

# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

## **Facultad de Ingeniería en Electricidad y Computación**

**Sistema de Monitoreo con Cerca Virtual para el Control del Rebaño  
en la Granja de ESPOL**

**INGE-2866**

## **PROYECTO INTEGRADOR**

Previo a la obtención del Título de:

**Ingeniero en Telecomunicaciones**

Presentado por:

Alexis Leonardo Salguero Paguay

Maritza Isabel Sampedro Núñez

Guayaquil – Ecuador  
Año: 2025

## **Dedicatoria**

---

A mis padres, **Rafael** e **Isabel** por enseñarme el valor del esfuerzo, por su amor, sacrificio y apoyo en cada etapa de mi vida, son mi columna vertebral. Su esfuerzo y constancia han sido mi aliento para seguir avanzando. Les agradezco todo lo que soy y seré.

A mi hermano **Marcelo** por siempre apoyarme, darme su cariño y recalcarme la importancia de nunca dejar de luchar.

A mi compañero de tesis **Leonardo**, por su incansable labor, paciencia y dedicación para alcanzar los mejores resultados en este proyecto.

A mis tíos, por sus palabras de aliento y fe en mí. Su confianza en lo que hago me ha dado valor para no rendirme.

Finalmente me dedico este logro a mí, que lo hice con miedo, dudas, pero con mucho esfuerzo, pude llegar a donde estoy.

**Maritza Isabel Sampedro Núñez**

## **Dedicatoria**

---

El presente proyecto en primer lugar se lo dedico a Dios por darme la sabiduría y fortaleza para culminar y cumplir esta meta, a mis padres **Rodrigo Salguero** y en especial a mi mamá **Lucia Paguay**, por brindarme su apoyo incondicional desde el día uno a lo largo de este camino el cual fue todo un reto de superación personal y dedicación académica.

A mi hermana **Roxana**, a mi sobrina **Alina**, mi novia, tíos y abuelos, quienes siempre me dieron su apoyo y palabras de motivación para seguir esforzándome en cada desafío que se me presento.

Finalmente termino esta dedicatoria citando una frase que me la dio un señor en uno de los momentos más difíciles: “Su esfuerzo de hoy, Dios y la vida se lo recompensara mañana”.

**Alexis Leonardo Salguero Paguay**

## **Agradecimientos**

---

Agradezco a Dios quien ha sido mi fuente de fortaleza, sabiduría y aliento. Gracias por no permitirme rendirme y haberme dado la fe necesaria para superar cada momento duro a lo largo de esta etapa. Todo viene de él.

A mis padres, por haberme dado todas las herramientas y palabras de aliento para cumplir mis sueños.

Al ingeniero **Edison Del Rosario** quien ha sido nuestra guía académica a lo largo de este proyecto, sus enseñanzas fueron importantes para poder cumplir nuestros objetivos.

A mis amigos de la universidad, **Leonardo, Paul, José, Carlos y Boris** por su apoyo, amistad y formar parte de esta etapa que recordare con mucho cariño. A todos ustedes, gracias por formar parte de esta historia y recorrer juntos este camino de aprendizaje para la vida.

**Maritza Isabel Sampedro Núñez**

## **Agradecimientos**

---

Agradezco a Dios por darme la fortaleza y sabiduría desde el día uno y más en los momentos difíciles.

A toda mi familia por siempre brindarme su apoyo incondicional a lo largo de este camino.

A mis amigos de la Universidad, **Maritza, Paul** y amigos de **Residencia ESPOL** quienes me brindaron su amistad, apoyo y muchos momentos compartidos que los recordare con alegría y cariño.

Un agradecimiento especial al Ingeniero **Edison Del Rosario** por compartirnos sus conocimientos y ser una guía fundamental en este proyecto integrador.

**Alexis Leonardo Salguero Paguay**

## Declaración Expresa

Nosotros Alexis Leonardo Salguero Paguay y Maritza Isabel Sampedro Núñez acordamos y reconocemos que:

La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores.

La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por nosotros durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que nos corresponda de los beneficios económicos que la ESPOL reciba por la explotación de nuestra innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique los autores que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, 30 de mayo de 2025.



Alexis Leonardo  
Salguero Paguay  
 Time Stamping  
Security Data

Alexis Leonardo  
Salguero Paguay



Maritza Isabel  
Sampedro Núñez  
 Time Stamping  
Security Data

Maritza Isabel Sampedro  
Núñez

## **Evaluadores**

---

**PhD. Patricia Chávez Burbano**  
Profesora de Materia

---

**PhD. German Vargas López**  
Tutor del Proyecto

## Resumen

El proyecto aborda la necesidad de mejorar el control del rebaño en la granja de ESPOL frente a los altos costos y limitaciones de las cercas físicas. El objetivo principal es diseñar e implementar un sistema de cerca virtual que permita monitorear en tiempo real el desplazamiento de los animales mediante comunicación inalámbrica y generar alertas al detectar salidas de la zona segura. Se plantea la integración de tecnologías de bajo consumo, junto con modelos predictivos, optimizar la gestión del ganado y reducir pérdidas.

El sistema se desarrolló utilizando nodos LoRa CubeCell instalados en collares, un gateway LoRaWAN conectado a ChirpStack y la plataforma Home Assistant para la visualización. Los datos de señal se almacenaron en una base SQL y se aplicaron algoritmos de aprendizaje automático, específicamente LSTM y Random Forest, para predecir la posición de los animales con mayor precisión.

Las pruebas realizadas en la granja mostraron que los prototipos ligeros y portátiles no afectaron la movilidad del ganado y permitieron delimitar zonas de pastoreo virtuales, emitiendo notificaciones inmediatas al personal mediante la interfaz web y dispositivos móviles.

En conclusión, el sistema constituye una alternativa eficiente, escalable y de bajo costo que mejora la seguridad del rebaño, optimiza recursos y representa un avance tecnológico aplicable al sector ganadero.

**Palabras Clave:** Cerca virtual, Ganadería, Monitoreo en tiempo real, LoRa

## Abstract

*The project addresses the need to improve herd control on the ESPOL farm in the face of high costs and limitations of physical fences. The main objective is to design and implement a virtual fence system that allows to monitor in real time the movement of animals through wireless communication and generate alerts when detecting exits from the safe zone. The integration of low-consumption technologies is proposed, along with predictive models, optimizing cattle management and reducing losses.*

*The system was developed using LoRa CubeCell nodes installed in collars, a LoRaWAN gateway connected to ChirpStack and the Home Assistant platform for visualization. The signal data was stored in an SQL database and machine learning algorithms, specifically LSTM and Random Forest, were applied to predict the position of the animals more accurately.*

*The tests carried out on the farm showed that the light and portable prototypes did not affect the mobility of the cattle and allowed the delimitation of virtual grazing areas, issuing immediate notifications to the staff through the web interface and mobile devices.*

*In conclusion, the system constitutes an efficient, scalable, and low-cost alternative that improves the safety of the herd, optimizes resources, and represents a technological advance applicable to the livestock sector.*

**Keywords:** *Virtual Fence, Livestock, Real time Monitoring, LoRa.*

# Índice General

## Contenido

Resumen.....	I
Abstract.....	II
Índice General .....	III
Abreviaturas .....	V
Simbología .....	VI
Índice de figuras .....	VII
Índice de Tablas.....	VIII
<b>Capítulo 1 .....</b>	<b>1</b>
1.1    Introducción .....	2
1.2    Descripción del Problema .....	4
1.3    Justificación del Problema .....	5
1.4    Objetivos .....	6
1.4.1 <i>Objetivo general</i> .....	6
1.4.2 <i>Objetivos específicos</i> .....	6
1.5    Marco teórico .....	7
1.5.1 <i>LoRa y LoRaWAN</i> .....	8
1.5.2 <i>Received Signal Strength Indicator (RSSI)</i> .....	8
1.5.3 <i>HELTEC HTM-01</i> .....	8
1.5.4 <i>Arduino IDE</i> .....	9
1.5.5 <i>ChirpStack</i> .....	9
1.5.6 <i>Home Assistant</i> .....	9
1.5.7 <i>Python (Spyder IDE)</i> .....	10
1.5.8 <i>Algoritmo de aprendizaje supervisado y red neuronal recurrente.</i> .....	11
<b>Capítulo 2 .....</b>	<b>12</b>
<b>2    Metodología .....</b>	<b>13</b>
2.1    Diseño conceptual y Metodología de Diseño.....	13
2.2    Recursos Utilizados.....	16
2.3    Diseño del Sistema .....	18
2.4    Desarrollo del Sistema .....	21
2.4.1 <i>Implementación de Nodos Lora</i> .....	24

2.4.2	<i>Captura y Envío de Parámetros de Medición</i> .....	24
2.4.3	<i>Gateway y Red LoRaWan</i> .....	24
2.4.4	<i>Decodificación de Tramas</i> .....	25
2.4.5	<i>Comunicación con MQTT</i> .....	25
2.4.6	<i>Incorporación del Modelo Predictivo LSTM</i> .....	26
2.4.7	<i>Visualización en Home Assistant</i> .....	26
Capítulo 3	.....	27
<b>3</b>	<b>Resultados y Análisis</b> .....	28
3.1	Desarrollo del Sistema .....	28
3.2	Entrenamiento del Modelo con Machine Learning.....	33
3.2.1	<i>Entrenamiento con Distancia</i> .....	34
3.2.2	<i>Entrenamiento sin Distancia</i> .....	38
3.2.3	<i>Entrenamiento con Distancia usando menos datos</i> .....	42
3.2.4	<i>Entrenamiento con Distancia usando más datos</i> .....	44
3.2.5	<i>Random Forest</i> .....	46
3.2.6	<i>Modelo LSTM con Random Forest</i> .....	48
3.3	Resultados Obtenidos .....	50
3.3.1	<i>Prototipo Final</i> .....	50
3.3.2	<i>Prototipo Posicionado</i> .....	52
3.3.3	<i>Alertas en Home Assistant</i> .....	53
3.4	Validación de alertas en Home Assistant .....	56
3.4.1	Posición del chivo dentro de la cerca virtual.....	56
3.4.2	Posición del chivo fuera de la cerca virtual.....	58
3.4.3	Visualización de la interfaz de Home Assistant desde dispositivo móvil .....	60
Capítulo 4	.....	65
<b>4</b>	<b>Conclusiones y Recomendaciones</b> .....	66
4.1	Conclusiones .....	66
4.2	Recomendaciones.....	67
5	Bibliografía .....	69

## Abreviaturas

ANN	Artificial Neural Networks
ESPOL	Escuela Superior Politécnica del Litoral
GIRNI	Grupo de Investigación de Redes de Información Inalámbricas
GEA	Granja Experimental Agrícola
GPS	Sistema de Posicionamiento Global
LoRa	Long Range
LoRaWAN	Protocolo para la gestión de comunicación de largo alcance entre dispositivos (Long Range Wide Area)
LSTM	Long Short-Term Memory
MAE	Error Absoluto Medio
ML	Machine Learning
MQTT	Protocolo de Mensajería ligero para IoT
RAM	Memoria de Acceso Aleatorio
RSSI	Indicador de Intensidad de la Señal Recibida
SNR	Relacion Señal/Ruido

## **Simbología**

GB	Gigabyte
GHz	Gigahercios
KHz	Kilohercio
MHz	Megahercio
m	Metro
min	minuto

## Índice de figuras

<b>Figura 2.1</b> Zona donde se realizarán las pruebas.....	15
<b>Figura 2.2</b> Sistema de monitoreo propuesto .....	19
<b>Figura 2.3</b> Etapas del sistema de posicionamiento.....	22
<b>Figura 3.1</b> Diagrama de Bloques del proyecto .....	28
<b>Figura 3.2</b> Área geográfica con la posición del Gateway y los nodos .....	29
<b>Figura 3.3</b> Vista del terreno donde estan ubicados los nodos estáticos.....	29
<b>Figura 3.4</b> Conexiones de Alimentación.....	30
<b>Figura 3.5</b> Base de Datos SQL.....	32
<b>Figura 3.6</b> Entrenamiento del modelo con ANN.....	34
<b>Figura 3.7</b> Entrenamiento del modelo con LSTM .....	34
<b>Figura 3.8</b> Curvas de Entrenamiento .....	35
<b>Figura 3.9</b> Predicción de Coordenadas Reales ANN.....	37
<b>Figura 3.10</b> Predicciones de Coordenadas Reales LSTM.....	37
<b>Figura 3.11</b> Entrenamiento del Modelo con ANN.....	38
<b>Figura 3.12</b> Entrenamiento del Modelo con LSTM .....	39
<b>Figura 3.13</b> Desempeño de los modelos.....	40
<b>Figura 3.14</b> Entrenamiento modelo ANN.....	42
<b>Figura 3.15</b> Entrenamiento modelo LSTM .....	43
<b>Figura 3.16</b> Entrenamiento modelo ANN.....	44
<b>Figura 3.17</b> Entrenamiento modelo LSTM .....	44
<b>Figura 3.18</b> Importancia de variables de entrenamiento .....	47
<b>Figura 3.19</b> Curva de entrenamiento .....	48
<b>Figura 3.20</b> Entrenamiento y evaluación del modelo LSTM .....	49
<b>Figura 3.21</b> Prototipo 1 con GPS como validador .....	50
<b>Figura 3.22</b> Prototipo 2 con GPS como validador .....	50
<b>Figura 3.23</b> Prototipo final sin GPS .....	51
<b>Figura 3.24</b> Prototipo ubicado en el Chivo.....	52
<b>Figura 3.25</b> Mapa de monitoreo en tiempo real en Home Assistant .....	53
<b>Figura 3.26</b> Área definida como Cerca Virtual .....	54
<b>Figura 3.27</b> Posición y ruta del primer dispositivo móvil.....	54
<b>Figura 3.28</b> Posición y ruta del segundo dispositivo móvil .....	55
<b>Figura 3.29</b> Chivo dentro de la cerca virtual, visualización en mapa .....	56
<b>Figura 3.30</b> Notificación de alerta de la posición del chivo .....	57
<b>Figura 3.31</b> Baliza en estado off .....	57
<b>Figura 3.32</b> Chivo fuera de la cerca virtual, visualización en mapa .....	58
<b>Figura 3.33</b> Notificación de alerta de la posición del chivo fuera de la cerca virtual .....	58
<b>Figura 3.34</b> Baliza en estado on .....	59
<b>Figura 3.35</b> Interfaz principal de Cerca Virtual .....	60
<b>Figura 3.36</b> Visualización de la posición del chivo dentro de la cerca virtual.....	61
<b>Figura 3.37</b> Posición del chivo fuera de la Cerca Virtual.....	62
<b>Figura 3.38</b> Mensajes de alerta.....	63

## Índice de Tablas

<b>Tabla 2.1</b> Lista de Requerimientos.....	13
<b>Tabla 2.2</b> Especificaciones técnicas de los dispositivos utilizados.....	17
<b>Tabla 3.1</b> Datos de lectura del GPS en el Área definida.....	31

# Capítulo 1

|

## 1.1 Introducción

A nivel mundial, la ganadería representa aproximadamente el 40 % del valor económico total de la producción agrícola y es fuente directa de sustento para más de 1.300 millones de personas (FAO, 2022). Sin embargo, este sector no está exento de dificultades: enfrenta retos ambientales vinculados con la sostenibilidad, demandas crecientes en cuanto al bienestar animal, limitaciones en la disponibilidad de mano de obra calificada y la presión constante de responder a una mayor necesidad de alimentos. (FAO, 2022)

Uno de los principales problemas que enfrenta la ganadería moderna es la dificultad para monitorear eficientemente grandes extensiones de terreno y garantizar un control adecuado sobre el movimiento del ganado, lo cual puede generar pérdidas económicas considerables por fuga, robo o accidentes.

La implementación de tecnologías como los sistemas de monitoreo remoto y las cercas virtuales ha cobrado fuerza en la última década como solución innovadora para estos desafíos. En países como Australia, se estima que más del 60 % de los productores ganaderos extensivos están considerando integrar soluciones de cercado virtual o geofencing en los próximos cinco años, debido a su potencial para reducir costos operativos y mejorar el manejo de los recursos naturales (CSIRO, 2024).

En América Latina, la adopción de estas tecnologías es aún limitada, pero creciente. Según datos de (Sotomayor, Ramirez, & Martinez, 2021) solo alrededor del 8 % de los productores agropecuarios en la región utilizan tecnologías digitales avanzadas, aunque países como Brasil, Chile y Uruguay han impulsado políticas de transformación digital rural.

En el caso de Ecuador, el sector ganadero representa aproximadamente el 10 % del PIB agropecuario y emplea directa o indirectamente a más de 300.000 personas, en su mayoría en zonas rurales (Pino, Vinueza, Muñoz, & Saravia, 2022).

Sin embargo, la modernización tecnológica del sector sigue rezagada: menos del 5% de las fincas ganaderas ecuatorianas cuentan con algún tipo de automatización o sistema de control remoto (Pino, Vinueza, Muñoz, & Saravia, 2022).

En este contexto, la Escuela Superior Politécnica del Litoral (ESPOL), a través de su Granja Experimental, se convierte en un espacio estratégico para validar e implementar soluciones tecnológicas adaptadas a las condiciones del trópico seco costero. Uno de los problemas persistentes en la granja es la dificultad para realizar un control eficiente y continuo del rebaño, especialmente durante jornadas nocturnas o en zonas de difícil acceso.

El uso de cercas físicas presenta limitaciones relacionadas con el mantenimiento, el costo y el riesgo de que los animales las evadan o sufran heridas; ante esta problemática, el presente proyecto propone el desarrollo de un Sistema de Monitoreo con Cerca Virtual para el Control del Rebaño, utilizando tecnologías de comunicación inalámbrica de bajo consumo energético como LoRaWAN y una plataforma web para la visualización en tiempo real.

Aunque se utilizará señal GPS únicamente como referencia puntual para validar el margen de error de ubicación reportada por los dispositivos LoRa, el sistema operará de forma autónoma con la información transmitida por estos últimos.

Esto permitirá definir zonas de pastoreo virtuales, recibir alertas ante cruces no permitidos, y realizar un seguimiento del comportamiento del ganado, con el objetivo de reducir pérdidas, optimizar recursos humanos y mejorar el bienestar animal.

La propuesta responde a tendencias globales de transformación digital en el agro, pero se contextualiza en las necesidades locales del entorno ecuatoriano, alineándose también

con los Objetivos de Desarrollo Sostenible (ODS), especialmente en los ámbitos de infraestructura sostenible (ODS 9), producción responsable (ODS 12) y conservación de ecosistemas terrestres (ODS 15).

## 1.2 Descripción del Problema

En la granja experimental de la Escuela Superior Politécnica del Litoral (ESPOL), la falta de un cercado físico completo ha generado dificultades en el control del rebaño de borregos. La ausencia de barreras eficaces ha permitido que los animales abandonen la zona asignada y se desplacen por otras áreas del campus, exponiéndose a situaciones de riesgo como accidentes o pérdidas, y generando posibles inconvenientes para la comunidad universitaria.

Esta problemática no es exclusiva de ESPOL ya que se presenta con frecuencia en diversas fincas del país, donde los altos costos asociados a la construcción y mantenimiento de cercas físicas limitan su implementación. Como consecuencia, el manejo del ganado suele depender de la vigilancia continua por parte del personal, lo cual incrementa la carga operativa y no siempre garantiza resultados efectivos.

Dado este contexto, se propone el desarrollo de un sistema de monitoreo con cerca virtual, que utilice tecnología inalámbrica para establecer límites digitales que permitan detectar la presencia del ganado fuera del área permitida y activar alertas oportunas. Esta solución prescinde del uso de GPS, lo que reduce su complejidad y costo, sin comprometer su funcionalidad.

El proyecto considera variables clave como el alcance y estabilidad de la comunicación inalámbrica, la eficiencia energética de los dispositivos, la facilidad de instalación en campo y la capacidad de respuesta en tiempo real.

Además, al tratarse de un sistema susceptible de observación, medición y análisis, se facilita la validación de su desempeño y su posterior mejora.

### 1.3 Justificación del Problema

La ausencia de un cercado físico en la granja de ESPOL permite que los borregos salgan del área segura, lo que los expone a riesgos de accidentes o extravíos, y también afecta las actividades del campus. Además, esto exige una supervisión constante del personal, conllevando a una mayor carga operativa. Estas dificultades operacionales y de seguridad fortalecen la necesidad de una solución eficiente (Vendramini, 2024).

Los cercos tradicionales tienen un costo elevado: por ejemplo, en ranchos de Nuevo México un kilómetro de cerca metálica puede costar hasta USD 9 300, sin contar su mantenimiento anual (Vitale, y otros, 2025). En contraste, las cercas virtuales se basan en collares GPS y estaciones base inalámbricas, eliminando la necesidad de infraestructura física, y pueden reducir los costos de cercado y mantenimiento hasta en un 76 %, especialmente en grandes extensiones (Vitale, y otros, 2025).

Diversos estudios recientes demuestran que las cercas virtuales contienen al ganado dentro de límites definidos en más del 90 % de los casos, con niveles de bienestar parecidos o mejores comparados con cercas tradicionales (Ruminants, 2025). Además, un análisis económico sobre sistemas VF en condiciones similares al sur de EE. UU. señala ahorros en

costos operativos de entre USD 40 y USD 120 por animal al año, principalmente al reducir labores de supervisión y el mantenimiento de cercas (Duval, y otros, 2025).

Las cercas virtuales también permiten delimitar zonas de pastoreo desde dispositivos móviles o plataformas web, emitir alertas en tiempo real y reconfigurar las áreas según sea necesario (System, 2025). Esto no solo mejora la seguridad del rebaño, sino que reduce la carga operativa del personal, al automatizar la detección de salidas no autorizadas, y facilita el replanteo de áreas sin incurrir en costos adicionales de infraestructura.

## 1.4 Objetivos

A continuación, presentamos los objetivos a desarrollar del proyecto cerca virtual.

### 1.4.1 *Objetivo general*

Diseñar e implementar un sistema de cerca virtual para la granja de ESPOL que permita monitorear el desplazamiento de los borregos mediante tecnologías de comunicación inalámbrica (LoRa) y notificar, mediante alertas en tiempo real (Telegram o WhatsApp) o interfaz web cuando alguno de ellos salga del área segura definida.

### 1.4.2 *Objetivos específicos*

- Realizar un levantamiento de información para determinar el perímetro requerido de la cerca virtual.
- Integrar dispositivos portátiles en los borregos que permitan detectar su proximidad o alejamiento de zonas seguras.
- Desarrollar un sistema de notificación automatizada que envíe alertas al encargado cuando un animal salga del área permitida.

## 1.5 Marco teórico

En diversas partes del mundo, el uso de tecnologías de cerca virtual y monitoreo de ganado ha sido implementado con éxito en la agricultura y ganadería. Estas soluciones utilizan tecnologías de comunicación como LoRaWAN y plataformas de automatización para mejorar el manejo del ganado, reducir costos operativos y aumentar la seguridad.

Por ejemplo, en Australia, el sistema Agersens ha sido desarrollado para gestionar la ubicación de ganado mediante collares electrónicos que emiten señales a un sistema de cerca virtual. Este sistema permite la delimitación de zonas de pastoreo sin la necesidad de cercas físicas, lo que reduce considerablemente los costos de instalación y mantenimiento, al tiempo que mejora la eficiencia en el manejo del ganado. Este sistema se basa en tecnología de radiofrecuencia similar a LoRaWAN, lo que proporciona un alcance de señal adecuado para grandes extensiones de tierra (Gallagher, 2021).

En Estados Unidos, el sistema Nofence ha ganado popularidad en la ganadería de precisión. Esta solución utiliza collares GPS y un sistema basado en LoRaWAN para gestionar el ganado en grandes ranchos (Nofence, 2025). El sistema permite monitorear la ubicación de los animales en tiempo real, con la posibilidad de generar alertas cuando un animal sale de la zona segura. Además, Nofence ha demostrado ser eficaz en condiciones climáticas extremas y en terrenos difíciles, lo que lo convierte en una opción viable para diversas regiones.

### ***1.5.1 LoRa y LoRaWAN***

LoRa es una tecnología de transmisión inalámbrica de largo alcance y bajo consumo. LoRaWAN es el protocolo que permite gestionar la comunicación entre múltiples nodos (dispositivos) y un gateway central (Roman, 2025).

Para este proyecto se utilizarán módulos LoRa en dispositivos portátiles sujetos a los animales. Estos módulos transmitirán datos a un gateway LoRaWAN ubicado en la granja, formando una red robusta que funciona sin conexión celular ni GPS.

### ***1.5.2 Received Signal Strength Indicator (RSSI)***

Es un indicador de la intensidad de la señal recibida por un receptor. Se utiliza como parámetro para estimar la distancia o calidad del enlace entre un nodo transmisor y el Gateway (Hardesty, 2025).

El valor de RSSI se utilizará para delimitar virtualmente la zona segura. Si un nodo transmite una señal con un RSSI inferior al umbral definido, el sistema interpretará que el animal ha salido del área permitida y generará una alerta.

### ***1.5.3 HELTEC HTM-01***

Es un sensor digital de temperatura y humedad que utiliza un único bus de datos para la comunicación. Basado en un sensor de humedad capacitivo y un termistor para medir la temperatura, este dispositivo es ampliamente utilizado en aplicaciones de monitoreo ambiental, como estaciones meteorológicas, sistemas de control de clima y proyectos de Internet de las Cosas (IoT) (Technologies, 2025).

Cada animal llevará un dispositivo HTM-01 como nodo transmisor, que enviará datos periódicos al gateway.

#### ***1.5.4 Arduino IDE***

Arduino IDE es un entorno de desarrollo de código abierto utilizado para programar microcontroladores. Permite configurar sensores, módulos de comunicación y rutinas de control mediante un lenguaje accesible (Arduino, 2024).

Se utilizará Arduino IDE para programar el comportamiento de los nodos HTM-01, definiendo la frecuencia de transmisión LoRa, la estructura de los datos enviados y la lectura de parámetros como el nivel de batería.

#### ***1.5.5 ChirpStack***

Es una plataforma de código abierto para la gestión de redes LoRaWAN. Administra los dispositivos, recibe sus transmisiones, y permite configurar reglas para eventos y almacenamiento de datos (Chirpstack, 2023).

Funcionará como la plataforma central receptora de los datos transmitidos por los nodos. A través de su interfaz, el personal podrá visualizar el estado de cada dispositivo, revisar los valores de RSSI y configurar notificaciones automáticas en caso de eventos críticos.

#### ***1.5.6 Home Assistant***

Es una plataforma de automatización del hogar de código abierto que permite la integración y control de dispositivos inteligentes desde una interfaz centralizada. Su enfoque

principal es la privacidad y el control local, evitando la dependencia de servicios en la nube.

La plataforma es compatible con una amplia variedad de dispositivos y protocolos, incluyendo LoRaWAN, Zigbee, Z-Wave, MQTT, entre otros. Además, ofrece la posibilidad de crear automatizaciones personalizadas y visualizar el estado de los dispositivos en tiempo real (Assistant, Home Assistant: La plataforma de automatización del hogar de código abierto, 2025).

Se usará Home Assistant como la plataforma central que gestionará los datos recibidos por los HTM01. A través de su interfaz de usuario, el personal de la granja podrá visualizar en tiempo real el estado de cada nodo, incluyendo la intensidad de la señal (RSSI), el nivel de batería de los dispositivos y otros parámetros críticos.

Home Assistant también permitirá configurar reglas de automatización que generen alertas cuando un nodo se desplace fuera de la zona segura, lo que proporcionará una respuesta rápida y eficiente ante cualquier incidente.

### **1.5.7 Python (Spyder IDE)**

Python es un lenguaje de programación ampliamente utilizado para análisis de datos, automatización y visualización. Spyder es un entorno de desarrollo (IDE) para Python, especialmente útil en tareas de ingeniería y análisis científico (Kosourova, 2024).

Se utilizará Python desde Spyder para procesar los datos recolectados desde ChirpStack, generar reportes, graficar los valores de RSSI, y analizar patrones de comportamiento del rebaño. Esta etapa permitirá validar el rendimiento del sistema, ajustar umbrales de alerta y obtener indicadores clave para la toma de decisiones.

### ***1.5.8 Algoritmo de aprendizaje supervisado y red neuronal recurrente.***

Un algoritmo de aprendizaje supervisado es un método de inteligencia artificial que permite a un modelo aprender a partir de ejemplos ya conocidos. Funciona con un conjunto de datos de entrada (características) y sus correspondientes salidas esperadas (etiquetas), de manera que el sistema va ajustando sus parámetros para minimizar el error entre sus predicciones y los resultados reales. Una vez entrenado, el algoritmo puede predecir la salida de nuevos datos que no ha visto antes.

Un ejemplo claro de algoritmo de aprendizaje supervisado es el Random Forest, el cual se basa en la construcción de múltiples árboles de decisión para realizar predicciones. En este enfoque, cada árbol se entrena con una muestra distinta de los datos y genera una predicción individual; luego, el modelo combina todas esas predicciones (mediante votación en clasificación o promedios en regresión) para obtener un resultado más robusto y preciso (Rodrigo, 2020).

Para este proyecto se utilizará un modelo híbrido utilizando un algoritmo de aprendizaje supervisado como lo es Random Forest y la red neuronal LSTM, ya que este último posee gran capacidad para recordar y regular información de secuencias largas, permitiendo efectividad en tareas que requieren mantener escenarios prolongados, es decir mientras más datos históricos de RSSI se obtenga, mejor será la predicción de la posición.

# Capítulo 2

## 2 Metodología

Este proyecto tiene como objetivo seguir la ubicación de los animales mediante módulos LoRa CubeCell que envían datos de señal y coordenadas a un Gateway LoRa. Estos datos se guardan en una base SQL y se usan para entrenar un modelo LSTM que predice la posición de los animales en tiempo real mejorando así el control y monitoreo del ganado.

### 2.1 Diseño conceptual y Metodología de Diseño

El diseño conceptual del sistema se basa en la implementación de una cerca virtual para monitorear el ganado mediante la tecnología LoRaWAN. En lugar de utilizar sistemas de localización como GPS, que resultan costosos y limitados en entornos rurales, el sistema utiliza la señal RSSI (Received Signal Strength Indicator) para determinar la proximidad de los animales a los límites de la zona segura.

El diseño comienza con un levantamiento de requerimientos como los presentados en la **Tabla 2.1**, donde se identificaron necesidades clave como la eficiencia energética, la escalabilidad y la capacidad de operar en condiciones de terreno rural como se visualiza en la **Figura 2.1**. A partir de estos requerimientos, se eligió LoRa como la tecnología de comunicación, dada su capacidad de largo alcance y bajo consumo de energía.

**Tabla 2.1** Lista de Requerimientos

Requerimiento	Descripción	Justificación
Eficiencia Energética	El sistema debe funcionar con bajo consumo de energía para prolongar la vida útil de los dispositivos	Da paso a un funcionamiento autónomo y duradero en zonas rurales sin necesidad constante de fuentes de energía

Escalabilidad	<p>El sistema debe poder adaptarse para agregar más nodos o dispositivos sin que se pierda su eficiencia.</p>	<p>Facilita el crecimiento del sistema a medida que aumente la cantidad de animales o el área de cobertura.</p>
Alcance de Comunicación	<p>Usar tecnología con alto alcance como LoRa que cubre grandes distancias sin alto costo.</p>	<p>LoRa permite la transmisión de datos en zonas rurales.</p>
Bajo Consumo de Energía	<p>Los dispositivos deben operar con baterías de larga duración.</p>	<p>Minimiza la necesidad de un mantenimiento frecuente y asegura la viabilidad en zonas con limitación a la electricidad.</p>



**Figura 2.1** Zona donde se realizarán las pruebas

La metodología de diseño sigue un enfoque iterativo. Primero se realiza una prueba inicial en una granja experimental con un área de 2 hectáreas, donde se instalan los nodos LoRa en los collares de los animales y las estaciones receptoras (gateways) en puntos estratégicos. A medida que se recogen datos del funcionamiento del sistema, se ajustan los parámetros y se optimizan los algoritmos de predicción. Además, se integra el sistema con Home Assistant, una plataforma de monitoreo en tiempo real, lo que permite a los administradores recibir alertas y visualizar la ubicación de los animales de manera remota. Este enfoque asegura que el sistema sea flexible y adaptable a diferentes condiciones de la granja.

El diseño también incorpora un modelo de aprendizaje automático basado en LSTM (Long Short-Term Memory), el cual se utiliza para predecir la proximidad futura de los animales. Este modelo es entrenado con datos históricos de las mediciones RSSI, junto con otros parámetros que se ajusten al modelo.

La predicción de la posición futura del ganado ayuda a mejorar la precisión del sistema, reduciendo el número de alertas falsas y permitiendo un monitoreo más efectivo.

## 2.2 Recursos Utilizados

Para el desarrollo de este sistema, se utilizaron distintos recursos de hardware y software, cuyas especificaciones técnicas se detallan en la **Tabla 2.2**. Esta tabla resume las principales características de los dispositivos empleados, incluyendo los módulos LoRa, el GPS, la Raspberry Pi utilizada como gateway, así como el software necesario para la gestión y visualización de los datos. En cuanto al hardware, se emplearon módulos LoRa HTM01, que fueron instalados en los collares de los animales para transmitir señales de RSSI a través de la red LoRaWAN.

Las estaciones receptoras, ubicadas en puntos estratégicos de la granja, recibieron estas señales y las transmitieron a un servidor central ubicado en una Raspberry Pi, donde se gestionó la comunicación mediante ChirpStack, un servidor LoRaWAN de código abierto. Estas estaciones receptoras permitieron extender la cobertura del sistema, asegurando que todo el perímetro de la granja estuviera cubierto.

El sistema de monitoreo fue integrado con Home Assistant, que proporcionó una plataforma de visualización en tiempo real de las posiciones de los animales. A través de esta interfaz, los administradores de la granja pueden recibir alertas si algún animal se acerca a la zona límite, permitiendo un control remoto del sistema y facilitando la gestión de la seguridad del ganado.

**Tabla 2.2** Especificaciones técnicas de los dispositivos utilizados

<b>Dispositivo</b>	<b>Modelo</b>	<b>Especificaciones técnicas clave</b>
<i>Módulo principal</i>		<ul style="list-style-type: none"> <li>- MCU: STM32L0</li> <li>- LoRa: SX1262</li> </ul>
<i>LoRa</i>	CubeCell HTCC-AB02	<ul style="list-style-type: none"> <li>- Frecuencia: 868/915 MHz</li> <li>- GPS integrado</li> <li>- Batería LiPo</li> </ul>
<i>Módulo GPS</i>	NEO-6M	<ul style="list-style-type: none"> <li>- Precisión: &lt; 2.5 m</li> <li>- Voltaje: 3–5 V</li> <li>- Interfaz: UART</li> <li>- Frecuencia: 1 Hz</li> </ul>
<i>Gateway</i>	Raspberry Pi 4	<ul style="list-style-type: none"> <li>- CPU: Quad-core ARM Cortex-A72</li> <li>- RAM: 2–8 GB</li> </ul>
<i>LoRaWAN</i>		<ul style="list-style-type: none"> <li>- Puerto SPI para LoRaHat</li> <li>- Sistema operativo: Raspberry Pi OS</li> </ul>
<i>Software de red</i>	ChirpStack	<ul style="list-style-type: none"> <li>- Código abierto</li> <li>- Soporte para múltiples dispositivos</li> <li>- Compatible con MQTT y HTTP</li> </ul>
<i>Dashboard de usuario</i>	Home Assistant	<ul style="list-style-type: none"> <li>- Plataforma domótica de código abierto</li> </ul>

- 
- Interfaz personalizable
  - Soporte para MQTT
- 

En cuanto al componente algorítmico de Machine Learning, se implementó un modelo LSTM (Long Short-Term Memory) para predecir la proximidad futura de los animales a la zona segura. Este tipo de red neuronal es especialmente eficaz para el tratamiento de datos secuenciales, como los provenientes de sensores de movimiento y señales inalámbricas, debido a su capacidad para retener información a largo plazo en series temporales (Sherstinsky, 2020).

El modelo fue entrenado en una Raspberry Pi 4B, utilizando datos históricos recolectados durante pruebas de campo, los cuales incluían valores de RSSI transmitidos por los nodos LoRa, así como variables ambientales como temperatura y humedad. Estudios recientes han demostrado que es posible realizar el entrenamiento y despliegue de modelos LSTM en dispositivos de bajo consumo como la Raspberry Pi para tareas de predicción basadas en RSSI. Estudios recientes han demostrado que es posible realizar el entrenamiento y despliegue de modelos LSTM en dispositivos de bajo consumo como la Raspberry Pi para tareas de predicción basadas en datos secuenciales. Por ejemplo, el sistema tinyRadar empleó una LSTM en una Raspberry Pi 4 para clasificación de actividades humanas en tiempo real, alcanzando una precisión del 93 % con baja latencia (Yadav, Anand, D, Nikitha, & Thakur, 2023).

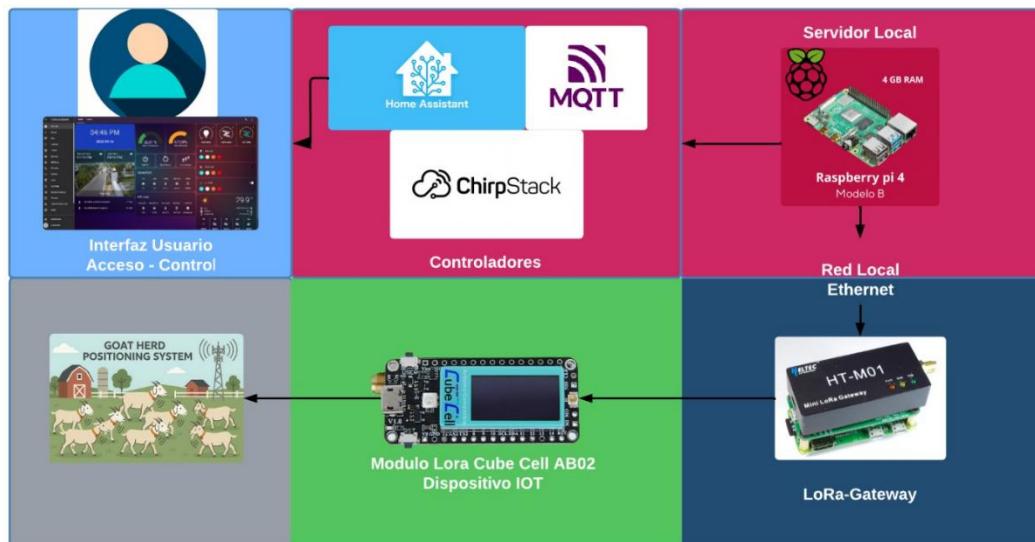
### 2.3 Diseño del Sistema

El diseño del sistema se centra en la integración de varios componentes clave para crear una red de monitoreo eficiente y precisa como se observa en la **Figura 2.2**. En primer

lugar, los nodos LoRa instalados en los collares de los animales funcionan como dispositivos emisores de señales, que envían periódicamente datos de RSSI a los Gateway receptores, ubicados en puntos estratégicos dentro del área de delimitación de la granja.

Estos receptores están ubicados en puntos estratégicos dentro del área de delimitación de la granja como: zonas de alimentación, bebederos o rutas de desplazamiento frecuente del ganado, con el fin de asegurar una cobertura óptima y detección precisa del movimiento.

Estas estaciones LoRaWAN (Gateways), administradas mediante el software ChirpStack, captan las señales y las envían a un servidor local donde se procesan para determinar si el animal se encuentra dentro o fuera de la zona segura. El sistema se configura de manera que, si el RSSI cae por debajo de un umbral predefinido, se genera una alerta para notificar al administrador de la granja.



**Figura 2.2** Sistema de monitoreo propuesto

El sistema también integra Home Assistant como plataforma de monitoreo, que recibe las alertas a través del protocolo de mensajería MQTT y presenta los datos en un dashboard

interactivo. Los administradores pueden visualizar la posición de cada animal, recibir alertas de proximidad y configurar los parámetros de la zona segura, todo desde una interfaz centralizada. Esta plataforma también ofrece la opción de configuración remota y ajustes en tiempo real, lo que permite optimizar el sistema a medida que se recopilan más datos.

En términos de la predicción y análisis de la posición del ganado, el sistema utiliza un modelo de red neuronal LSTM (Long Short-Term Memory), diseñado para procesar secuencias de datos temporales como los valores de RSSI (Indicador de Intensidad de Señal Recibida) y parámetros ambientales relacionados.

Este tipo de red es particularmente eficaz cuando se requiere capturar patrones a lo largo del tiempo, como el movimiento repetitivo de los animales o sus cambios de comportamiento ante ciertas condiciones del entorno. El modelo predice la proximidad futura del ganado y ajusta las alertas en función de la evolución de las señales captadas. Su eficacia depende de una correcta correlación entre los datos históricos y las condiciones actuales; por ejemplo, funciona bien cuando el entorno tiene patrones relativamente estables o repetitivos.

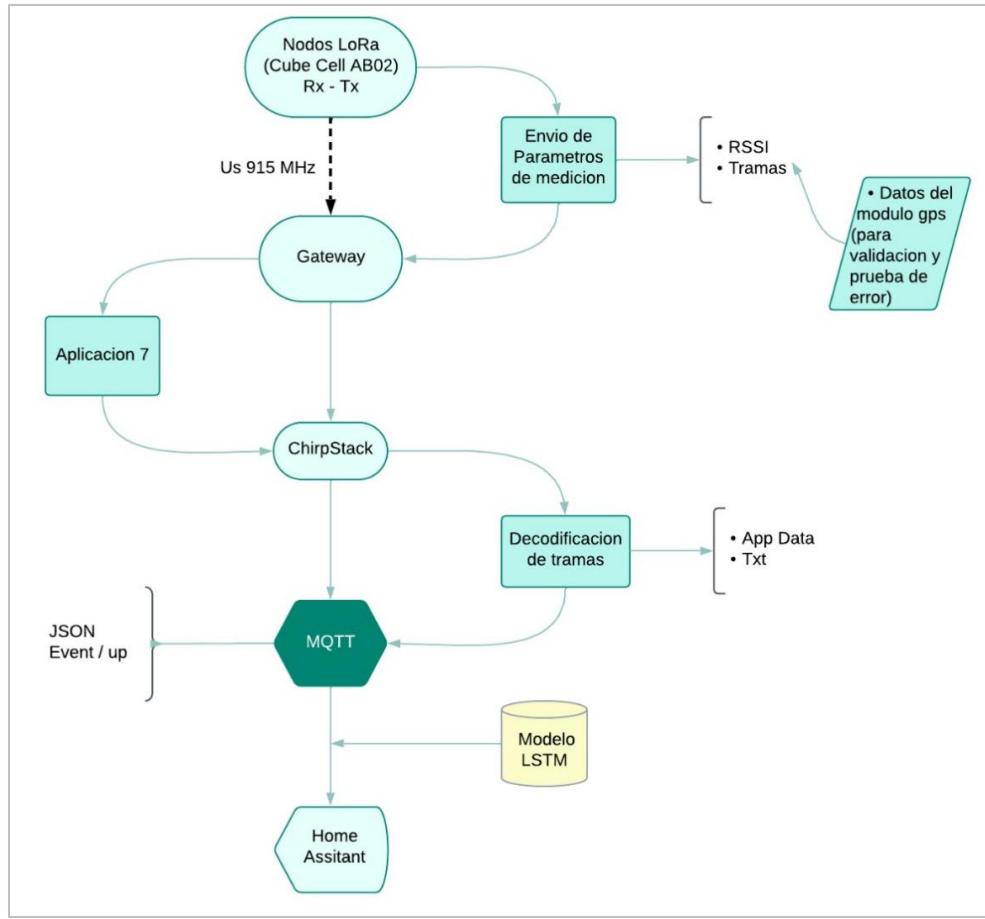
Sin embargo, podría fallar si existen interferencias en la señal RSSI, si los datos de entrada tienen demasiado ruido, o si ocurren cambios abruptos en el ambiente que no se hayan representado durante el entrenamiento. Estudios recientes han demostrado que este enfoque puede aplicarse de forma eficiente en dispositivos de bajo consumo como Raspberry Pi, logrando una predicción fiable de proximidad mediante análisis secuencial de RSSI (Yadav, Anand, D, Nikitha, & Thakur, 2023).

Este diseño no solo mejora la precisión del sistema, sino que también permite que el sistema se adapte de manera dinámica al comportamiento de los animales y las fluctuaciones en las condiciones de propagación de la señal.

## 2.4 Desarrollo del Sistema

El sistema desarrollado tiene como objetivo principal monitorear en tiempo real la posición de los chivos en la granja de ESPOL, utilizando tecnología LoRa para la transmisión de datos. Se diseñó una arquitectura compuesta por nodos, un gateway LoRa, una red central basada en ChirpStack y un sistema de visualización integrado en Home Assistant.

Adicionalmente, se incorporó un modelo predictivo basado en redes neuronales LSTM para estimar la posición de los animales a partir de parámetros como el RSSI como se observa en la **Figura 2.3**.



**Figura 2.3** Etapas del sistema de posicionamiento

El flujograma presentado describe el recorrido completo de los datos desde su origen en los dispositivos colocados en los animales hasta su visualización e interpretación por el usuario. El proceso comienza con los nodos LoRa, específicamente módulos Cube Cell AB02, que están equipados con capacidades de transmisión y recepción (Rx – Tx). Estos dispositivos envían información utilizando la frecuencia US 915 MHz, y los datos transmitidos incluyen tanto la intensidad de la señal (RSSI) como tramas estructuradas que contienen parámetros de medición, como temperatura, humedad y datos GPS. Estos últimos se utilizan principalmente como referencia para validar el rendimiento del sistema y detectar posibles errores en la predicción.

La señal enviada desde los nodos es captada por un gateway, el cual actúa como receptor central del sistema LoRa. Su función es recibir la información transmitida por los nodos y redirigirla a través de red IP a los sistemas encargados de su procesamiento. En este punto, interviene la plataforma ChirpStack, una herramienta de gestión de redes LoRaWAN que se encarga de decodificar las tramas recibidas. Esta decodificación genera dos tipos de datos: uno estructurado para aplicaciones (App Data) y otro en texto plano (Txt), ambos fundamentales para alimentar los sistemas de análisis y visualización.

Una vez que los datos han sido tratados, son enviados a través del protocolo MQTT. Este protocolo ligero y eficiente permite transmitir la información codificada en formato JSON, específicamente en paquetes del tipo “event/up”, lo cual facilita su suscripción por distintos servicios y plataformas en tiempo real.

Uno de los componentes que consume esta información es el modelo LSTM (Long Short-Term Memory), un tipo de red neuronal recurrente entrenada previamente con datos históricos de RSSI y condiciones ambientales. Este modelo analiza la secuencia temporal de señales para detectar patrones en el comportamiento del ganado, y en función de ellos, puede predecir su proximidad futura a zonas definidas como seguras o restringidas. La capacidad del modelo para adaptarse a cambios graduales en el entorno hace posible ajustar las alertas de forma dinámica

Finalmente, el resultado del análisis se integra con la plataforma Home Assistant, que permite al usuario monitorear en tiempo real la posición y comportamiento del animal. Desde allí, también se pueden configurar acciones automáticas, emitir alertas visuales o sonoras y generar informes. Todo este flujo convierte al sistema en una herramienta no solo de

monitoreo, sino también de predicción y gestión inteligente de los animales dentro del área determinada.

#### ***2.4.1 Implementación de Nodos Lora***

Se utilizaron dispositivos CubeCell AB02 con capacidad de transmisión y recepción (Rx-Tx), los cuales fueron montados en collares que portan los chivos. Estos dispositivos incluyen un módulo GPS integrado y son capaces de enviar periódicamente datos de posicionamiento y parámetros de red como el RSSI, utilizando la banda ISM US915 MHz.

#### ***2.4.2 Captura y Envío de Parámetros de Medición***

Cada nodo LoRa instalado en los collares de los chivos recopila y transmite de forma periódica datos esenciales como las coordenadas GPS y el nivel de señal RSSI, empaquetados en tramas que también incluyen identificadores para rastrear posibles errores.

Estos datos se envían mediante la banda de 915 MHz, aprovechando las ventajas del protocolo LoRa para lograr una comunicación de largo alcance y bajo consumo energético, adecuada para entornos rurales. Esta información es recibida por el gateway, dando paso al siguiente nivel de procesamiento dentro del sistema.

#### ***2.4.3 Gateway y Red LoRaWan***

El gateway actúa como puente entre los nodos LoRa y la red central, capturando las tramas enviadas por los collares a través de la frecuencia 915 MHz. Una vez recibidas, estas tramas son reenviadas al servidor de red ChirpStack, el cual gestiona la arquitectura LoRaWAN, autentica a los dispositivos y organiza los paquetes entrantes según la aplicación correspondiente.

Esta etapa garantiza la continuidad del flujo de información y la correcta administración de los dispositivos registrados en la red.

#### ***2.4.4 Decodificación de Tramas***

Una vez que ChirpStack ha recibido y organizado las tramas, se ejecuta un proceso de decodificación mediante un script personalizado (habitualmente en JavaScript) que traduce los datos en bruto a información legible como latitud, longitud y valores de RSSI.

Esta decodificación convierte los bytes en variables estructuradas que luego se empaquetan en formato JSON. Este paso es clave para transformar los datos técnicos en información útil para su visualización o análisis posterior.

#### ***2.4.5 Comunicación con MQTT***

Una vez decodificados los datos por ChirpStack, estos se publican automáticamente en un broker MQTT bajo el tema event/up, utilizando el formato JSON. MQTT, un protocolo de mensajería ligero y eficiente para dispositivos IoT, permite distribuir la información de forma flexible a múltiples suscriptores en tiempo real.

Gracias a esta arquitectura, aplicaciones como Home Assistant y módulos de análisis predictivo pueden acceder a los datos sin interferir en el flujo de comunicación del sistema.

#### **2.4.6 Incorporación del Modelo Predictivo LSTM**

Con el objetivo de predecir la ubicación de los chivos en función de la intensidad de señal, se integró un modelo de red neuronal tipo LSTM (Long Short-Term Memory) entrenado exclusivamente con datos históricos de RSSI. Este modelo permite que el sistema aprenda a estimar la posición de los animales sin depender de las coordenadas GPS, lo cual es especialmente útil en zonas con cobertura limitada o interferencias.

El modelo se implementa en la misma Raspberry Pi que aloja Home Assistant y se conecta al flujo de datos mediante MQTT, generando predicciones que complementan la información real capturada por los nodos.

#### **2.4.7 Visualización en Home Assistant**

Home Assistant, una plataforma de automatización de código abierto (Team, 2024) se configura como suscriptor del canal MQTT para recibir los datos emitidos por los nodos. Gracias a la integración de sensores MQTT, puede procesar mensajes (estado o atributos JSON) incluyendo valores como RSSI, ubicación y otros datos ambientales (Assistant, Home Assistant - MQTT Sensor, 2024) (Assistant, Home Assistant - Device Tracker, 2024).

Una vez integrados, los valores de ubicación, RSSI y predicción de posicionamiento, son visualizados a través de dashboards personalizados, que permiten al usuario monitorear la posición de cada chivo en tiempo real y mostrar alertas si salen del área limitada. Esta visualización facilita la toma de decisiones rápidas y el seguimiento del rebaño de manera intuitiva. Además, con la última actualización de Home Assistant se tienen mejoras en la creación de dashboards, añadiendo diseños intuitivos y mayor flexibilidad en la organización de la información añadida.

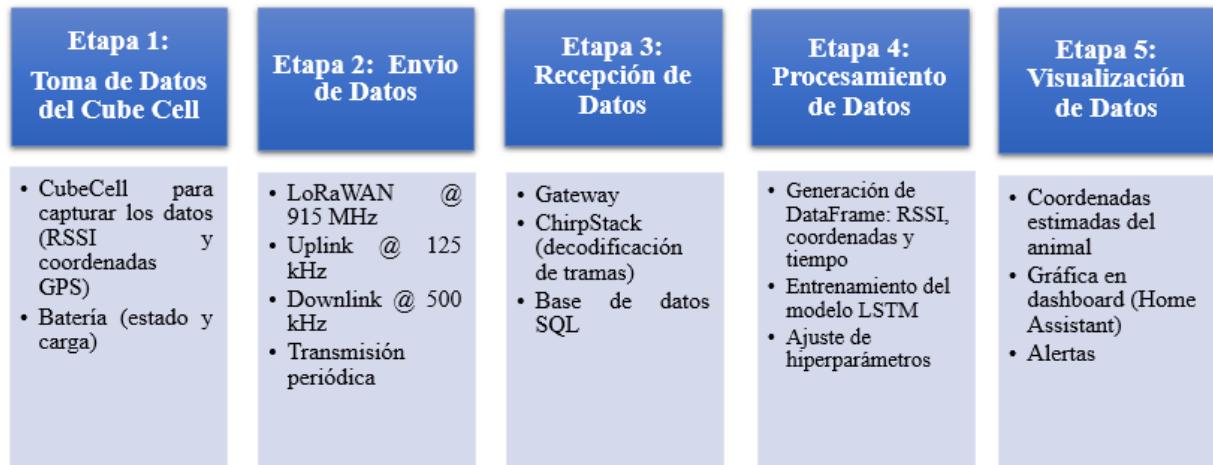
# Capítulo 3

### 3 Resultados y Análisis

En este capítulo se mostrarán los primeros resultados obtenidos a partir de pruebas realizadas con los módulos Lora Cube Cell que fueron colocados en puntos fijos dentro del área definida. Las coordenadas registradas permitieron construir una base de datos SQL que sirve como punto de partida para el análisis de los datos recogidos.

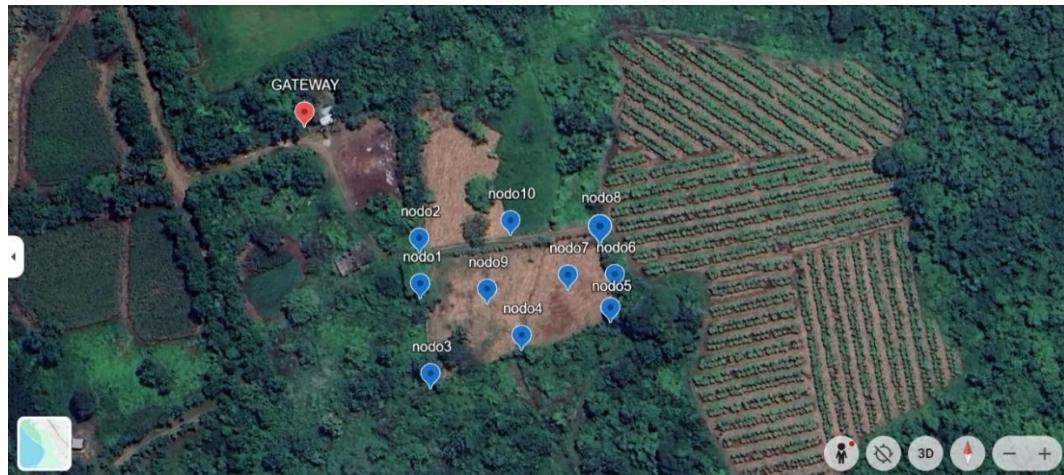
#### 3.1 Desarrollo del Sistema

Para el entendimiento del sistema se presenta un diagrama de bloques con el orden de los pasos aplicados para el desarrollo del proyecto (**Figura 3.1**) en el cual se identifican 5 etapas.



*Figura 3.1 Diagrama de Bloques del proyecto*

En la primera etapa se recolectan datos del CubeCell ubicados en 10 puntos dentro del área definida (**Figura 3.3**) para la cerca virtual. Su función es capturar dos tipos de información: potencia de señal recibida (RSSI) y las coordenadas. Estos datos permiten saber a qué distancia se encuentra un nodo del Gateway y su posición geográfica tal como se aprecia en la **Figura 3.2** la cual es una captura del área con la posición de los nodos referente al Gateway en la aplicación de Google Earth.



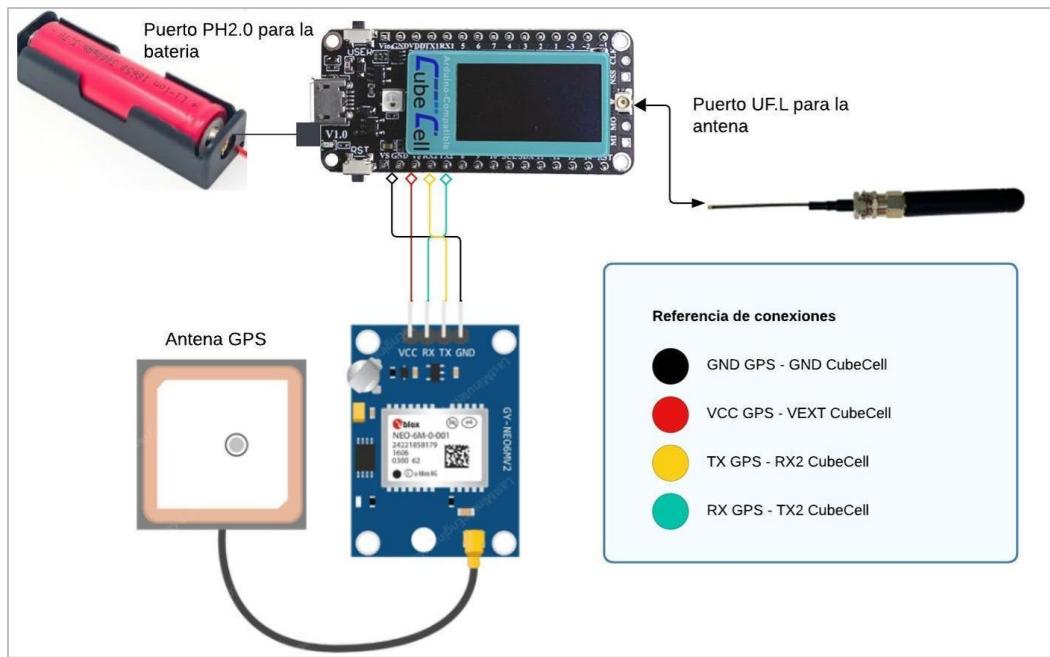
**Figura 3.2** Área geográfica con la posición del Gateway y los nodos



**Figura 3.3** Vista del terreno donde estan ubicados los nodos estáticos

En la etapa 2 se tiene el envío de datos recolectado por el CubeCell en donde se aplica la tecnología LoRaWAN (Long-Range) que permite la transmisión de datos a largas distancias sin consumir mucha energía. Este proceso se da en dos canales: uplink a 125kHz

de ancho de banda y downlink a 500kHz a una frecuencia de 915MHz. La transmisión se da cada 30 segundos por cada nodo, luego el módulo entra en sleep (duración de 1 min aproximadamente) y vuelve a tomar datos.



*Figura 3.4 Conexiones de Alimentación*

Se visualiza en la **Figura 3.4** el esquema de conexión entre el módulo CubeCell y el Módulo GPS NEO-6M junto con la batería (fuente de voltaje) y antena LoRa. Los nodos LoRa empleados son HTCC AB02 los cuales para las mediciones estuvieron conectados a la batería y a la antena omnidireccional de 915MHz conectada al puerto UF. L en el CubeCell.

Para la etapa 3 se tiene la información enviada por el CubeCell, GPS y batería que es recibida por el Gateway que es un HTM-01. Este Gateway se encuentra conectado a ChirpStack el cual es el encargado de interpretar y decodificar los paquetes de datos recibidos.

Una vez decodificados los datos se almacenan en una base de datos SQL que permite mantener un registro organizado para consultas y análisis posteriores.

Para mejor entendimiento se presenta una tabla (**Tabla 3.1**) con las coordenadas (latitud, longitud y cantidad de satélites en el punto) recogidas y su hora respectiva de cada lectura.

*Tabla 3.1 Datos de lectura del GPS en el Área definida*

Lectura	Hora	Coordenadas
Primera	10:47am	Lat: -2.145489 Lon: -79.960556 Sat: 5
Segunda	10:49am	Lat: -2.139788 Lon: -79.960689 Sat: 8
Tercera	10:51am	Lat: -2.139962 Lon: -79.9604857 Sat: 7
Cuarta	10:53am	Lat: -2.139948 Lon: -79.962584 Sat: 9
Quinta	10:55am	Lat: -2.140045 Lon: -79.96 Sat: 7
Sexta	10:57am	Lat: -2.140027 Lon: -79.96026 Sat: 7
Séptima	10:58am	Lat: -2.140077 Lon: -79.960323 Sat: 9
Octava	11:00am	Lat: -2.140054 Lon: -79.96448

		Sat: 11
Novena	11:02am	Lat: -2.140201 Lon: -79.960410 Sat: 8
Decima	11:04am	Lat: -2.140150 Lon: -79.960356 Sat: 8

Estas lecturas se almacenan en la base de datos que tiene por defecto Home Assistant, aquí llegan todos los datos recogidos por los nodos que están integrados en la interfaz. Los datos que son recolectados por los nodos (puntos ubicados dentro del área) se almacenan en una base de datos SQLite que viene integrada en Home Assistant (**Figura 3.5**).

The screenshot shows the DB Browser for SQLite interface with a database named 'home-assistant\_v2\_20250718\_14h45.db'. The 'state\_attributes' table is selected, displaying a list of rows with columns: attributes\_id, hash, and sharedAttrs. The 'sharedAttrs' column contains JSON objects representing sensor data. An 'Edit Database Cell' dialog is open on the right, showing a single row's data in JSON format. The JSON object includes fields like applicationID, applicationName, deviceName, devEUI, latitude, longitude, and various sensor values. The dialog also shows the 'Mode: JSON' dropdown and some status information at the bottom.

**Figura 3.5 Base de Datos SQL**

En la cuarta etapa se tiene el procesamiento de datos que consiste en organizar una estructura de tipo DataFrame con el objetivo de usar una red neuronal, en específico el

modelo LSTM (Long Short-Term Memory) para entrenar este modelo y que aprenda patrones, comportamientos y movimientos de los animales.

Durante la realización de esta etapa de entrenamiento del modelo de ML (Machine Learning) se aplican ajustes de parámetros para mejorar la precisión del modelo y de esta manera asegurar que las predicciones resulten en lo más exactas posibles.

Para la última etapa se presentan los resultados del modelo traducidos a información legible para el usuario final. Se generan y se visualizan las coordenadas en tiempo real del animal junto con el rssi de su posición en el dashboard de Home Assistant.

De igual manera se podrá ver su ubicación en un mapa geográfico y el sistema genera alertas si el animal cruza el área permitida, facilitando así el monitoreo remoto y una toma de decisiones rápida por parte del personal encargado. Esta etapa convierte los datos complejos en resultados claros y visuales para tomar acciones precisas.

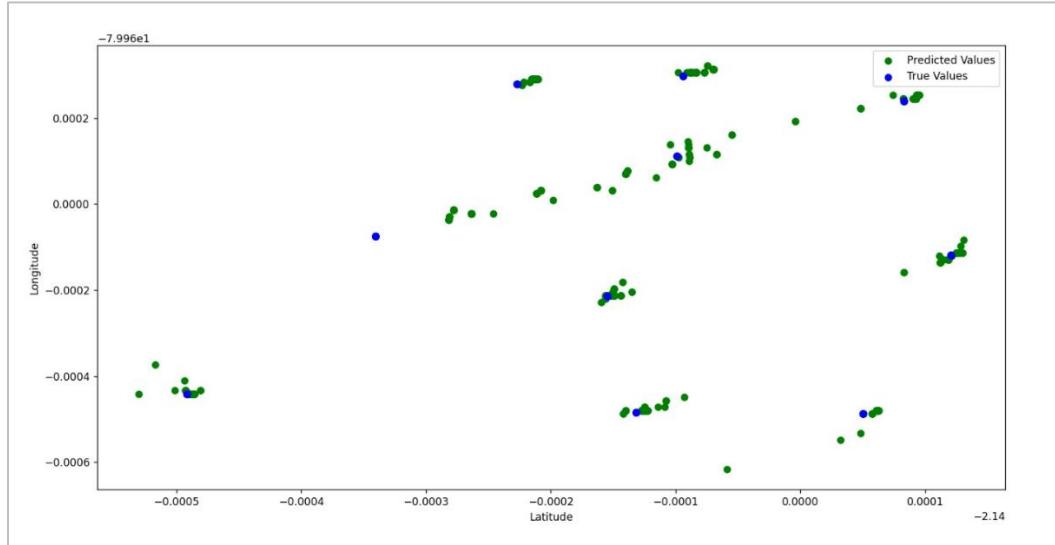
### 3.2 Entrenamiento del Modelo con Machine Learning

Para lograr un sistema con localización con resultados confiables era necesario contar con un modelo que interprete la relación entre el RSSI (intensidad de la señal) y la posición real de los animales, ya que la señal se ve afectada por muchos factores como el movimiento, las condiciones ambientales y la línea de vista en relación al Gateway.

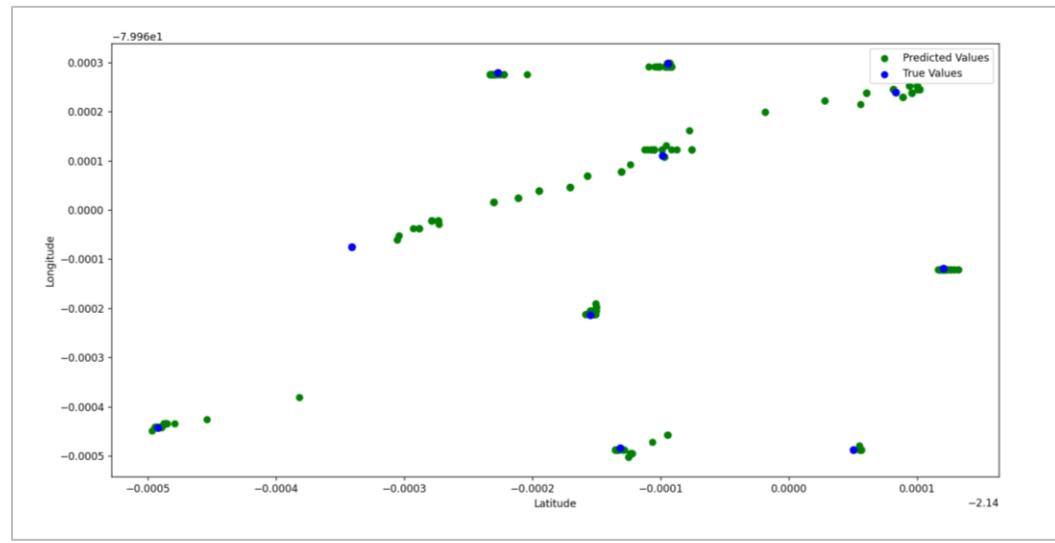
Se optó por el uso de técnicas de **Machine Learning** (ML) de aprendizaje no supervisado el cual se basa en encontrar patrones ocultos a través de la organización de datos similares con el objetivo de entrenar modelos a partir de datos reales recogidos en el área de trabajo, aprender patrones de comportamiento de la señal y predecir coordenadas con mayor

precisión. Esta implementación ofrece mejorar de manera progresiva la precisión del sistema según más datos vaya recibiendo, convirtiéndola así en una base sostenible para un cercado virtual eficiente.

### 3.2.1 Entrenamiento con Distancia



*Figura 3.6 Entrenamiento del modelo con ANN*

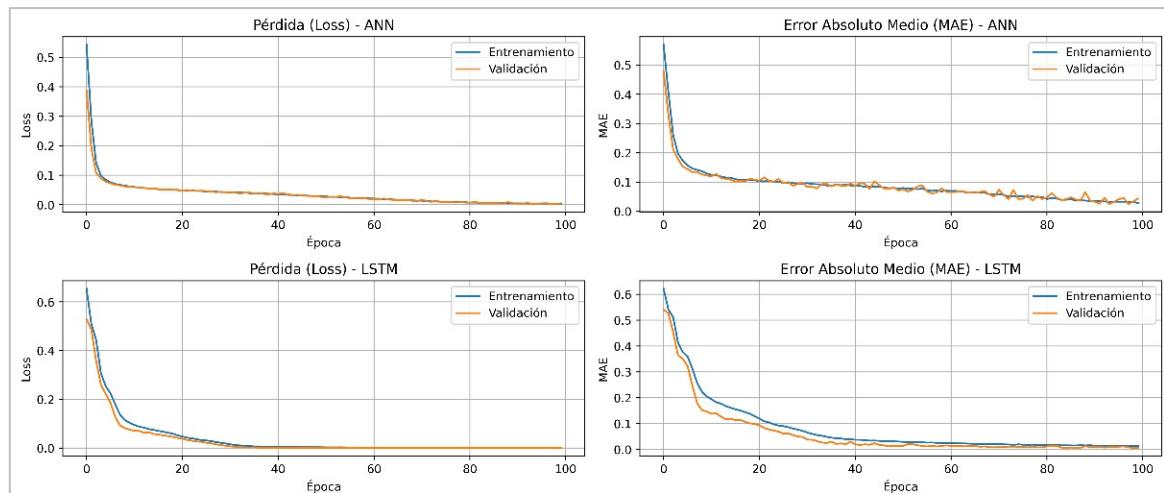


*Figura 3.7 Entrenamiento del modelo con LSTM*

Para ambos entrenamientos se usaron dos capas: una de 40 neuronas y la segunda de 20 neuronas que permitió capturar las relaciones temporales presentes en los datos. El

entrenamiento que incorpora a la distancia como variable arroja un mejor rendimiento tanto para el modelo ANN como para el LSTM. Las entradas para este entrenamiento fueron: distancia, RSSI, latitud y longitud (Gateway) y de salida: latitud y longitud referente a los nodos estáticos.

El modelo ANN presentado en la **Figura 3.6** alcanzo un coeficiente  $R^2 = 0.945$  con un error promedio de *3.79metros* en la predicción de coordenadas. Por otra parte, el modelo LSTM (**Figura 3.7**) presenta un mejor desempeño con  $R^2 = 0.953$  con un error promedio de *3.02metros* en la predicción de coordenadas.



**Figura 3.8** Curvas de Entrenamiento

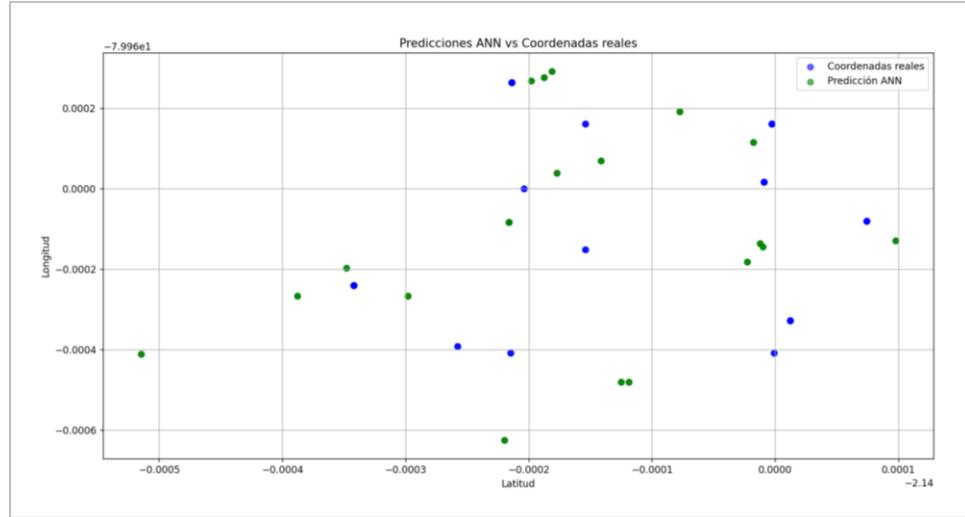
En la **Figura 3.8** se pueden observar las curvas de entrenamiento y validación de dos modelos, en la parte superior ANN y de la parte inferior LSTM, usados para predecir la posición de los 10 nodos estáticos que fueron usados para el entrenamiento. Los gráficos muestran cómo evolucionan dos métricas clave: la pérdida (Loss) y el Error Absoluto Medio (MAE) a lo largo de 100 épocas de entrenamiento.

La curva de pérdida del modelo ANN baja rápidamente en las primeras épocas, lo cual es una buena señal. Después de unas 30 épocas, se estabiliza, y aunque la pérdida de entrenamiento sigue mejorando, la de validación se mantiene constante. Esto sugiere que el modelo podría estar comenzando a sobre ajustarse un poco, ya que no mejora tanto con los datos de validación.

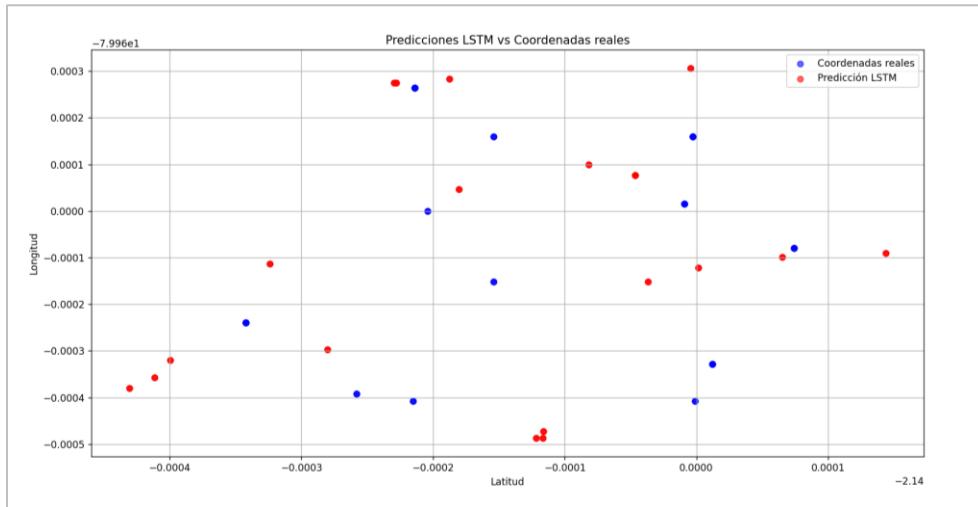
En cuanto al MAE, se ve una tendencia similar. El error de entrenamiento es siempre más bajo que el de validación, lo que refuerza la idea de que el modelo ANN podría estar ajustándose demasiado a los datos de entrenamiento y no generalizando lo suficientemente bien.

Para el modelo LSTM, la pérdida disminuye de forma más pronunciada, y lo interesante es que las curvas de entrenamiento y validación siguen una trayectoria casi idéntica. Esto indica que el modelo LSTM tiene un buen rendimiento en general y parece estar adaptándose mejor a los datos, sin los indicios de sobreajuste que se ven en el modelo ANN.

El MAE del modelo LSTM también muestra una caída estable, con una pequeña diferencia entre los datos de entrenamiento y validación. Esto sugiere que el modelo LSTM está generalizando de manera eficiente y no está sobre ajustándose.



**Figura 3.9** Predicción de Coordenadas Reales ANN



**Figura 3.10** Predicciones de Coordenadas Reales LSTM

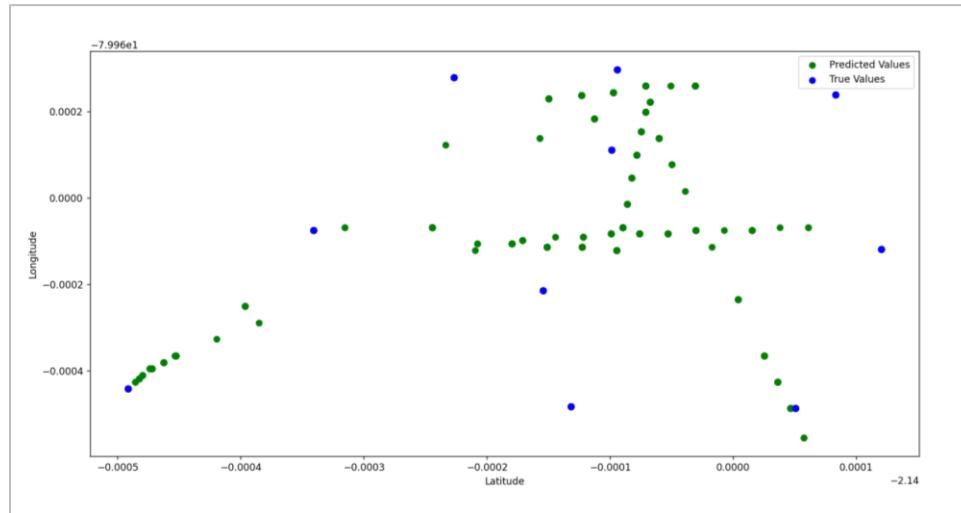
Se puede visualizar con claridad en la **Figura 3.9** y **Figura 3.10** el comportamiento de ambos modelos: ANN y LSTM frente a una ruta real usada como referencia para el entrenamiento. En ambas imágenes se visualiza que la trayectoria mantiene un recorrido próximo a las coordenadas reales con mínimas variaciones a lo largo de la ruta.

Para hacer una comparación se tiene que el modelo LSTM presenta una trayectoria más estable y menos dispersa en los puntos de predicción lo que evidencia la capacidad de

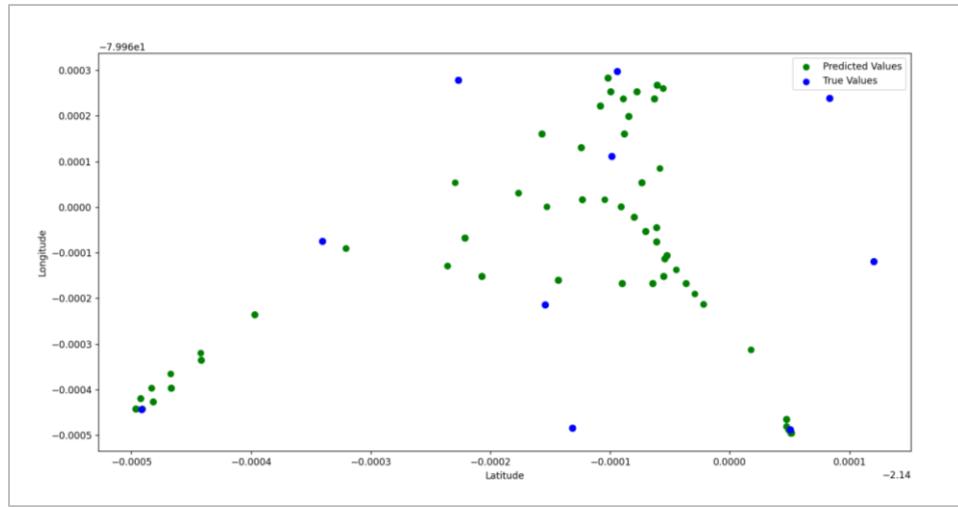
manejar secuencias temporales y “aprender” de patrones previos de variación de RSSI junto con la variable distancia.

El modelo ANN, aunque también es preciso, muestra pequeñas desviaciones que se traducen en menos estabilidad frente a la variación de la señal. Esta diferencia remarca ya que un sistema de localización basado en LSTM se adapta mejor a las variaciones producidas por el movimiento de los animales y las condiciones externas que se encuentran en constante cambio, son aspectos que se reflejan en los datos con los que el modelo siguió una ruta real.

### 3.2.2 *Entrenamiento sin Distancia*



**Figura 3.11** Entrenamiento del Modelo con ANN



**Figura 3.12** Entrenamiento del Modelo con LSTM

Para este entrenamiento en específico se eliminó la variable distancia y se entrenó únicamente con el valor de RSSI y en comparación al entrenamiento previo, el rendimiento se redujo significativamente. Las entradas para este entrenamiento fueron: RSSI, latitud y longitud (Gateway) y de salida: latitud y longitud referente a los nodos estáticos.

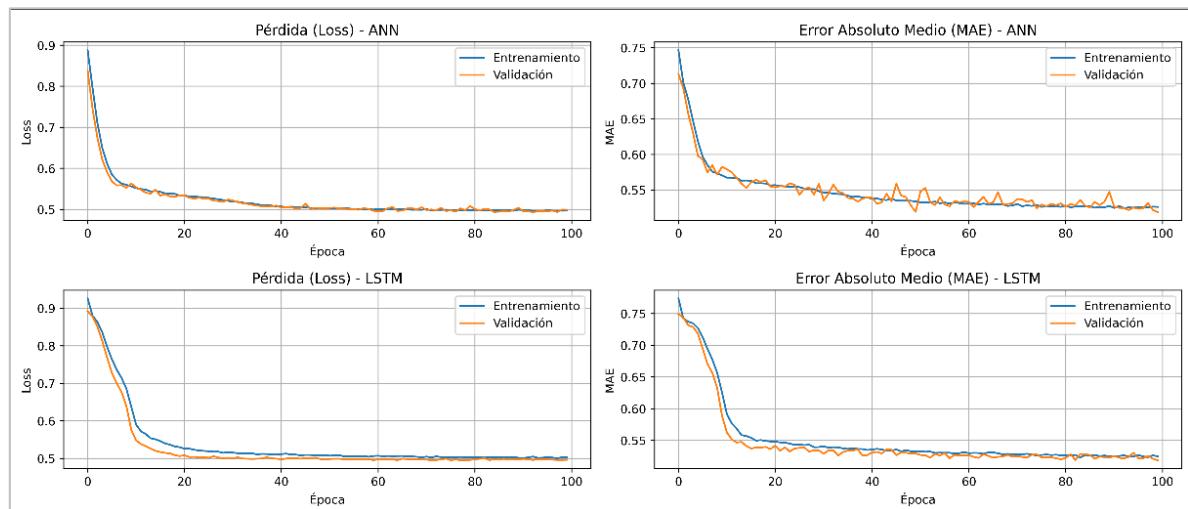
Se tiene en la **Figura 3.11** que el modelo ANN obtuvo un coeficiente de  $R^2 = 0.537$  y un error promedio de *21.84metros*; por otro lado, en el modelo LSTM presentado en la **Figura 3.12** alcanzo un coeficiente de  $R^2 = 0.579$  con un error promedio de *21.09metros*.

Ambas graficas muestran trayectorias dispersas e inestables con relacion a la ruta real. Esto confirma que el RSSI por sí solo no brinda información suficiente para estimar la ubicación con precisión ya que los obstáculos del camino, condiciones ambientales y reflexiones de señal representan una afectación.

Aunque el modelo LSTM siga mostrando un rendimiento ligeramente mejor respecto al modelo ANN para este entrenamiento, la ausencia de la variable distancia provoca que ambos modelos pierdan precisión considerablemente, resultado que se aprecia en las gráficas

al comparar su dispersión con respecto a las predicciones del entrenamiento realizado con distancia.

El rendimiento de LSTM frente al ANN es debido a la capacidad de manejar datos secuenciales a diferencia de las redes ANN que tratan cada entrada como independiente, el LSTM integra información temporal de lecturas previas lo que permite reconocer patrones y tendencias en la señal conforme cambia el recorrido de los animales, obstáculos y variaciones del entorno. Esto se traduce en una mejor estabilidad en las predicciones y una trayectoria más limpia y ajustada a la ruta real.



**Figura 3.13** Desempeño de los modelos

La **Figura 3.13** se observa el desempeño de dos modelos de redes neuronales en la tarea de predicción de posición de los 10 nodos estáticos que se usaron como entrenamiento. Se emplearon métricas de pérdida (Loss) y error absoluto medio (MAE), tanto para los conjuntos de entrenamiento como para validación, a lo largo de 100 épocas de entrenamiento.

El modelo ANN muestra una pérdida que comienza con un valor de aproximadamente 0.88 en la primera época (eje Y, parte superior izquierda). A lo largo de las primeras épocas,

la pérdida disminuye rápidamente, alcanzando un valor cercano a 0.5 al final del entrenamiento (época 100). Sin embargo, se observa una pequeña desviación en la curva de validación que permanece en torno a 0.55, lo que indica una ligera diferencia en el rendimiento entre entrenamiento y validación.

El MAE para el modelo ANN empieza en aproximadamente 0.75 al inicio del entrenamiento. Después, muestra una disminución significativa hasta llegar a valores cercanos a 0.55 al final de las 100 épocas, tanto en el conjunto de entrenamiento como en el de validación. Sin embargo, la curva de validación muestra oscilaciones pequeñas, lo que sugiere una mayor variabilidad en las predicciones de validación.

En el caso del modelo LSTM, la pérdida de entrenamiento también comienza en un valor cercano a 0.88 y desciende hasta 0.5 (parte inferior izquierda), similar al comportamiento del modelo ANN. Sin embargo, las curvas de entrenamiento y validación son más consistentes, manteniendo una diferencia menor entre sí, con la validación estabilizándose alrededor de 0.55 y el entrenamiento acercándose a 0.5.

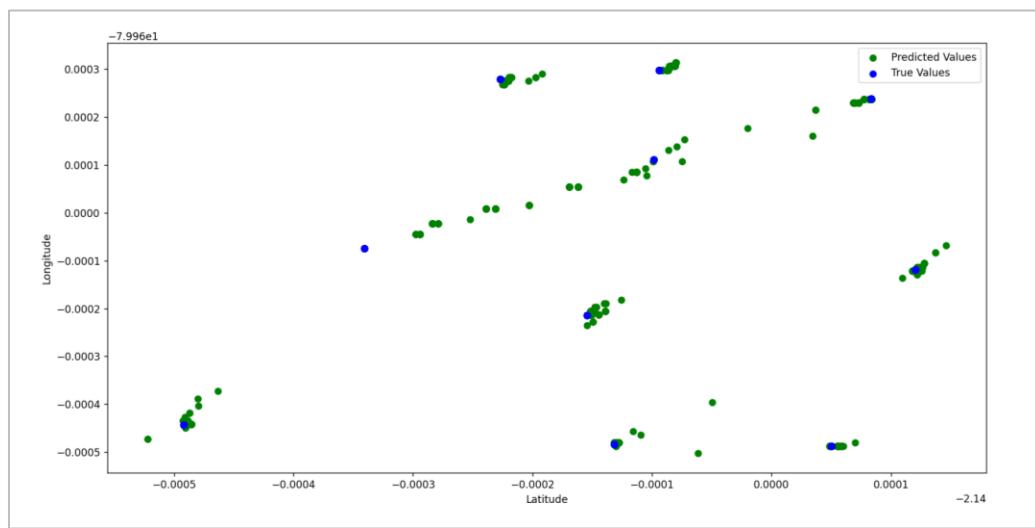
En el modelo LSTM, el MAE comienza en torno a 0.75 y desciende de manera similar, alcanzando un valor cercano a 0.55 tanto en el conjunto de entrenamiento como en el de validación (parte inferior derecha). La diferencia entre ambos conjuntos es mínima, lo que sugiere que el modelo generaliza mejor a lo largo del entrenamiento.

Por lo que se concluye que cuando se utiliza el parámetro distancia, el rendimiento de los modelos ANN y LSTM mejora notablemente. Los modelos entrenados con esta variable consiguen adaptarse mejor a los datos, reduciendo los errores y logrando una mejor capacidad para generalizar. Sin embargo, al eliminar este parámetro, especialmente en el

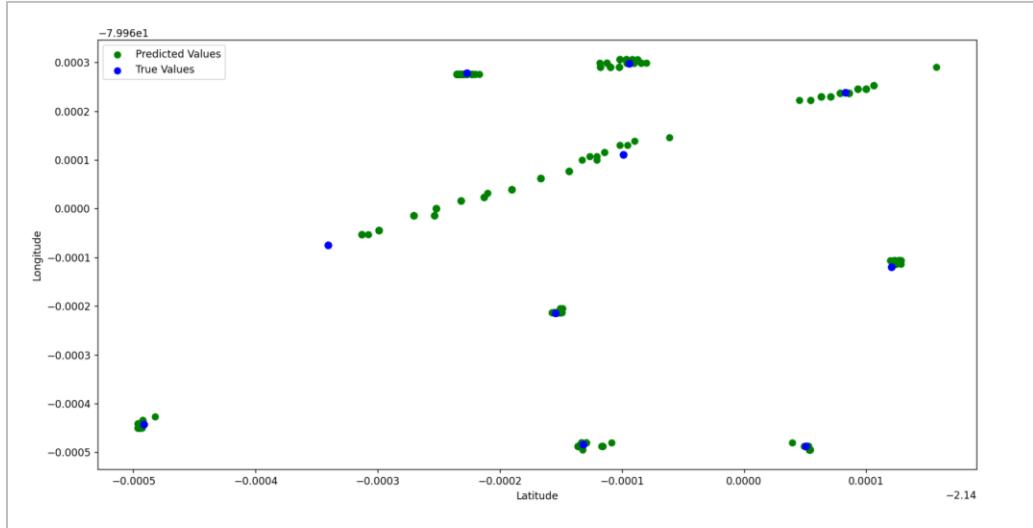
ANN, los modelos tienden a sobre ajustarse más y su rendimiento general disminuye. Esto muestra claramente la importancia de incluir características clave, como la distancia, para hacer las predicciones más precisas reduciendo el error promedio en metros de las coordenadas reales respecto a las predichas.

A continuación, se presenta una comparación del entrenamiento con distancia con más o menos datos (lecturas) recolectados para seguir mejorando el modelo de predicción.

### 3.2.3 *Entrenamiento con Distancia usando menos datos*



**Figura 3.14** Entrenamiento modelo ANN



**Figura 3.15** Entrenamiento modelo LSTM

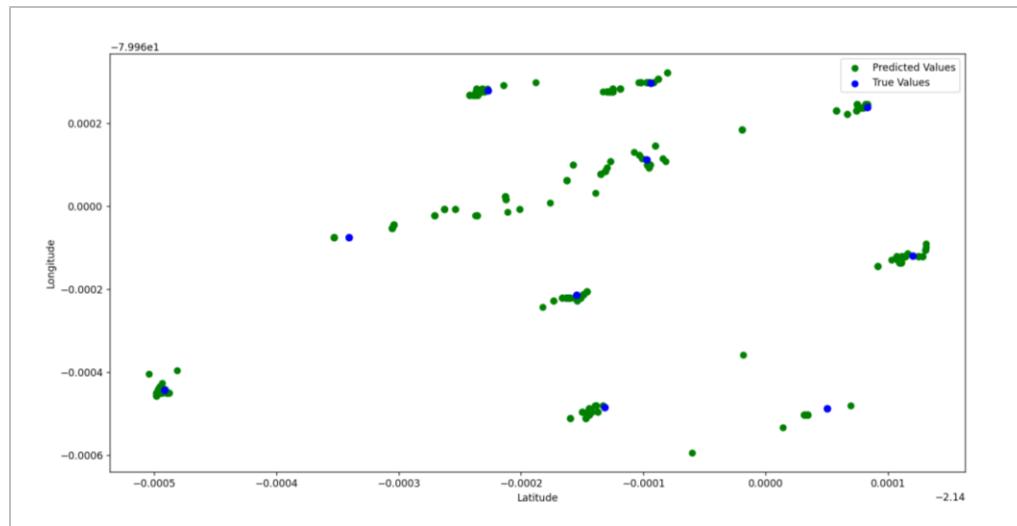
Se tiene un dataset de 2860 datos de 10 nodos estáticos recopilados durante varios días donde cada nodo enviaba datos cada 2 minutos.

El modelo ANN alcanzo un coeficiente de  $R^2 = 0.957$  con un error promedio de predicción de 3.39metros. La **Figura 3.14** muestra que la trayectoria predicha sigue la tendencia de la ruta real, pero con oscilaciones y posiciones que se desvían significativamente de la posición real; demostrando que a pesar de que el modelo trabaja bien con los datos de RSSI, distancia y coordenadas, su arquitectura no logra predecir con la misma precisión las variaciones que afectan la señal.

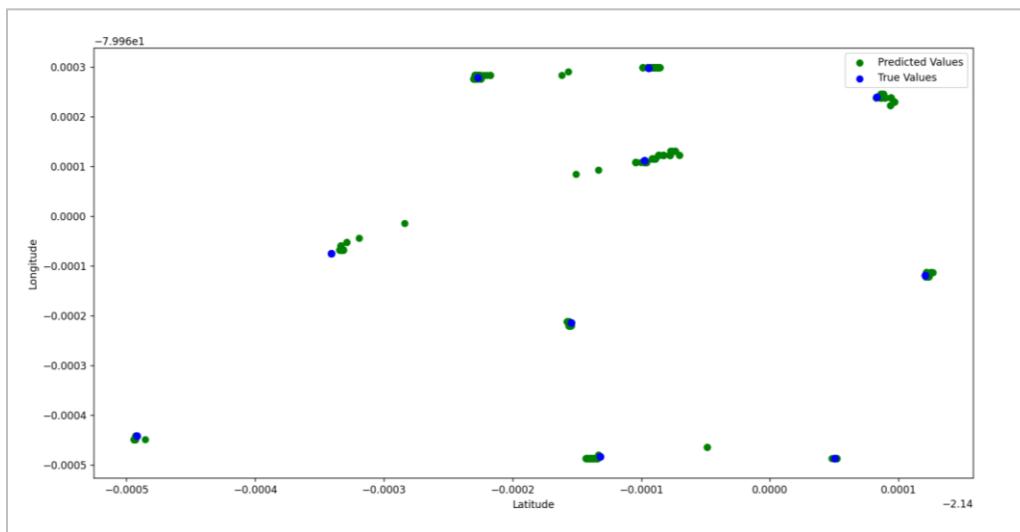
El modelo LSTM alcanzo un coeficiente de  $R^2 = 0.976$  con un error promedio de 2.69metros. La **Figura 3.15** muestra de igual manera la trayectoria predicha cercana a la ruta real, aunque también con mínimas desviaciones evidentes en ciertos tramos. Con una cantidad reducida de datos el modelo logro captar patrones temporales con mejor estabilidad que el ANN lo que reafirma la precisión del modelo.

Se aprecia en la misma grafica una menor dispersión en comparación al modelo anterior ya que la capacidad de LSTM para manejar datos secuenciales da paso a reconocer patrones en evolución de la señal RSSI y la distancia, resultando en predicciones más precisas.

### 3.2.4 Entrenamiento con Distancia usando más datos



*Figura 3.16 Entrenamiento modelo ANN*



*Figura 3.17 Entrenamiento modelo LSTM*

Para esta parte del entrenamiento se tiene un aumento significativo de datos, un dataset de 5000 registros recolectados de la misma forma que el entrenamiento anterior. Al aumentar los datos el modelo ANN mostrado en la **Figura 3.16** mejoró su desempeño alcanzando un  $R^2 = 0.966$  y un error promedio de *3.27 metros*. La grafica muestra una leve reducción en su dispersión de puntos, aunque siguen presentes las desviaciones en zonas donde la señal presenta cambios repentinos.

Esto demuestra que, aunque el incremento de datos ayuda a que el modelo mejore, sus limitaciones permanecen presentes.

Para el siguiente caso el modelo LSTM presento una mejora significativa al trabajar con más datos, alcanzando un coeficiente de  $R^2 = 0.996$  y su error promedio se redujo a *0.77 metros*. En la **Figura 3.17** se visualiza que la trayectoria predicha se superpone con la ruta real, sin desviaciones perceptibles. Esto refleja la capacidad que tiene el modelo para aprovechar el incremento de datos y así ajustar con mayor precisión los patrones de propagación de señal, logrando un incremento visiblemente optimo.

Se evidencia que, aunque ambos modelos mejoran con más datos, el LSTM se beneficia significativamente y su error promedio pasa de *2.69 metros* a *0.77 metros* mientras que el modelo ANN reduce el error de *3.39 metros* a *3.27 metros*.

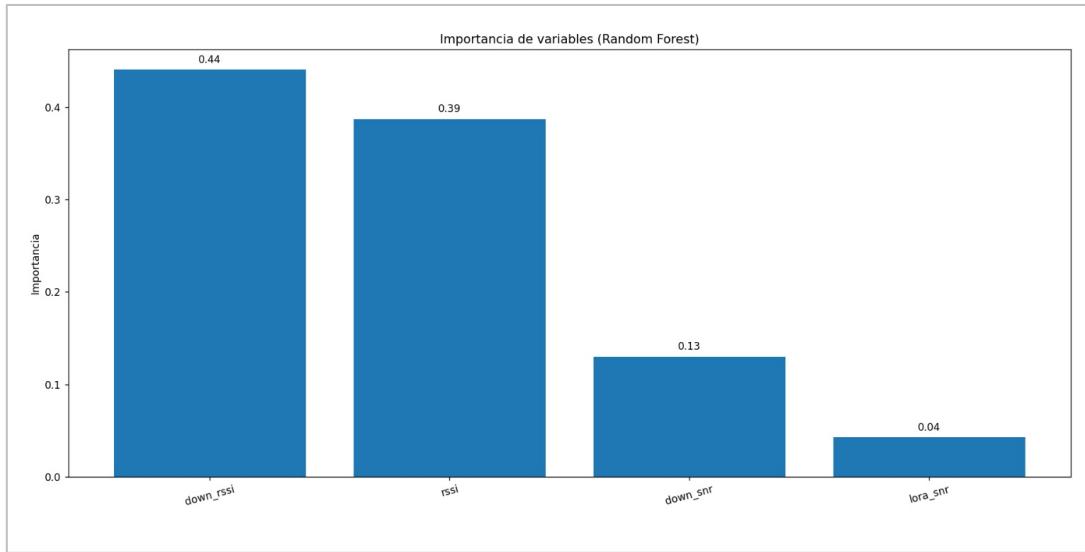
Esta diferencia reafirma que el modelo LSTM no solo tiene mejor arquitectura para este tipo de situaciones, sino que también responde de manera eficiente al incremento del volumen de datos, logrando que el sistema de localización sea más confiable y preciso para la realización del proyecto.

En conclusión, se decidió a usar el modelo LSTM debido a su capacidad de manejar un mayor volumen de datos lo que aporto a una mejora significativa en la predicción junto con el uso de la distancia. Posteriormente se entrenará el modelo con otro algoritmo llamado Random Forest lo que permitirá optimizar aún más su rendimiento y precisión en la predicción, resultando así en un mejor enfoque para el desarrollo del proyecto.

### **3.2.5 *Random Forest***

Como se demostró en modelos anteriores que el entrenamiento con la variable distancia y la cantidad de datos empleada (mientras más datos se emplean más preciso es el modelo) es significativa para la predicción de coordenadas; se usó un modelo adicional para poder estimar la distancia de cada nodo móvil (nodos colocados en los chivos) a través del RSSI de subida y RSSI de bajada para poder reducir el error de predicción. Esto es importante ya que la distancia no es parámetro conocido con los nodos móviles, por ende, la predicción de esta nos permite incluir la variable distancia en el modelo LSTM.

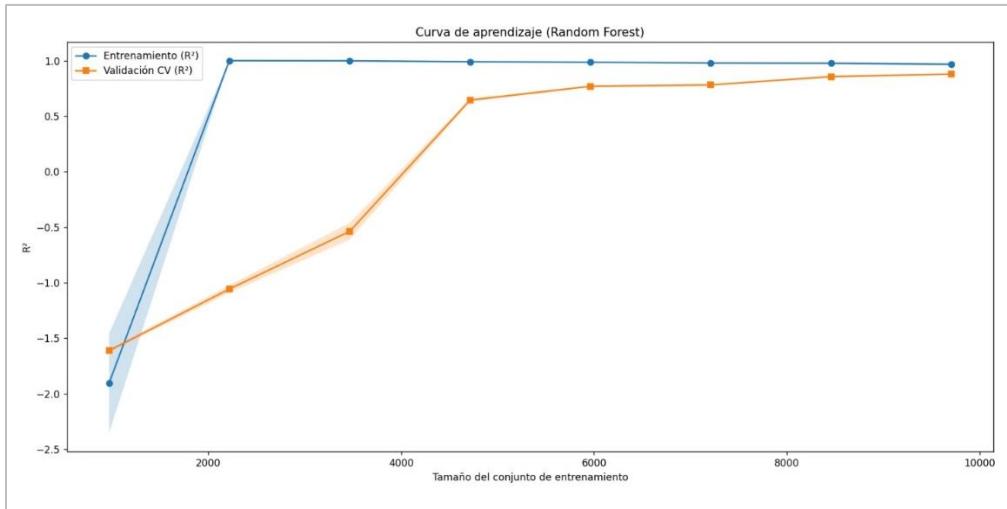
Se utiliza para la estimación de distancia el Modelo de Random Forest ya que la relación entre el RSSI de subida y bajada y la distancia real no siempre sigue una ecuación simple, porque la señal está afectada por ruido, obstáculos como vegetación y variaciones del entorno como irregularidad del terreno. El modelo permite aprender patrones complejos y no lineales a partir de datos reales, mejorando la precisión en la estimación de la distancia en comparación con un cálculo puramente teórico.



**Figura 3.18** Importancia de variables de entrenamiento

Para el entrenamiento se recopilaron 4 datos RSSI de subida, RSSI de bajada, SNR de subida y SNR de bajada como se muestra en la **Figura 3.18**. También se muestra la importancia de cada variable en el proceso de entrenamiento, esto con el fin de que en futuras muestras poder optimizar el uso de parámetros y proceso de entrenamiento.

El modelo se procesó con 12 mil datos recopilados durante 3 días aproximadamente de los 10 nodos estáticos, de los cuales un 80% se utilizó para entrenamiento, donde el algoritmo aprende la relación entre las señales y la distancia y 20% prueba donde se valida que tan bien generaliza el modelo.



**Figura 3.19** Curva de entrenamiento

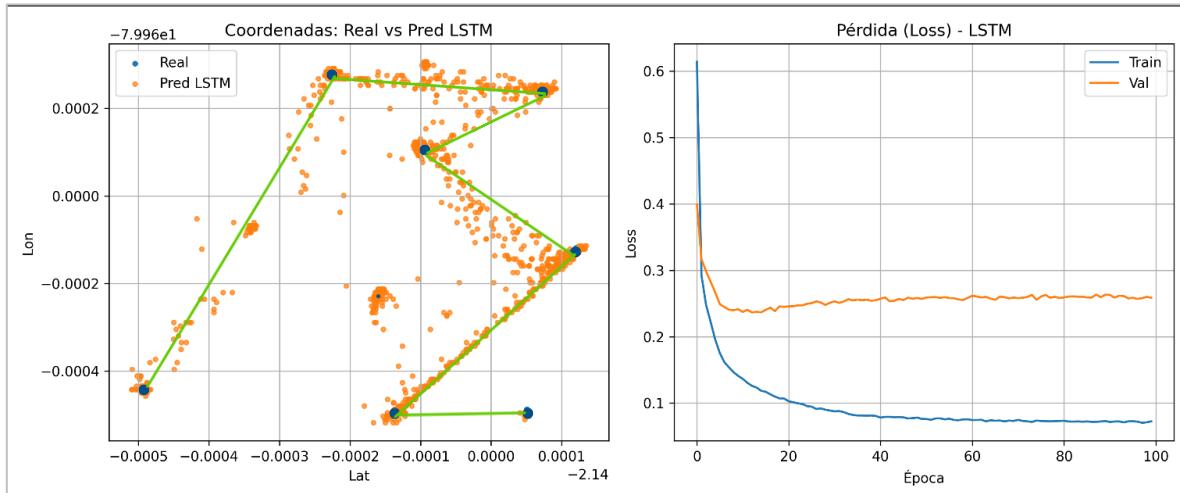
En la **Figura 3.19** se observa muestra una curva de aprendizaje de un modelo Random Forest, donde en el eje horizontal representa el tamaño del conjunto de entrenamiento y en el eje vertical el desempeño medido con la métrica  $R^2$ . La línea azul corresponde al rendimiento en entrenamiento, que rápidamente alcanza valores cercanos a 1, indicando un muy buen ajuste a los datos de entrenamiento. La línea naranja refleja la validación cruzada, que inicia con valores bajos cuando hay pocos datos, pero mejora de manera progresiva conforme aumenta la cantidad de muestras utilizadas, hasta estabilizarse en un rendimiento cercano al del entrenamiento.

Esto evidencia que el modelo necesita un número suficiente de ejemplos para generalizar adecuadamente, y que al disponer de más datos la brecha entre entrenamiento y validación se reduce, mostrando un buen equilibrio entre ajuste y capacidad predictiva.

### 3.2.6 **Modelo LSTM con Random Forest**

Finalmente se utiliza la combinación de dos modelos de aprendizaje para la estimación como primer punto la distancia, para posteriormente predecir posición a través de las coordenadas en grados decimales. Este modelo híbrido utiliza 2 capas, la primera con 40

neuronas y la segunda con 20 neuronas de memoria y entrenado con 12 mil datos recopilados de los 10 nodos estáticos.



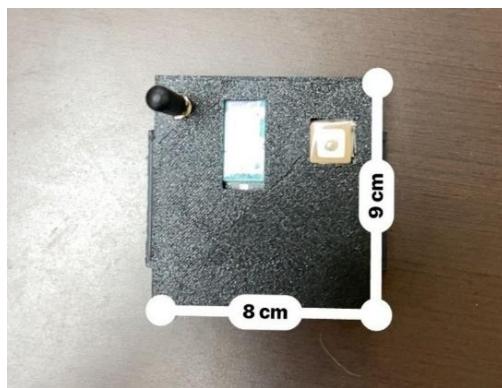
**Figura 3.20** Entrenamiento y evaluación del modelo LSTM

La **Figura 3.20** presenta los resultados del entrenamiento y evaluación del modelo LSTM con Random Forest en dos gráficos. En el panel izquierdo se observa la comparación entre las coordenadas reales y las predichas por el modelo: los puntos azules representan los valores reales de la ruta de un chivo con el nodo móvil y los naranjas las predicciones. Se aprecia que las predicciones siguen en general la trayectoria de los datos reales con un error promedio de 8.72 *metros*, lo que indica que el modelo capta la tendencia de forma favorable considerando que el error promedio del GPS tradicional es de 5 a 10 metros. En el panel derecho se muestra la evolución de la función de pérdida (loss) durante 100 épocas; la curva azul corresponde al conjunto de entrenamiento y la naranja al de validación. La pérdida en entrenamiento disminuye de manera constante hasta estabilizarse en valores de entre 0.2 a 0.3 de perdida.

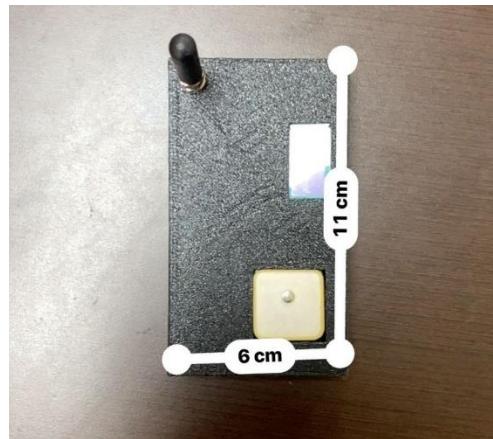
### 3.3 Resultados Obtenidos

Luego de realizar todas las pruebas necesarias, modificaciones pertinentes y prototipos de baja para ubicarlos en los chivos se tiene los resultados alcanzados durante la implementación del sistema de cercado virtual. Estos resultados abarcan tanto la parte de visualización de las alertas a través de Home Assistant como el desarrollo físico del prototipo y su aplicación en el entorno real (la granja de ESPOL) al colocarlo en el chivo.

#### 3.3.1 Prototipo Final



*Figura 3.21 Prototipo 1 con GPS como validador*

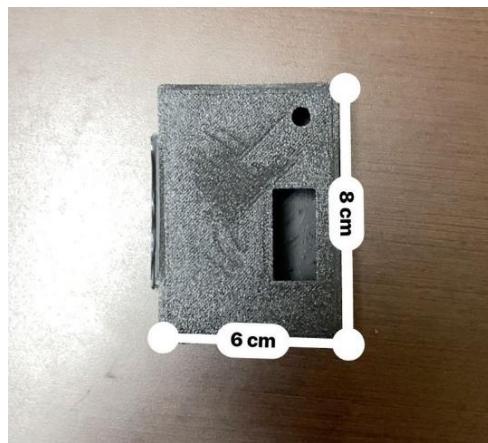


*Figura 3.22 Prototipo 2 con GPS como validador*

Se puede apreciar en la **Figura 3.21** y **Figura 3.22** los prototipos finales diseñados con el módulo GPS, el módulo CubeCell y la batería en su interior; se evidencia variación de tamaño ya que el módulo GPS del prototipo 1 es un poco más grande que el prototipo 1.

También se puede visualizar que las dimensiones de cada prototipo no resultan demasiado grandes lo cual corresponde a la meta de mantener un peso (aproximadamente 120 gramos) y tamaños adecuados para que puedan ser ubicados en el lomo de cada chivo sin generar daños ni incomodidad.

Las medidas y el hecho de que la caja sea una impresión 3D hace que resulte ligero y seguro para que el sistema de monitoreo sea funcional en campo sin afectar el bienestar ni la rutina de los animales garantizando también el monitoreo de su ubicación.



*Figura 3.23 Prototipo final sin GPS*

Por otro lado, en la **Figura 3.23** se observa el prototipo final usado para monitoreo sin modulo GPS. Aquí la caja (impresión 3D) únicamente contiene como sistema el módulo CubeCell con su respectiva antena y la batería lo que permite a diferencia de los prototipos anteriores, reducir notablemente sus dimensiones y obtener un diseño más pequeño, portátil y ligero pensado para ajustarse con mayor facilidad al lomo del chivo y garantizar de igual manera el monitoreo sin afectar su movilidad ni comodidad.

### 3.3.2 Prototipo Posicionado



**Figura 3.24** Prototipo ubicado en el Chivo

En la **Figura 3.24** se puede apreciar el prototipo colocado sobre el lomo del chivo para su respectivo monitoreo. Este modelo mantiene su diseño compacto y posicionado ya en el lomo del chivo se observa que no le causa incomodidad ni afecta su movilidad para seguir con su rutina; de igual manera la pechera no se encuentra demasiado ajustada con el mismo propósito de no afectar sus actividades.

Se eligió que el prototipo se encuentre en el lomo ya que la antena del LoRa debe permanecer orientada hacia arriba lo que garantiza una mejor propagación y recepción de la señal. Esta posición no solo favorece la eficiencia en la transmisión de datos hacia el Gateway, sino que también permite mantener una distancia uniforme respecto al suelo, reduciendo interferencias y perdidas de línea de vista. Además de que también evita que la

antena del CubeCell este en contacto directo con el cuerpo del chivo lo cual podría generar un efecto espejo y afectar la calidad de transmisión de datos.

### 3.3.3 Alertas en Home Assistant

En esta sección se muestra la interfaz de visualización dentro de Home Assistant que tendrá el usuario final para monitorear y tener el control del rebaño.



**Figura 3.25** Mapa de monitoreo en tiempo real en Home Assistant



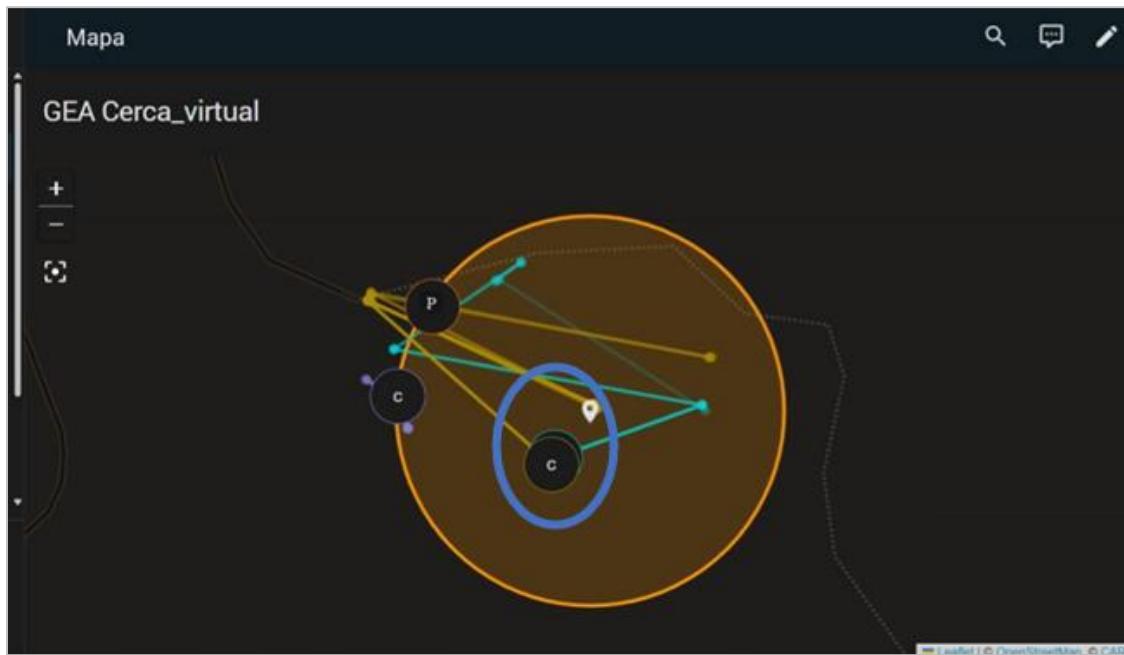
**Figura 3.26** Área definida como Cerca Virtual

En la **Figura 3.25** se puede observar la interfaz de Home Assistant donde se monitorea la ruta y posición de los chivos dentro del mapa donde se delimita la cerca virtual con un círculo naranja tomando como referencia el mapa de Google Earth de la **Figura 3.26**.



**Figura 3.27** Posición y ruta del primer dispositivo móvil.

Para fines prácticos se utilizaron dos dispositivos en chivos adultos para las pruebas, donde se puede visualizar en la **Figura 3.27** dos puntos del primer dispositivo, encerrados con un círculo verde. El punto “C” corresponde a las coordenadas emitidas por el modelo GPS como validación y el punto “P” corresponde a las coordenadas predichas por el modelo LSTM.



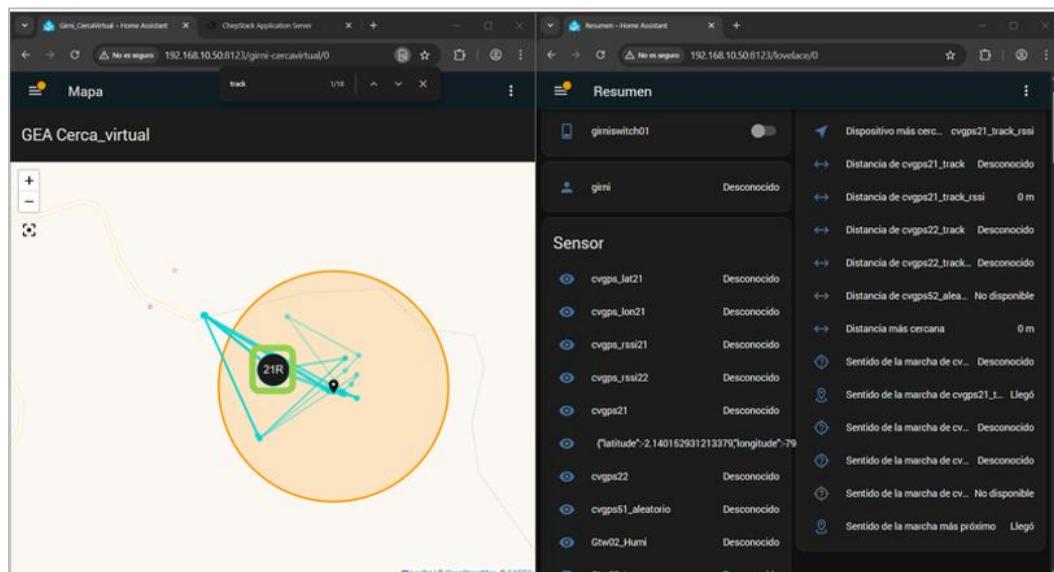
**Figura 3.28** Posición y ruta del segundo dispositivo móvil

De igual forma para el dispositivo 2, como se observa en la **Figura 3.28**, existen dos puntos encerrados con el círculo azul. El punto “C” que corresponde a las coordenadas del GPS como validador y el punto “P” que corresponde a la coordenada predicha por el modelo LSTM que se encuentra relativamente en el mismo punto del GPS.

### 3.4 Validación de alertas en Home Assistant

Una vez validado el modelo de predicción de posición en un entorno real se optó por cuestiones prácticas simular posiciones aleatorias del chivo dentro de la Granja para validar el funcionamiento de las alertas en Home Assistant.

#### 3.4.1 Posición del chivo dentro de la cerca virtual



**Figura 3.29** Chivo dentro de la cerca virtual, visualización en mapa

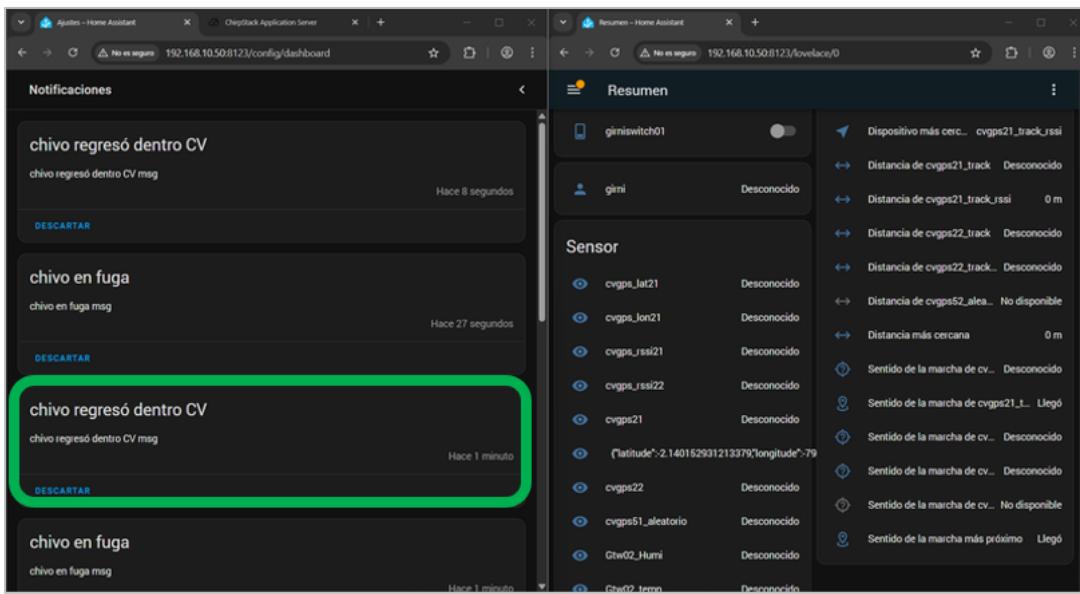


Figura 3.30 Notificación de alerta de la posición del chivo

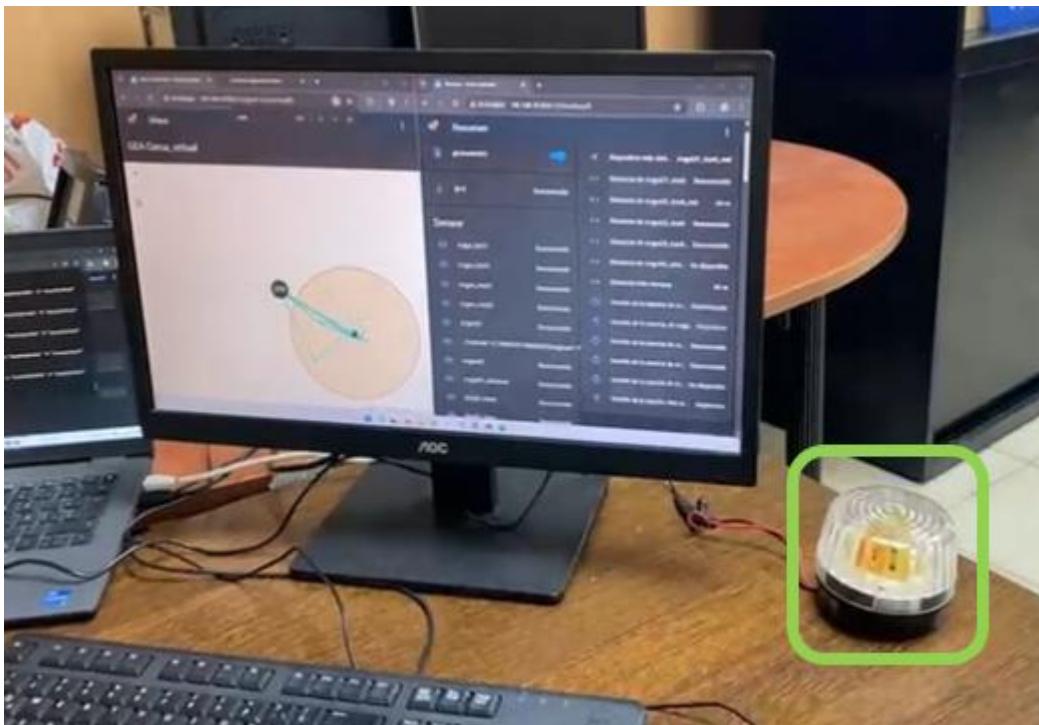


Figura 3.31 Baliza en estado off

### 3.4.2 Posición del chivo fuera de la cerca virtual

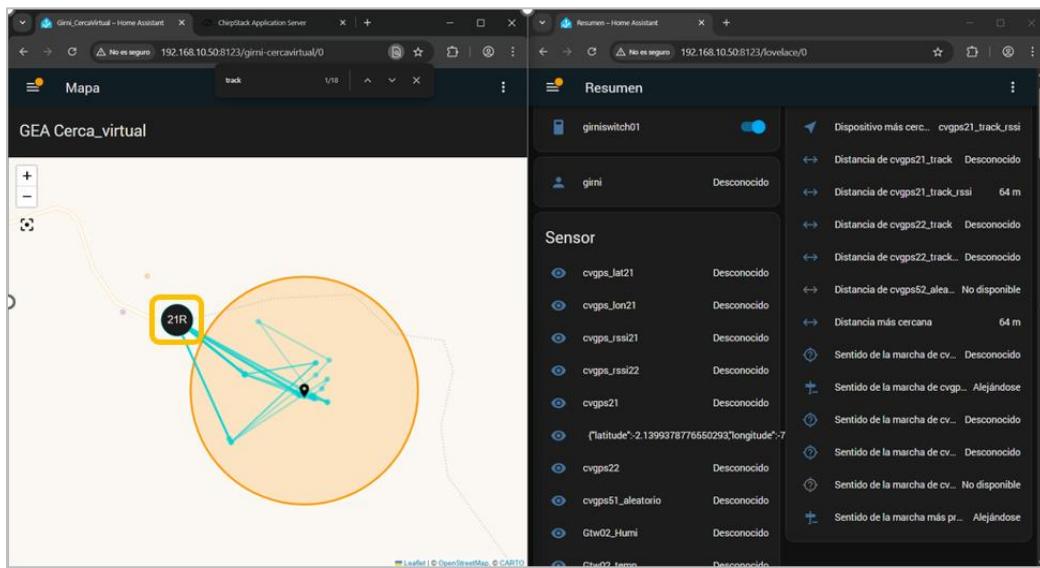


Figura 3.32 Chivo fuera de la cerca virtual, visualización en mapa

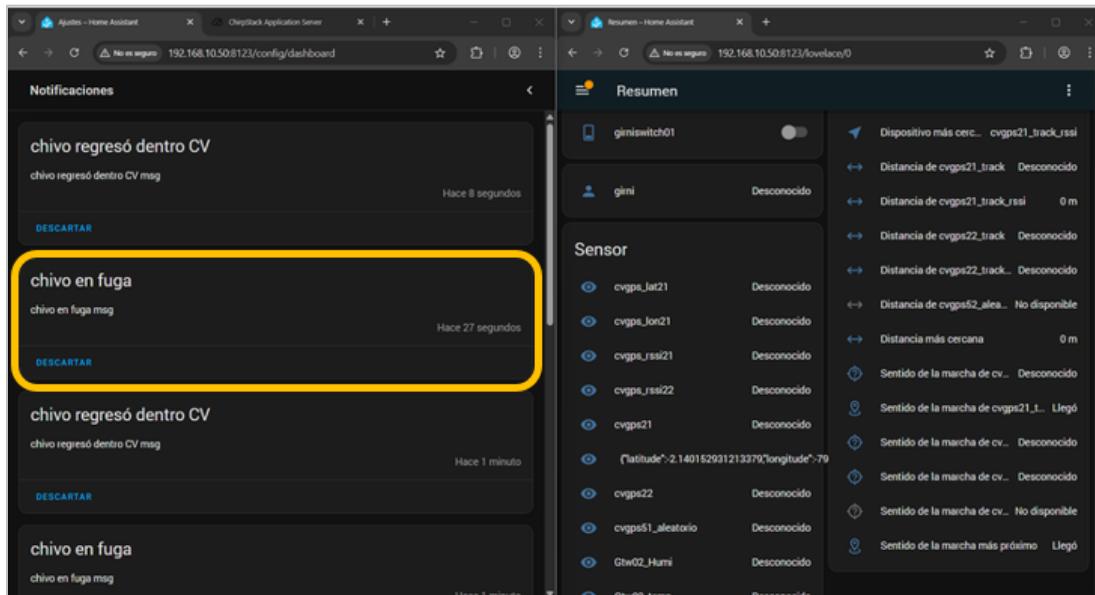
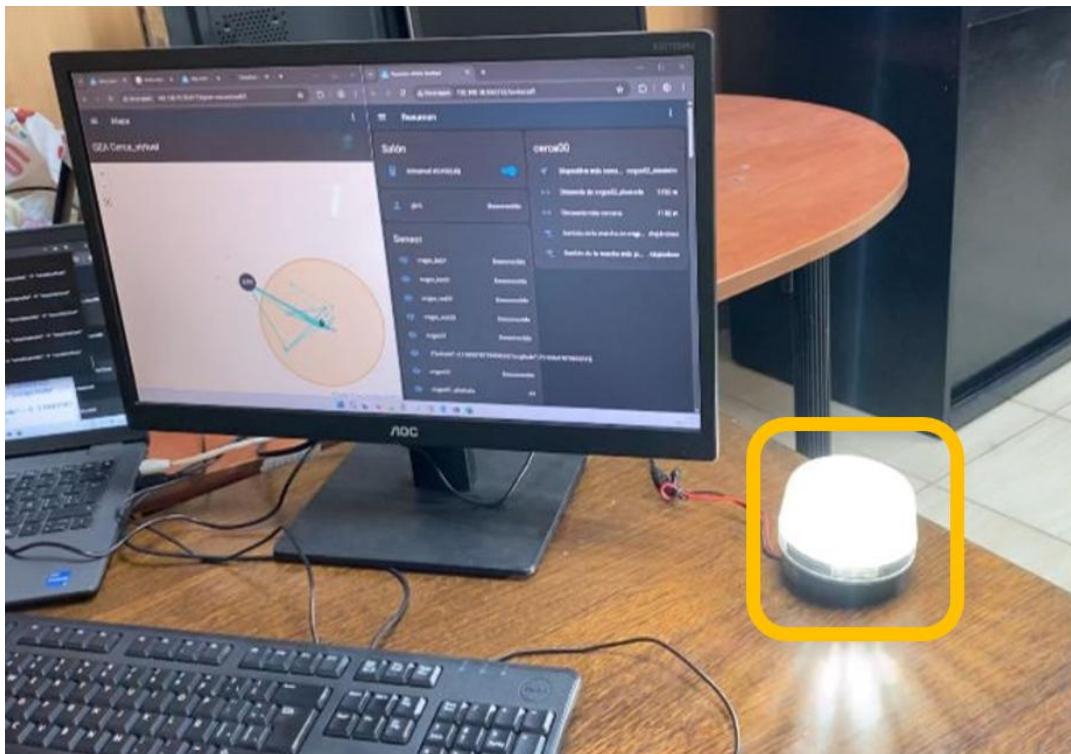


Figura 3.33 Notificación de alerta de la posición del chivo fuera de la cerca virtual

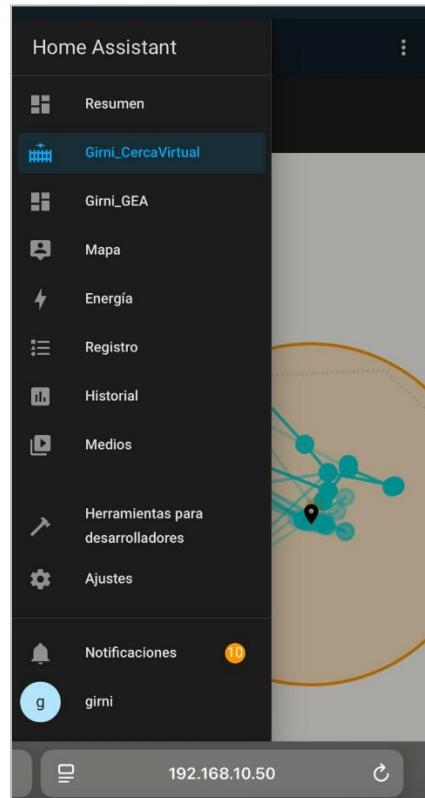


*Figura 3.34 Baliza en estado on*

La interfaz de Home Assistant cuenta con dos sistemas de alertas, el primero basado en el tracking de la posición del chivo como se puede observar en la **Figura 3.29**, donde el círculo etiquetado como 21R indica en tiempo real la ubicación del chivo dentro de la Granja Experimental Espol. El segundo sistema se basa en mensajes de alertas indicando si el chivo se encuentra dentro o fuera de la cerca virtual. Finalmente se agregó una alerta física visual como se observa en la **Figura 3.34**, usando una baliza que se enciende si el chivo se encuentra fuera de la cerca y si apaga cuando este regresa o se mantiene dentro de la cerca. Estos sistemas de alertas se activan o desactivan cada vez que el dispositivo lora ubicado en el chivo envía el RSSI y el modelo de LSTM predice la posición basada en esa lectura de RSSI.

### 3.4.3 Visualización de la interfaz de Home Assistant desde dispositivo móvil

En esta sección se presenta la visualización de la interfaz de Home Assistant desde un dispositivo móvil. Además, se incluyen capturas de pantalla que muestran cómo se interactúa con la interfaz en tiempo real.



*Figura 3.35 Interfaz principal de Cerca Virtual*

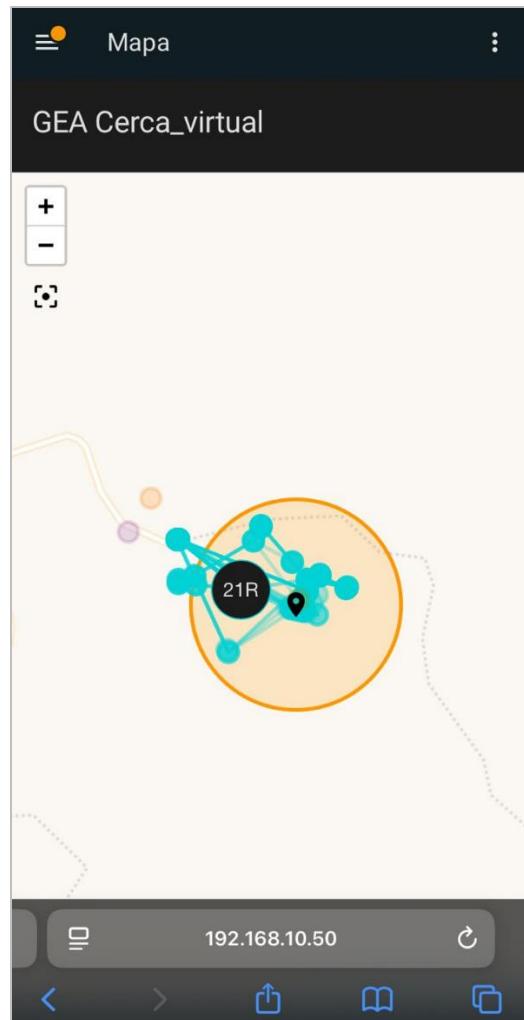
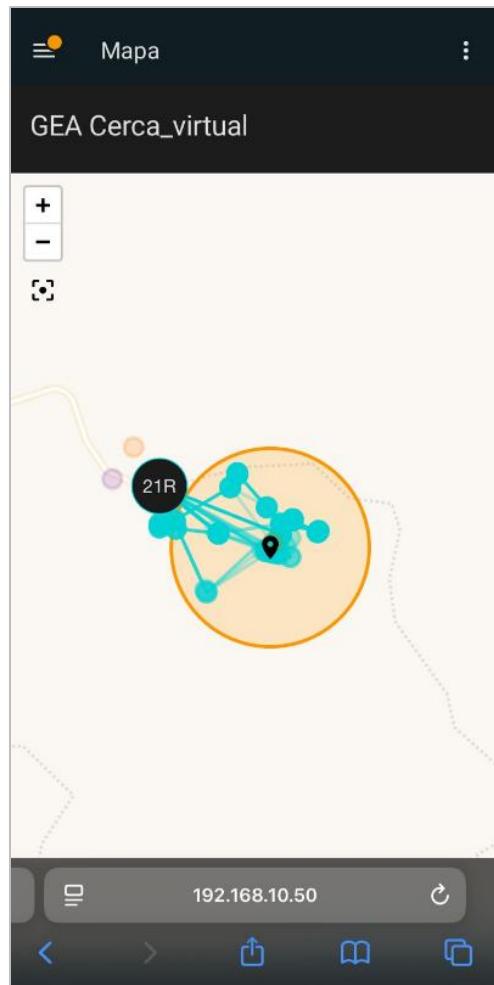


Figura 3.36 Visualización de la posición del chivo dentro de la cerca virtual



**Figura 3.37** Posición del chivo fuera de la Cerca Virtual



*Figura 3.38 Mensajes de alerta*

Para constancia del funcionamiento del servidor de Home Assistant, se probaron los accesos y visualización de la interfaz con un dispositivo móvil como se puede observar en las **Figura 3.35, Figura 3.36, Figura 3.37 y Figura 3.38**, en las cuales se muestra la interfaz inicial y la visualización del sistemas de alertas explicadas previamente.

Para concluir, la integración de alertas en Home Assistant permite una experiencia de monitoreo eficiente tanto desde la interfaz web como desde dispositivos móviles, lo que facilita la gestión y supervisión en tiempo real. Al configurarse las alertas, los usuarios reciben notificaciones visuales directamente en su pantalla, ya sea en el navegador web o en

sus teléfonos inteligentes, asegurando que las acciones necesarias puedan tomarse de inmediato. Además, cuando se activa una alerta, una baliza se enciende automáticamente, proporcionando una señal visual clara de que el chivo salió de la zona segura, lo que refuerza la respuesta rápida y mejora la seguridad del entorno monitoreado

# Capítulo 4

## 4 Conclusiones y Recomendaciones

Una vez finalizadas todas las pruebas, así como los ajustes y modificaciones del presente proyecto podemos establecer las siguientes conclusiones sobre la implementación y el uso de las tecnologías aplicadas en ambientes agrícolas. Así mismo, los resultados del producto final permiten evaluar su desempeño en condiciones reales, identificando mejoras potenciales que podrán aplicarse en futuras implementaciones y proyectos afines.

### 4.1 Conclusiones

- El proyecto culminó con el diseño e implementación de un sistema de cerca virtual para la granja de ESPOL, capaz de monitorear en tiempo real el desplazamiento de los chivos mediante tecnología LoRa. El sistema también genera alertas inmediatas a través de la interfaz web de Home Assistant al detectar la salida de un animal fuera del área segura, cumpliendo con el propósito planteado y aportando una herramienta práctica para la gestión y seguridad del ganado.
- Se realizaron modificaciones en el algoritmo de localización basado en aprendizaje automático, empleando una base de datos con una mayor cantidad de registros para fortalecer el entrenamiento del modelo. También al entrenar el modelo de Machine Learning con más datos permite disminuir el error en las predicciones.
- El sistema de cerca virtual desarrollado para la granja de ESPOL ha demostrado ser una solución efectiva para el monitoreo y la seguridad del ganado. Al integrar tecnología LoRa y sistema de alertas automatizadas el cual consta de 3 fases capaces de proporcionar un control eficiente y garantizar una respuesta inmediata. Esta innovación no solo mejora la gestión del ganado, sino que también proporciona una

herramienta práctica y accesible para los encargados de la granja, facilitando el control y cuidado del animal en todo momento.

- La implementación del sistema de cercado virtual basado en tecnología LoRa demostró ser una solución eficaz para la localización y monitoreo del ganado, logrando una comunicación estable y de largo alcance sin depender de redes celulares convencionales.
- La incorporación del modelo LSTM para la predicción de posiciones, junto con una arquitectura de bajo consumo y basada en software de código abierto, aportó un valor significativo al sistema, permitiendo anticipar desplazamientos y facilitar acciones preventivas. Esta combinación demostró ser escalable, adaptable a distintos entornos y con potencial para futuras implementaciones en otros escenarios productivos.

## 4.2 Recomendaciones

- Para desarrollar y entrenar algoritmos de aprendizaje automático es importante contar con un equipo de cómputo con suficientes recursos, se propone disponer de mínimo 16GB de RAM, 960 GB de disco y 3.00 GHZ de velocidad para procesamiento de información. Estos procesos implican la ejecución de códigos complejos durante largos periodos, cuyo tiempo puede variar según la cantidad de datos empleados y el número de iteraciones necesarias para afinar los resultados. Un hardware adecuado no solo facilita la ejecución, sino que también aporta estabilidad y eficiencia durante el entrenamiento del modelo.
- Se recomienda incrementar la distancia de separación entre los nodos respecto a la configuración actual, de manera que la variación del RSSI registrada por cada uno

sea más clara y distinguible. Un patrón de señal más definido facilita que el modelo de predicción identifique mejor las lecturas, reduciendo el solapamiento entre nodos y aumentando la precisión en la estimación de posiciones. Este ajuste ayudaría a minimizar errores derivados de la proximidad excesiva y optimizaría el rendimiento general del sistema.

## **5 Bibliografía**

- Arduino. (2024). *Arduino IDE v1 Basics*. Obtenido de <https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics/>
- Assistant, H. (2024). *Home Assistant - Device Tracker*. Obtenido de <https://www.home-assistant.io/integrations/sensor.mqtt/>?
- Assistant, H. (2024). *Home Assistant - MQTT Sensor*. Obtenido de [https://www.home-assistant.io/integrations/device\\_tracker.mqtt/](https://www.home-assistant.io/integrations/device_tracker.mqtt/)?
- Assistant, H. (2025). *Home Assistant: La plataforma de automatización del hogar de código abierto*. Obtenido de <https://www.home-assistant.io/>
- Chirpstack. (2023). *Chirpstack*. Obtenido de <https://www.chirpstack.io/>
- CSIRO. (Abril de 2024). Obtenido de <https://www.csiro.au/en/research/technology-space/it/virtual-fencing>
- Duval, D., Audoin, F., Dalke, A., Mayer, B., Antaya, A., Quintero, J., . . . Lien, A. (2025). *Foundations of Virtual Fencing: Economics of Virtual Fence (VF) Systems*. Arizona.
- FAO. (13 de Octubre de 2022). *FAO (Food and Agriculture Organization)*. Obtenido de [https://www.fao.org/cfs/cfs-hlpe/insights/news-insights/news-detail/livestock-engine-for-economic-growth-and-sustainability/en?utm\\_source](https://www.fao.org/cfs/cfs-hlpe/insights/news-insights/news-detail/livestock-engine-for-economic-growth-and-sustainability/en?utm_source)
- Gallagher. (14 de Junio de 2021). *Gallagher*. Obtenido de <https://am.gallagher.com/knowledge-hub/articles/news/virtual-fencing-leader-agersens-acquired-by-gallagher>
- Hardesty, G. (2 de Abril de 2025). *Data - Alliance*. Obtenido de <https://www.data-alliance.net/es/rssi-y-rendimiento-de-antenas-y-aplicaciones-inalambricas-de-iot>
- Kosourova, E. (11 de Diciembre de 2024). *DataCamp*. Obtenido de <https://www.datacamp.com/es/blog/what-is-python-used-for>
- Nofence. (2025). *Nofence*. Obtenido de <https://www.nofence.no/en-us/what-is-nofence>
- Pino, M. L., Vinueza, R., Muñoz, I., & Saravia, E. A. (2022). *Plan Estratégico Institucional*. Quito: Ministerio de Agricultura y Ganadería de Ecuador.
- Rodrigo, J. A. (2020). *Ciencia de Datos*. Obtenido de [https://cienciadedatos.net/documentos/py08\\_random\\_forest\\_python.html](https://cienciadedatos.net/documentos/py08_random_forest_python.html)
- Roman, C. (18 de Marzo de 2025). *Monolitic*. Obtenido de <https://www.monolitic.com/lorawan-que-es-como-funciona>
- Ruminants. (2025). MDPI. Obtenido de <https://www.mdpi.com/journal/ruminants/about>

- Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *Physica D: Nonlinear Phenomena*. Obtenido de <https://www.sciencedirect.com/science/article/abs/pii/S0167278919305974?>
- Sotomayor, O., Ramirez, E., & Martinez, H. (2021). *Digitalizacion y Cambio tecnologio en las mipymes agricolas y agroindustriales en America Latina*. Santiago: CEPAL. Obtenido de <https://repositorio.cepal.org/server/api/core/bitstreams/787ce64b-7f95-4a27-aad9-0a3dc9a3bb70/content>
- System, C. (22 de Mayo de 2025). *The Magazine of the Colorado State University System*. Obtenido de [https://magazine.csusystem.edu/2025/05/22/virtual-fencing/?utm\\_source=](https://magazine.csusystem.edu/2025/05/22/virtual-fencing/?utm_source=)
- Team, H. A. (2024). *Home Assistant*. Obtenido de <https://www.home-assistant.io/>
- Technologies, H. (2025). *Model HTM-01 Technical Specifications*". Obtenido de <https://www.habitattechnologies.com/documents/model-htm-01-technical-specifications>
- Vendramini, J. (2024). *Virtual Fence – A potential new tool for grazing lands management in Florida*.
- Vitale, P., Ward, M., Vitale, J., Gifford, C., Utsumi, S., & Spackman, C. (2025). *Virtual fencing using e-collars vs. physical fencing cost comparison on New Mexico cattle ranches*. Las Cruces, Nuevo México: New Mexico State University. Obtenido de [https://pubs.nmsu.edu/\\_b/B133.pdf](https://pubs.nmsu.edu/_b/B133.pdf)
- Yadav, S. S., Anand, S., D, A. M., Nikitha, D. S., & Thakur, C. S. (2023). tinyRadar: LSTM-based Real-time Multi-target Human Activity Recognition for Edge Computing. *IEEE International Symposium on Circuits and Systems (ISCAS) – también fue publicado luego en IEEE Trans. on Biomedical Circuits and Systems*. Obtenido de [https://labs.dese.iisc.ac.in/neuronics/wp-content/uploads/sites/16/2024/05/Human\\_activity\\_Recognition\\_LSTM\\_ISCAS\\_2024-1.pdf?](https://labs.dese.iisc.ac.in/neuronics/wp-content/uploads/sites/16/2024/05/Human_activity_Recognition_LSTM_ISCAS_2024-1.pdf?)