESP32
- Torregroza, A. D. (23 de Abril de 2024). *Implementación de tecnología LoRa para mejorar la fiabilidad de la transmisión de datos en aplicaciones de monitoreo energético*. Obtenido de repositorio.cuc: <https://repositorio.cuc.edu.co/server/api/core/bitstreams/a4f90397-a7f4-4648-9570-0cdc48045cac/content>
- Tulka, T. (22 de 07 de 2022). *Explicación del factor de propagación LoRa*. Obtenido de ttulka: <https://blog.ttulka.com/lora-spreading-factor-explained/>

U-BLOX. (s.f.). *NEO-6 - Data Sheet*. Obtenido de U-BLOX: https://content.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf?

ublox. (s.f.). *ublox*. Obtenido de <https://www.u-blox.com/en/product/neo-6-series>

UNIT Electronics. (s.f.). *Bateria LiPo 3.7V 2500mAh 104050*. Obtenido de <https://uelectronics.com/producto/bateria-lipo-3-7v-2500mah/>

uPesy Electronics. (s.f.). *Using UART and GPS modules with ESP32*. Obtenido de <https://www.upesy.com/>

World Sensing. (Enero de 2020). *Explicación de LoRa y razones clave para su uso*. (World Sensing) Recuperado el 2025, de <https://www.worldsensing.com/es/news/explicacion-de-lora-y-razones-clave-para-su-uso/>

Zhou, L. (9 de abril de 2025). *Clasificación IP: diferencias entre IP65, IP66, IP67*. Obtenido de <https://www.eabel.com/es/diferencias-de-clasificacion-ip-entre-ip65-ip66-e-ip67-2/>

ANEXOS

Anexo A

Código de dispositivo receptor

```

1  #include <SPI.h> // Librerías para LoRa, WiFi, Telegram, Blynk y pantalla OLED
2  #include <RadioLib.h>
3  #include <Wire.h>
4  #include <WiFi.h>
5  #include <WiFiClientSecure.h>
6  #include <UniversalTelegramBot.h>
7  #include <ArduinoJson.h>
8  #include <Adafruit_GFX.h>
9  #include <Adafruit_SSD1306.h>
10
11 #define BLYNK_TEMPLATE_ID "TMPL2fEqbXWnl" // Credenciales del proyecto Blynk
12 #define BLYNK_TEMPLATE_NAME "Collar01"
13 #define BLYNK_AUTH_TOKEN "4nYl6mZKwVjSihoQ10RuNcZSDs1g1nW-"
14 #include <BlynkSimpleEsp32.h>
15
16 #define SCREEN_WIDTH 128 // Configuración de la pantalla OLED
17 #define SCREEN_HEIGHT 64
18 #define SDA_PIN 17
19 #define SCL_PIN 18
20 #define OLED_POWER 21
21 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
22
23 char ssid[] = "CNT_PILLASAGUA"; // Credenciales de la red Wifi
24 char password[] = "2400124976fp";
25
26 #define BOT_TOKEN "8661594565:AAGXQm2RW8MqBt6YRDs16w-tnNVikzcXYsQ" // Token del bot y chat donde se enviarán las notificaciones
27 #define CHAT_ID "7081633636"
28 WiFiClientSecure client;
29 UniversalTelegramBot bot(BOT_TOKEN, client);
30
31 #define NSS 8 // Pines utilizados por el módulo SX1262
32 #define DIO1 14
33 #define RST 12
34 #define BUSY 13
35 SX1262 lora = new Module(NSS, DIO1, RST, BUSY);
36
37 String lastLat = ""; // Variables para almacenar la última ubicación y controlar alertas
38 String lastLon = "";
39 bool alertaDistanciaEnviada = false;
40 double latReferencia = -1.845495;
41 double lonReferencia = -80.743345;
42
43 double calcularDistancia(double lat1, double lon1, double lat2, double lon2) { // Calcula la distancia entre dos coordenadas
44   const double R = 6371000.0; // Radio de la Tierra en metros
45   double dLat = radians(lat2 - lat1);
46   double dLon = radians(lon2 - lon1);
47   double a =
48     sin(dLat / 2) * sin(dLat / 2) +
49     cos(radians(lat1)) * cos(radians(lat2)) *
50     sin(dLon / 2) * sin(dLon / 2);
51   double c = 2 * atan2(sqrt(a), sqrt(1 - a));
52   return R * c;
53 }
54 void setup() {
55   Serial.begin(115200); // Inicializa el monitor serial
56   delay(2000);
57   pinMode(OLED_POWER, OUTPUT); // Enciende e inicializa la pantalla OLED
58   digitalWrite(OLED_POWER, HIGH);
59   delay(300);
60   Wire.begin(SDA_PIN, SCL_PIN);
61   if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
62     Serial.println("OLED no encontrado");
63     while(true);
64   }
65   display.ssd1306_command(SSD1306_SETCONTRAST);
66   display.ssd1306_command(255);
67   display.clearDisplay();
68   display.setTextSize(1);
69   display.setTextColor(SSD1306_WHITE);
70   display.setCursor(0,10);
71   display.println("RECEPTOR LORA");
72   display.println("Iniciando...");
73   display.display();
74   delay(2000);
75   WiFi.begin(ssid, password); // Conecta el receptor a la red Wifi

```

```

76 while (Wifi.status() != WL_CONNECTED) {
77     delay(500);
78     Serial.print(".");
79     display.clearDisplay();
80     display.setCursor(0,10);
81     display.println("Conectando WiFi");
82     display.display();
83 }
84 Serial.println("WiFi conectado");
85 display.clearDisplay();
86 display.setCursor(0,10);
87 display.println("WiFi conectado");
88 display.display();
89 delay(1000);
90 Blynk.config(BLYNK_AUTH_TOKEN); // Establece la conexión con la plataforma Blynk
91 bool conectado = Blynk.connect(10000);
92 if(conectado){
93     Serial.println("Blynk conectado");
94     display.clearDisplay();
95     display.setCursor(0,10);
96     display.println("Blynk conectado");
97     display.display();
98 }
99 else{
100     Serial.println("Error Blynk");
101     display.clearDisplay();
102     display.setCursor(0,10);
103     display.println("Error Blynk");
104     display.display();
105 }
106 delay(2000);
107 client.setInsecure(); // Configura el cliente seguro para Telegram
108 SPI.begin(9, 11, 10, 8); // Inicializa el módulo LoRa
109 int state = lora.begin(915.0);
110 if(state != RADIOLIB_ERR_NONE){
111     Serial.println("Error LoRa");
112     display.clearDisplay();
113     display.setCursor(0,10);
114     display.println("ERROR LORA");
115     display.display();
116     while(true);
117 }
118 lora.setOutputPower(25); // Configura los parámetros de comunicación LoRa
119 lora.setSpreadingFactor(12);
120 lora.setBandwidth(125.0);
121 lora.setCodingRate(8);
122 lora.setCurrentLimit(140);
123 lora.setPreambleLength(12);
124 display.clearDisplay();
125 display.setCursor(0,10);
126 display.println("LoRa LISTO");
127 display.println("Esperando GPS...");
128 display.display();
129 }
130 void loop() {
131     Serial.print("WiFi: ");
132     Serial.println(Wifi.status());
133     Serial.print("Blynk: ");
134     Serial.println(Blynk.connected());
135     Blynk.run(); // Mantiene activa la conexión con Blynk
136     String mensaje; // Espera la recepción de un paquete LoRa
137     int state = lora.receive(mensaje);
138     if(state == RADIOLIB_ERR_NONE){
139         Serial.println("Paquete recibido:");
140         Serial.println(mensaje);
141         if(mensaje.startsWith("SLEEP,")){ // Verifica si el transmisor entró en modo Deep Sleep
142             int coma1 = mensaje.indexOf(','); // Extrae la latitud, longitud y número de satélites recibidos
143             int coma2 = mensaje.indexOf(',', coma1 + 1);
144             String wake = mensaje.substring(coma1 + 1, coma2);
145             String tiempoSleep = mensaje.substring(coma2 + 1);
146             Serial.println("Transmisor entro en Deep Sleep");
147
148             display.clearDisplay(); // Pantalla OLED
149             display.setCursor(0,0);
150             display.println("COLLAR SLEEP");
151             display.println("-----");
152             display.print("Wake:");
153             display.println(wake);
154             display.print("Tiempo:");
155             display.println(tiempoSleep);
156             display.println(" s");
157             display.display();
158
159             String texto = // Telegram
160                 "Collar GPS\n\n"
161                 "Transmission completada\n\n"
162                 "Wake #" + wake +
163                 "\n\nEntrando en Deep Sleep\n\n"
164                 "Duracion: " + tiempoSleep + " segundos";
165             bot.sendMessage(CHAT_ID, texto, "");
166             return;
167         }

```

```

168
169     int coma1 = mensaje.indexOf(',');// Calcula la distancia respecto al punto de referencia
170     int coma2 = mensaje.indexOf(',', coma1 + 1);
171     String lat = mensaje.substring(0, coma1);
172     String lon = mensaje.substring(coma1 + 1, coma2);
173     String satelites = mensaje.substring(coma2 + 1);
174     double latActual = lat.toDouble();
175     double lonActual = lon.toDouble();
176
177     double distancia = calcularDistancia(// Calcula la distancia respecto al punto de referencia
178     latReferencia,
179     lonReferencia,
180     latActual,
181     lonActual
182 );
183 if (distancia > 600.0) { // Envía una alerta si la distancia supera los 600 metros
184     if (!alertaDistanciaEnviada) {
185         String msgAlerta = "⚠ ALERTA DE SEGURIDAD\n";
186         msgAlerta += "El ganado ha superado los 600 metros de distancia.";
187         bot.sendMessage(CHAT_ID, msgAlerta, "");
188         alertaDistanciaEnviada = true;
189     }
190 } else {
191     alertaDistanciaEnviada = false; // Reinicia cuando el ganado vuelve al área permitida
192 }
193 String link = "https://www.google.com/maps?q=" + lat + "," + lon; // Genera el enlace de Google Maps
194 Serial.println(link);
195
196 Serial.print("Estado Blynk antes de enviar: "); // Verifica la conexión con Blynk y reconecta si es necesario
197 Serial.println(Blynk.connected());
198 if (!Blynk.connected()) {
199     Serial.println("Reconectando Blynk...");
200     if (Blynk.connect(10000)) {
201         Serial.println("Blynk reconectado");
202     }
203     else {
204         Serial.println("No se pudo conectar a Blynk");
205         return;
206     }
207 }
208
209 String distanciaTexto; // Envía las coordenadas y la distancia a Blynk
210 if (distancia < 1000) {
211     distanciaTexto = String(distancia, 0) + " m";
212 }
213 else {
214     distanciaTexto = String(distancia/1000.0, 2) + " km";
215 }
216 Blynk.virtualWrite(V0, lat);
217 Blynk.virtualWrite(V1, lon);
218 Blynk.virtualWrite(V2, link);
219 Blynk.virtualWrite(V3, distanciaTexto);
220 Blynk.run();
221     delay(200);
222     Blynk.run();
223     Serial.println("Datos enviados a Blynk");
224     Serial.print("Estado Blynk despues: ");
225     Serial.println(Blynk.connected());
226
227     display.clearDisplay(); // Muestra la información recibida en la pantalla OLED
228     display.setCursor(0,0);
229     display.println("UBICACION RECIBIDA");
230     display.println("-----");
231     display.print("Lat:");
232     display.println(lat);
233     display.print("Lon:");
234     display.println(lon);
235     display.print("Dist:");
236     if (distancia < 1000) {
237         display.print(distancia, 0);
238         display.println(" m");
239     }
240     else {
241         display.print(distancia / 1000.0, 2);
242         display.println(" km");
243     }
244     display.display();
245
246     if (lat != lastLat || lon != lastLon) { // Envía la ubicación a Telegram únicamente si hubo cambios
247         Serial.println("Ubicacion nueva -> enviando Telegram");
248         String texto = " Ubicacion del ganado\n\n";
249         texto += " Latitud: ";
250         texto += lat;
251         texto += "\n Longitud: ";
252         texto += lon;
253         texto += "\n Satélites: ";
254         texto += satelites;
255         texto += "\n\n Distancia desde el punto de referencia: ";
256         if (distancia < 1000) {
257             texto += String(distancia, 0);

```

```

258     texto += " m";
259 }
260 else{
261     texto += String(distancia / 1000.0, 2);
262     texto += " km";
263 }
264 texto += "\n\n Google Maps:\n";
265 texto += link;
266     bot.sendMessage(CHAT_ID, texto, "");
267
268     lastLat = lat; // Guarda la última ubicación recibida para evitar mensajes repetidos
269     lastLon = lon;
270     display.setCursor(0,50);
271     display.println("Telegram enviado");
272     display.display();
273 }
274 else{
275     Serial.println("Ubicacion igual");
276     display.setCursor(0,50);
277     display.println("Sin cambio GPS");
278     display.display();
279 }
280 }
281 }

```

Anexo B

Código de dispositivo transmisor (collar)

```

1  #include <SPI.h> // Librerías para LoRa, GPS, OLED y gestión del modo Deep Sleep
2  #include <RadioLib.h>
3  #include <Wire.h>
4  #include <TinyGPS++.h>
5  #include <HardwareSerial.h>
6  #include <Adafruit_GFX.h>
7  #include <Adafruit_SSD1306.h>
8  #include "esp_sleep.h"
9
10 #define SCREEN_WIDTH 128 // Configuración de la pantalla OLED y sus pines de comunicación
11 #define SCREEN_HEIGHT 64
12 #define SDA_PIN 17
13 #define SCL_PIN 18
14 #define OLED_POWER 21
15 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
16
17 TinyGPSPlus gps; // Inicializa el receptor GPS y define los pines de comunicación
18 HardwareSerial GPSSerial(1);
19 #define GPS_RX 39
20 #define GPS_TX 38
21
22 #define NSS 8 // Configuración de los pines utilizados por el módulo LoRa SX1262
23 #define DIO1 14
24 #define RST 12
25 #define BUSY 13
26
27 SX1262 lora = new Module(NSS, DIO1, RST, BUSY);
28
29 RTC_DATA_ATTR uint32_t wakeCount = 0; // Variable almacenada en memoria RTC para contar los despertares del ESP32
30
31 // Define los tiempos de reprogramación, búsqueda GPS y Deep Sleep
32 #define TIEMPO_REPROGRAMACION 30000 // 30 s
33 #define TIEMPO_BUSQUEDA_GPS 300000 // 5 min
34 #define TIEMPO_DEEP_SLEEP 120 // 2 min
35
36 void setup() {
37     Serial.begin(115200); // Inicializa la comunicación serial
38
39     delay(TIEMPO_REPROGRAMACION); // Espera unos segundos para permitir una nueva programación del ESP32
40     wakeCount++; // Incrementa el contador de activaciones después de cada despertar
41
42     pinMode(OLED_POWER, OUTPUT); // Inicializa la pantalla OLED y muestra el número de activación
43     digitalWrite(OLED_POWER, HIGH);
44     delay(300);
45     Wire.begin(SDA_PIN, SCL_PIN);
46     if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
47         Serial.println("OLED no encontrado");
48         while(true);
49     }
50     display.ssd1306_command(SSD1306_SETCONTRAST);

```

```

51 display.ssd1306_command(255);
52 display.clearDisplay();
53 display.setTextSize(1);
54 display.setTextColor(SSD1306_WHITE);
55 display.setCursor(0,0);
56 display.println("COLLAR GPS");
57 display.print("Wake #");
58 display.println(wakeCount);
59 display.display();
60 delay(2000);
61
62 GPSserial.begin(9600, SERIAL_8N1, GPS_RX, GPS_TX);// Inicializa el módulo GPS
63 gps = TinyGPSPlus();
64
65 SPI.begin(9, 11, 10, 8);// Inicializa el módulo LoRa
66 int state = lora.begin(915.0);
67 if(state != RADIOLIB_ERR_NONE){
68     display.clearDisplay();
69     display.setCursor(0,10);
70     display.println("ERROR LORA");
71     display.display();
72     while(true);
73 }
74
75 lora.setOutputPower(25);// Configura los parámetros de comunicación LoRa
76 lora.setSpreadingFactor(12);
77 lora.setBandwidth(125.0);
78 lora.setCodingRate(8);
79 lora.setCurrentLimit(140);
80 lora.setPreambleLength(12);
81 display.clearDisplay();// Muestra en la OLED el inicio de la búsqueda de la señal GPS
82 display.setCursor(0,0);
83 display.println("LoRa OK");
84 display.println("Buscando GPS");
85 display.display();
86 unsigned long gpsStart = millis();// Inicia el temporizador para limitar el tiempo de búsqueda del GPS
87
88 while ((millis() - gpsStart) < TIEMPO_BUSQUEDA_GPS) { // Lee continuamente los datos recibidos por el módulo GPS
89     while (GPSserial.available()) {
90         gps.encode(GPSserial.read());
91     }
92     uint32_t satellites = gps.satellites.value();// Obtiene el número de satélites detectados
93
94     if (gps.location.isValid() && satellites > 0) { // Verifica que exista una posición GPS válida antes de transmitir
95
96         float lat = gps.location.lat();// Obtiene las coordenadas geográficas actuales
97         float lon = gps.location.lng();
98         String mensaje = String(lat, 6) + "," + String(lon, 6) + "," + String(satellites);// Construye el paquete con lat, long y num de satélites
99         Serial.println("Enviando: " + mensaje);
100         state = lora.transmit(mensaje); // Envía la información GPS mediante comunicación LoRa
101
102         display.clearDisplay();// Muestra en la pantalla OLED la información transmitida
103         display.setCursor(0,0);
104         display.println("GPS ENVIADO");
105         display.print("SAT:");
106         display.println(satellites);
107         display.print("Lat:");
108         display.println(lat, 4);
109         display.print("Lon:");
110         display.println(lon, 4);
111         if(state == RADIOLIB_ERR_NONE){
112             display.println("TX OK");
113         } else {
114             display.println("TX ERROR");
115         }
116         display.print("Wake:");
117         display.println(wakeCount);
118         display.display();
119         delay(10000);
120     }
121     display.clearDisplay();// Informa en la pantalla OLED que el dispositivo entrará en Deep Sleep
122     display.setCursor(0,10);
123     display.println("DEEP SLEEP");
124     display.println("2 MINUTOS");
125     display.display();
126     delay(3000);
127
128     String sleepMsg = "SLEEP," +
129         String(wakeCount) + "," +
130         String(TIEMPO_DEEP_SLEEP);
131     lora.transmit(sleepMsg);// Envía el mensaje de Deep Sleep al receptor mediante LoRa
132     Serial.println("Mensaje Sleep enviado:");
133     Serial.println(sleepMsg);
134     delay(1000);
135     esp_sleep_enable_timer_wakeup((uint64_t)TIEMPO_DEEP_SLEEP * 1000000ULL);
136 }
137 esp_deep_sleep_start();// Coloca el ESP32 en modo Deep Sleep para reducir el consumo de energía
138 }

```

```

139     display.clearDisplay();// Actualiza la pantalla OLED con el estado de búsqueda del GPS
140     display.setCursor(0,0);
141     display.println("BUSCANDO GPS");
142     display.print("Wake:");
143     display.println(wakeCount);
144     display.print("SAT:");
145     display.println(satellites);
146     display.print("Seg:");
147     display.println((millis() - gpsStart)/1000);
148     display.display();
149     Serial.print("SAT: ");// Muestra por el monitor serial el número de satélites y el tiempo transcurrido
150     Serial.print(satellites);
151     Serial.print(" | Tiempo: ");
152     Serial.println((millis() - gpsStart)/1000);
153     Serial.println(" s");
154     delay(1000);
155 }
156 // Si no se obtiene una posición válida, informa el estado en la OLED
157 display.clearDisplay();
158 display.setCursor(0,0);
159 display.println("GPS NO FIX");
160 display.println("DEEP SLEEP");
161 display.display();
162 delay(5000);
163 esp_sleep_enable_timer_wakeup// Configura el temporizador para despertar automáticamente del modo Deep Sleep
164 | (uint64_t)TIEMPO_DEEP_SLEEP * 1000000ULL
165 );
166 esp_deep_sleep_start();// Coloca el ESP32 en modo Deep Sleep para reducir el consumo de energía
167 }
168
169 void loop() { // La lógica principal se ejecuta en setup() debido al funcionamiento con Deep Sleep
170 }

```

Anexo C

Dispositivo transmisor (collar)

|



Anexo D

Dispositivo receptor



Anexo E

Presupuesto de proyecto

Cantidad	Componente	Precio Unitario	Precio Total
2	ESP32 LoRa V4 placa de desarrollo + batería de 3000 mAh, ESP32-S3 SX1262 LoRa WiFi 2MB PSRAM 16MB Flash 915MHz Antena OLED Soporte GPS Solar para Arduino Meshtastic LoRaWAN	\$39	\$78
1	GPS NEO-6M	\$10	\$10
2	Carcasas impresas en 3D	\$17,5	\$35
1	Collar	\$1	\$1
TOTAL:			\$124

Anexo F

Autorización del ingreso al centro de apoyo Manglaralto



FACULTAD DE CIENCIAS AGRARIAS

CARRERA DE AGROPECUARIA

OF-CA-2026-081

La Libertad, 1 de julio de 2026

Ing. Carlos Efraín Andrade Caicho
DOCENTE DE LA CARRERA DE TELECOMUNICACIONES

Agr. Ángel Rodolfo León Mejía, Mgtr.
COORDINADOR DE LOS CENTROS DE APOYO DE LA UPSE

MVZ. Joffre Javier Masaquiza Aragón, Mgtr.
DOCENTE DE LA FACULTAD DE CIENCIAS AGRARIAS

Ing. Agrop. Maura Judith De La A Rodríguez, Mgtr.
COORDINADORA DEL CENTRO DE APOYO MANGLARALTO

De mis consideraciones:

En atención a oficio s/n, suscrito por el Ing. Carlos Andrade Caicho, Docente Tutor de Tesis de la Carrera de Telecomunicaciones, de la Facultad de Sistemas y Telecomunicaciones de la UPSE, que en la parte pertinente refiere:

...
Por medio de la presente, me permito solicitar de la manera más comedida se nos conceda el permiso para ingresar a las instalaciones de la sede Manglaralto de la UPSE, con la finalidad de desarrollar actividades relacionadas con el trabajo de titulación (tesis) titulado: "Monitoreo de ganado en zonas rurales mediante comunicación LoRa y geolocalización para aplicaciones de seguridad".
Para tales efectos, tenemos previsto realizar las pruebas de cobertura de alcance y pruebas rápidas con una vaca, mediante la colocación de un collar GPS para validar el funcionamiento del sistema en condiciones reales, los días jueves y viernes en un horario de 8:30 a. m. a 3:00 p.m. ...

De lo antes citado, se autoriza el ingreso la estudiante Fiorella Pillasagua Yagual y de Gabriela Aracely Tomalá Pozo de la Carrera de Telecomunicaciones para que realicen la investigación para el trabajo de titulación: "Monitoreo de ganado en zonas rurales mediante comunicación LoRa y geolocalización para aplicaciones de seguridad", en el Centro de Apoyo UPSE Manglaralto, de acuerdo con la siguiente planificación:

Cédula Identidad	Estudiantes	Días	Horarios
2400124976	Fiorella Verónica Pillasagua Yagual	Jueves y	08:30 a.m. -
2450298308	Gabriela Aracely Tomalá Pozo	viernes	15:00 p.m.

Asimismo, comunico que se designa al MVZ. Joffre Javier Masaquiza Aragón, Mgtr., Docente de la Facultad de Ciencias Agrarias para el acompañamiento en el desarrollo de la investigación. Celular: 096 128 6951

Sin otro particular, me suscribo de usted.

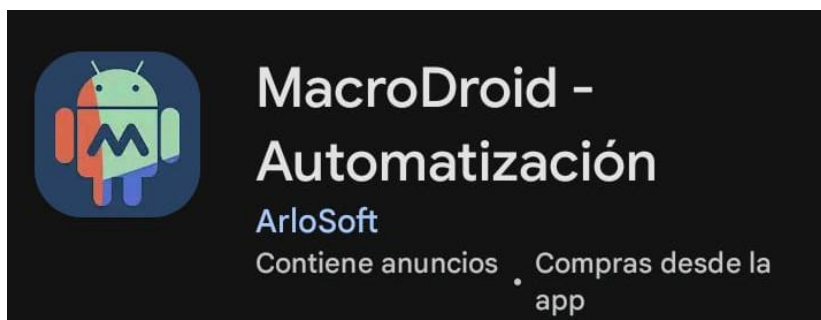
Atentamente.



Ing. Verónica Cristina Andrade Yucailla, Ph.D.
DIRECTORA CARRERA AGROPECUARIA

Anexo G

Aplicación para sonido de alerta



Notificación recibida
Las notificaciones, como las UIP, pueden bloquearse por motivos de privacidad y, por lo tanto, no se activarán. Puedes desactivar este comportamiento desactivando 'Notificaciones mejoradas' en la configuración de notificaciones de tu dispositivo.

Aplicaciones

Telegram 

☐ Utilizar filtros separados para el título y el texto

Contenido del texto:

☐ Cualquier ☐ Coincide
☒ Contiene ☐ Excluye

ALERTA 

caracteres especiales * y ? son soportados

☐ Activar la coincidencia de expresiones regulares

☒ Ignorar mayúsculas

☐ Omitir las notificaciones persistentes

☒ Prevenir multiples disparadores

Sonido de notificación

Cualquier valor ▼

CANCELAR **ACEPTAR**

Anexo H

Manual básico de instalación y operación del sistema de monitoreo de ganado

1. Componentes que se utilizaron para el sistema.

Para el dispositivo transmisor (collar):

- Heltec WiFi LoRa 32 V4 (incluye microcontrolador ESP32 y Pantalla OLED)
- Módulo GPS NEO-6M
- Antena LoRa
- Batería recargable

Para el dispositivo receptor:

- Heltec WiFi LoRa 32 V4 (incluye microcontrolador ESP32 y Pantalla OLED)
- Antena LoRa
- Batería recargable
- Conexión Wifi

Se utilizan las siguientes aplicaciones:

- Telegram
- Blynk

2. Pasos para instalación.

- 1) Revisar estado de carga de las baterías.
- 2) Verificar que las antenas LoRa estén correctamente conectadas.
- 3) Ubicar el collar sobre el ganado asegurando un ajuste firme y cómodo.
- 4) Instalar el receptor en un punto elevado con acceso a la red Wifi.
- 5) Encender ambos dispositivos.

3. Carga de la batería.

- Utilizar un cargador de baterías de litio, o en su defecto una fuente de 5v.
- Cargar por completo las baterías antes de cada jornada de trabajo.
- Evitar el uso de los dispositivos cuando el nivel de batería esté bajo.

4. Encendido.

- Encender el transmisor.
- Esperar la adquisición de señal GPS.
- Encender el receptor.
- Constatar que ambos dispositivos estén ejecutándose correctamente mediante la información de la pantalla OLED.

5. Configuración.

- Configurar las credenciales de la red Wifi en el dispositivo receptor.
- Verificar el Token del bot de Telegram.
- Configurar las credenciales de Blynk.
- Utilizar para ambos dispositivos una frecuencia LoRa de 915 MHz.

6. Flujo de operación.

- Correcto encendido de los dispositivos.
- El collar obtiene las coordenadas GPS.
- La información se transmite mediante LoRa.
- El receptor recibe los datos.
- La ubicación se envía automáticamente a Telegram y Blynk.
- Verificar periódicamente que las coordenadas sean actualizadas correctamente.

7. Solución de fallos.

Problema	Posible causa	Solución
No obtiene coordenadas	Pocos satélites u obstrucción de la señal satelital	Colocar el collar en un espacio abierto y esperar la adquisición de señal
No existe comunicación LoRa	Antena desconectada o fuera de alcance	Verificar la antena y reducir la distancia entre los dispositivos
No llegan mensajes a Telegram	Sin conexión Wifi o configuración incorrecta	Comprobar la conexión WiFi y el token del bot
Blynk no actualiza datos	Problemas de conexión MQTT o Wifi	Revisar la conexión a Internet y reiniciar el receptor
Pantalla OLED sin información	Error de alimentación o reinicio del dispositivo	Verificar la batería y reiniciar el equipo