

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE
MONITOREO DE GEOPOSICIÓN Y TEMPERATURA CORPORAL
PARA GANADO VACUNO MEDIANTE DISPOSITIVOS DE
COMUNICACIÓN LORA.**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA Y CONTROL**

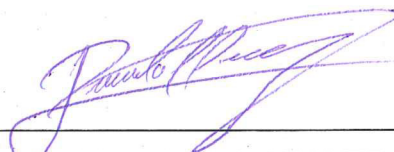
EDUARDO ISMAEL CHANGO ANALUISA

DIRECTOR: Dr. GEOVANNY DANILO CHAVEZ GARCIA

Quito, junio 2022

AVAL

Certifico que el presente trabajo fue desarrollado por Eduardo Ismael Chango Analuisa, bajo mi supervisión.



GEOVANNY DANILO CHAVEZ GARCIA
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Eduardo Ismael Chango Analuisa, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.



EDUARDO ISMAEL CHANGO ANALUISA

DEDICATORIA

Quiero dedicar este trabajo de titulación a mis padres, por su apoyo incondicional a lo largo de mi vida. A mi papá Jorge por su voz de aliento y sus consejos, a mi mamá Margoth, Dios permitió que se quedará en este mundo para brindarme su amor en mis horas mas oscuras. A mi hermanita, sé que estarías orgullosa de mí.

Agradezco a mis mejores amigos, Juan Carlos, Edgar, Salito, Lesly, Henry, Vivi, Edi personas únicas con quienes pude compartir mis logros y fracasos, no me alcanzaría la vida para agradecerles, este logro también es de ustedes. A los chachos, sin lugar a duda el mejor grupo de amigos y profesionales.

Finalmente, dedico este trabajo de titulación a quien fue mi más grande apoyo en gran parte del camino, Fátima Cortez (la osa), por creer en mí, por haber sido mi amiga, mi novia y por ser mi apoyo cuando quise rendirme incluso sin saberlo.

Eduardo Chango

(14 de marzo 2022)

AGRADECIMIENTO

Agradezco por este logro obtenido en primer lugar a Dios, pese a no ser muy cercanos estoy seguro de que ha sido fundamental en este proceso. A mi tutor el Dr. Danilo Chávez García quien ha sido mi guía para la culminación de este trabajo de titulación. A la Escuela Politécnica Nacional y a todos mis profesores por sus enseñanzas a lo largo de mi carrera. A mis amigos sin ustedes esto no sería posible.

Eduardo Chango

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	IX
ABSTRACT	X
1. INTRODUCCIÓN	1
1.1 OBJETIVOS	2
1.2 ALCANCE	2
1.3 MARCO TEÓRICO.....	5
1.3.1 MONITOREO DE GEOPOSICIÓN EN GANADERIA.....	5
1.3.2 MONITOREO DE TEMPERATURA EN GANADERIA	7
1.3.2.1 Temperatura corporal en el ganado.....	8
1.3.2.2 Temperatura y Humedad Relativa ambiente y Temperatura corporal. ..	9
1.3.2.3 Temperatura corporal y su relación con el estado de salud	11
1.3.2.4 Métodos de medición de temperatura corporal.....	12
1.3.3 LORA	13
1.3.3.1 Lora y otras tecnologías	13
1.3.3.2 Lora y LoRaWan.....	14
1.3.3.3 Mensaje LoRa	16
1.3.3.4 Parámetros LoRa	16

1.3.3.4.1	Spreading factor	17
1.3.3.4.2	Duty cycle.....	18
1.3.3.4.3	Received Strength Signal Indicator (RSSI)	19
1.3.3.4.4	Signal-to-noise ratio(snr)	20
1.3.3.4.5	Frecuencia de operación	20
1.3.4	TECNOLOGÍAS DISPONIBLES	21
1.3.4.1	Moovement.....	21
1.3.4.2	Sodaq.....	22
1.3.4.3	LightBug	23
1.3.5	COMPONENTES ELECTRÓNICOS A UTILIZARSE EN EL PROTOTIPO .	23
1.3.5.1	GPS.....	24
1.3.5.2	Sensor de temperatura corporal	25
1.3.5.3	Sensor de Temperatura Ambiente y Humedad Relativa	25
1.3.5.4	Chip LoRa	26
1.3.5.5	Raspberry Pi.....	27
1.3.5.6	ESP32	29
1.3.5.7	Batería.....	31
1.3.6	SOFTWARE A UTILIZARSE EN EL PROTOTIPO.....	32
1.3.6.1	Arduino IDE	33
1.3.6.2	Python.....	33
1.3.6.3	Flask.....	33
1.3.6.4	Open Layers.....	34
1.3.6.5	Herramientas de Desarrollo Web.....	34
1.3.6.5.1	JavaScript	34
1.3.6.5.2	HTML	34
1.3.6.5.3	CSS.....	35
1.3.6.6	SQLITE 3.....	35

2. METODOLOGÍA	36
2.1. DESARROLLO DE HARDWARE	37
2.1.1 Dispositivo Remoto.....	38
2.1.2 Implementación Nodo Final	40
2.1.3 Dispositivo Receptor.....	42
2.1.4 Implementación del Receptor	42
2.2. DESARROLLO DE SOFTWARE	46
2.2.1 Nodo final	46
2.2.2 Dispositivo Receptor.....	50
2.2.2.2 Programa principal	53
2.2.2.3 Programa interfaz	55
2.2.2.3.1 Despliegue del mapa y de geo posición	57
2.2.2.3.2 Cerca virtual	59
2.2.2.3.3 Configuración de parámetros de la cerca	60
2.2.2.3.4 Apagado y reinicio del sistema	61
2.2.2.3.5 Gráficas de temperatura y humedad relativa	61
2.2.2.3.6 Gráficas de temperatura corporal	63
2.2.2.4 Implementación de la interfaz	66
3. RESULTADOS Y DISCUSIÓN	69
3.1 PRUEBAS DE GEO-POSICIÓN	69
3.2 PRUEBAS DE COBERTURA.....	70
3.3 PRUEBAS DE TEMPERATURA CORPORAL	72
3.4 PRUEBAS DE TEMPERATURA Y HUMEDAD RELATIVA AMBIENTE .	75
3.5 INCONVENIENTES	77
3.6 COSTO DE IMPLEMENTACIÓN	78
4. CONCLUSIONES Y RECOMENDACIONES.....	80

4.1. CONCLUSIONES.....	80
4.2. RECOMENDACIONES	81
5. REFERENCIAS BIBLIOGRÁFICAS	83
ANEXOS:	89
ANEXO A	90
A.1. INTRODUCCIÓN	90
A.1.1. Objetivo	90
A.2.1. Hardware	90
A.2.1.1 Dispositivo remoto	91
A.2.1.2 Dispositivo receptor	92
A.2.2. Software	93
A.3.1. Dispositivo Remoto.....	94
A.3.3. Interfaz de Usuario	95

RESUMEN

En el presente trabajo de titulación se desarrolla un prototipo de monitoreo de ganado vacuno, dentro de un área de la zona de crianza. En primer lugar, está un dispositivo remoto que va colocado en el cuello del animal mediante un collar, este se encarga de adquirir datos de geo-posición y temperatura, mediante un chip GPS y un sensor infrarrojo de temperatura, los datos son acondicionados en un microcontrolador y enviados utilizando LoRa como capa física de datos. En segundo lugar, se encuentra el receptor colocado en un punto fijo dentro de la zona de crianza, este dispositivo recibe la información desde el nodo final, y mediante protocolos TCP/IP permite mostrarla en una interfaz de usuario en cualquier dispositivo con un navegador web que este dentro de la misma red de área local.

PALABRAS CLAVE: LoRa, GPS, Temperatura, Monitoreo, Rastreo

ABSTRACT

In the following degree project, a prototype for monitoring cattle in an area of the breeding is developed. First, there is a remote device that is placed on the animal's neck through a collar, this is responsible for acquiring geo-position and temperature data through a GPS chip and an infrared temperature sensor, the data are conditioned in a microcontroller and sent using LoRa as the physical data layer. Secondly, the receiver is placed at a fixed point within the breeding area, this device receives the information from the end node, and through TCP/IP protocols allows displaying it in a user interface on any device with a web browser that is within the same local area network.

KEYWORDS: LoRa, GPS, Temperature, Monitoring, Tracking

1. INTRODUCCIÓN

Durante los últimos seis años en el Ecuador el sector ganadero libra una batalla contra el hurto de ganado, el cual pone a productores grandes y pequeños contra las cuerdas al no contar con una alternativa de protección para cuidar su inversión. Las pérdidas solo por robo de ganado en este periodo ascienden a cerca de 13 millones de dólares. Las denuncias por esta actividad ilegal son de 11.167 casos de acuerdo con registros de la Policía Nacional, con una mayor tendencia en la costa en especial en Manabí, Esmeraldas y Guayas [1]. De acuerdo con investigaciones realizadas por la Policía Nacional, los robos ocurren durante la madrugada y debido a la extensión de terreno en fincas ganaderas, resulta casi imposible la intervención de testigos que permitan el reconocimiento de los cuatrerros. Al momento este delito representa una pena privativa de libertad que en el menor de los casos puede ser de 15 a 30 días de acuerdo con el artículo 211 del COIP [2].

Ahora bien, este fenómeno no ocurre solamente en Ecuador, en países de África el robo de ganado en zonas rurales es común y en Senegal por ejemplo representa una pérdida de 3,6 millones de dólares anuales [3]. El robo de ganado, por supuesto, no envuelve solamente a países en vías de desarrollo, en Reino Unido según el Rural Crime Report 2020, el robo de ganado representó una pérdida de 3 millones de euros en 2019 [4]. Todos estos reportes tanto en Ecuador como en el resto de los países concuerdan que el abigeato es llevado a cabo por bandas de crimen organizado.

Muchos países desarrollados empiezan a optar por soluciones tecnológicas que les permitan prevenir el robo de ganado, una de estas soluciones es emplear sistemas de rastreo GPS que permitan conocer la ubicación del ganado dentro del perímetro de sus granjas, establecer cercas virtuales y de esta manera alertar si los animales se encuentran en zonas en las cuales podrían ser fácilmente sustraídos. [5]

Paralelamente desde hace varios años se llevan a cabo investigaciones acerca de las consecuencias que tienen los factores internos y externos de crianza de estos animales respecto a su bienestar y la producción de leche. Así como el empleo de tecnologías que permitan un mejor manejo de las zonas de crianza, faciliten la toma de decisiones y aumenten la productividad. Un caso de estudio realizado por Libelium, una empresa dedicada a IoT, arroja que, tras el monitoreo de parámetros como temperatura corporal del ganado y temperatura ambiente, permitió gestionar de manera eficiente la alimentación del ganado ahorrando 380 mil euros en seis meses e incrementando la

producción de leche de 25 a 33 litros en una granja productora en Voshazhnikovo-Rusia.
[6]

En base a estos precedentes el proyecto actual pretende construir un prototipo que se convierta en una herramienta para el granjero. Por medio de un dispositivo compuesto por un GPS, sensor de temperatura y a través de tecnología de comunicación LoRa, le permita conocer la ubicación geográfica dentro del área de crianza, establecer un cerco virtual que despliegue una alarma en caso de que el animal abandone el perímetro y monitorear la temperatura superficial para que el granjero pueda tomar decisiones bajo su criterio y experiencia.

1.1 OBJETIVOS

El objetivo general de este Proyecto Técnico es:

Diseñar e implementar un prototipo de monitoreo de geo posición y temperatura corporal para ganado vacuno mediante dispositivos de comunicación LoRa.

Los objetivos específicos del Proyecto Técnico son:

- Recopilación de información relevante respecto a investigaciones relacionadas con la implementación de dispositivos de geolocalización y monitoreo de temperatura de ganado vacuno que emplean tecnología de comunicación LoRa para el envío de datos.
- Diseñar un dispositivo remoto que permita la obtención de datos desde un módulo GPS, y de un sensor de temperatura y los envíe a través de LoRa como capa física.
- Implementar una interfaz que permita visualizar la geo posición del dispositivo remoto, así como los cambios de temperatura de manera periódica cada 10 minutos.
- Realizar pruebas para validar el equipo

1.2 ALCANCE

Los alcances del presente proyecto de titulación son:

- Revisión bibliográfica acerca del monitoreo de geo posición y temperatura en aplicaciones ganaderas, para establecer la importancia del monitoreo de geo posición como herramienta de gestión del ganado y determinar el tipo de sensor de temperatura que se utilizará en el prototipo.
- Revisión bibliográfica acerca de tecnología LoRa, sus características y condiciones de uso para el envío y manejo de datos. De esta manera se establecerá las características del chip LoRa que se usará en el prototipo como niveles de voltaje, consumo de energía, frecuencia de operación e instrucciones de programación para establecer los tiempos de espera entre mensajes.
- Revisión bibliográfica para obtener características de elementos electrónicos disponibles en el mercado, tarjetas microcontroladoras basadas en ESP32 y transceptores de señal LoRa, sensores que permitan la adquisición de datos GPS y temperatura, Raspberry Pi para aplicaciones web, herramientas de programación para la adquisición, envío y recepción de datos mediante LoRa y el despliegue de estos dentro de una interfaz gráfica.
- Se diseñará el dispositivo remoto que contará con un microcontrolador ESP 32 para acondicionar los datos del sensor de temperatura, discriminar los datos del GPS y enviarlos a través de LoRa.
- Se considerarán para el diseño las características del GPS, sensor de temperatura, batería, chip transceptor LoRa para distribuir las entradas y salidas del microcontrolador ESP32.
- Se diseñará una carcasa de plástico para alojar los elementos electrónicos del dispositivo remoto.
- Se diseñará el hardware del dispositivo receptor compuesto por la tarjeta de desarrollo LoRa y una Raspberry Pi que será el servidor de datos y contendrá la interfaz de usuario, fuente de alimentación y transmisión de datos mediante ethernet dentro de la red de área local.
- Se diseñará la interfaz como una página WEB, que se encuentra alojada en la Raspberry y a la cual se puede acceder desde dispositivos que tengan instalados un navegador web, como una PC, siempre y cuando estén conectados a la misma red de área local que el dispositivo receptor. Los datos se almacenan en la

memoria de la Raspberry siendo esta tarjeta el servidor de datos y la PC un cliente.

- Se construirán y programarán los dispositivos remoto y receptor con los elementos de software y hardware considerados en el diseño, de manera que envíen datos de forma periódica, el dispositivo se activa cada cierto tiempo para obtener los datos de los sensores, se envían al receptor y se desactiva para ahorrar energía. Este proceso se realiza cada 10 minutos, con el fin de reducir el consumo de batería durante la desactivación del dispositivo y para cumplir con las recomendaciones de uso de tecnología LoRa, la cual especifica un tiempo de espera entre mensajes.
- Se implementarán los programas de la interfaz de usuario, almacenamiento de datos en la Raspberry y en la tarjeta de desarrollo LoRa que recibe los datos desde el dispositivo remoto.
- Se obtendrán los datos desde el transceptor LoRa conectado al puerto USB de la Raspberry y esta se encargará de ingresar la información a una base de datos, gestionar el intercambio de información desde y hacia la interfaz.
- En la interfaz, la geo posición será visible en un mapa de la zona, y permitirá dibujar un cerco virtual en el mapa delimitando de esta manera el área en la que debería estar la res, en caso de que la res junto con el dispositivo remoto abandone esta área delimitada se mostrará una alerta visual.
- Los datos de temperatura serán almacenados en una base de datos y estarán disponibles de acuerdo con la necesidad del cuidador. La interfaz le permitirá al cuidador conocer las variaciones de temperatura corporal de la res.
- El dispositivo remoto se colocará en el cuello de una res de ganado vacuno, el dispositivo receptor se colocará en una posición fija dentro del área de pruebas.
- Se comprobará que los datos del dispositivo remoto se visualicen en la interfaz en el computador conectado a la misma red de área local que el receptor.
- Se registrará el rango de cobertura a través de la geo posición e indicadores de ruido e intensidad de señal disponibles en los chips transceptores, se comprobará que los datos de geo posición correspondan a la ubicación dentro de la zona de crianza y que ante un abandono del perímetro en la interfaz se observe una alerta

visual; el error máximo de geo posición estará determinado por el fabricante del sensor.

- Se validará la relación que existe entre temperatura corporal y rectal mediante muestras obtenidas de forma tradicional mediante termómetro rectal y mediante el dispositivo prototipo, el error máximo en la medición de temperatura será de 0.8 grados.

1.3 MARCO TEÓRICO

En el presente apartado se realiza un estudio bibliográfico acerca del monitoreo de geo posición en ganado vacuno como una herramienta para mantener el control de este dentro de área de crianza y de esta manera reducir la posibilidad que este pueda ser sustraído. Se estudia el monitoreo de temperatura como una herramienta de soporte para el granjero de tal manera que esta ayude a predecir posibles anomalías en la salud del ganado.

Por otra parte, se abarca la revisión bibliográfica de LoRa acerca del envío de datos, y la configuración de todos los parámetros que esta requiere de acuerdo con la región.

Finalmente se estudian los dispositivos comerciales para definir las características que se desea que tenga el prototipo, así como los componentes electrónicos de los que se dispone en el mercado y las herramientas de programación para cumplir con el objetivo principal del presente proyecto.

1.3.1 MONITOREO DE GEOPOSICIÓN EN GANADERIA

Parte del manejo de ganado involucra conocer su ubicación dentro de la zona de crianza, debido a que las extensiones de terreno son de varias hectáreas conocer la geo posición resulta una herramienta útil para el ganadero ya que de esta manera puede conocer patrones de movimiento y mantenerse informado sobre desplazamientos inusuales.

Una de las aplicaciones del monitoreo de geo-posición del ganado, es la reducción del índice de robos, un estudio realizado por [7] establece que mediante el patrón de desplazamiento se puede determinar si el ganado es arreado. En la figura 1.1. se puede

observar dos tipos de desplazamiento; en primer lugar, la línea verde indica el desplazamiento habitual del ganado sin un vaquero que lo guíe, mientras que la línea roja son datos de geo posición tomados durante el arreo del ganado de regreso hacia un pozo de agua, se puede observar que el desplazamiento en este último caso se asemeja más a una línea recta. Se puede entonces determinar si el ganado es arreado al observar su patrón de desplazamiento a lo largo del tiempo y al no existir un arreo programado se puede tratar de un robo en progreso.



Figura 1.1. Patrón de movimiento del ganado en libertad (línea verde), patrón de movimiento del ganado cuando es arreado (línea roja) [7]

Es posible combinar la geo-posición con un algoritmo de cercado virtual, que no es más que una delimitación mediante datos de latitud y longitud del área en la que idealmente debería desplazarse el ganado, el estudio llevado a cabo por [8] empleó este concepto para hacer que la res por sí misma regrese dentro de la cerca usando choques eléctricos, estímulos visuales y auditivos en caso que esta abandone su perímetro permitido, el experimento analizó la viabilidad de reemplazar las cercas de alambre físicas por esta tecnología, en la figura 1.2. se puede observar el dispositivo. Otro estudio realizado por [9], analizó el comportamiento de ocho vacas ante estímulos sonoros provenientes de un dispositivo colocado en el cuello que contenía un GPS y un amplificador de audio, cuando el animal abandonaba la cerca virtual el dispositivo emitía sonidos con el objetivo de disuadir el reingreso del animal.



Figura 1.2. Dispositivo de estímulos eléctrico, auditivo y visual de experimentación de cercado virtual. [8]

Los resultados de las investigaciones anteriores demostraron que existe reacción por parte del ganado ante los estímulos, sin embargo, no fue la esperada y se observó que el mayor grado de contención del ganado se presentó cuando existía una cerca física.

En base a los estudios anteriores se puede establecer la importancia de conocer la ubicación del ganado dentro del terreno y analizar su desplazamiento. En este caso particular, no conviene dotar al dispositivo de mecanismos de disuasión como choques eléctricos, sistema de audio o estímulos visuales ya que los resultados no serían los deseados y reducirían la autonomía del dispositivo remoto. Por lo tanto, es mejor informar al granjero si el animal abandono el área designada y que este tome las acciones pertinentes para reingresar a la res a su lugar designado.

1.3.2 MONITOREO DE TEMPERATURA EN GANADERIA

Dentro del cuidado y crianza de ganado bovino la temperatura juega un papel fundamental en el diagnóstico del estado de salud de estos animales. Mediante la temperatura el ganadero puede determinar si las vacas están en periodo de celo, o decidir si requieren un chequeo médico.

El principal método de medición de temperatura es a través de la temperatura rectal, y bajo este método se determina que la temperatura normal del ganado oscila entre 36.5

°C y 39.5 °C [10]. Temperaturas sobre 44.0 °C y por debajo de 25.0 °C pueden ser fatales y ocasionar daño cerebral [11]. La precisión de la medición de la temperatura rectal depende del operador del termómetro, puesto que este método resulta incómodo para el animal, puede causar estrés y por ende variar la medición [12].

1.3.2.1 Temperatura corporal en el ganado

Un estudio realizado por la Universidad de Suecia de Ciencias Agrícolas [10] realizó pruebas para determinar los cambios de temperatura interna en el ganado durante un año, se tomaron muestras tanto de temperatura rectal como temperatura sobre la piel del animal entre ellas el cuello mediante un sensor infrarrojo y un sensor de contacto.

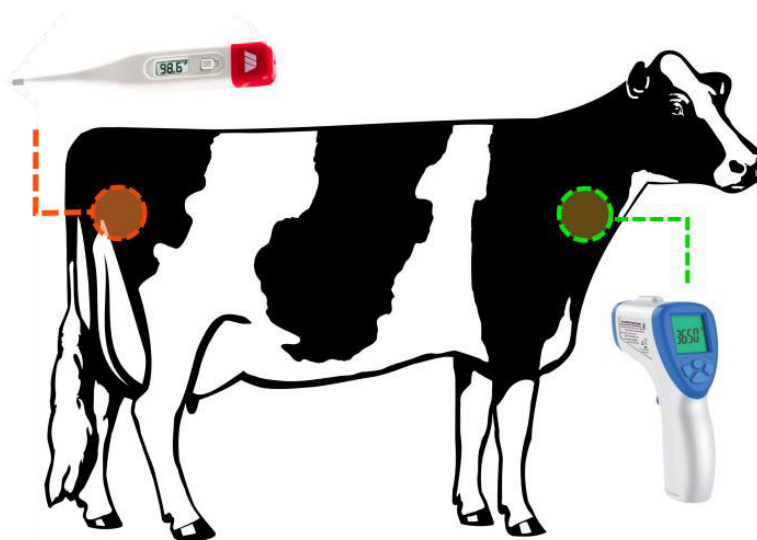


Figura 1.3. Ubicaciones termómetro rectal e infrarrojo [Elaboración Propia].

En la gráfica de barras de la figura 1.4. se muestran los resultados, la temperatura ambiente representada por las barras azules varió entre 12.4 °C y 23.7 °C. Al observar las barras que representan la temperatura ambiente y las barras rojas que representan la temperatura superficial medida con el sensor infrarrojo, se observa que este método registra de mejor manera los cambios debidos a las variaciones de temperatura del entorno de crianza. La temperatura rectal varía de igual manera que la temperatura superficial, pero de una forma muy discreta, con una variación máxima de 0.3 °C en contraste con la variación de 7.7 °C de la temperatura superficial.

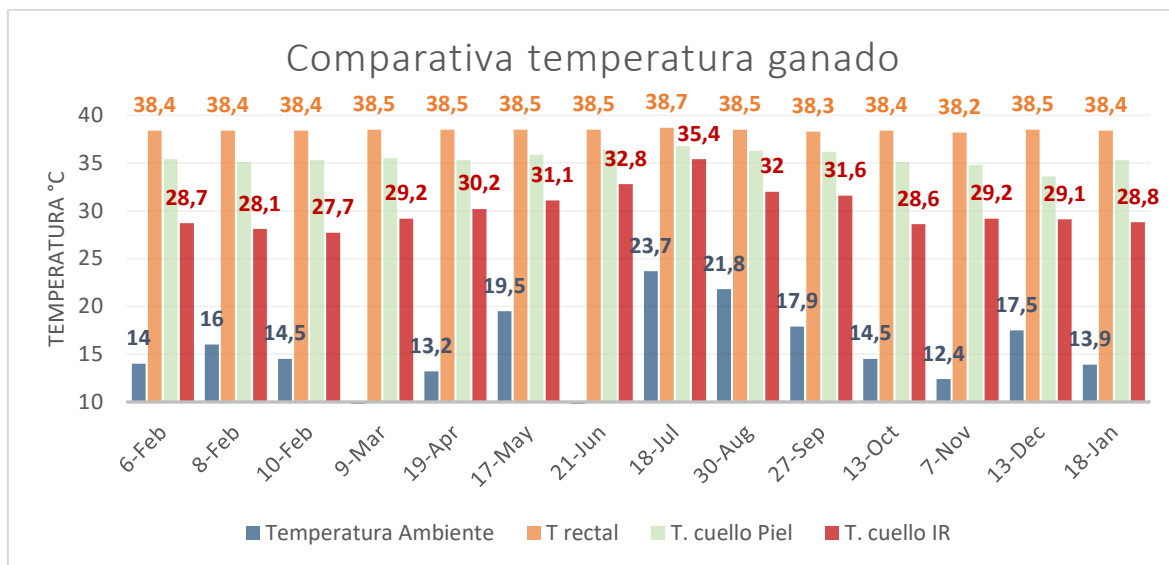


Figura 1.4. Gráfica de resumen del estudio “Changes in Dairy Cows’ temperature”, Swedish University of Agricultural Sciences [10].

Por otro lado, al analizar los datos se observa que la relación que existe entre la temperatura rectal y la temperatura piel e infrarroja en el cuello son 0.78 y 0.92 respectivamente, es decir la temperatura de un sensor de contacto se acerca más a la temperatura rectal. Un estudio realizado en la India [13], introdujo durante diez días a cuatro vacas en una cámara con temperatura ambiente y humedad relativa controlada para analizar la temperatura corporal en el recto y otras partes del cuerpo mediante un sensor infrarrojo antes y después del experimento, la temperatura ambiente varió entre 25 °C y 40 °C con una humedad relativa constante de 50%, el resultado fue que la correlación entre la temperatura rectal y la temperatura medida en el cuello es de 0.7, lo cual se aproxima a la correlación obtenida en el estudio de la Universidad de Suecia.

1.3.2.2 Temperatura y Humedad Relativa ambiente y Temperatura corporal.

En base a lo mencionado anteriormente se puede establecer que la temperatura interna del ganado se ve afectada por la temperatura del ambiente. La Universidad de Nebraska-Lincoln [14] mediante un estudio determinó que la temperatura ambiente ideal para el ganado esta entre -3 °C y 18 °C. De igual manera, a partir de 22 °C de temperatura ambiente la humedad relativa influye en el ganado de acuerdo con [15][16], en la figura 1.5. se observa el Índice de Temperatura y Humedad Relativa THI donde se presenta el grado de afectación en el ganado debido a estas dos variables. A medida que aumenta la

humedad relativa y la temperatura ambiente el ganado eleva su ritmo cardiaco, transpira más para reducir el impacto de la temperatura y entra en un estado conocido como estrés térmico, en un nivel de estrés inicial la producción de leche empieza a disminuir [10], y cuando la temperatura ambiente sobrepasa los 33.5 °C los efectos pueden ser mortales dependiendo de la humedad relativa, si está aumenta las consecuencias serían graves.

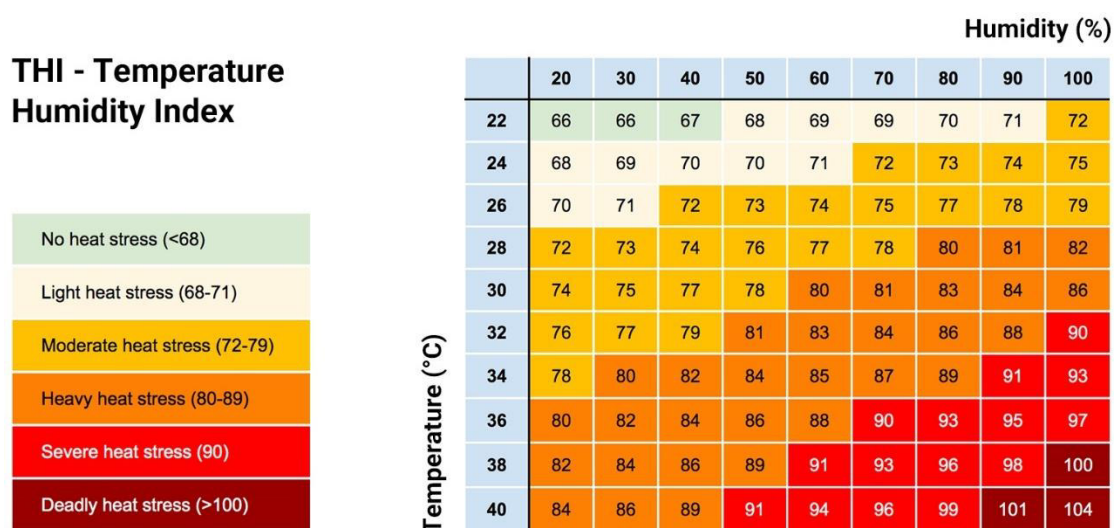


Figura 1.5. Índice de Temperatura y Humedad Relativa THI [16].

Los resultados de esos estudios son importantes considerando la cantidad de ganado que existe en Ecuador y donde se concentra. Según la Encuesta de Superficie y Producción Agropecuaria Continua (ESPAC) y el Instituto Nacional de Estadística y Censos (INEN) correspondiente al 2019 [17], en el país existen 4.31 millones de cabezas de ganado vacuno de estas el 39.71% se ubican en la costa con alrededor de 930 mil cabezas de ganado en la provincia de Manabí, a esta le siguen Azuay, Chimborazo, Cotopaxi y Pichincha. Por otro lado, de acuerdo con el informe climatológico del INAMHI en su última actualización [18] la temperatura media y humedad relativa de estas provincias es la que se muestra en la tabla 1.1., el dato más notable se presentó durante el mes de septiembre en Manabí con una temperatura máxima de 32.8 °C y humedad relativa de 80% la cual según el índice THI corresponde a un alto grado de estrés en el ganado.

Tabla 1.1. Resumen existencias de cabezas de ganado y factores climáticos en las principales provincias ganaderas del Ecuador [18].

Provincia	Cabezas de Ganado [miles]	Temperatura [°C]	Humedad relativa [%]
Manabí	930	19.6 – 32.8	77-87
Azuay	330	11.9-18.5	84-94
Chimborazo	315	7.3-13.8	-
Cotopaxi	309	7.3-13.8	74-97
Pichincha	286	7.3-18.5	85-90

Las condiciones climáticas en el Ecuador pueden ubicar al ganado en situaciones de estrés térmico moderado y alto, si se considera que estos factores influyen en la producción de leche es deseable monitorear la temperatura del ganado para saber cómo reacciona este a las variaciones de su entorno y de esta manera que el granjero decida qué acciones tomar para reducir la carga de estrés.

1.3.2.3 Temperatura corporal y su relación con el estado de salud

Los cambios en el estado de salud en los mamíferos se presentan con variaciones de temperatura. En el caso de los bovinos existen investigaciones que tienen como objetivo estudiar la influencia de la temperatura en diferentes etapas a lo largo de su vida, posibles infecciones, celo y como se observó en el apartado anterior posibles afectaciones debido a la temperatura y humedad del ambiente.

El estudio realizado por [19], estudia la variación de temperatura en 24 vacas durante 54 ciclos reproductivos y en la figura 1.6. se presenta un resumen de los cambios de temperatura que oscila entre 38.3 °C y 38.6 °C, el día que se presenta el celo está determinado como día cero, antes de ese día se presenta una reducción de temperatura para incrementarse rápidamente en el día cero, posterior a ese día se reduce y se incrementa nuevamente pero más lentamente.

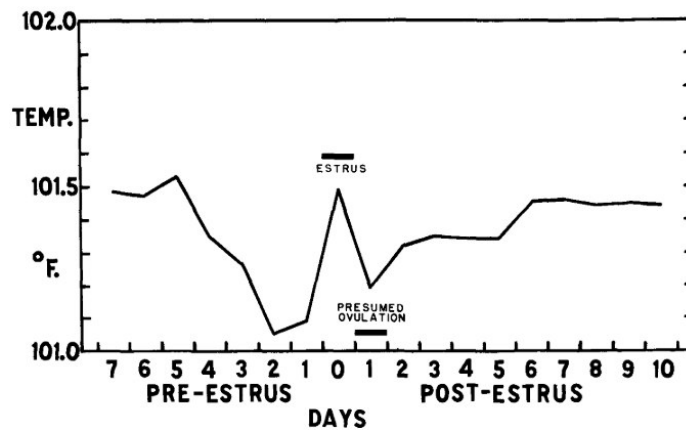


Figura 1.6. Variaciones de temperatura antes, durante y después del celo [19].

Un estudio realizado por [20], analiza el costo y efectividad de varios métodos para detectar el celo en un grupo de 100 vacas, de estos sobresalen la medición de temperatura con una efectividad del 55% y un costo siete veces menor respecto al método visual con una efectividad del 96%. El estudio concluye que la efectividad de la medición de temperatura se puede ver afectada debido a que el aumento de temperatura puede darse por una patología o factores ambientales, por lo tanto, requiere de la inspección visual lo cual concuerda con el propósito del prototipo el cual es ser una herramienta de monitoreo para el cuidador.

1.3.2.4 Métodos de medición de temperatura corporal

Para realizar mediciones de temperatura es posible usar métodos de contacto con la piel y métodos sin contacto. Los métodos de contacto se consiguen a través de termocuplas o termistores, mientras que para el método sin contacto se puede emplear sensores infrarrojos y cámaras térmicas. En la tabla 1.2 se muestran los resultados del estudio comparativo entre diferentes métodos de medición de temperatura corporal en animales en agricultura de precisión realizado por [21].

Tabla 1.2. Comparativa métodos de medición de temperatura [21].

	Termómetro y cámaras térmicas	Termocuplas	Termistores
Problemas ambientales	Influye la radiación ambiente/ polvo/ temperatura local/ u obstáculos.	Baja influencia del ambiente	
Consideraciones para el correcto uso del sensor	Respetar la distancia hacia la superficie, de cada sensor.	Menor precisión, estabilidad y sensibilidad.	Frágiles, pueden sufrir calentamiento y

	Considerar el ángulo y el error asociado a este y la emisividad del cuerpo Hay que considerar que la emisividad del pelaje de los mamíferos esta entre 0.98 – 1.		no son lineales
		Requiere una referencia con la que debe calibrarse constantemente	
Distancia	Desde unos pocos metros hasta 3 Km	Necesitan estar en contacto	
Tiempo de respuesta	De 2 a 3 segundos	Aproximadamente un segundo	De 1 ms a 250 ms

En base a las características de los tipos de sensores, se decide emplear un sensor infrarrojo para medir la temperatura corporal en el cuello de la res, esto debido principalmente a que el dispositivo no estará ajustado completamente al cuello del animal para no producir molestias. Termocuplas y termistores requieren estar en contacto con el objeto a medir, y dado que el dispositivo será colocado en un collar, para el confort del animal este no puede estar demasiado ajustado. Otra de las razones por las cuales no se usará este tipo de sensores es por su precisión y estabilidad en comparación con los termómetros infrarrojos.

1.3.3 LORA

1.3.3.1 Lora y otras tecnologías

Existen diferentes tecnologías que permiten la transmisión de datos de forma inalámbrica cada una de ellas con diferentes características y aplicaciones. En la tabla 1.3. se presenta una comparación entre ellas [22]. Como se puede observar entre las características de cada tecnología, sobresalen principalmente el costo, rango de cobertura y potencia de transmisión como ejes de comparación.

Tabla 1.3. Comparación entre tecnologías inalámbricas de transmisión [22].

Tecnología	Comunicación inalámbrica	Rango	Potencia de transmisión
Bluetooth	Rango corto	10m	2.5mW
Wifi	Rango corto	50m	80mW
3G/4G	Celular	5Km	5000mW

LoRa	LPWAN	2-5Km (Áreas Urbanas)	20mW
------	-------	-----------------------	------

Partiendo del rango de cobertura, se encuentran a la cabeza tanto la tecnología 3G/4G y LoRa, sin embargo, LoRa requiere del 0.4% de energía para el envío de información en contraste con la tecnología 3G/4G. En términos de energía sobresale la tecnología bluetooth que requiere el 12.5% en comparación con LoRa, pero bluetooth cae en la categoría de Red de Área Personal, y cubre distancias de tan solo diez metros.

En cuanto a costo la tecnología celular representa el costo más alto, debido a que las frecuencias para su transmisión están a cargo del Estado, el cual concede permisos a las operadoras de red móvil. Por lo cual de usar esta tecnología es necesario estar suscrito a una operadora.

LoRa, se diferencia de la tecnología celular, en que está diseñado para operar en bandas de frecuencia ISM [22] (Industrial, Científica y Médica), estas bandas son de uso libre, por lo cual no es necesario suscribirse a un operador comercial.

En base a este análisis y al considerar antecedentes como la envergadura del terreno para la crianza de ganado y que el prototipo al ser móvil requiere batería y por tanto es deseable mayor autonomía LoRa figura como la mejor opción debido a su alcance, bajo costo y bajo consumo de energía.

1.3.3.2 Lora y LoRaWan

LoRa (Long Range) es una tecnología de modulación de radio frecuencia desarrollada por Semtech, cae dentro de la categoría de una red de área completa y bajo consumo de energía. En condiciones ideales LoRa permite una autonomía de 10 años para dispositivos que emplean baterías y un rango de cobertura de:

- Hasta 5 Km en áreas urbanas
- Hasta 15 Km o más en áreas rurales.

La topología empleada es estrella donde los nodos finales se comunican con un dispositivo central (receptor). Es importante aclarar que LoRa y LoRaWan son diferentes. LoRa es la tecnología propiamente, es decir la capa física que abarca los dispositivos y sus componentes que permiten transmitir datos dentro de la frecuencia ISM la cual puede

variar en diferentes regiones del mundo [22]. LoRaWan por otro lado es un estándar de comunicación que permite la interoperabilidad de los dispositivos LoRa y en última instancia las aplicaciones desarrolladas para el análisis de datos provenientes de los dispositivos como se observa en la figura 1.7.

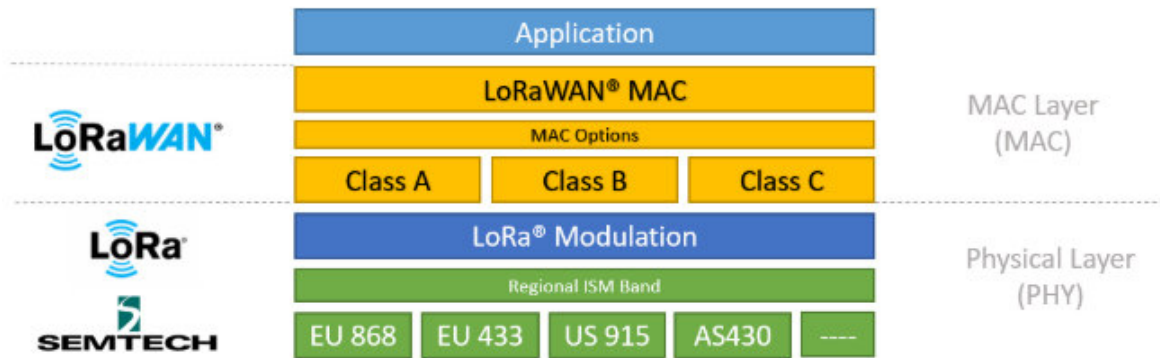


Figura 1.7. Stack LoRa [23].

Como se observa la figura 1.8. de izquierda a derecha, en primer lugar, se encuentran los nodos finales que constan del chip LoRa para la transmisión de información, pero también están conformados por sensores y actuadores para aplicaciones específicas, elementos pasivos, amplificadores operacionales para el acondicionamiento de señales analógicas, circuitos de potencia y microcontroladores para la ejecución de tareas. En segundo lugar, se encuentran los concentradores de señal también compuestos por chips LoRa de uno o más canales, los cuales se encargan de recibir, enviar información desde o hacia los nodos finales y permiten el intercambio de información a través de otra tecnología como ethernet mediante TCP/IP, de esta manera los datos recolectados, se pueden monitorear en app web en una red local o una app web sobre un servidor en internet.

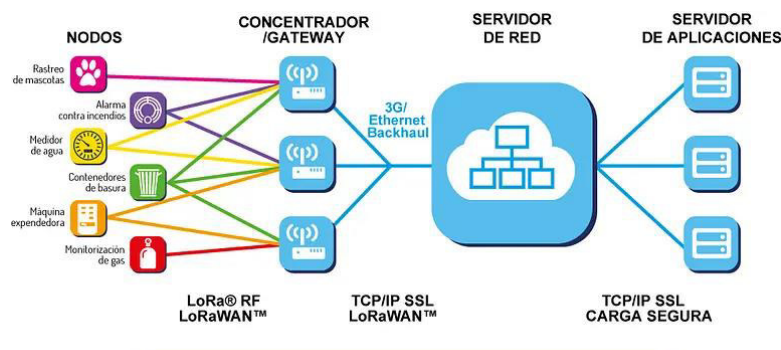


Figura 1.8. Esquema general de LoRaWan [23].

1.3.3.3 Mensaje LoRa

Un mensaje LoRa tiene la siguiente estructura en la capa física se compone de los elementos mostrados en la figura 1.9. Un preámbulo que está compuesto por 10 símbolos sin modulación, una cabecera seguida de la verificación de la cabecera a continuación el payload que contiene los datos del mensaje, y por último el CRC de verificación de todo el mensaje, solo en el caso de mensajes que son enviados o mensajes de uplink [24]. Se consideran mensajes de uplink aquellos que son enviados desde el nodo final al receptor, mientras que downlink son aquellos que recibe el nodo final desde el receptor.

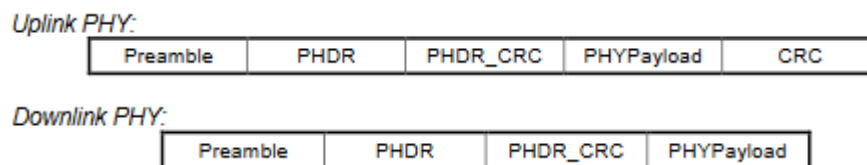


Figura 1.9. Estructura de un mensaje LoRa [24].

1.3.3.4 Parámetros LoRa

LoRa basa la transmisión de datos en la modulación de frecuencia pulsada (chirp), la frecuencia aumenta o disminuye linealmente (up/down chirp) oscilando dentro del ancho de banda asignado a partir de 125KHz como se muestra en la figura 1.10. De esta manera se pasa la señal que originalmente son bits binarios a una señal analógica para ser transmitido [25].

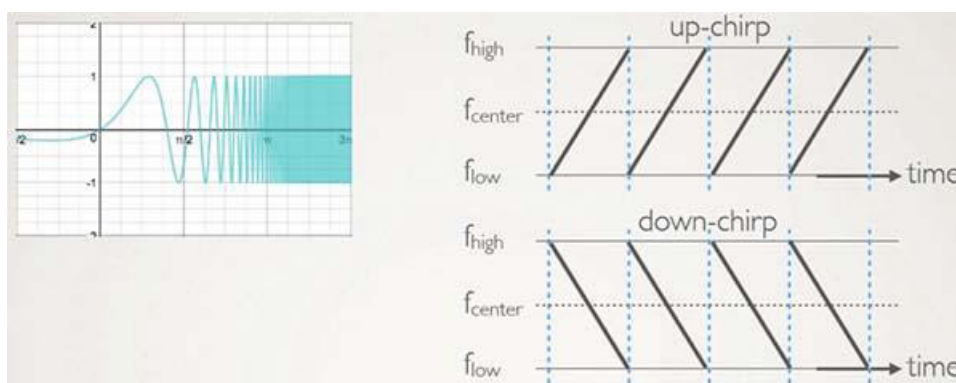


Figura 1.10. Modulación de frecuencia pulsada CHIRP [25].

Existen diferentes parámetros que intervienen en la comunicación LoRa, los cuales establecen lineamientos para configurar un buen uso de la red y aquellos que

representan información sobre la calidad de la comunicación. A continuación, se definen estos parámetros. [26]

1.3.3.4.1 Spreading factor

Un chirp también es conocido como un símbolo y en ellos están decodificados los datos. El spreading factor (SF) se encarga de controlar el número de datos contenido en estos símbolos y por tanto la velocidad de transmisión. Cuando menor sea el SF los chirps tienen una menor duración, y por tanto los datos tardarán menos tiempo en enviarse [25]. Por otro lado, por cada incremento de SF la duración del chirp se duplica como se muestra en la figura 1.11., por lo tanto, se requiere de más tiempo para enviar toda la información. LoRa tiene 6 posibilidades de SF del 7 al 12.

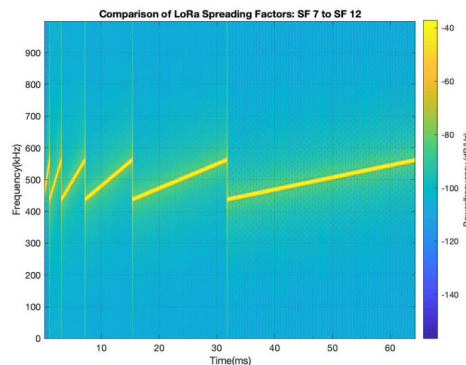


Figura 1.11. Modulación de frecuencia para diferentes valores de SF [25].

En resumen, cuanto más aumenta el SF mejora el procesamiento y por tanto existe menor posibilidad de errores. Por lo tanto, se puede mejorar la distancia de transmisión, sin embargo, al incrementarse el tiempo requerido para enviar cada chirp, se requiere más tiempo para la transmisión de información ToA como se observa en la figura 1.12. [27], esto a su vez influye en el duty cycle al reducir la cantidad de mensajes diarios. El SF influye directamente también en el tamaño de payload que se puede enviar por mensaje.

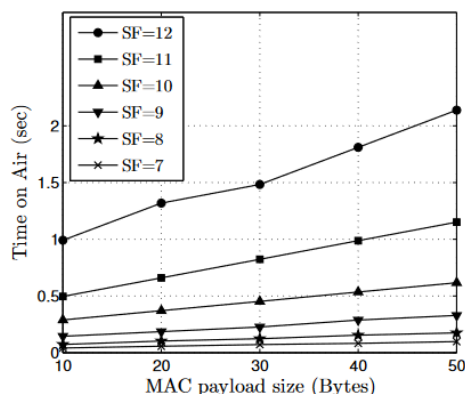


Figura 1.12. ToA vs la cantidad de bytes de payload [27].

1.3.3.4.2 Duty cycle

Uno de los inconvenientes de LoRa es que el estándar establece lineamientos respecto a cada cuanto tiempo es posible enviar un mensaje. Cada país recomienda cuanto tiempo un dispositivo puede hacer uso de esta banda libre y en general se establece que cada equipo puede usar el 1% diario de ciclo de trabajo [27].

El tiempo de espera entre mensajes se calcula mediante la ecuación 1.1, tomando en cuenta el duty cycle d de 0.01 p.u., con el ToA sugerido por TTN de 1.2 segundos, si por ejemplo se considera un SF de 11 y un payload de 50 bytes como se observa en la figura 1.9. el tiempo de espera mínimo entre mensajes sería de 119 segundos.

$$T_s = T_{oA} \left(\frac{1}{d} - 1 \right) \quad (1.1)$$

De igual manera el duty cycle tiene incidencia sobre la autonomía de los dispositivos LoRa que emplean baterías, si los mensajes se envían con el T_s mínimo la autonomía es menor en contraste con un T_s 5 veces mayor. Esto se vuelve evidente por ejemplo en dispositivos comerciales, en el dispositivo de monitoreo GPS desarrollado por LightBug, en la web del fabricante [28] se especifica que si los datos se actualizan cada 2 minutos como en el ejemplo mencionado en el párrafo anterior el tiempo de autonomía es de hasta 12 días, mientras que si la actualización se realiza cada 10 minutos la autonomía aumenta hasta un mes y medio. Esta información es importante puesto que a partir de estas consideraciones se establecerá el tiempo de actualización de los datos de GPS del prototipo de monitoreo.

1.3.3.4.3 Received Strength Signal Indicator (RSSI)

El RSSI, es un indicador que mide la cobertura general de un transceptor LoRa, como su nombre lo indica informa sobre la intensidad de la señal recibida. El RSSI se mide en dBm (decibelios - mili watt) y oscila entre -80 dbm a -120 dbm mientras menor sea este indicador menor es la intensidad de la señal, y esta se reduce dependiendo de la distancia desde el nodo final hasta el receptor o concentrador, también se ve afectada por el terreno. El estudio realizado por [29] analiza el desempeño de una LPWAN (Low Power Wide Área Network) que emplea tecnología LoRa para aplicaciones IoT, con la configuración mostrada en la tabla 1.4.

Tabla 1.4. Parámetros para el análisis de desempeño de LPWAN mediante tecnología LoRa [29].

Parámetro	Valor
Frecuencia	923 – 925 MHz
Ancho de Banda	125kHz
Spreading Factor	7, 8, 9, 10, 11, 12
Tamaño del Payload	19 bytes

El análisis se realizó en una zona rodeada de varios edificios, y como se observa en la figura 1.13, se tiene que para distancias menores de 200 metros el RSSI es más alto cuando se emplea un SF menor pero solo alcanzo cobertura hasta 300 metros, para SF mayores a partir de 200 metros se alcanza un RSSI estable entre -100 y -110 dBm.

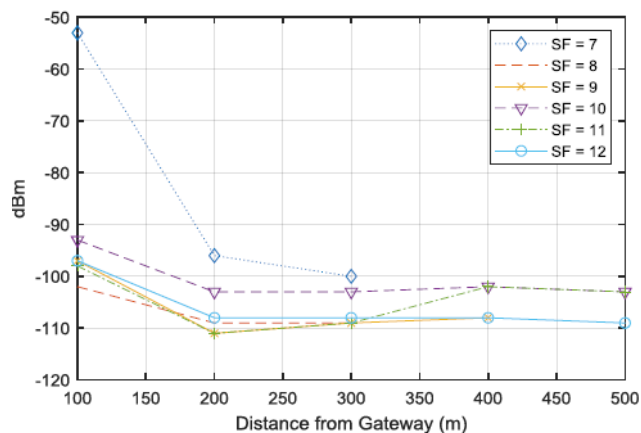


Figura 1.13. RSSI en función de la distancia desde el Gateway diferentes SF [29],

1.3.3.4.4 *Signal-to-noise ratio(snr)*

Este indicador es una relación entre la intensidad de la señal y el nivel de ruido [30], no tiene unidades, como se observa en la ecuación 1.2. Mientras mayor sea el valor de SNR existe menos probabilidad de error, puesto que una señal con menor contenido de ruido es más sencilla de decodificar, y varía entre +10 y -20 [22].

$$SNR = \frac{P_{signal}}{P_{noise}} \quad (1.2)$$

El siguiente estudio [31], determinó que existe una relación entre SNR, RSSI con SF, mientras menor sea el SF empleado tanto RSSI como el SNR mejoran, desde luego esto tiene que ver con la distancia de cobertura. El resultado se muestra en la figura 1.14.

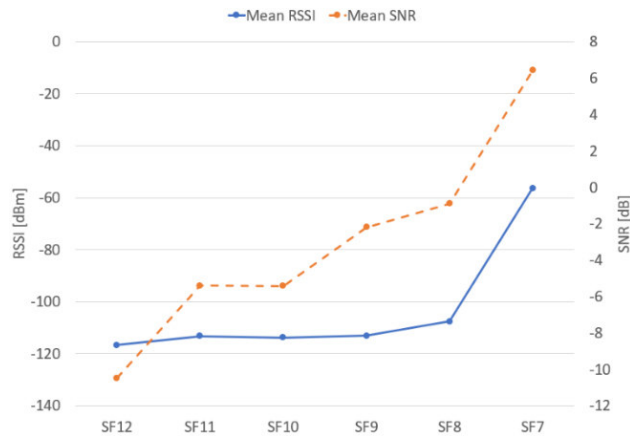


Figura 1.14. Relación entre RSSI, SNR con SF [31].

1.3.3.4.5 *Frecuencia de operación*

Uno de los parámetros principales es la frecuencia en la que opera LoRa. Debido a que LoRa opera en la denominada banda ISM de espectro libre industrial, científica y médica la cual es una banda de espectro libre, de acuerdo con LoRa Alliance en su reporte [32] se establece que la banda asignada a Ecuador está entre 902 a 928 MHz y se recomienda el uso de la configuración de la banda AU915-928MHz. Esta banda permite configurar los siguientes parámetros mostrados en la tabla 1.5 [33].

Tabla 1.5. Parámetros permitidos para la banda AU915 – 928 MHz [33].

Data Rate	SF	Bit Rate [bit/sec]	Payload máximo [bytes]	ToA máx. [ms]
0	12	240	59	2793.5

1	11	440	59	1560.5
2	10	980	59	698.4
3	9	1760	123	676.9
4	8	3125	230	707.1
5	7	5470	230	399.6

1.3.4 TECNOLOGÍAS DISPONIBLES

Existen varias alternativas de dispositivos en el mercado internacional que tienen como característica principal el monitoreo de geo posición, otros que integran también medición de temperatura y existen investigaciones que integran acelerómetros para analizar la rumia del ganado. En este apartado se analiza aquellos que emplean LoRa como tecnología para el envío de datos.

1.3.4.1 Moovement

Este dispositivo es fabricado por MOOVEMENT con sede en Australia emplea tecnología LoRa para la transmisión de datos [34], el dispositivo integra un panel solar, transmite datos GPS cada dos horas, las características del dispositivo se detallan a continuación:

- Instalación en la oreja
- Rango cobertura: 8 [Km], dependiendo de las características del terreno.
- Peso: 30 [gr]
- Precisión del GPS: 15 [m]
- Autonomía: 5 años
- Costo en Australia: 60 \$



Figura 1.15. Dispositivo de rastreo mOOvement [34].

1.3.4.2 Sodaq

El fabricante de Países Bajos diseño este dispositivo para rastrear la posición de vacas, caballos y renos [35], se coloca alrededor del cuello del animal como se muestra en la figura 1.16. El dispositivo permite modificar su funcionalidad debido a que está basado en el microcontrolador SAMD21 de 32 bits. Las características del dispositivo se muestran a continuación [36].

- Microcontrolador empleado SAMD21
- GPS EVA 8M Ublox
- Panel solar de 0.5 [W] y una batería LiPo de 2200 [mAh]
- Autonomía: 2 semanas
- Precisión: 30 [m]
- Costo en países bajos: 232.05 \$



Figura 1.16. Dispositivo de rastreo de ganado SODAQ [35].

1.3.4.3 LightBug

Desarrollado por el fabricante inglés LightBug. El dispositivo añade un acelerómetro para detección de movimiento, y permite observar los datos tanto en aplicaciones en un dispositivo móvil, así como en una aplicación WEB [37].

- Autonomía: 2.6 meses [28].
- Actualización de datos cada 15 minutos
- Peso aproximado 30 [gr]
- Batería 550 [mAh]
- Cobertura hasta 10 [Km]
- Precisión del GPS 0.5 - 5 [m]

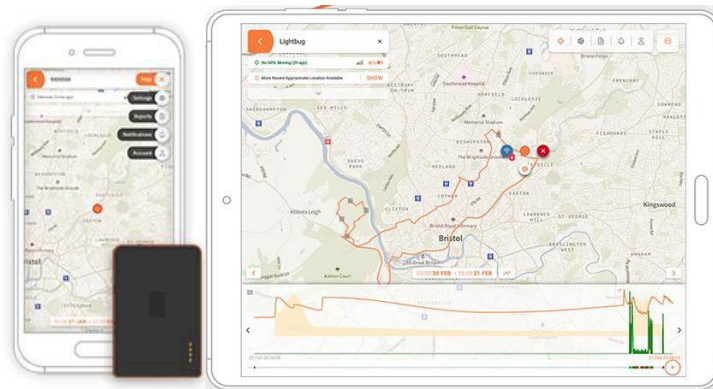


Figura 1.17. Dispositivo de rastreo LightBug Zero [37].

1.3.5 COMPONENTES ELECTRÓNICOS A UTILIZARSE EN EL PROTOTIPO

Después de analizar la importancia de medir geo posición, temperatura y considerando las características que deberían tener los elementos de comunicación que emplean LoRa es importante buscar elementos que se encuentren disponibles en el mercado que permitan desarrollar el prototipo dentro de los parámetros esperados, de manera que pueda tener características similares a los dispositivos comerciales actualmente.

Los dispositivos comerciales tienen una característica en común, el hardware se divide en dos partes, por un lado, está el nodo final que es el dispositivo remoto y en otro el dispositivo receptor o gateway, que recibe la información y se encarga de volverla utilizable ya sea en una interfaz local o a través de internet. Es así como, se consideran para la implementación de este proyecto los elementos que se detallan a continuación.

1.3.5.1 GPS

En el mercado los chips GPS más extendidos son los desarrollados por la compañía UBLOX, y que se emplean en diferentes dispositivos como drones. Existen diferentes versiones entre las cuales sobresalen NEO 6M, NEO 7M y NEO 8M [38], [39], [40]. En la tabla 1.6 se muestran las diferencias de estos elementos.

Tabla 1.6. Características principales GPS UBLOX [38], [39], [40].

	NEO- 6M	NEO-7M	NEO-8M
Canales	50	56	72
Máxima frecuencia de actualización	1 Hz	10 Hz	10 Hz
Exactitud posición horizontal	<1 m – 2 m	2 m - 2.5 m	2.5 m – 8 m
Tiempo de arranque en frío	27 s	30 s	26 s – 57 s
Costo aproximado	12.0 \$	20.0 \$	28.5 \$

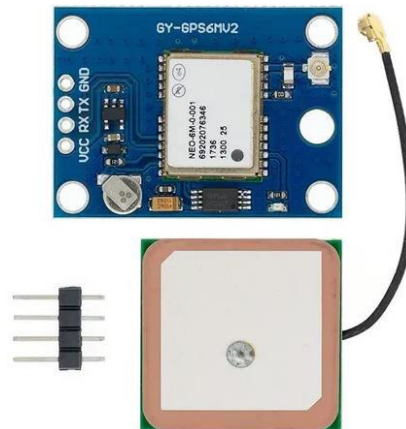


Figura 1.18. Módulo GPS NEO-6M [38].

El dispositivo que se utilizará es el GPS NEO-6M, debido principalmente al costo del dispositivo a costa de una actualización de datos más lenta. El módulo considerado para el diseño es el mostrado en la figura 1.18, y tiene las siguientes características técnicas.

- Consumo de corriente promedio: 39 [mA]
- Voltaje de alimentación: 2.7 – 3.6 [V]
- Comunicación serial TTL

1.3.5.2 Sensor de temperatura corporal

Como se analizó en los métodos de medición de temperatura, es conveniente utilizar un sensor de temperatura infrarrojo, ya que este sensor no requiere estar en contacto directo con la piel el sensor de fácil acceso es la termopila MLX 90614 [41]. La serie que se considera para utilizar en este proyecto es MLX90614BAA con las siguientes características.

- Rango de temperatura: -40 a 125 [°C]
- Voltaje de alimentación: 3.6 [V]
- Precisión: 0.5 [°C]
- Consumo de corriente máximo: 25 [mA]
- Resolución: 0.02 [°C]
- Modo de ahorro de energía
- Comunicación I2C



Figura 1.19. Termopila MLX90614 [41].

1.3.5.3 Sensor de Temperatura Ambiente y Humedad Relativa

En base a lo que se observó respecto al monitoreo de temperatura, la temperatura y humedad relativa del ambiente que rodea al ganado influye en su temperatura corporal. El INAMHI (Instituto Nacional de Meteorología e Hidrología) tiene distribuidos a lo largo del país una red de estaciones meteorológicas [42], sin embargo, no todas las haciendas cuentan con acceso a Internet para obtener esta información, por esta razón se considera

agregar un sensor que permita monitorear estas variables del ambiente el cual puede ser medido en el mismo lugar en el que se ubicará el dispositivo receptor.

El sensor considerado es el DHT22/ AM2303 [43], es un sensor de salida digital de 8 bits que permite medir humedad relativa y temperatura, sus características se detallan a continuación.

- Voltaje de alimentación: 3.3- 6 [Vdc]
- Rango: 0 - 100% RH, -40 – 125 [°C]
- Precisión: +-2% RH, +- 0.2 [°C]
- Resolución: 0.1% RH, 0.1 [°C]

1.3.5.4 Chip LoRa

El fabricante de chips LoRa es SEMTECH, los chips empleados en nodos finales son especialmente los modelos SX127x [44], mientras que los concentradores se basan en los chips SX1301 [45] que poseen hasta ocho canales de recepción, sin embargo, también se puede usar los chips SX127x para emplearlos como concentradores de un solo canal.

Tabla 1.7. Frecuencia y Spreading Factor de los chips LoRa más utilizados [44] [45].

	Rango de frecuencia	Spreading Factor	Número de Canales
SX1276	137 – 1020 MHz	6-12	1
SX1277	137 – 1020 MHz	6-9	1
SX1278	137 – 525 MHz	6-12	1
SX1279	137 – 960 MHz	6-12	1
SX1301	137 – 960 MHz	7-12	8

Existen concentradores LoRa diseñados como módulos de expansión para Raspberry. Un ejemplo es el RAK2245 [46] mostrado en la figura 1.20, está basado en el chip LoRa SX1301 con ocho canales de recepción, soporta las diferentes bandas globales, sin embargo al momento de realizarse este trabajo de titulación el módulo no se encuentra disponible en la web del fabricante.

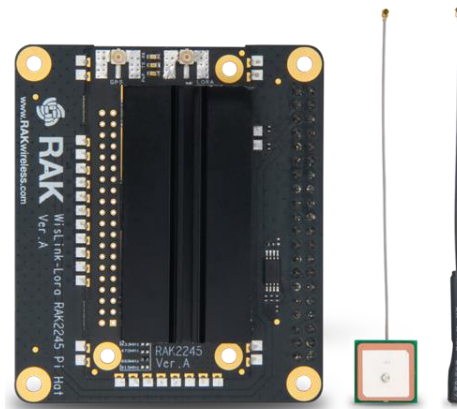


Figura 1.20. Rak2245 Pi HaT compatible con Raspberry 3B+ [46].

Al ser posible usar un chip LoRa destinado principalmente para su aplicación como nodo como un concentrador se puede utilizar el chip RFM95W, que está basado en el chip LoRa SX127x, pero con una técnica de modulación diferente. El chip RFM95W mostrado en la figura 1.21. fabricado por HOPERF [47] al estar basado en el SX127x, posee los mismos registros internos y de esta manera es compatible con las librerías desarrolladas para este chip LoRa empleadas en ESP32.

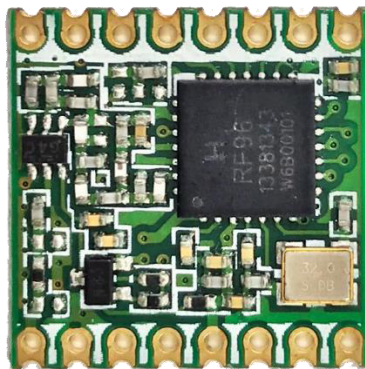


Figura 1.21. RFM95W [47].

1.3.5.5 Raspberry Pi

Raspberry Pi es un minicomputador, a pesar de su reducido tamaño permite desarrollar aplicaciones orientadas a IoT, existen diferentes modelos de los cuales se tienen 3B+ y 4 como los más potentes. Posee los mismos periféricos de un computador convencional, como puerto HDMI para conectar una pantalla en el modelo 3 B+ o dos en el modelo Pi 4, puertos USB, bluetooth, wifi y puerto Ethernet. La principal diferencia es que emplea una memoria flash en lugar del disco duro magnético de una PC, que podría compararse con

un disco solido actual, pero de menor capacidad y rendimiento, en el modelo Pi 3 B+ la memoria RAM es de 1 GB mientras que en el modelo Pi 4 puede ser de hasta 8 GB [48], [49]. Los periféricos y componentes se pueden observar en la figura 1.22.

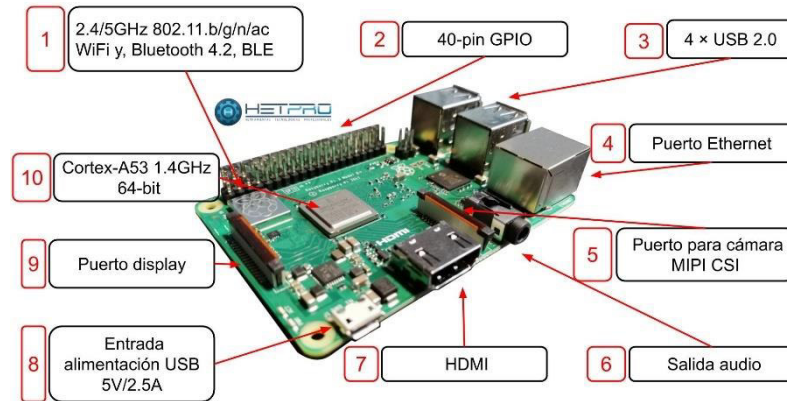


Figura 1.22. Raspberry 3 B+ [48].

La tarjeta es capaz de correr sistemas operativos, como se muestra en la figura 1.23, el principal es Raspbian que es una distribución de Linux diseñada especialmente para Raspberry, el sistema operativo tiene diferentes versiones la más completa aquella que presenta una interfaz de usuario similar a las distribuciones de Linux y otra versión mucho más ligera que solo cuenta con la consola de comandos, el uso de las versiones depende del programador.

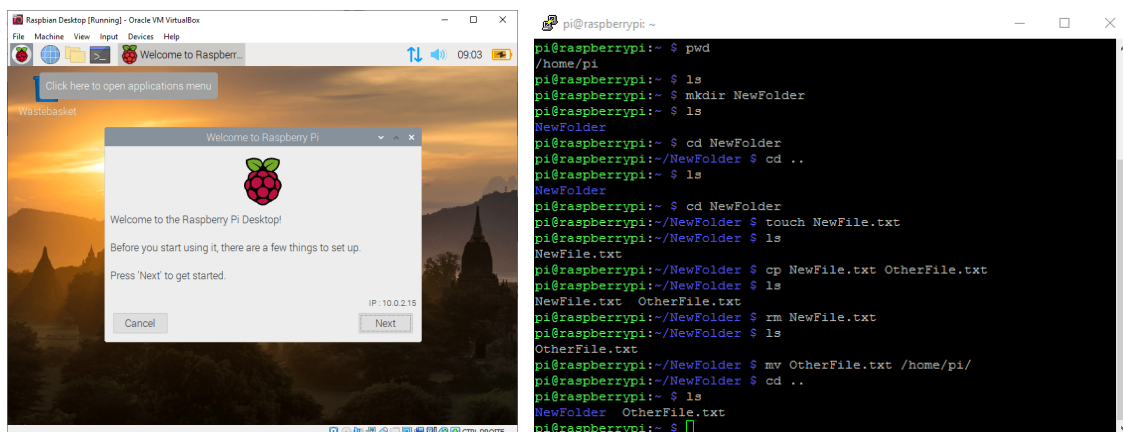


Figura 1.23. Raspbian OS versión de escritorio y versión lite [Elaboración Propia]..

La principal ventaja de Raspberry es que es posible correr sobre el sistema operativo programas desarrollados en diferentes lenguajes de programación, como Python y sus librerías [50], C, C++, así como el uso de bases de datos, interfaces desarrolladas en

Python, JavaScript, CSS y HTML. Es posible montar sobre la tarjeta un servidor de datos, y debido a que tiene puertos de entrada y salida, realizar acciones como monitorear variables y controlar actuadores a través de programación de forma remota.

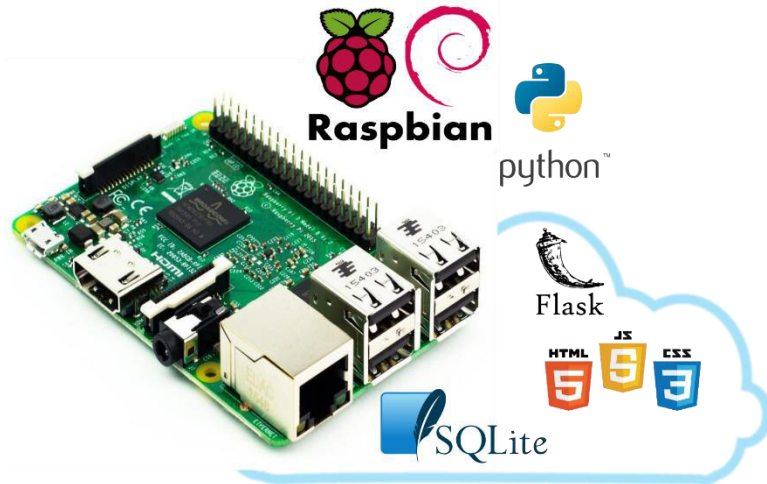


Figura 1.24. Conjunto de herramientas de programación Raspberry [Elaboración Propia].

1.3.5.6 ESP32

Para poder gestionar el envío, recepción de datos y demás funciones del chip LoRa se requiere de un elemento micro controlado o micro procesado, desde este punto de vista existen varias alternativas en el mercado desde tarjetas sencillas como Arduino, ST Electronics hasta otras más complejas como Raspberry, Jetson Nano y Texas Instruments. Sin embargo, actualmente existen tarjetas comerciales que integran los chips LoRa y dejan las labores de gestión de datos desde y hacia la tarjeta al microcontrolador ESP-32 del fabricante Expressif el modelo propuesto se encuentra en la figura 1.20 y en esa medida existe mayor desarrollo de librerías en torno al manejo de datos LoRa a través de este microcontrolador, a continuación, se listan características de este chip [51].

- 2 DAC de 8 bits
- Comunicación SPI, I2S, I2C, UART
- Conexión WI-FI, Bluetooth
- Corriente de operación 50 – 130 [mA]
- Chip extensa doble núcleo con 32 bits.
- Oscilador interno 8 [MHz]
- 34 entradas y salidas digitales

- Conversor ADC de 12 bits hasta 18 canales
- 10 sensores táctiles

Actualmente existen tarjetas que integran el chip LoRa SX127x y el microcontrolador ESP32, como la tarjeta desarrollada por HELTEC, en la figura 1.25 se muestra la tarjeta HELTEC V1, el principal problema de esta tarjeta es que integra una pantalla OLED y el chip LoRa siendo estos conectados directamente a la salida del regulador de 3.3 voltios integrado en la tarjeta, si no es requerido el uso de la pantalla, o si se requiere que el micro controlador pase a estado de ahorro de energía persiste un consumo de energía de los elementos electrónicos aunque es mínimo el consumo acumulado no permite que la tarjeta reduzca al máximo el consumo de energía.

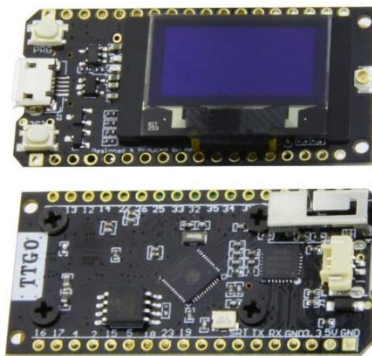


Figura 1.25. Módulo LoRa basado en ESP32 [52]

Una alternativa es emplear el microcontrolador ESP32 por separado o una tarjeta microcontroladora que lo contenga, es así como se puede emplear la tarjeta Firebeetle mostrada en la figura 1.26. desarrollada por DFRobot [53], una de las diferencias principales entre esta tarjeta y la HELTEC V1 es el regulador de 3.3v, el cual cuando el microcontrolador está en modo sueño tiene una corriente de inactividad (Quiescent Current) de 2uA [54] mientras que la HELTEC V1 tiene una corriente de 55uA [55]. Los errores se reducen en la siguiente versión de la tarjeta Heltec V2 [52], puesto que agrega un regulador de menor corriente de inactividad, y permite controlar la conexión de la pantalla OLED hacia la fuente de alimentación mejorando la eficiencia en ahorro de energía, pero mantiene conectado el chip LoRa y por ende no reduce el consumo de corriente lo suficiente en estado de sueño profundo.

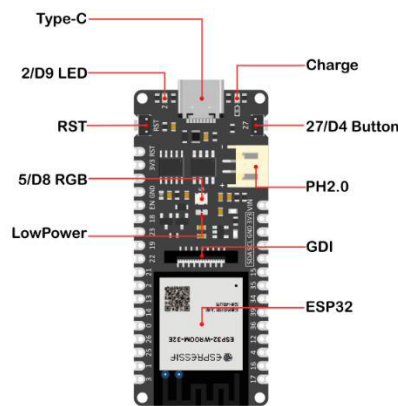


Figura 1.26. Firebeetle [53].

1.3.5.7 Bateria

Para la autonomía del dispositivo remoto se considera utilizar una batería, en el mercado sobresalen las baterías de ion litio del modelo 18650 por su capacidad, si bien es cierto, existen baterías que sugieren capacidad de más de 3000mAh en especial de fabricantes chinos, son falsas y no alcanzan la capacidad anunciada. Muchas de estas baterías tienen un peso inferior en contraste con aquellas producidas por fabricantes confiables como Samsung o Sony, y aquellas que tienen un peso aceptable están rellenas de arena u otros componentes[56]. La capacidad actual de las baterías 18650 comerciales es de 3500mAh, en la figura 1.27 se observa las diferencias entre baterías falsas (naranja y amarilla) y una batería confiable (negra), internamente las baterías falsas no ocupan todo el espacio disponible y contienen una batería de menor capacidad, a pesar ello en el caso de la batería amarilla promete 9900mAh.



Figura 1.27. Baterías 18650 ion litio de diferentes fabricantes [56].

En la figura 1.28 se observa la batería que se considera como fuente de alimentación para el dispositivo remoto. Es una batería de ion litio de 3000mAh del fabricante Sony [57], las características se muestran a continuación:

- Modelo: 18650
- Voltaje de carga completa: 4.2 ± 0.05 [V]
- Corriente de carga: 3.0 [A]
- Tiempo de carga 2.5 [h]



Figura 1.28. Batería Sony VTC6

1.3.6 SOFTWARE A UTILIZARSE EN EL PROTOTIPO

Para poder llevar a cabo el desarrollo del prototipo, en el lado del nodo final es necesario configurar los parámetros LoRa, obtener datos desde el GPS, del sensor de temperatura, enviarlos y gestionar el funcionamiento de cada elemento que compone el nodo final para administrar el consumo de energía. En el caso del receptor LoRa es necesario gestionar los datos recibidos mediante LoRa para almacenarlos en una base de datos y publicarlos en la interfaz de usuario, de manera que el cuidador de ganado pueda realizar acciones con la información suministrada.

1.3.6.1 Arduino IDE

Las tarjetas LoRa basadas en ESP-32 son compatibles con la programación en Arduino IDE y de la misma manera es extendido el desarrollo de diferentes librerías para el control de GPS neo 6m, MLX90614 y el chip LoRa.

1.3.6.2 Python

Python es un lenguaje de programación de alto nivel, soporta programación orientada a objetos y maneja múltiples librerías, incluso librerías desarrolladas entorno a minicomputadores con puertos de entrada y salida de uso general como en el caso de Raspberry Pi, incluso librerías que permiten desarrollar aplicaciones web [58].

1.3.6.3 Flask

Flask es un entorno de trabajo (framework) web que no es más que un conjunto de librerías y módulos que permiten que un desarrollador web no necesite preocuparse acerca de detalles de bajo nivel tales como el protocolo, manejo hilos de programación entre otros [59]. Flask está escrito en Python y en esta media es un framework WSGI (Web Server Gateway Interface), que a diferencia de un servidor tradicional permite la ejecución de aplicaciones desarrolladas en Python [60], como se observa en la figura 1.29, el servidor WSGI es el que se encarga de enlazar la página web almacenada en el archivo HTML, para que este pueda ser ejecutado en un web browser, el web server pasa la solicitud del web browser hacia el WSGI y retorna la respuesta de este en formato HTML, esto debido a que el WSGI se ejecuta en un puerto diferente al del web server.

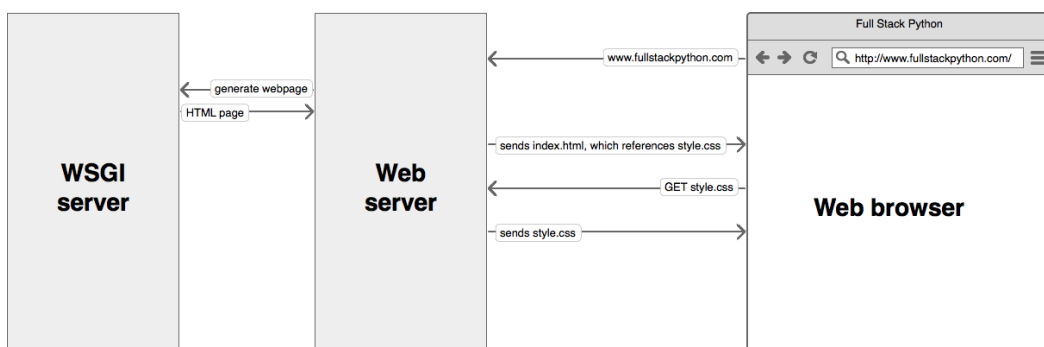


Figura 1.29. Interacción entre el WSGI y el Web Browser [60]

1.3.6.4 Open Layers

Los dispositivos comerciales en sus interfaces integran un mapa del entorno de crianza, en vista de esta característica y dado que los datos recibidos del nodo final son de georreferenciación es decir latitud y longitud, se considera usar una herramienta que permita reflejar estos datos sobre un mapa previamente desarrollado.

Open Layers es una librería de código abierto desarrollada principalmente para aplicaciones web la cual permite desplegar e interactuar con mapas y datos geoespaciales. Está diseñado para operar en todos los buscadores web modernos que soporten HTML5 y por tanto puede usarse en computadoras, portátiles y dispositivos móviles [61].

1.3.6.5 Herramientas de Desarrollo Web

Como se detalla en el literal anterior Open Layers está diseñado para montarse sobre aplicaciones web principalmente, por lo tanto, para el desarrollo de la interfaz en el cual se mostrará la geo-posición de una res de ganado dentro de su área de crianza, se debe tomar en cuenta las herramientas de programación que permitan el uso de la librería Open Layers. Es así como se definen las siguientes herramientas de Desarrollo Web para este proyecto técnico [62].

1.3.6.5.1 JavaScript

Es un lenguaje de programación el cual sirve para establecer las tareas que se llevan a cabo en una página web, se encarga de la parte dinámica de la página, tareas repetitivas, mapas interactivos, animaciones en 2d y 3d y demás tareas de interacción entre la página y el usuario.

1.3.6.5.2 HTML

HTML es un lenguaje que se emplea para estructurar y darle orden al contenido de una página web [63]. En un script HTML se definen los elementos que formaran parte de la interfaz como párrafos, tablas, mapas, imágenes y videos, también es aquí donde se hace referencia a otros scripts que contiene las reglas de estilo de la página y los scripts que contienen librerías y el programa que contiene las instrucciones desarrolladas en JavaScript que ejecutará la página.

1.3.6.5.3 CSS

Es un lenguaje que contiene reglas de estilo, se usa para configurar la apariencia del contenido HTML, como color de fondo, fuente y tamaño de letras, así como la posición y tamaño de objetos [64].

JavaScript, HTML y CSS son tres lenguajes básicos encargados de funcionalidad, estructura y estilo respectivamente que en conjunto permiten la construcción de una página web.

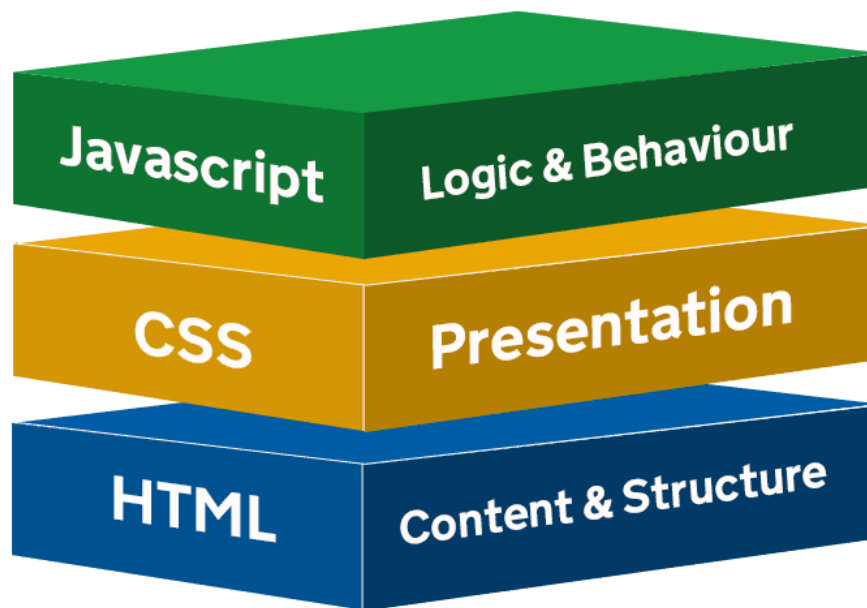


Figura 1.30. Lenguajes básicos para desarrollo de una página web [62].

1.3.6.6 SQLITE 3

Para poder almacenar y clasificar datos, se requiere de una herramienta que permita contener la información de manera ordenada en este sentido se puede emplear una base de datos. Una base de datos es una recopilación de datos estructurados, para consultar o manipular los datos almacenados en ella emplean un lenguaje de programación llamado SQL [65] (Structured Query Language).

Existen diferentes bases de datos empezando por las principales comerciales como Oracle y SQL server, así como bases de datos de código abierto. Debido a que se pretende desarrollar un prototipo y la cantidad de información que se pretende almacenar es pequeña una alternativa es el empleo de una base de datos sencilla, y en ese contexto

SQLITE 3 es una buena alternativa. SQLite es una base de datos de código abierto relacional [66], lo cual significa que los datos están organizados en un conjunto de tablas, de varias filas y columnas lo cual permite un acceso flexible y eficiente a los datos. Pese a ser una base de datos sencilla es ampliamente utilizada, al punto de estar presente en smartphones basados en iOS y Android [67].

2. METODOLOGÍA

Este proyecto se desarrolla en tres fases. En el primer capítulo, correspondiente a la primera fase, se hace énfasis en la teoría, mediante la referencia a investigaciones previas que resaltan la importancia de conocer geo posición y monitorear temperatura en ganado vacuno para su crianza, así como los conceptos detrás de los componentes tanto de software como de hardware a través de los cuales es posible realizar estas tareas de monitoreo. La segunda fase, correspondiente al diseño e implementación, que parte desde el diseño hardware y software para continuar con la implementación mediante el ensamblaje de cada componente del prototipo, así como el enlace entre cada uno de los programas que permiten obtener el producto final demostrable. Finalmente se tiene la fase de resultados en la cual se observa el funcionamiento del prototipo y se analizan variables asociadas a su desempeño.

En la figura 2.1 se muestra un esquema general del prototipo, el cual se indica que se obtiene información desde el nodo final colocado sobre una vaca, esta información es recolectada por el dispositivo receptor y la cual se puede visualizar en un computador, de modo que si la vaca sale del rango de la zona de crianza un empleado de la hacienda pueda ir a buscarla.

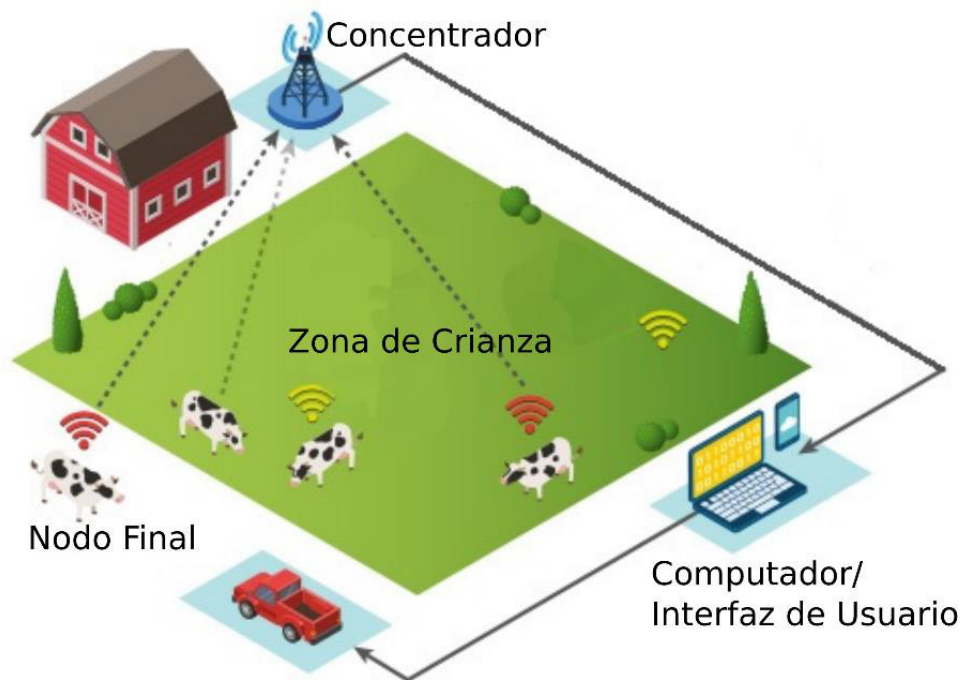


Figura 2.1 Esquema general del prototipo [Elaboración Propia].

2.1. DESARROLLO DE HARDWARE

En el capítulo anterior se expuso la importancia de conocer geo-posición y temperatura corporal en el ganado, los beneficios que LoRa ofrece para este tipo de aplicaciones de monitoreo, y aquellos elementos que tienen las características necesarias para realizar las tareas descritas. A partir del análisis bibliográfico se llevará a cabo el desarrollo del hardware del prototipo, su diseño y construcción.

En la figura 2.2 se observa un esquema general del hardware, en el lado izquierdo está el dispositivo remoto o nodo final y en el lado derecho el dispositivo receptor con sus respectivos componentes.

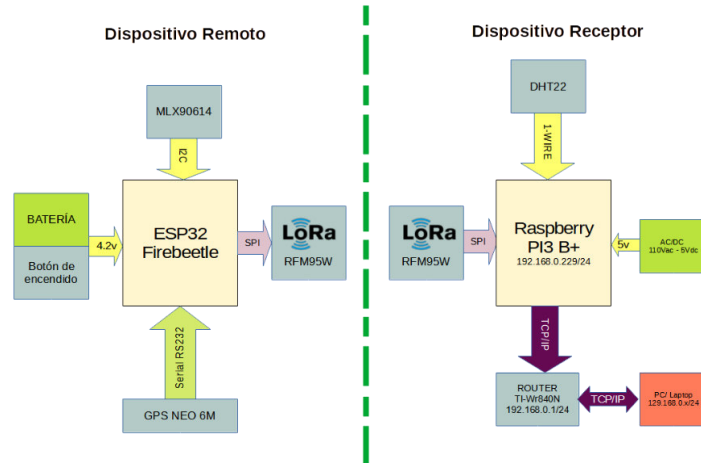


Figura 2.2 Esquema general del hardware [Elaboración Propia].

2.1.1 Dispositivo Remoto

Para la construcción del nodo final o dispositivo remoto, se establecerá la distribución de pines del microcontrolador para la conexión del GPS, el sensor de temperatura y el sensor de batería, basados en las hojas de datos de los fabricantes de estos dispositivos. En la figura 2.3 se muestra los pines disponibles en la tarjeta Firebeetle de DFRobot los cuales corresponden a los pines del microcontrolador ESP-32 [53].

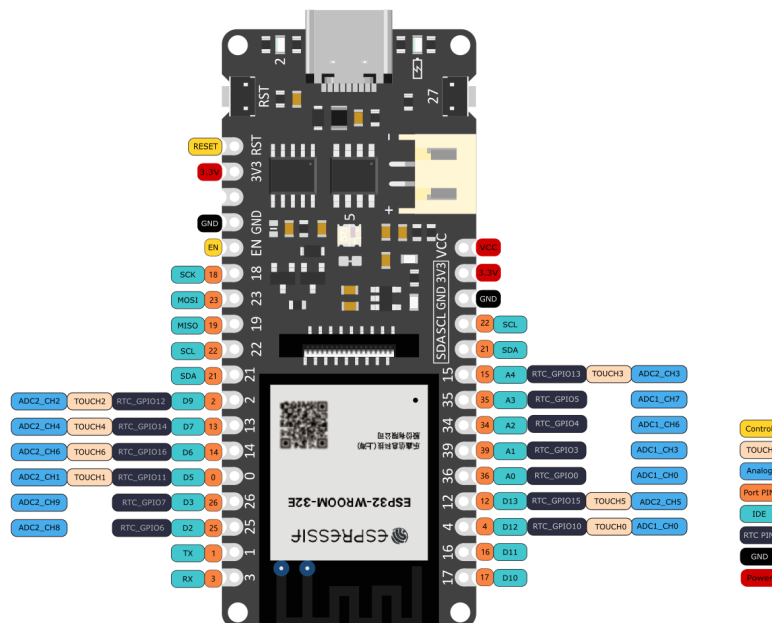


Figura 2.3. Pines Tarjeta de desarrollo Firebeetle [53].

Se requiere para la conexión del GPS pines RX y TX, para lo cual se dispone de tres puertos de comunicación serial RS232 en el ESP-32, el puerto 1 se utiliza en la tarjeta

internamente para la comunicación con la memoria flash y no tiene pines externos en la tarjeta, el puerto 0 está disponible externamente, sin embargo, internamente se conecta con el convertidor de RS232 a USB (CP2102) que permite la comunicación de la tarjeta con un ordenador y permite programar el dispositivo. En base a lo anterior se decide emplear el puerto 2, no se empleará el puerto 0 ya que en caso de reprogramar el nodo se necesita desconectar el GPS y resultaría incomodo. Por otra parte, en el caso del sensor de temperatura se requiere de comunicación serial I2C, para lo cual se usarán los pines GPIO 21 y GPIO 22 para las señales de datos SDA y la señal de reloj SCL.

La tarjeta de desarrollo, está diseñada para aplicaciones que requieren alimentación de batería y cuenta con un circuito de carga TP4056x para baterías de Ion Litio [68]. La tarjeta cuenta con un circuito para monitorear el estado de carga de batería conectado internamente al pin GPIO34. De igual manera es deseable implementar un circuito de activación para los sensores de modo que cuando el dispositivo no esté operativo los sensores se apaguen y se reduzca el consumo de energía. En la figura 2.4 se muestra la arquitectura del hardware y en la tabla 2.1 se detallan los pines que se emplearán para conectar los diferentes componentes.

Tabla 2.1. Asignación de pines ESP-32.

Periférico	Pin periférico	GPIO ESP-32 Firebeetle
GPS	RX	16
	TX	17
SENSOR DE TEMPERATURA	SCL	22
	SDA	21
SENSOR BATERIA	A0	34
CONTROL DE ENCENDIDO SENSORES	DO1	15
CHIP LORA SX1276	MOSI	23
	MISO	19
	RST	14
	SCK	18
	CS	13

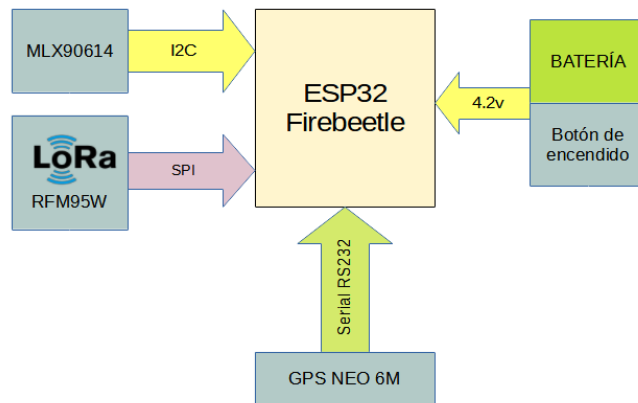


Figura 2.4. Arquitectura de Hardware del dispositivo remoto [Elaboración Propia].

2.1.2 Implementación Nodo Final

Una vez definidos los elementos y la distribución de pines de la tarjeta de desarrollo Firebeetle, se llevará a cabo del diseño de una placa electrónica en la cual se soldarán los diferentes elementos que componen este dispositivo. En la figura 2.5 se muestra el diseño del circuito impreso, de autoría propia y desarrollado en Altium.

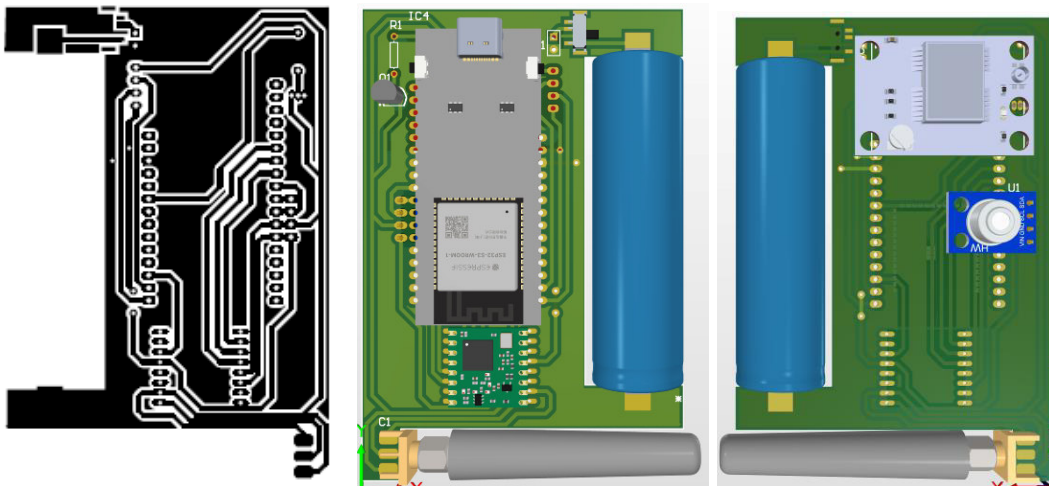


Figura 2.5. Diseño tarjeta electrónica para el nodo final [Elaboración Propia].

Una vez implementada la placa se procede a colocar los elementos en la placa, en la figura 2.6 se observa el resultado final.

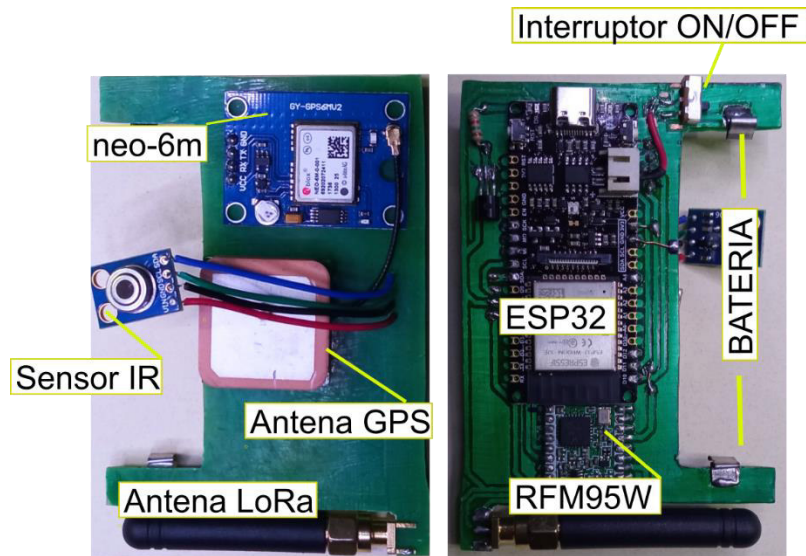


Figura 2.6. PCB implementada [Elaboración Propia].

Para alojar los elementos electrónicos se requiere de una carcasa, gracias a la impresión 3D y a software CAD se desarrolló una carcasa que pueda alojar los elementos y a su vez se pueda montar sobre un collar, en la figura 2.7 se muestra el resultado final, el dispositivo se monta en el collar mediante tornillos M3 distribuidos en la periferia de la carcasa, mientras que el collar está conformado por un broche y reguladores de plástico.

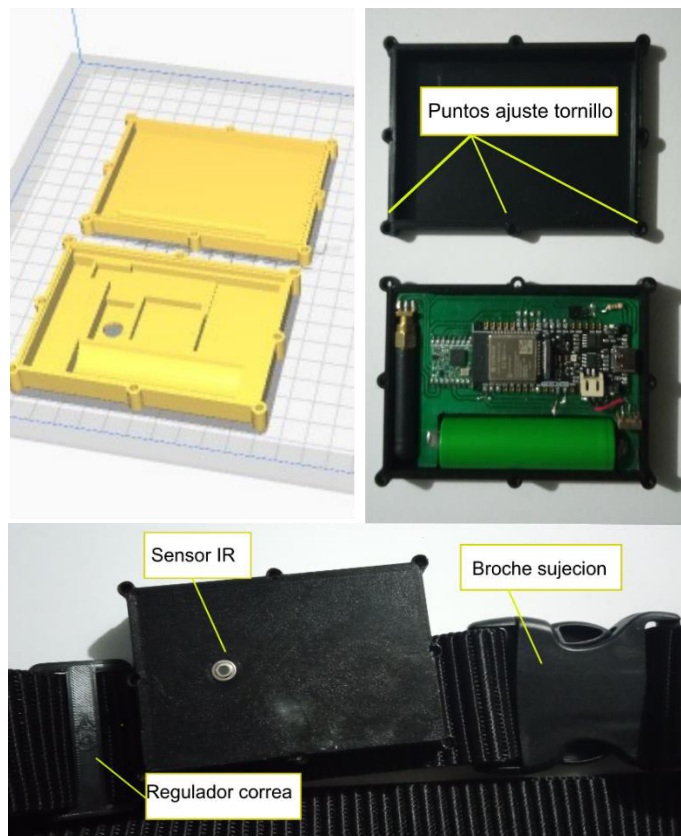


Figura 2.7. Estructura para el nodo final [Elaboración Propia].

2.1.3 Dispositivo Receptor

El siguiente paso es diseñar el receptor o concentrador de señal, de acuerdo con lo expuesto en la fase teórica el principal componente será una Raspberry Pi 3B+ para poder llevar a cabo el desarrollo de software tanto de la lectura del sensor de temperatura y humedad relativa, así como el chip LoRa.

En primer lugar, se requiere incluir el chip LoRa para obtener los datos desde el nodo final, como se analizó en la fase teórica se puede emplear un módulo diseñado especialmente para Raspberry como es el RAK2245, sin embargo, al momento de desarrollarse este trabajo de titulación el elemento no se encuentra disponible por parte del fabricante; en vista de este inconveniente, queda la posibilidad de implementar un concentrador basado en un chip LoRa de un solo canal como el SX1276. Es así, que se decide emplear el módulo RFM95W que esta basado en el chip SX1276 [47].

Por otra parte, se requiere conectar el sensor de temperatura y humedad relativa ambiente DHT22, para lo cual se emplea el GPIO 4 de la Raspberry Pi. Los pines destinados para la conexión de este módulo en la Raspberry son los mostrados en la tabla 2.2.

Tabla 2.2. Asignación de pines Raspberry Pi 3B+.

Periférico	Pin periférico	GPIO Raspberry
DHT 22 Sensor de Temperatura y Humedad relativa	OUT	4
CHIP LORA RFM95W (SX1276)	DIO0	22
	DIO1	23
	DIO2	24
	RST	25
	CLK	SCK
	MISO	MISO
	MOSI	MOSI
NSS	CE1	

2.1.4 Implementación del Receptor

Una vez definidos los elementos y la distribución de pines de la Raspberry se diseña una placa electrónica en la cual se conectará el DHT22 y se soldará el chip RFM95W. En la figura 2.8 se muestra el diseño del circuito impreso.

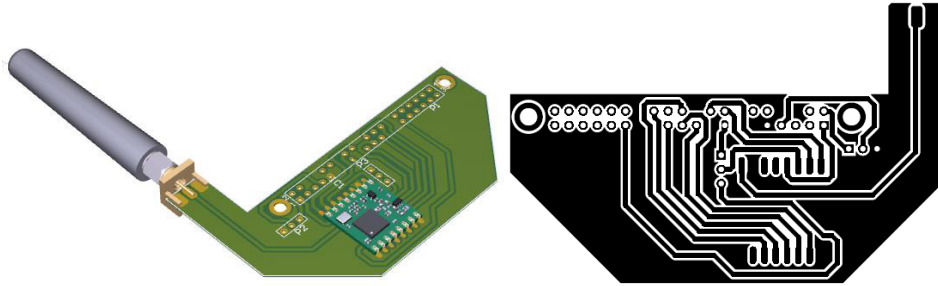


Figura 2.8. Diseño tarjeta electrónica que se colocará sobre la Raspberry Pi [Elaboración Propia].

Una vez implementada la placa se procede a colocar los elementos, en la figura 2.9 se observa el resultado final.



Figura 2.9. PCB módulo receptor para Raspberry Pi [Elaboración Propia].

Para alojar los elementos electrónicos se requiere de una carcasa, en este caso se empleó una caja de paso plástica y para la conexión de los cables de alimentación y ethernet, así como la colocación de la antena se emplea prensa estopa para evitar que el agua ingrese a la caja. El resultado se muestra en la figura 2.10.

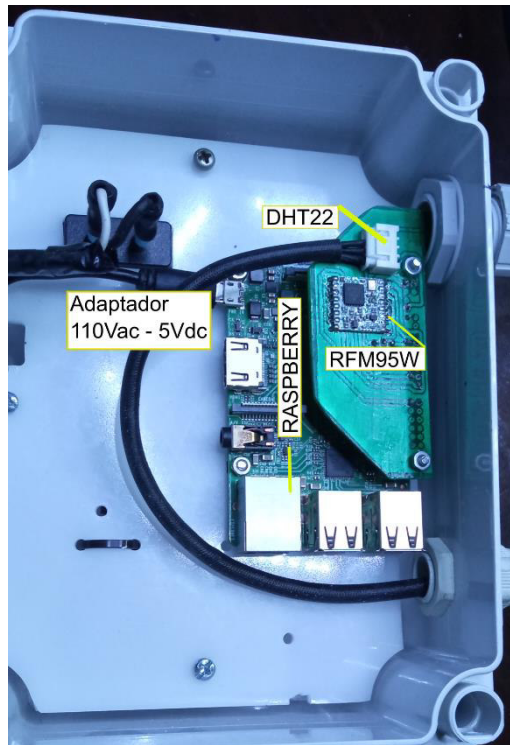


Figura 2.10. Módulo receptor LoRa [Elaboración Propia].

Parte del hardware es el sensor de temperatura y humedad relativa ambiente, el sensor comercial no está diseñado para ubicarse en exteriores, sin embargo, necesita estar en contacto con el aire del entorno que se desea medir, debido a esto se requiere de una carcasa que permita que el sensor permanezca en contacto con el ambiente exterior, pero esté protegido en caso de lluvia. En la figura 2.11 se observa una carcasa que puede ser impresa en 3D de manera que la lluvia no pueda causar daño permanente al sensor, diseñada por [69] y de libre acceso mediante la plataforma Thingiverse.

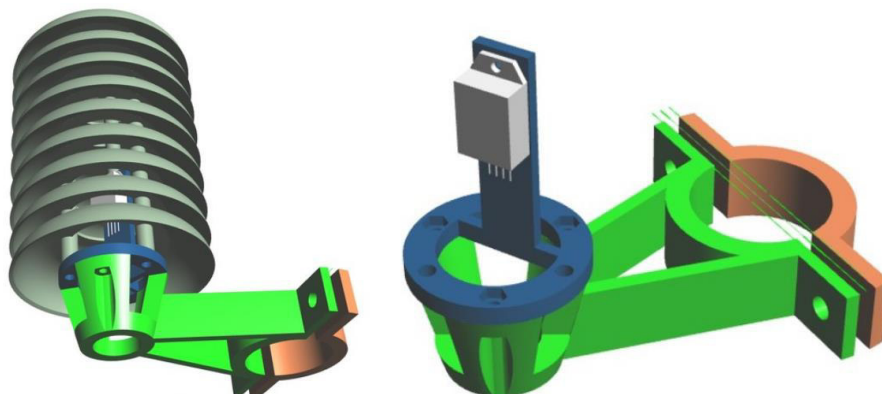


Figura 2.11. Carcasa DHT22 para sujetarse en tubo [69].

Finalmente se dispone de un router comercial el cual será el punto central de la red local, permitiendo que los dispositivos conectados a la red local mediante ethernet o wifi accedan a la interfaz de usuario desde el web browser Mozilla Firefox. El router seleccionado para este proyecto técnico es el TP-LINK TL-WR840N [70], de esta manera se forma una red local de topología estrella la cual se muestra en la figura 2.12.



Figura 2.12. Módulo receptor LoRa [70].

El resultado de la construcción del Hardware se muestra en la figura 2.13, en ella se puede observar tanto el receptor como el sensor DHT22 colocados sobre un tubo galvanizado.

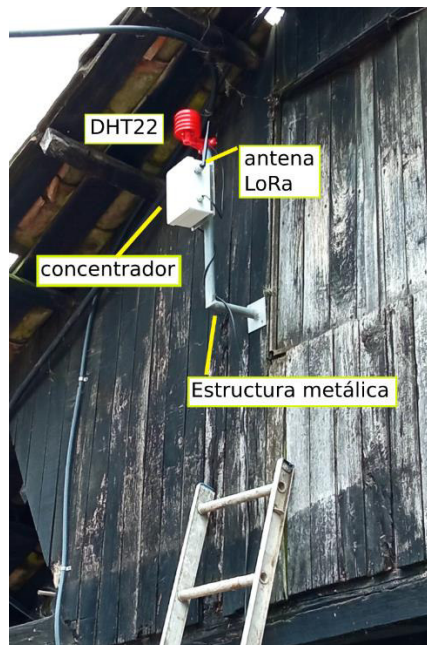


Figura 2.13. Implementación del receptor Hacienda Tambillo [Elaboración Propia].

2.2. DESARROLLO DE SOFTWARE

Una vez diseñados e implementados tanto el nodo final como el receptor se requiere programar estos dos dispositivos con las instrucciones que se encargan de ejecutar para el monitoreo de temperatura y geo posición de la res de ganado sobre la cual se realizarán las pruebas. Es así como en esta sección se abordará la programación del nodo final y del dispositivo receptor.

2.2.1 Nodo final

El nodo final es el dispositivo móvil que se encarga de obtener y enviar los datos del GPS, sensor de temperatura infrarrojo, estado de carga de la batería y controlar el encendido de los sensores mencionados anteriormente. En la figura 2.14. se observa el diagrama de flujo del programa del nodo final.

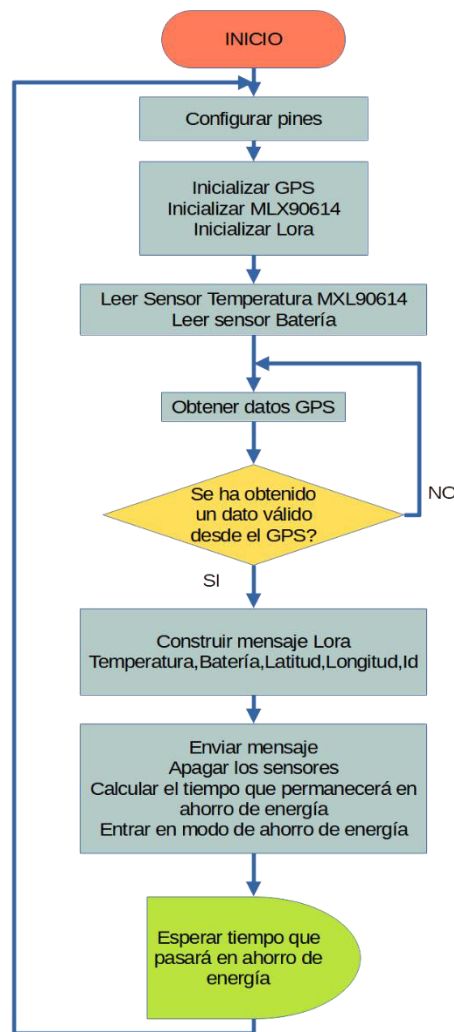


Figura 2.14. Diagrama de flujo del programa principal Nodo Final [Elaboración Propia].

La principal condición que debe cumplir el dispositivo es obtener los datos de los sensores, los fabricantes del GPS Neo-6m y del sensor de temperatura MLX90614 tienen librerías desarrolladas entorno al microcontrolador seleccionado para ser utilizadas en Arduino IDE, y son las siguientes:

- *TinyGPS.h*: es una librería de código abierto desarrollada por Mikal Hart [71] que permite convertir los datos en formato NMEA (National Marine Electronics Association), un formato creado incluso antes de inventarse el GPS, a un formato de datos más simple de entender como latitud, longitud y tiempo. El principal mensaje NMEA es el GPSSGA, que tiene la estructura mostrada en la tabla 2.3.

Tabla 2.3. Estructura del mensaje GPSSGA [72]

\$--GGA,hhmmss.ss,IIII.II,a,yyyyy.yy,a,x,xx,x.x,x.x,M,x.x,M,x.x,xxxx	
Elemento	Definición
hhmmss.ss	Tiempo (hora, minuto, segundo)
IIII.II	Latitud
a	Norte o Sur (N, S)
yyyyy.yy	Longitud
a	Este u Oeste (E, W)
x	Indicador satélite GPS (0 = no fijado, 1 = fijo)
xx	Numero de satélites en uso
x.x	Precisión horizontal
x.x	Altura
M	Unidades (metros)
x.x	Separación geoidal: Diferencia entre la altura elipsoidal y la altura ortométrica en un punto dado de la superficie de la tierra.
M	Unidades (metros)
x.x	Segundos pasados después del último dato adquirido
xxxx	ID de la estación de referencia

- *LoRa.h*: esta librería desarrollada por Sandeep Mistry permite leer y escribir en los registros del chip LoRa, de esta manera se pueden establecer los parámetros principales de la comunicación y enviar los datos mediante esta capa física [73].
- *SPI.h*: es una biblioteca a través de la cual se hace uso de los pines SPI en el microcontrolador, es una biblioteca nativa preinstalada de Arduino compatible con ESP32.

- *Wire.h*: es una librería que permite hacer uso de los pines I2C disponibles en el microcontrolador, y a igual que la biblioteca anterior esta preinstalada en el IDE de Arduino.
- *Adafruit_MLX90614.h*: esta librería en conjunto con la librería *Wire.h* permite leer los registros del sensor de temperatura MLX90614. La librería esta desarrollada por Adafruit y es de código abierto [74].

Otra de las condiciones es el ahorro de energía, el GPS neo-6m tiene un tiempo de arranque al encenderse de máximo 27 segundos de acuerdo con el fabricante [38], este tiempo puede aumentar dependiendo de si el GPS está en campo abierto o si hay suficientes satélites en órbita. A partir de que se establece conexión con los satélites el GPS actualiza los datos cada segundo, con este precedente el dispositivo necesita tiempo para establecer la conexión con los satélites y en obtener un dato GPS válido. Una vez que se obtiene un dato válido, simplemente los datos se envían mediante LoRa y después de realizada esta tarea se calculará el tiempo que el dispositivo permanecerá en modo de ahorro de energía, se suspende el paso de energía hacia los sensores y el dispositivo suspende sus funciones de procesamiento.

Es importante el modo de ahorro de energía ya que el envío de datos desde el nodo final hacia el receptor se hará cada 10 minutos, y debido a que el tiempo estimado de espera para conectar con los satélites es de 27 segundos no es necesario que el nodo final se mantenga operativo durante todo el tiempo, de esta manera se puede ahorrar energía desactivando los sensores y las funciones del microcontrolador. El microcontrolador en el modo ahorro de energía de menor consumo solamente tiene activo un procesador de ultra bajo consumo y el timer principal para controlar el tiempo que este se mantiene en este modo [75], como se muestra en la figura 2.15 el consumo del ESP-32 se reduce de 50 mA hasta 10 uA.

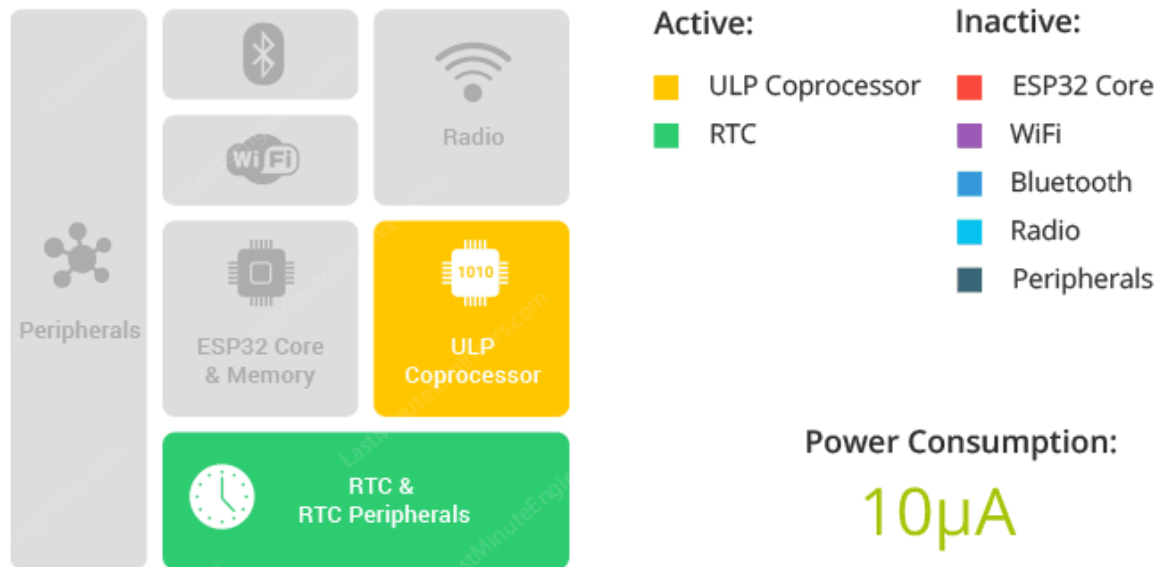


Figura 2.15. Elementos activos durante el modo de ahorro de energía [75].

Por último, el circuito que emplea la tarjeta Firebeetle para determinar el nivel de carga de la batería es mediante un divisor de voltaje formado por dos resistencias de $1\text{M}\Omega$ cada una, el dato se obtiene desde el pin analógico 34, y como se observa en la fase teórica la batería VTC6 tiene un voltaje de carga completa de $4.2 \pm 0.5\text{Vdc}$ mientras que la tarjeta Firebeetle requiere de un mínimo de 3.3Vdc para su operación al igual que el GPS neo-6m, el sensor MLX90614 y el chip RFM95W, por tanto se considerará 3.3Vdc como el 0% de carga y 4.15 como el 100%, se realizaron mediciones del voltaje de batería y la lectura analógica del pin 34 y los resultados se observan en la figura 2.16.

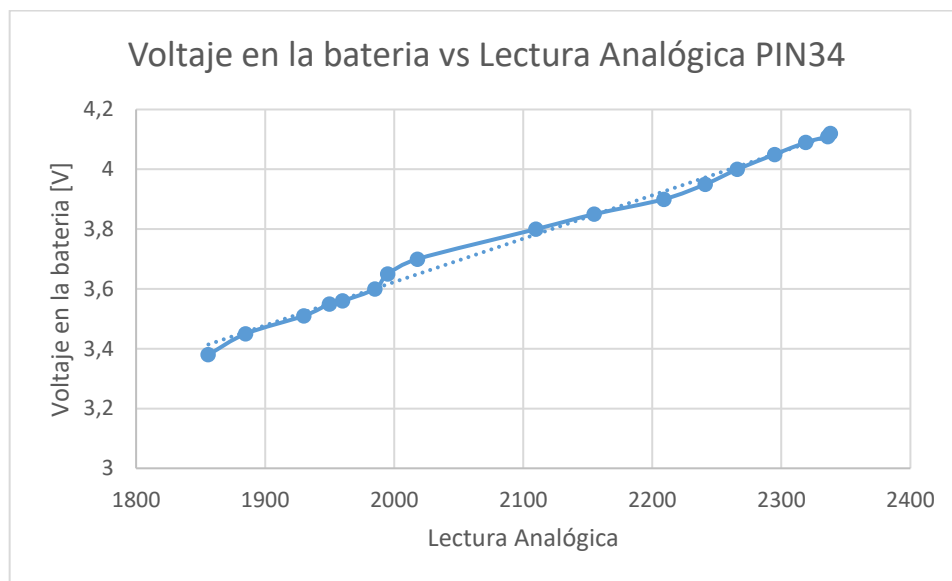


Figura 2.16. Relación entre el voltaje en la batería y la lectura del pin analógico de 16bits [Elaboración Propia].

En base a lo obtenido de forma experimental se puede obtener la siguiente ecuación de acondicionamiento del sensor que se implementará dentro del programa principal del dispositivo remoto o nodo final.

$$\text{porcentajeCarga} = 0.1768 * \text{LecturaAdc} - 314.31 \quad (2.1)$$

2.2.2 Dispositivo Receptor

En este punto se describe la teoría detrás del funcionamiento del receptor y como son diseñados los programas que se ejecutan en este dispositivo. De igual manera se definen cuales variables se almacenarán en la base de datos, y aquellas que se presentarán en la interfaz de usuario.

Como se observó en el apartado de hardware el componente principal es una Raspberry Pi en la cual están conectados tanto el sensor de temperatura y humedad relativa como el chip receptor LoRa. En la figura 2.17 se observa la arquitectura de software con los programas a ejecutarse en la Raspberry Pi.

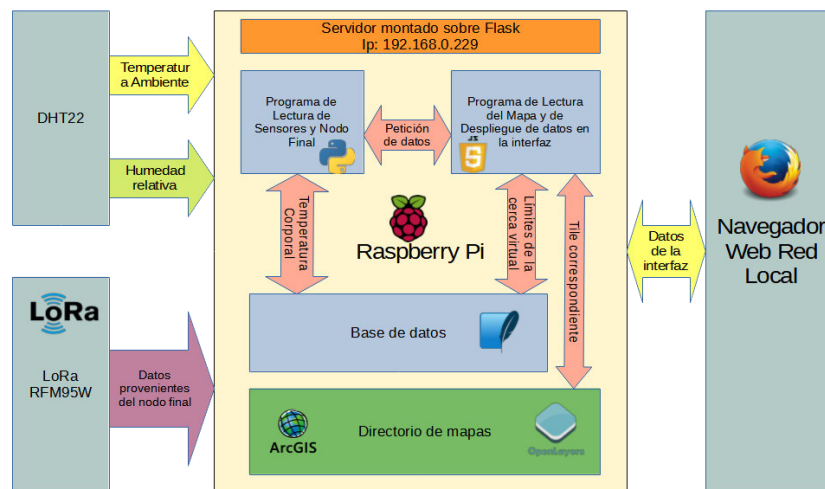


Figura 2.17. Arquitectura de software en el receptor [Elaboración Propia].

Los archivos que componen la arquitectura de software almacenados en la memoria de la Raspberry se reparten en diferentes directorios, a continuación, en la figura 2.18 se muestra la organización de estos archivos los cuales serán referenciados en los siguientes apartados. Todos los archivos se encuentran en el siguiente directorio.

/home/pi/

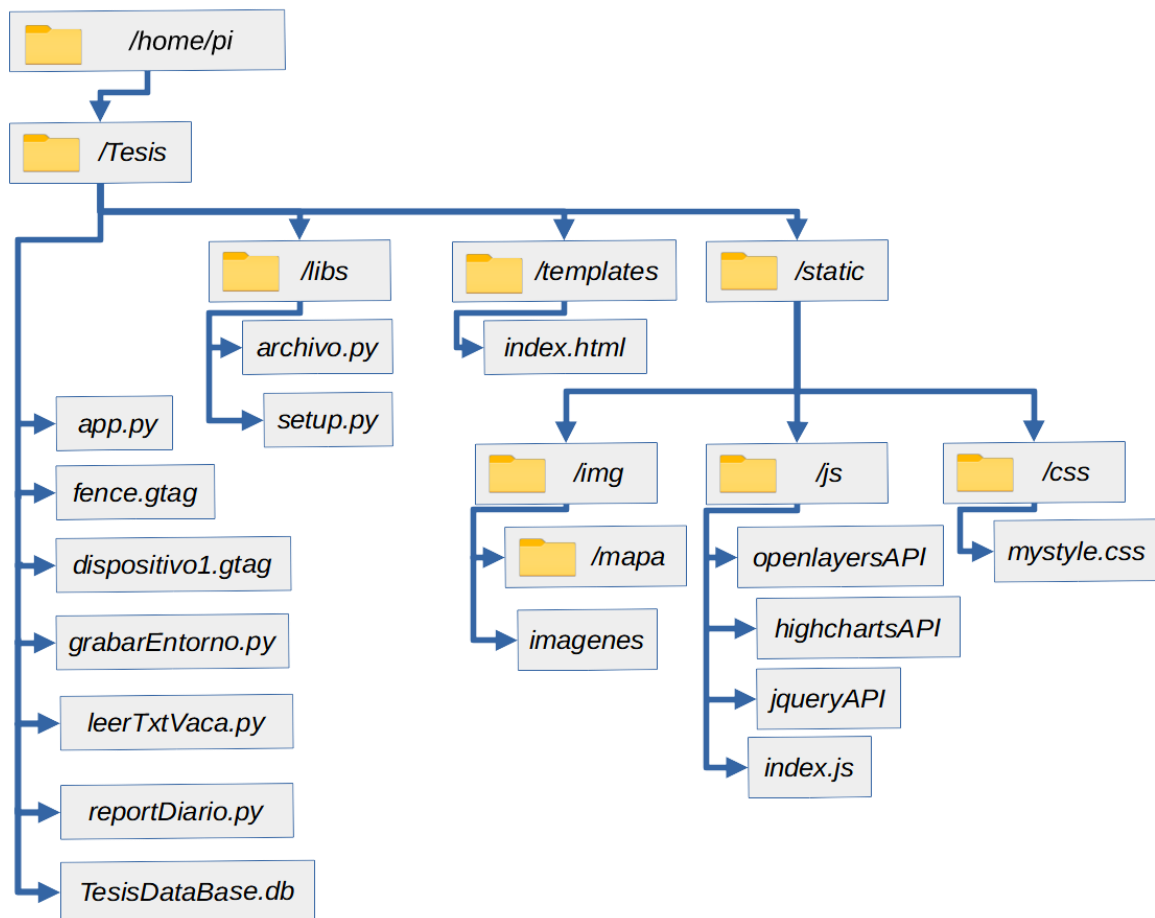


Figura 2.18. Organización archivos en la memoria flash de la Raspberry Pi [Elaboración Propia].

El archivo *app.py* es el programa que se ejecuta desde el arranque del receptor, mientras que el archivo *grabarEntorno.py* se ejecuta cada hora para almacenar los datos de temperatura y humedad relativa ambiente. Por otra parte, los archivos *fence.gtag* y *dispositivo1.gtag* son archivos de texto que contienen los datos de la cerca virtual del dispositivo y los datos recibidos desde el dispositivo remoto respectivamente.

En el directorio */libs* se encuentran dos librerías desarrolladas específicamente para este proyecto, mediante las clases dentro de esta librería se permite acceso a comandos del sistema operativo, y tienen instrucciones de acceso a la base de datos.

Para la utilización de la librería Flask, los archivos involucrados en el desarrollo de la interfaz se dividen en: */templates* y */static*, en el primer directorio se almacena el archivo HTML de la interfaz, mientras que en el segundo se encuentran los archivos CSS, JS con sus librerías, y los archivos multimedia de la interfaz como imágenes y mapas. En los siguientes apartados se detallan cada uno de los archivos mencionados.

2.2.2.1 Configuración inicial de la Raspberry[76]

Debido a que la interfaz de usuario será diseñada como una página web a la cual se puede acceder desde los dispositivos dentro de una red local, se debe permitir el acceso de la raspberry a la red en caso de usarse wifi se debe agregar las credenciales de la red (nombre y contraseña) en el archivo `/etc/wpa_supplicant/wpa_supplicant.conf`, y mediante el editor de texto en consola `nano`.

```
network = {  
    ssid = "nombredelared"  
    psk = "clave"  
}
```

En caso de un reinicio del sistema, el sistema operativo por defecto asigna una dirección IP dentro de la red mediante lo que se conoce como DHCP (Dinamic Host Configuration Protocol), si se mantiene esta configuración durante un reinicio o desconexión del dispositivo este puede cambiar de dirección dentro de la red, lo cual dificulta el acceso desde el navegador, por lo tanto, se establecerá una dirección fija introduciendo el siguiente comando en el archivo `/etc/dhcpd.conf`, si la conexión es mediante wifi la interface es `wlan0`, si por el contrario se emplea ethernet se utiliza `eth0`.

```
interface wlan0  
static ip_address = 192.168.0.229/24  
static routers = 192.168.0.1  
static domain_name_server = 192.168.0.1 8.8.8.8
```

Finalmente es necesario para poder hacer uso de los puertos SPI y One Wire de la Raspberry y que el programa principal pueda leer los registros del chip LoRa y el DHT22 respectivamente se deben habilitar estas interfaces. también es importante habilitar la interfaz SSH para tener acceso a la consola de la raspberry desde un PC, conectado a la red local. Mediante el comando `sudo raspi-config`, se abre en el terminal el menú de configuración que se muestra en la figura 2.19, y al seleccionar la opción *Interfacing Options* se desplegara un menú con los periféricos disponibles en la Raspberry, donde se habilitarán los ya mencionados.

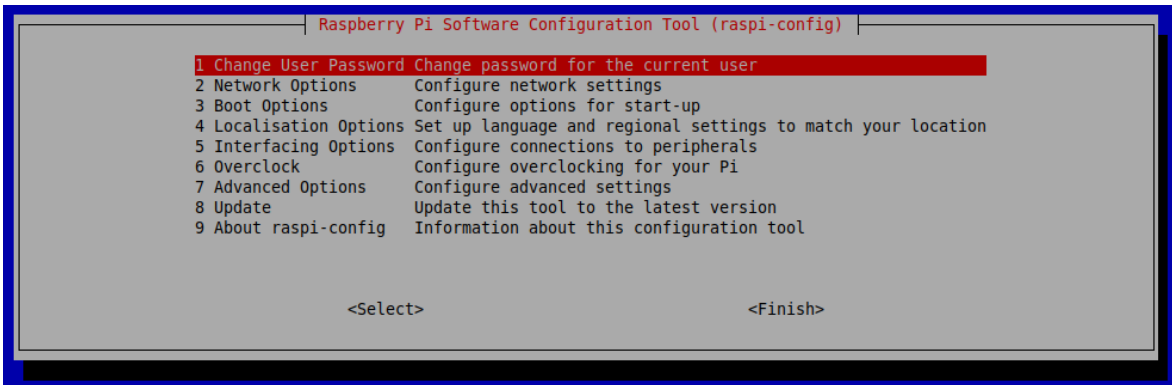


Figura 2.19. Menú de configuración del sistema [76].

2.2.2.2 Programa principal

El programa principal se desarrolla en Python y se encarga de gestionar toda la información que proviene desde los sensores conectados a la raspberry, así como la información del nodo final colocado sobre una res de ganado que ingresa a través del módulo RFM95W. Este programa obtiene la información y se encarga de clasificarla e ingresarla a la base de datos, también de atender y realizar las peticiones de la interfaz de usuario, cuando estas se generan en la interfaz. En la figura 2.20 se observa el diagrama de flujo del programa principal.

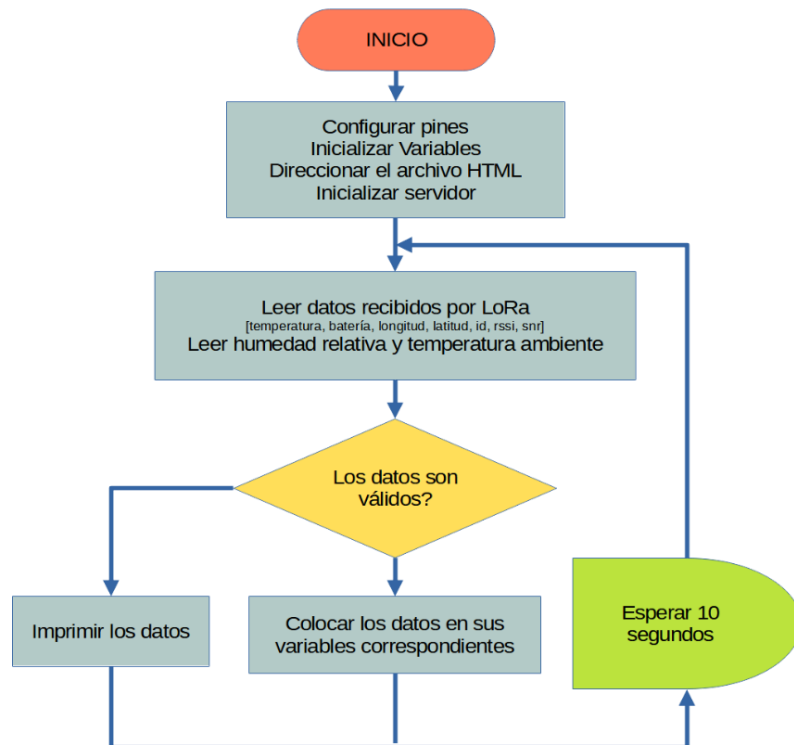


Figura 2.20. Diagrama de flujo del programa principal [Elaboración Propia].

Las librerías involucradas en el desarrollo de la interfaz son las siguientes:

- *OS*: Es un módulo que permite usar funcionalidades que dependen directamente del sistema operativo, permite obtener acceso a los directorios en la memoria del sistema al igual que introducir instrucciones de igual manera que se pueden introducir en la línea de comandos del sistema operativo, de esta manera desde la interfaz de usuario se puede ejecutar una instrucción de reinicio, o de apagado de la Raspberry [77].
- *Adafruit_rfm9x*: esta librería de python es una similitud de la librería *LoRa.h* empleada en Arduino, permite leer y escribir los registros en el chip RFM95W conectado al puerto SPI de la Raspberry, la conexión con el puerto se realiza mediante *circuitpython* [78].
- *Adafruit_DHT22*: esta librería de código abierto desarrollada por Adafruit permite obtener datos desde el sensor DHT22, mediante *circuitpython* [79].
- *Flask*: Esta librería permite acceder a las funcionalidades del WSGI Flask, esta librería se convierte en el eje central de la interfaz, permite establecer el nexo entre los principales programas de la interfaz, el archivo *app.py* y los archivos de desarrollo web encabezados por el archivo *index.html*. De esta manera las solicitudes que se producen desde la interfaz son atendidas y ejecutadas por el programa *python* [80].
- *SQLite3*: Es un módulo escrito por Gerhard Häring [81], el cual permite crear una conexión con una base de datos SQLite, el módulo permite ejecutar comandos en lenguaje SQL [66], esto permitirá tener acceso a las tablas, obtener y leer los datos que en ellas se encuentran.
- *Time*: Es una librería que permite manipular el tiempo, para generar esperas en la ejecución, al igual que obtener información del tiempo del sistema (hora y fecha) [82].
- *Numpy*: es una librería para realizar cálculos computacionales en Python, en especial operaciones con vectores y matrices. Desde la base de datos se obtiene la información como una matriz de N filas y una columna, sin embargo, para construir la gráfica se necesita un vector, por lo tanto se emplea la librería *numpy* para convertir la matriz en un vector [83].

- *Multiprocessing*: es un módulo similar a *threading*, la diferencia es que en *threading* se puede manejar solo un hilo a la vez [84], mientras que *multiprocessing* permite crear subprocesos que se ejecutan paralelamente, el uso de *multiprocessing* en este caso particular es para ejecutar la lectura del chip LoRa y la lectura de temperatura y humedad relativa en procesos diferentes [85].
- *Json*: es una librería que permite convertir los datos de Python en una sintaxis literal de un objeto JavaScript y de esta manera poder utilizarlo en un programa desarrollado en este lenguaje, por esta razón la librería se emplea para enviar información que es solicitada desde el programa de la interfaz al programa principal [86].

El programa principal se ejecuta apenas se inicia el sistema operativo en la Raspberry, mediante la instrucción programada en el archivo `/etc/rc.local` nativo del sistema operativo Raspbian, dentro del archivo se coloca la siguiente instrucción antes de la sentencia `exit 0`.

```
sudo python3 /home/pi/tesis/app.py &
```

2.2.2.3 Programa interfaz

El programa de la interfaz consiste principalmente en la adquisición de datos a través de solicitudes al programa principal mediante la librería *jQuery.js* de la cual se emplea la función *ajax* [87] mediante la cual se puede crear una solicitud HTTP al servidor que en este caso es la Raspberry. En la figura 2.21 se muestra el programa principal de la interfaz.

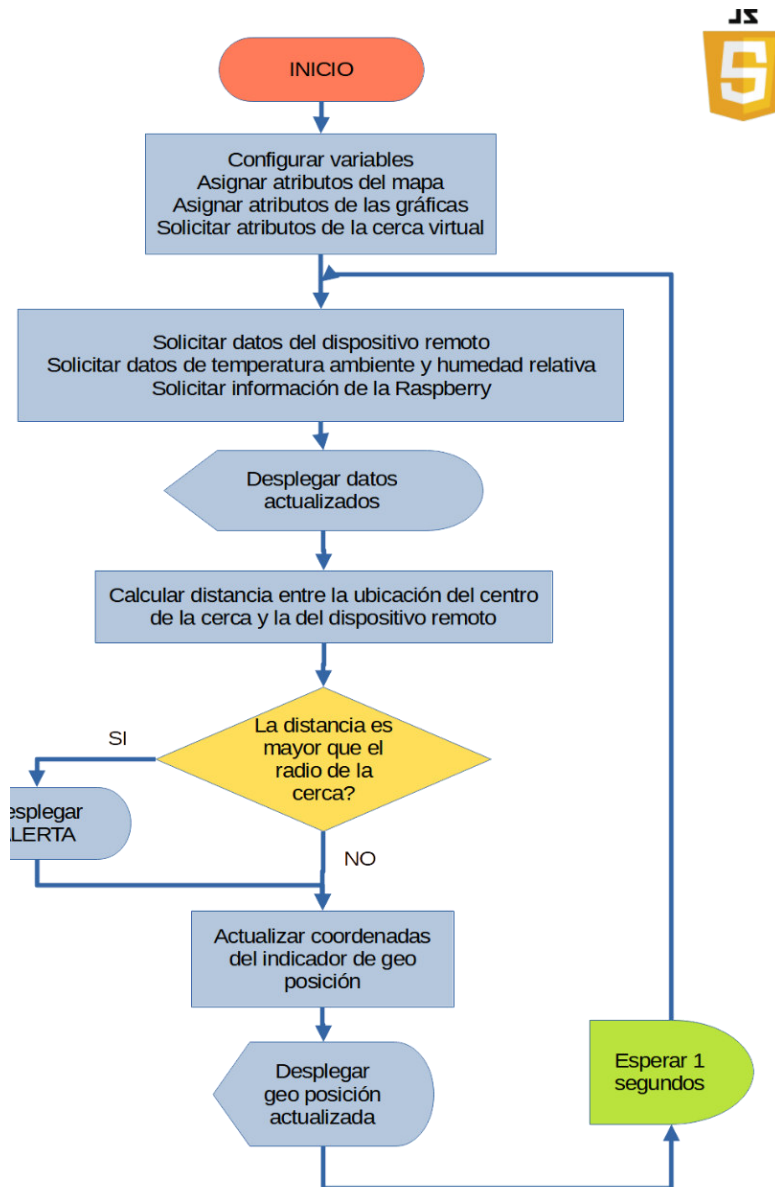


Figura 2.21. Diagrama de flujo del programa principal [Elaboración Propia].

En primer lugar, es importante definir los elementos que componen la interfaz. Para cumplir con los objetivos de este proyecto de titulación se mostrará la geo posición como un marcador en el mapa de la zona de crianza, así como la temperatura corporal, estos datos provienen desde el dispositivo remoto y se actualizan cada diez minutos; adicional a estos datos desde el nodo final se obtiene el estado del dispositivo mediante el estado de carga de la batería, rssi y snr para evaluar la calidad de la conexión LoRa. Además, existen otros datos que provienen del propio receptor, como datos de fecha y hora, así como humedad relativa y temperatura ambiente obtenidos desde el sensor DHT22.

Una vez definidos los componentes de la interfaz a continuación se desglosan el desarrollo de sus partes.

2.2.2.3.1 Despliegue del mapa y de geo posición

Como se determinó en la parte teórica la librería Open Layers permite desplegar mapas en una interfaz desarrollada en Java Script. Este apartado abarca el diseño desde la obtención y despliegue del mapa, sus características, la creación de la cerca virtual que limita el espacio de desplazamiento para el marcador. Para poder emplear Open Layers en una aplicación web, se debe referenciar la librería desde el archivo HTML, si la aplicación se ejecuta en un dispositivo conectado a internet se puede emplear la URL [88], caso contrario también es posible descargar desde el repositorio de la API la librería. Para referenciar en el archivo HTML se emplea la siguiente instrucción antes de cerrar la etiqueta *body* del archivo.

```
<script src="../../static/js/ol/build/ol.js"></script>
```

Open Layers al ser una API desarrollada principalmente para ejecutarse en aplicaciones web permite la obtención del mapa a partir de un repositorio en línea por esta razón como primera medida se requiere de conexión a internet. Ahora bien, el prototipo está pensado para aplicaciones en ambientes rurales, en los cuales es muy probable no exista una conexión a internet, es por esto por lo que se analizó la posibilidad de agregar un mapa previamente descargado y que se encuentre alojado en la memoria del dispositivo receptor.

Un mapa desplegado en Open Layers está compuesto de una matriz de tiles que son simplemente imágenes separadas que en conjunto forman el mapa. Existen varios repositorios de código abierto que permiten el acceso a estos tiles, el acceso simplemente se hace a través de una URL, pero en el caso de requerir descargar el tile para el uso sin conexión a internet se puede realizar mediante una herramienta de software llamada QGIS que es un sistema de información geográfica de código abierto [89], utilizado para la visualización, captura, análisis de datos geográficos. En la figura 2.22 se observa la interfaz de usuario de QGIS.

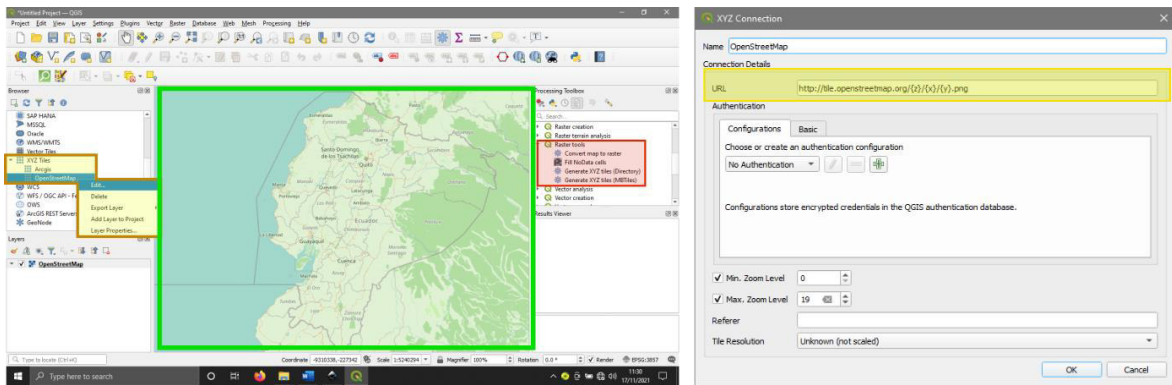


Figura 2.22. Interfaz de usuario y sus componentes QGIS [Elaboración Propia].

Las herramientas de QGIS utilizadas en este proyecto son tres, para empezar, se encuentra el buscador en la sección XYZ Tiles en la cual se permite agregar mapas de distintos proveedores a través de un URL, para este proyecto se emplea al mapa provisto por ArcGIS que presenta una vista satelital del terreno. En la sección de la mitad en verde se puede visualizar el mapa y enfocarlo en el área de interés. Finalmente se encuentran las herramientas de procesamiento de las cuales se emplea Generate XYZ Tools (Directory) en Raster Tools el cual se muestra en la figura 2.23, esta herramienta permite obtener los tiles en formato png organizados en una serie de carpetas de acuerdo con el nivel de zoom preseleccionado, mientras mayor sea el nivel de zoom más imágenes se requieren para armar el mapa y por lo tanto la carpeta ocuparía más espacio de almacenamiento.

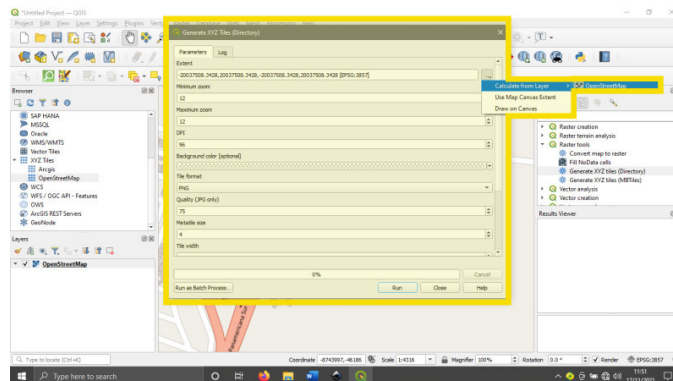


Figura 2.23. Herramienta de generación de Tiles [Elaboración Propia].

Una vez descargado el mapa se hace referencia al directorio en el cual se encuentra almacenado dentro de la Raspberry en el directorio que se observa en la figura 2.18, para acceder al mapa dentro del archivo JS y mostrarlo se emplean las funciones de Open Layers *Map*, *layer.Tile*, *source.XYZ* [90] mediante la siguiente instrucción.

```

var map= new ol.Map({layers: [new ol.layer.Tile({source: new ol.source.XYZ({
url: '../static/img/Mapa/{z}/{x}/{y}.png',
})  }) ],
});

```

2.2.2.3.2 Cerca virtual

Uno de los elementos que forman parte de la interfaz es la delimitación del área dentro de la cual el dispositivo debe permanecer, en caso de que el dispositivo salga de esa área delimitada en la interfaz se desplegará una alerta para informar que el dispositivo abandono la misma.

Dentro de Open Layers existe el elemento *geom* el cual posee diferentes geometrías con sus distintas herramientas [90]. La geometría escogida para dibujar la cerca virtual es un círculo debido a que tiene una menor cantidad de parámetros a configurar en el caso del operador, los parámetros que se requieren para crear un círculo son: latitud, longitud del centro y radio en metros, esto se realiza mediante la función que se muestra a continuación.

```

var circle = new ol.geom.Circle(ol.proj.fromLonLat([longitud,latitud]),radio)

```

Una vez definida la cerca dentro de la interfaz, periódicamente se compara el radio de la cerca con la distancia entre el punto recibido desde el dispositivo remoto y la ubicación del centro de la cerca. Para obtener esta distancia se aplica una fórmula matemática denominada *Haversine*, una ecuación muy utilizada en navegación la cual da como resultado la distancia entre dos puntos ubicados sobre una esfera a partir de los ángulos latitud y longitud de dos puntos y 6378 [m] que es el radio de la tierra en este caso particular [91].

$$D = 2 \cdot R \cdot 1000 \cdot \sin^{-1} \left\{ \sqrt{\sin^2 \left(\frac{lat_1 - lat_2}{2} \right) + \cos(lat_1) \cdot \cos(lat_2) \cdot \sin^2 \left(\frac{long_1 - long_2}{2} \right)} \right\} \quad (2.2)$$

2.2.2.3.3 Configuración de parámetros de la cerca

Una de las funcionalidades de la interfaz que tiene relación con la cerca virtual es la configuración de los parámetros mencionados en el apartado anterior, los datos ingresados en la configuración de la cerca son modificados y almacenados en un archivo de texto, en la figura 2.24 se observa el diagrama de flujo que corresponde a la solicitud generada desde la interfaz.

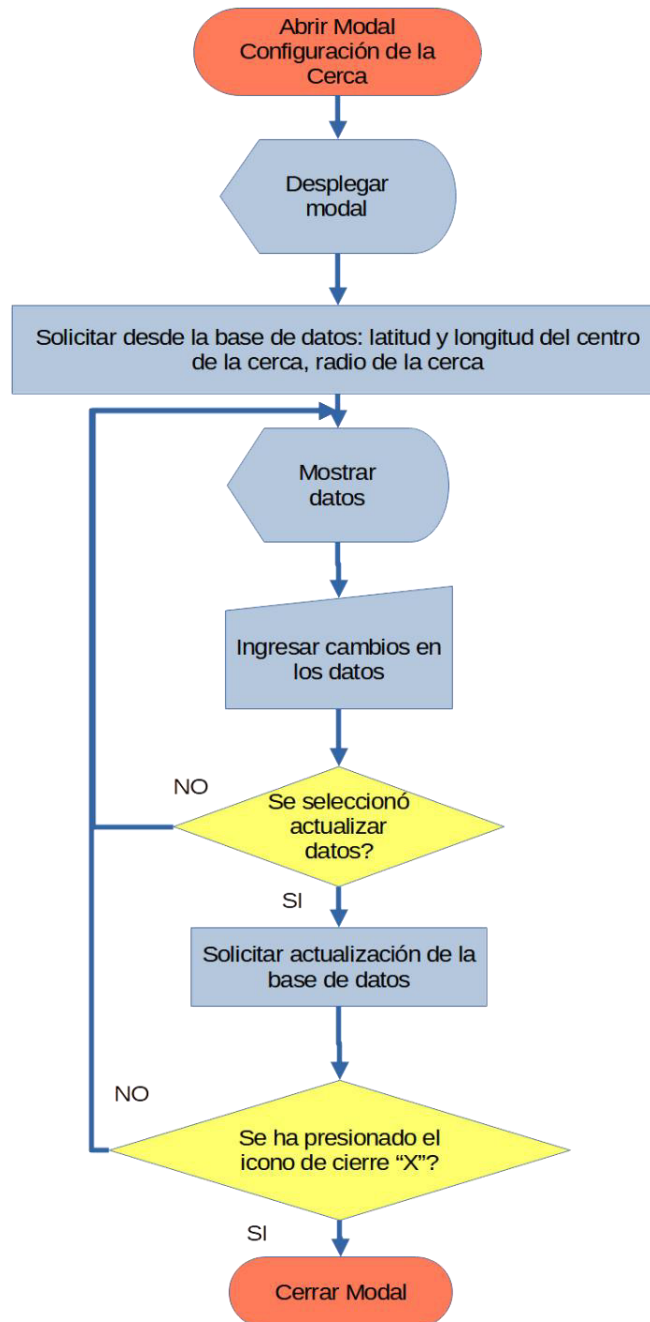


Figura 2.24. Flujograma del modal de actualización de la cerca [Elaboración Propia].

2.2.2.3.4 Apagado y reinicio del sistema

En caso de que se requiera el reinicio o el apagado del receptor existe la posibilidad de solicitar estos comandos al sistema mediante el programa principal, en especial debido a que en caso de requerir una desconexión del equipo receptor es recomendable apagar el equipo correctamente para evitar daños en la tarjeta microSD que contiene el sistema operativo. En la figura 2.25 se observa el diagrama de flujo que genera la solicitud de apagado o reinicio del sistema.

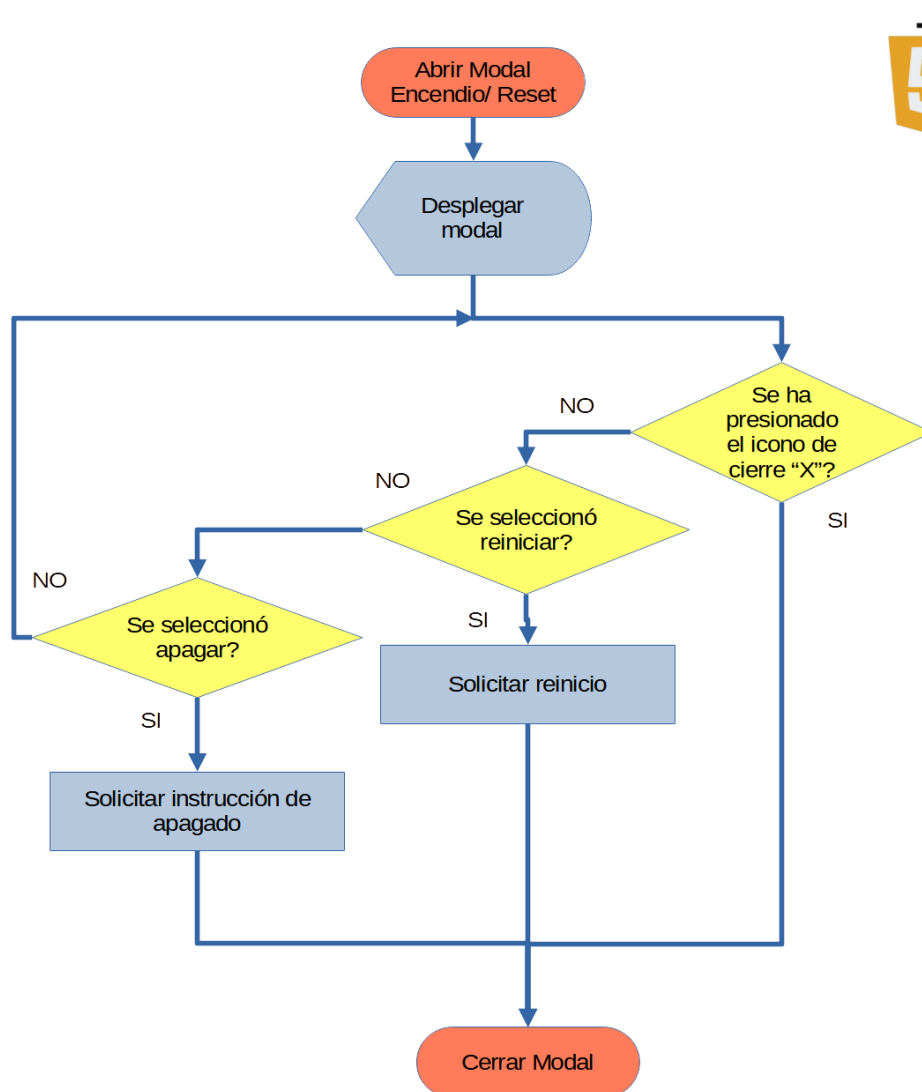


Figura 2.25. Flujograma del modal de opciones de equipo [Elaboración Propia].

2.2.2.3.5 Gráficas de temperatura y humedad relativa

Otro de los componentes de la interfaz es una herramienta gráfica que permita conocer los datos del entorno los cuales son almacenados en la base de datos mediante la

ejecución del archivo python *grabarEntorno.py* el archivo se ejecuta cada hora debido a que está programada mediante el comando *crontab -e* un comando de Unix que se usa para programar tareas [92], el comando a ejecutarse es el siguiente.

```
0 * * * * sudo python3 /home/pi/tesis/grabarEntorno.py
```

El comando se ejecutará al minuto 0 en cualquier hora, día, mes y día de la semana. Los datos de fecha, hora, temperatura y humedad relativa se almacenan en orden dentro de la base de datos *tesisDataBase* en la tabla *entorno* cada hora se actualiza la tabla de tal manera que siempre existan los datos de las últimas 24 horas, como se observa en la figura 2.26.

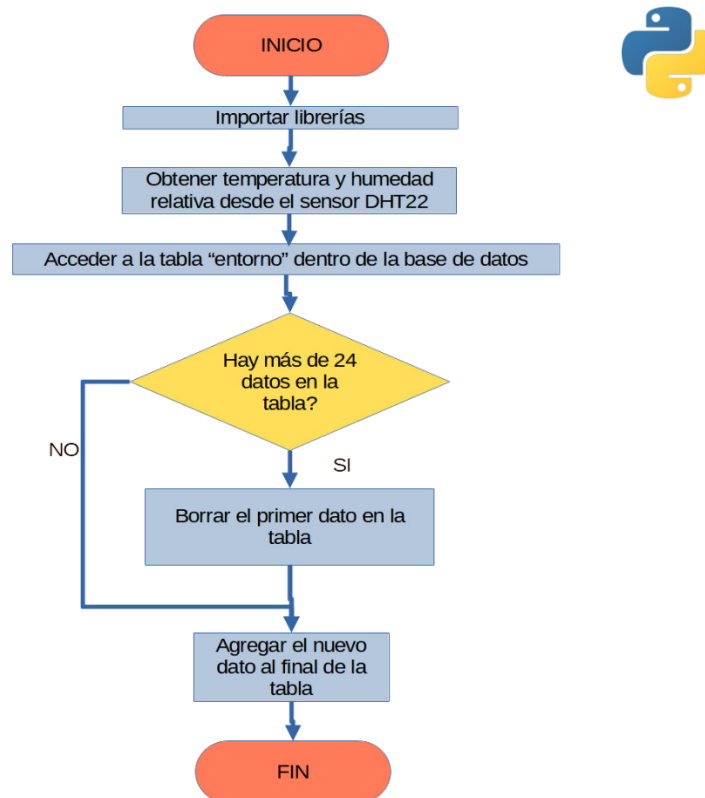


Figura 2.26. Flujograma del programa en Python para actualización de temperatura y humedad relativa ambiente en la base de datos. [Elaboración Propia].

Los datos almacenados en la tabla *entorno* son accedidos desde el archivo JavaScript mediante una solicitud al programa principal *app.py* los datos son entregados como tres vectores los cuales se despliegan en la gráfica mediante la biblioteca *Highcharts.js* [93], esta librería permite crear gráficas estáticas o dinámicas, en este caso se emplea para

crear una gráfica sencilla en dos dimensiones. En la figura 2.27 se observa el diagrama de flujo de la solicitud de datos desde la interfaz.

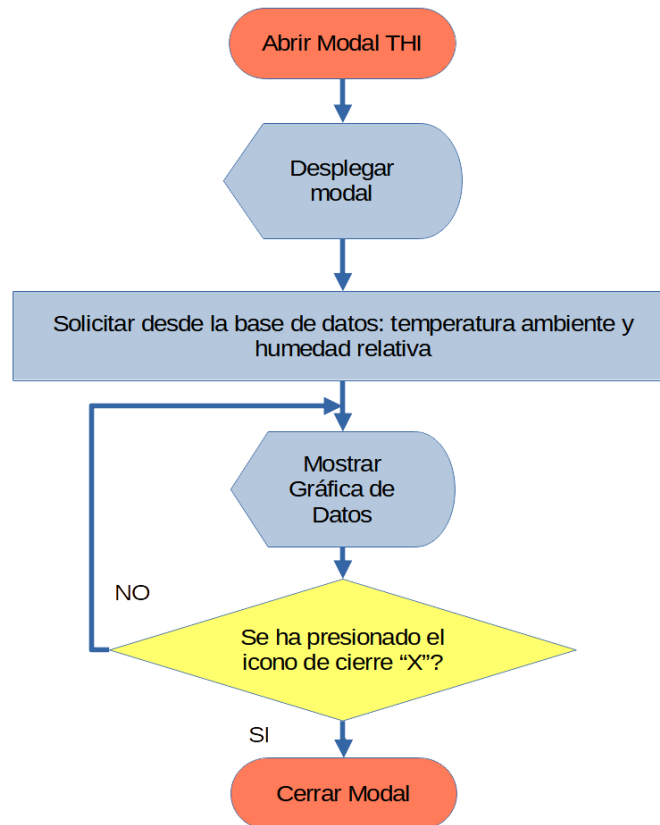


Figura 2.27. Diagrama de flujo para el despliegue de datos de temperatura y humedad relativa ambiente en la interfaz de usuario. [Elaboración Propia].

2.2.2.3.6 Gráficas de temperatura corporal

El valor numérico de temperatura corporal es presentado de forma continua, sin embargo, en la interfaz es posible hacer un seguimiento de la temperatura del ganado a lo largo de los días a través de una gráfica. Para obtener estos datos, mediante un sketch de Python se almacena en la base de datos en la tabla *reportTemp*, el dato de temperatura corporal con la misma periodicidad en la cual debería el dispositivo remoto enviar esta información. Al final del día a partir de los datos almacenados en *reportTemp*, se obtiene los valores: máximo, mínimo y promedio de ese día, los cuales se almacenan en la tabla *reportDiario* los cuales pueden ser visualizados en la interfaz en forma de un gráfico de columnas, finalmente los datos de *reportTemp* son eliminados.

Para almacenar los datos en la tabla *reportTemp* mediante el comando *crontab -e* se ejecuta el programa de Python *leerTxtVaca.py*. Este programa, como se observa en la figura 2.28, se encarga de leer el último dato disponible en el archivo de texto *dispositivo1.gtag* el cual es actualizado constantemente por el programa principal, una vez que obtiene el dato lo ingresa en la tabla correspondiente; el comando introducido es el siguiente:

```
* /10 * * * * sudo python3 /home/pi/tesis/leerTxtVaca.py
```

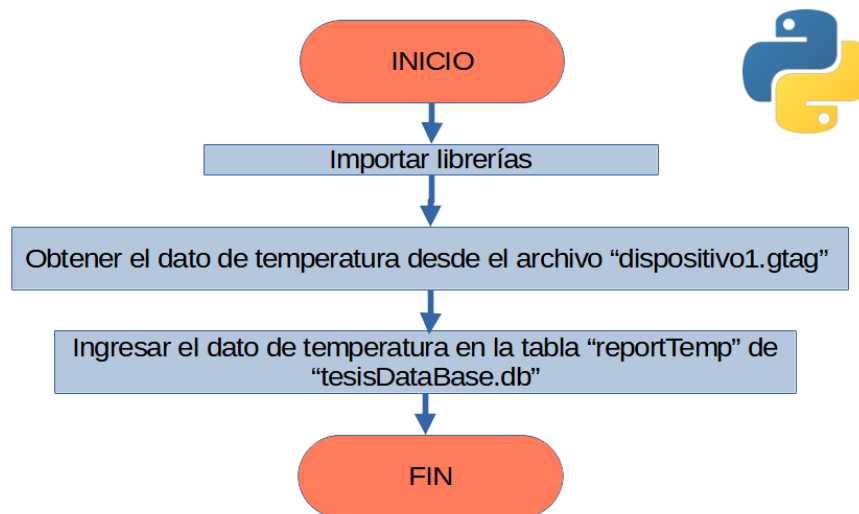


Figura 2.28. Diagrama de flujo almacenamiento de datos de temperatura corporal.
[Elaboración Propia].

Por otro lado, para almacenar los datos que se obtendrán desde la interfaz, se emplea el programa de Python *reportDiario.py*. Este programa, como se observa en la figura 2.29, a partir de los datos disponibles en la tabla *reportTemp*, obtiene los valores: máximo, mínimo y promedio, y los almacena en la tabla *reportDiario*, para finalmente borrar los datos originales dejando libre la primera tabla. El programa se ejecuta cada 24 horas, mediante el siguiente comando dentro de *crontab -e*.

```
0 0 * * * sudo python3 /home/pi/tesis/reportDiario.py
```

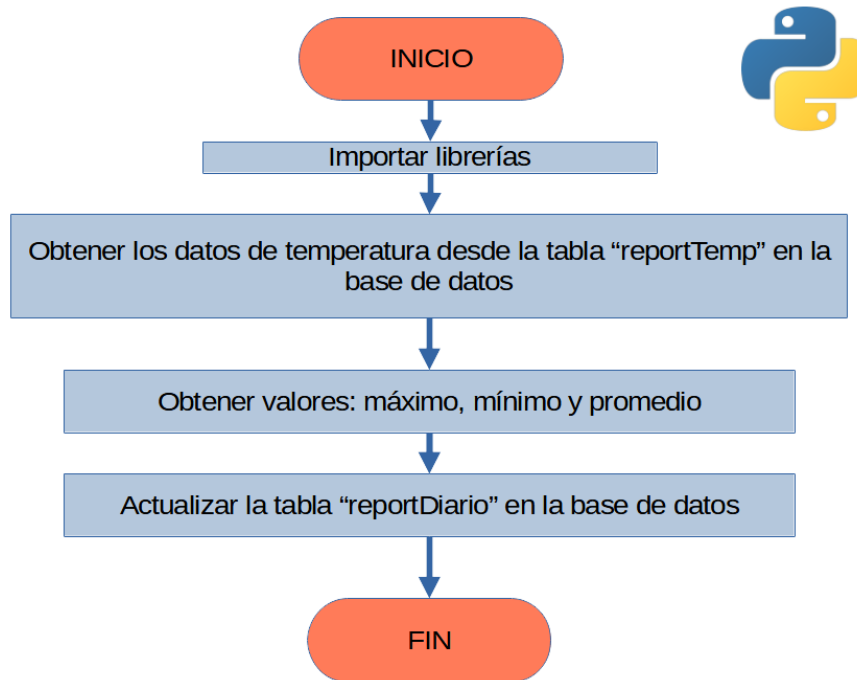


Figura 2.29. Diagrama de flujo actualización de datos de temperatura corporal para mostrar en la interfaz. [Elaboración Propia].

Para desplegar los datos se realiza una petición desde el archivo JavaScript y los datos se obtienen como cuatro vectores, que contienen la fecha, temperatura máxima, promedio y mínima respectivamente, para la construcción de las gráficas se emplea la librería *Highcharts.js* [93], de igual manera que para mostrar los datos de THI.

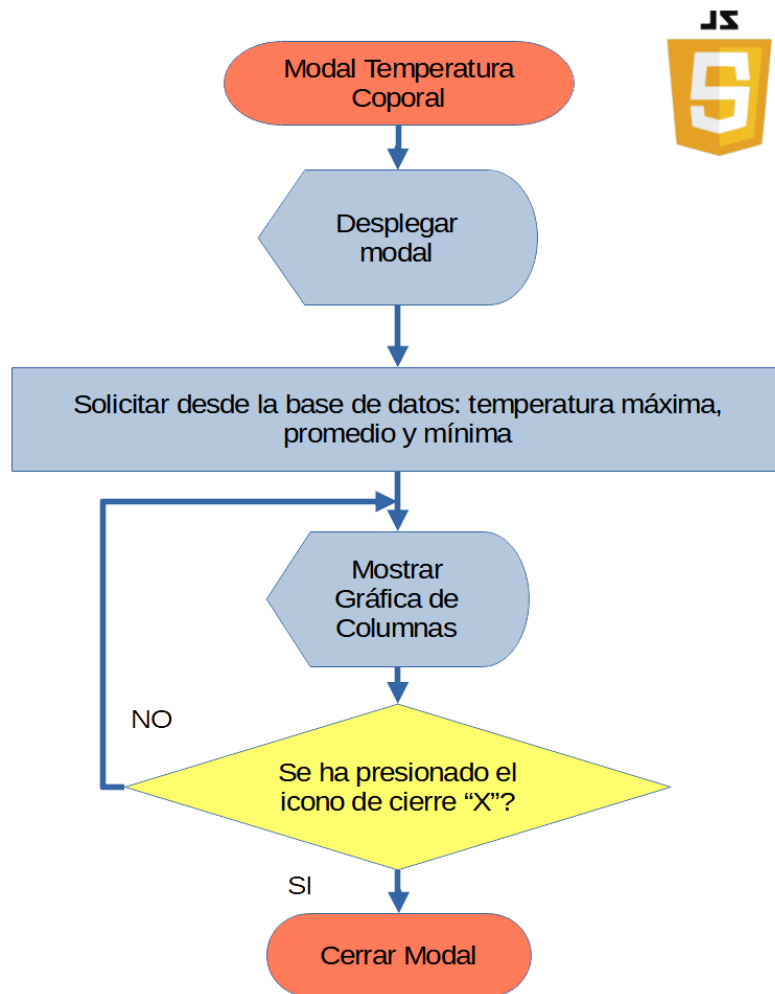


Figura 2.30. Diagrama de flujo interrupción para desplegar modal. [Elaboración Propia].

2.2.2.4 Implementación de la interfaz

El conjunto de programas descritos en esta sección permite desplegar la interfaz, a la cual se accede desde una PC mediante un navegador web, en este caso se emplea Mozilla Firefox, también se puede acceder desde el navegador de un teléfono celular o Tablet siempre que estén dentro de la red local gestionada por el router, sin embargo, la interfaz se desarrolló orientada a visualizarse en la pantalla de un computador.

Para ejecutarse se requiere que el dispositivo receptor esté conectado a la fuente de alimentación principal. En la barra de búsqueda del navegador se debe colocar la siguiente dirección configurada por defecto.

192.168.0.229:5000

La pantalla principal, figura 2.31, contiene una barra de menú desde la cual se permite ordenar el apagado o reinicio del sistema, configurar las características de la cerca virtual y obtener información acerca del prototipo. Por otro lado, contiene dos barras laterales informativas, en la barra izquierda se puede observar datos relacionados con el dispositivo remoto, mientras que en la derecha esta la información del sensor de temperatura y humedad relativa instalado junto con el dispositivo receptor. Finalmente, en la parte central se encuentra el mapa de la zona donde se puede determinar la posición del dispositivo remoto y la cerca virtual.

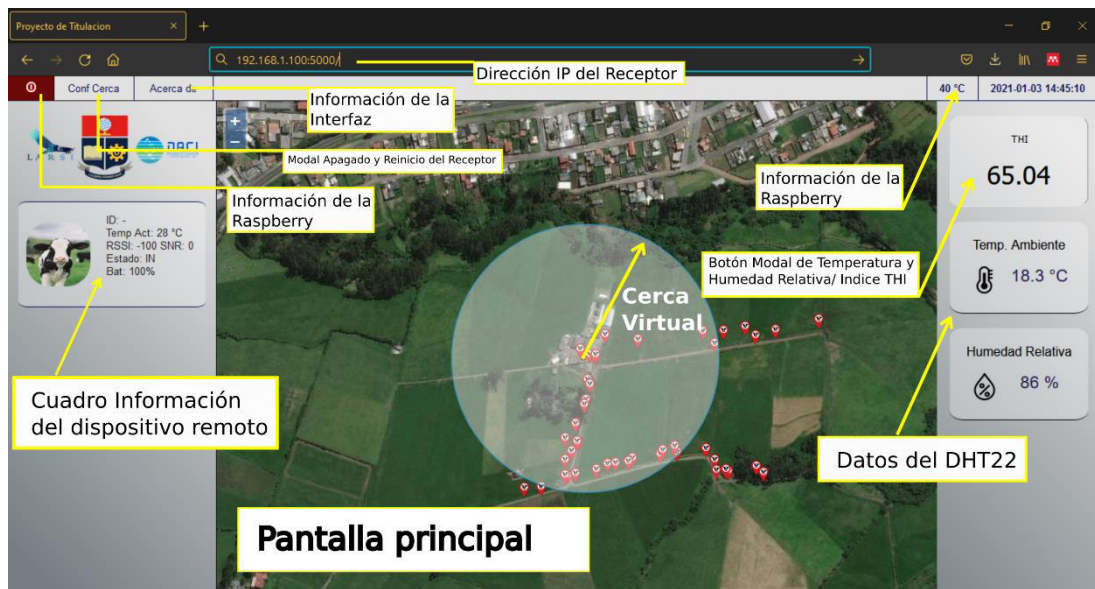


Figura 2.31. Interfaz pantalla principal [Elaboración Propia].

Dentro de la barra derecha se encuentran los datos obtenidos desde el sensor de temperatura y humedad relativa, además, el THI calculado a partir estos datos el cual si presenta un resultado superior a 68 significa que el ganado podría sufrir de estrés térmico. El indicador de THI funciona además como un botón que abre un modal que contiene una gráfica de la variación de temperatura y humedad relativa de las últimas 24 horas como se observa en la figura 2.32.

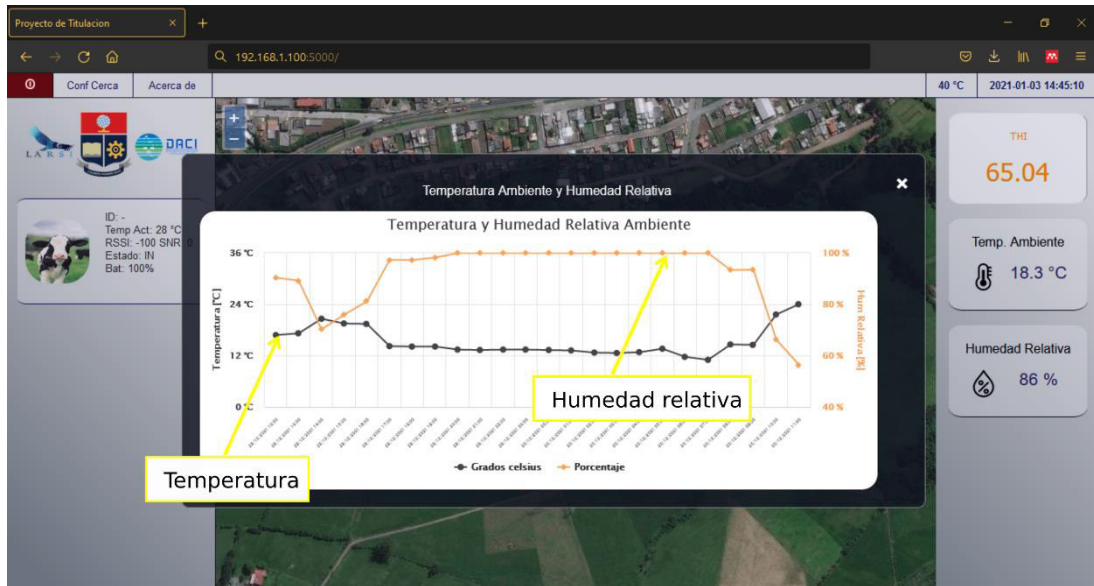


Figura 2.32. Modal de gráficas de temperatura y humedad relativa [Elaboración Propia].

Dentro de la barra de menú esta la opción de controlar el apagado y reinicio del dispositivo remoto, como se observa en la figura 2.33, para cerrar el menú basta con presionar *NO* en cualquiera de las dos opciones.

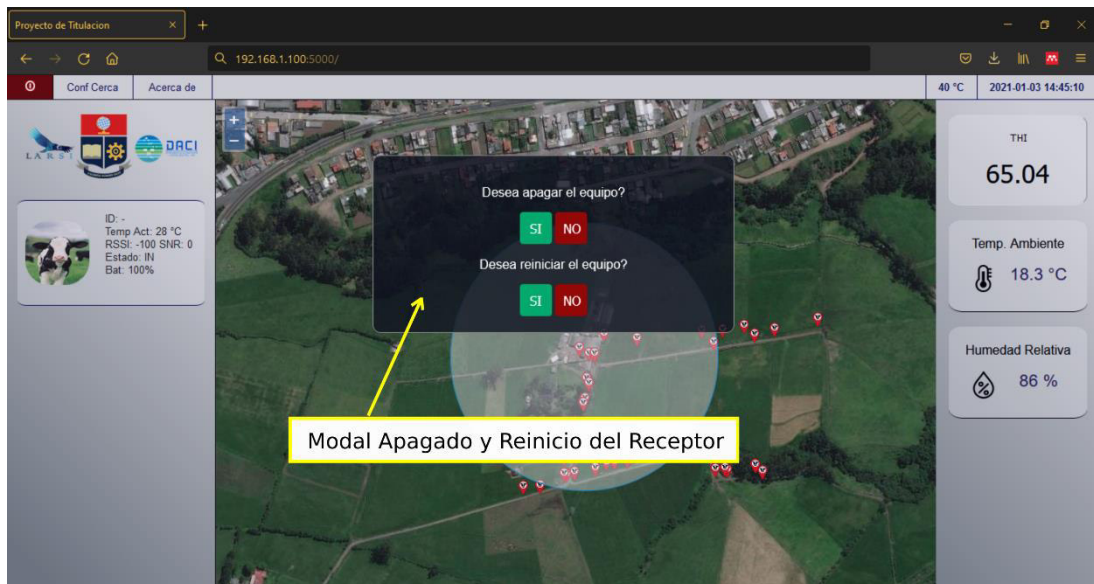


Figura 2.33. Modal de apagado y reinicio del sistema [Elaboración Propia].

Otro de los componentes del menú es la opción de configurar los parámetros de la cerca que en este caso es un círculo, figura 2.34, desde aquí se puede manipular el centro de la ubicación de la cerca y el radio de esta.

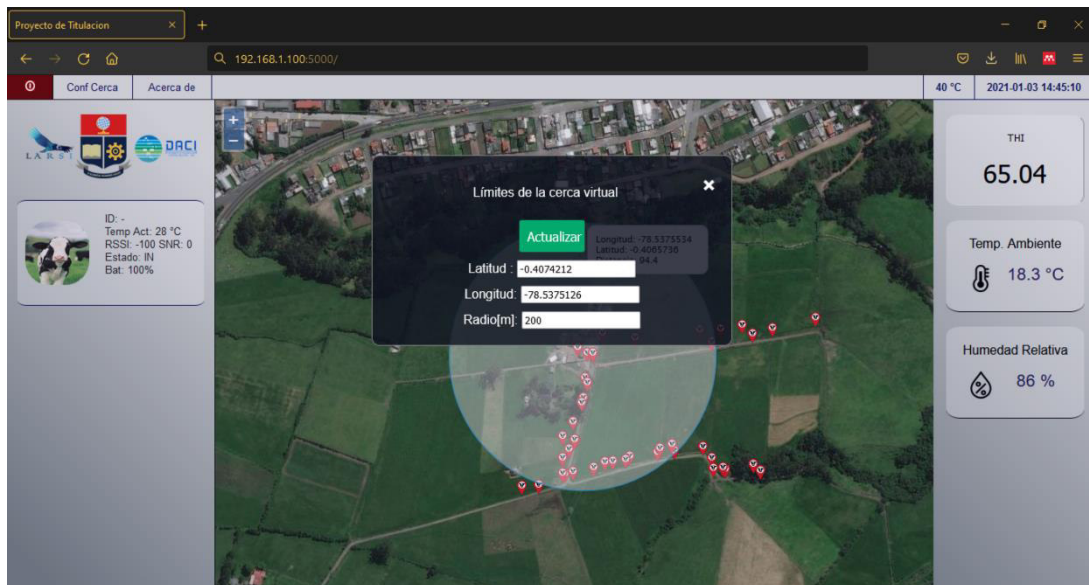


Figura 2.34. Modal de configuración de la cerca virtual [Elaboración Propia]

3. RESULTADOS Y DISCUSIÓN

Después de llevar a cabo el diseño, construcción y programación de las partes que conforman el prototipo se procede a medir el desempeño, se analizan diferentes variables, como la precisión de la geo-posición, del sensor de temperatura, el rango de cobertura del dispositivo remoto y la precisión en las medidas de temperatura y humedad relativa ambiente. Las pruebas son llevadas a cabo en *Hacienda San Miguel* ubicada en el sector de Tambillo a las afueras de la ciudad de Quito.

3.1 PRUEBAS DE GEO-POSICIÓN

El primer parámetro que se analiza es la precisión de los datos de geo-posición obtenidos del GPS en campo, para este efecto se tomaron cuatro muestras en diferentes puntos en lugares abiertos, y mediante el uso de un teléfono celular en cada punto se obtuvo el valor de referencia de latitud y longitud para poder comparar con las muestras obtenidas del dispositivo remoto en ese mismo punto. Ahora bien, mediante la ecuación 2.2 se obtiene la distancia entre el punto de referencia y cada una de las cuatro muestras, resultando así una distancia promedio, la cual sería el error promedio de posición en ese punto.

Otro de los datos obtenidos es el número de satélites a los cuales está conectado el dispositivo remoto en el momento de tomar cada medición. En la figura 3.1 se observa los resultados obtenidos a partir de estas mediciones.

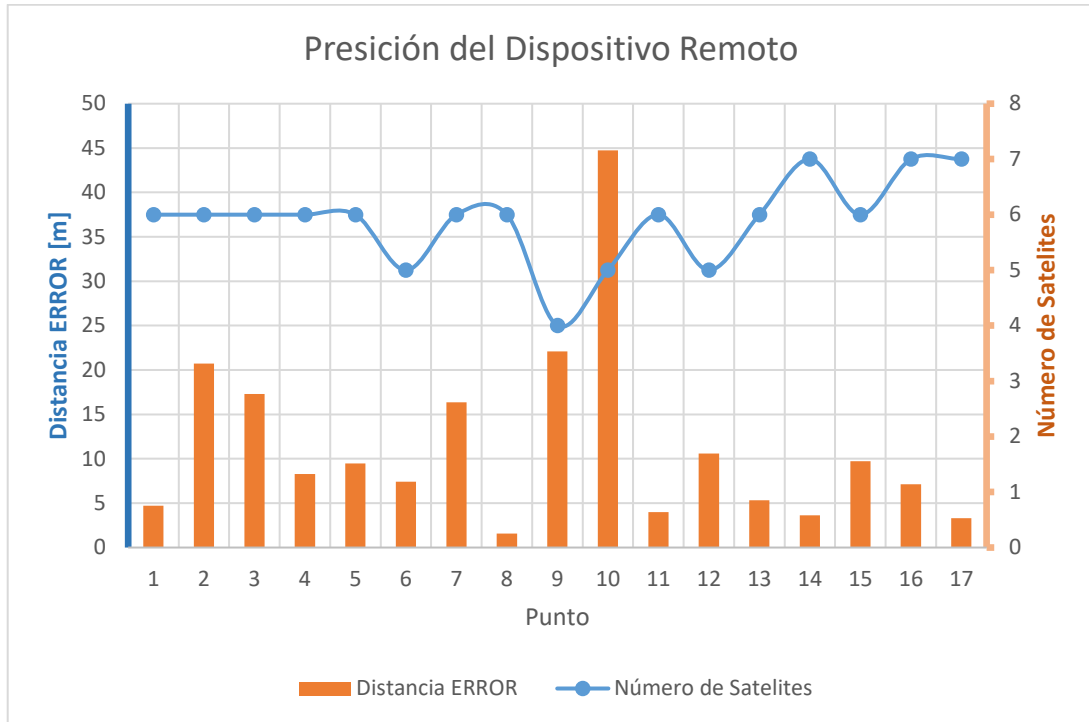


Figura 3.1 Distancia error y número de satélites entre las muestras de geo-posición. [Elaboración Propia].

En base a los resultados se puede observar que los errores máximos son de 22 y 45 metros, con la similitud que ambos en ese momento tenían conexión con 4 y 5 satélites, mientras que el error disminuye a medida que el número de satélites aumenta.

El fabricante del módulo GPS neo-6m promete una precisión de 2.5 metros [38], sin embargo, esta precisión está determinada por las condiciones del entorno donde se toma la medición la cual influye en el número de satélites de los cuales se alcanza a establecer la conexión.

3.2 PRUEBAS DE COBERTURA

El siguiente parámetro que se estudia es la distancia que cubre el dispositivo LoRa, como se observó en [29], el rango de cobertura está sujeto a condiciones del terreno, como la geografía, edificaciones o árboles de la zona, elementos que pueden alterar la línea de

vista entre el receptor y el dispositivo remoto. Para realizar las pruebas de cobertura, una vez instalado el receptor en un lugar fijo, se activa el dispositivo remoto, este dispositivo obtiene los datos del GPS y los envía hacia el receptor, donde se almacenan en una base de datos para posteriormente obtener la distancia entre el punto obtenido y la posición fija del receptor. En la figura 3.2 se observan diferentes posiciones del dispositivo remoto dentro de la hacienda. En los resultados se obtuvo una distancia de cobertura de hasta 353 metros calculados mediante la ecuación 2.2, y como se observa las distancias de mayor cobertura corresponden a aquellas que no presentan obstáculos como árboles o edificios.

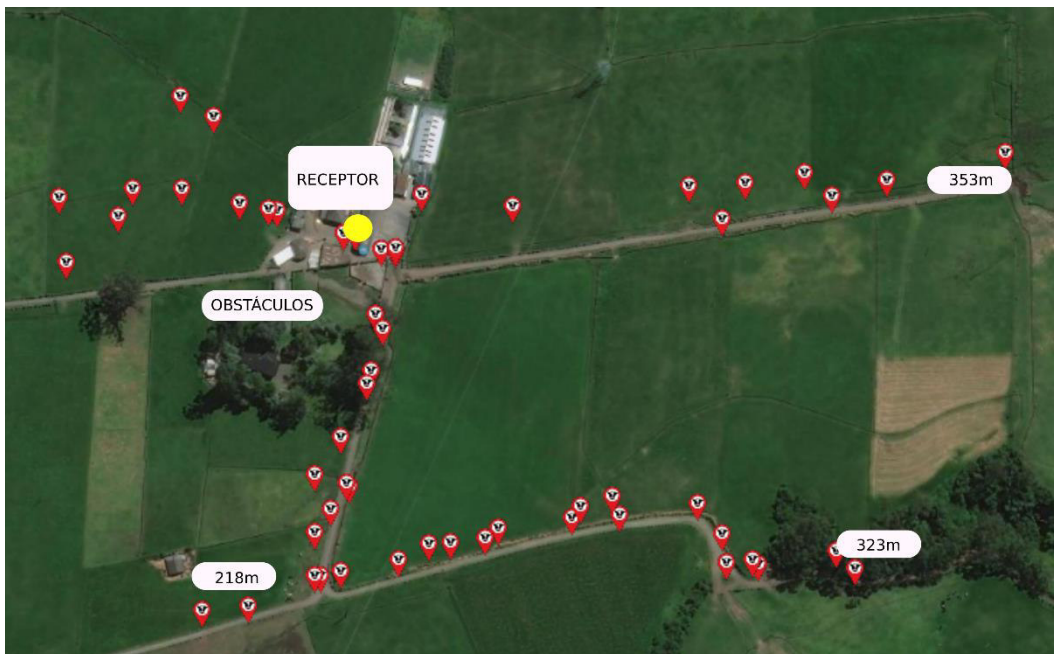


Figura 3.2 Puntos de geo-posición, muestras tomadas en los senderos de la Hacienda ubicada en Tambillo [Elaboración Propia].

Para medir la cobertura se emplea el indicador RSSI, el cual determina la intensidad de la señal recibida, como se observa en la figura 3.3. el RSSI para las pruebas en campo se mantiene entre -100 y -120 dBm, estos resultados tienen concordancia con las pruebas realizadas por [29], en la cual se observa el desempeño de una red LoRa. Se observa que a una distancia entre 0 y 50 metros el RSSI varía entre -80 y -100 dBm. Además, se registró el SNR a partir de lo cual se observa que en todos los casos se obtiene un valor mayor a cero lo cual indica una baja cantidad de ruido en los datos recibidos.

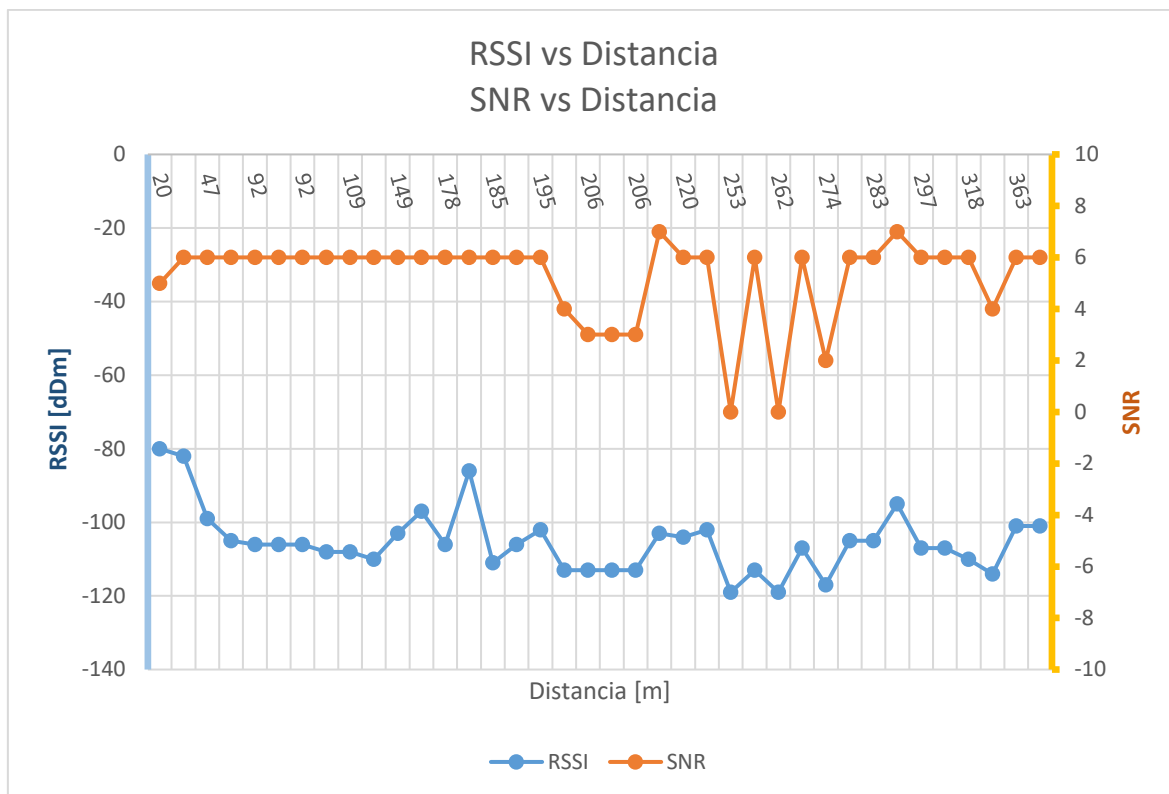


Figura 3.3 RSSI y SNR vs Distancia, valores medidos dentro de la zona de crianza Hacienda San Miguel -Tambillo [Elaboración Propia].

3.3 PRUEBAS DE TEMPERATURA CORPORAL

Monitorear la temperatura corporal es uno de los objetivos principales del presente proyecto, y como se manifestó en la fase teórica para obtener estos datos se emplea el sensor infrarrojo MLX90614, el cual tiene una precisión de 0.5 °C [41], este sensor viene calibrado de fábrica con un cuerpo negro de emisividad uno y por defecto tiene esta emisividad en su memoria; en el caso del ganado vacuno respecto a la emisividad del pelaje diferentes autores coinciden en un valor de 0.95 [94][21][95].

Con respecto a la medición de temperatura corporal en ganado mediante el uso de sensores infrarrojos, estos autores [21][95][96] emplearon en parte de sus estudios pirómetros y cámaras térmicas para medir la temperatura en diferentes partes del cuerpo de las vacas. Por esta razón, para comparar la medición de temperatura del dispositivo remoto se emplea el pirómetro comercial FLUKE 62 MAX con las siguientes características:

- Visualización del valor: máximo, mínimo, promedio, diferencial, función HOLD
- Ajuste de emisividad
- Resolución 0.1 [°C]
- Rango de medición -30 a 500 [°C]
- Precisión: 1.5% respecto a la medición



Figura 3.4 FLUKE 62 MAX [97]

Para comparar las mediciones de temperatura se tomaron muestras de temperatura corporal cerca del hueso de la cadera y a la altura del cuello como se muestra en la figura 3.5, tanto con el dispositivo remoto como con el FLUKE 62 MAX. Además, se tomaron muestras de temperatura rectal mediante un termómetro de mercurio provisto por el cuidador de la Hacienda San Miguel.

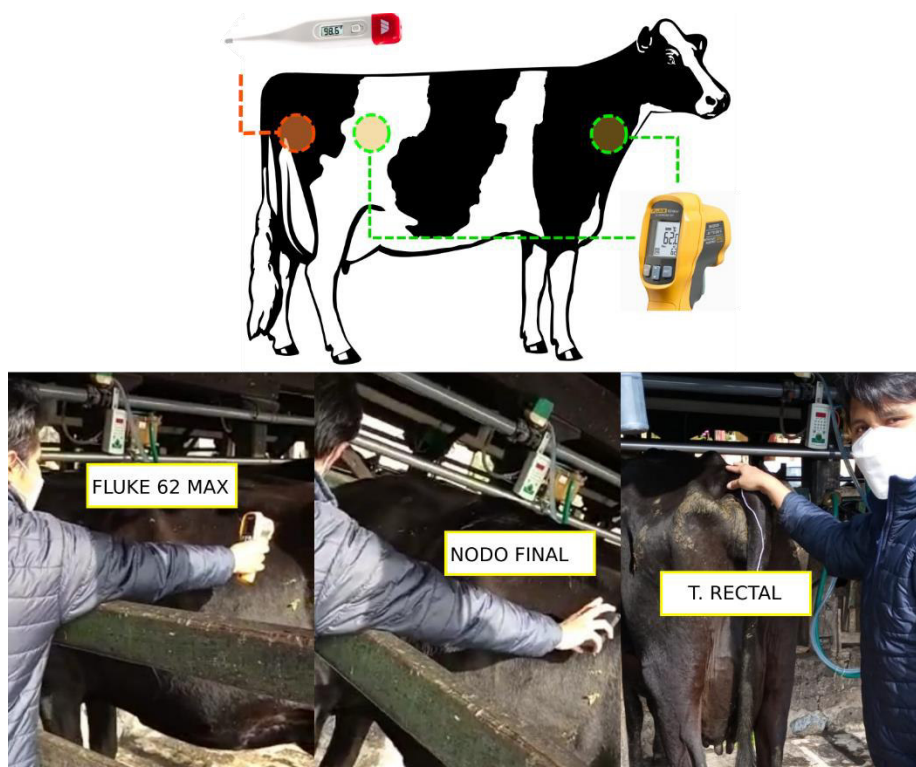


Figura 3.5 Mediciones de temperatura corporal mediante un pirómetro comercial, dispositivo remoto y temperatura rectal. [Elaboración Propia].

Las muestras fueron obtenidas de diez vacas preñadas durante un chequeo veterinario. Como se observa en la figura 3.6 la temperatura varía en el caso del termómetro FLUKE 62 MAX entre 31.5 °C hasta 34.9 °C, mientras que el sensor MXL90614 registra una variación de entre 28.5 °C hasta 32.2 °C, y se determinó una diferencia entre ambos sensores de 2.895 °C y por consiguiente se compensa este desfase mediante la programación del dispositivo remoto.

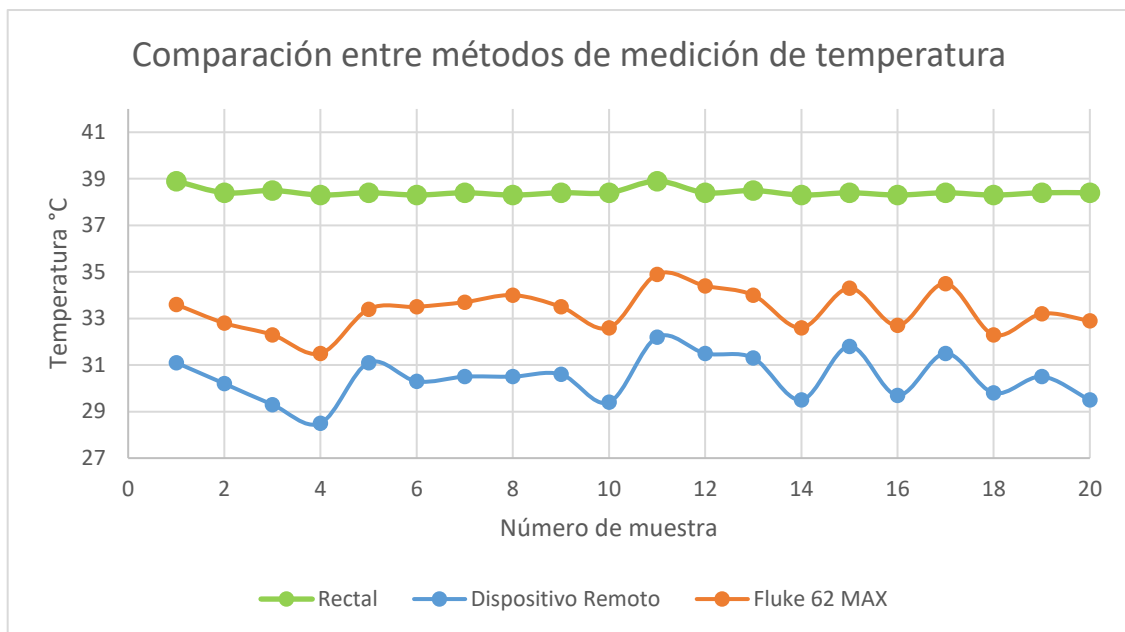


Figura 3.6 Comparación entre medición de temperatura corporal mediante un pirómetro comercial, dispositivo remoto y temperatura rectal. [Elaboración Propia].

Las mediciones fueron realizadas durante las ocho hasta las diez de la mañana, con una temperatura ambiente que varió durante ese periodo entre 14 °C y 16 °C, y los resultados se compararon con los obtenidos por parte de la Universidad de Suecia [10] en la cual se estudió la variación de temperatura corporal ingresando al ganado en una habitación con temperatura controlada, obteniendo así una variación entre 31 °C y 33 °C para las muestras tomadas en el cuello como en el hueso de la cadera. En el caso de la temperatura rectal se mantiene dentro del rango de 38.3 °C y 38.7 °C. Al comparar los datos obtenidos al realizar mediciones se observa concordancia con estudios realizados por trabajos de investigación previos, tanto para la temperatura corporal como la temperatura rectal y de esta manera se puede validar el monitoreo de temperatura.

3.4 PRUEBAS DE TEMPERATURA Y HUMEDAD RELATIVA AMBIENTE

Dentro de los componentes de hardware en el receptor se incluyó un sensor de temperatura y humedad relativa, puesto a que como se menciona en el apartado teórico estas variables tiene una relación con la temperatura del ganado la cual está representada por el THI. Con la finalidad de validar este sensor, en este apartado se compara las muestras obtenidas en la Hacienda San Miguel con datos oficiales proporcionados por la base de datos del INAMHI [42] de dos de las estaciones meteorológicas más cercanas a esta ubicación las cuales son Cutuclagua y Machachi que se presentan en la tabla 3.1 y en la figura 3.7

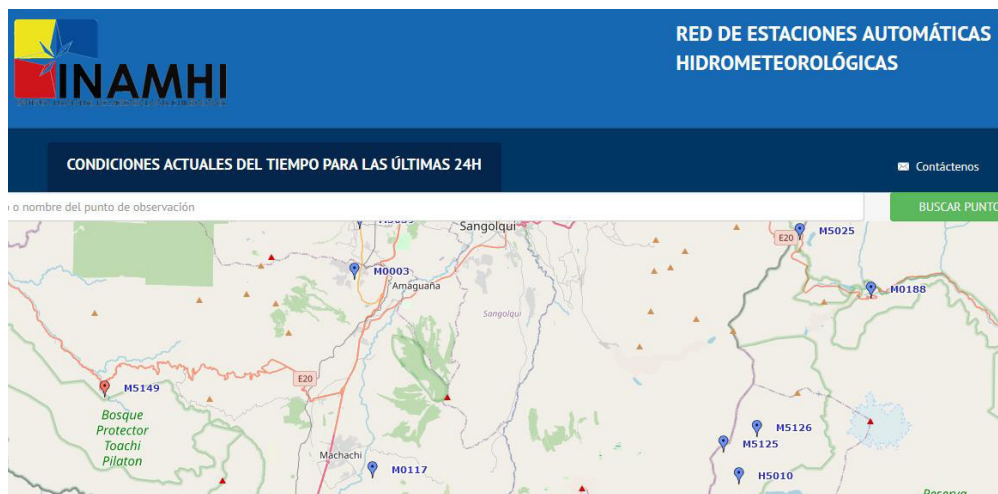


Figura 3.7 Pagina de la Red de Estaciones Automáticas Hidrometeorológicas INAMHI. [42]

Tabla 3.1. Estaciones meteorológicas en la cercanía de la Hacienda Tambillo

Código	Nombre	Latitud	Longitud	Altitud [msnm]
M0003	IZOBAMBA	-0.365833	-78.555000	3058
M0117	MACHACHI	-0.524167	-78.540556	2944

En la figura 3.8 se muestra la comparación de una selección de medidas de temperatura tomadas en un intervalo de 24 horas, como se observa el sensor de temperatura ubicado en la hacienda sigue la tendencia de variación de temperatura de las dos estaciones meteorológicas. Para medir el error se tomaron 96 muestras y se determinó un error

promedio de 3 °C sobre la temperatura obtenida de estaciones oficiales, por tanto, se realizó una compensación en la programación del dispositivo remoto.

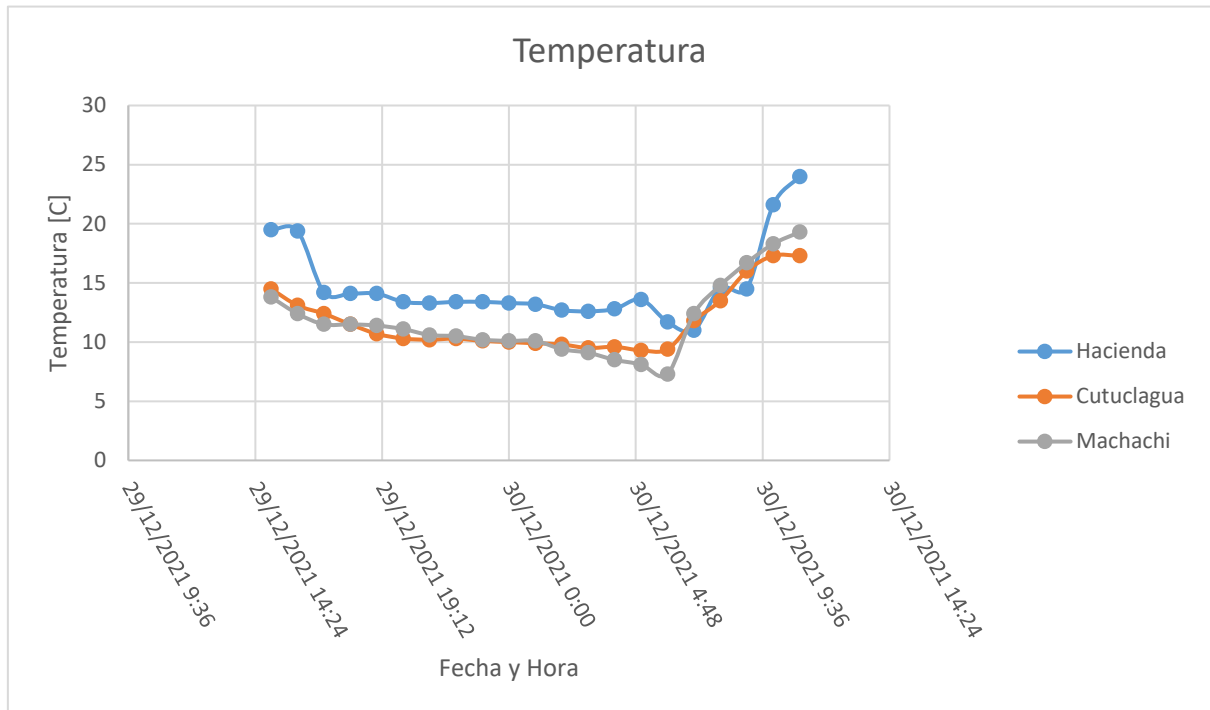


Figura 3.8 Temperatura Ambiente monitoreada en Hacienda San Miguel y estaciones meteorológicas automáticas Machachi y Cutuclagua (INAMHI) [Elaboración Propia].

En el caso de la humedad relativa el sensor presenta problemas ante los aumentos de temperatura, y en esa medida se tarda más tiempo en alcanzar la humedad relativa como se observa en la figura 3.9, sin embargo, las disminuciones de temperatura las sigue correctamente, y en los puntos máximos de humedad relativa el error entre el prototipo y las estaciones meteorológicas es cero.

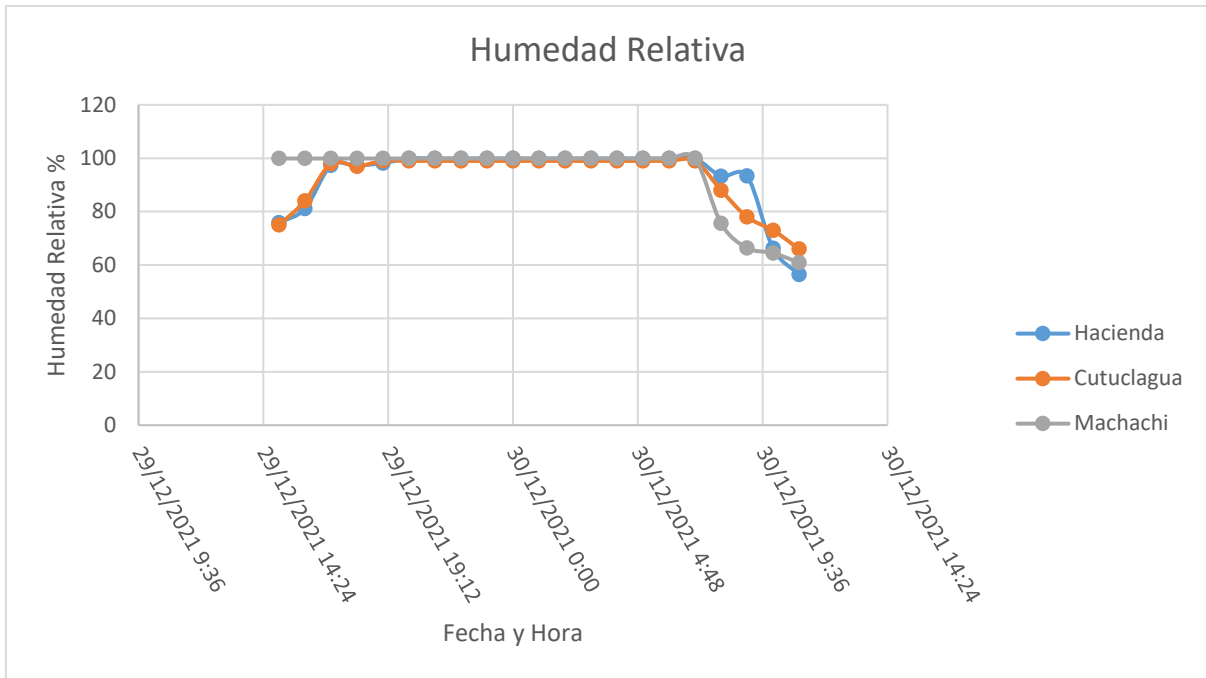


Figura 3.9. Humedad Relativa Ambiente monitoreada en Hacienda y estaciones meteorológicas automáticas Machachi y Cutuclagua (INAMHI) [Elaboración Propia].

3.5 INCONVENIENTES

Existen diversos factores que influyen en el rango de cobertura de un dispositivo LoRa, el principal es la línea de vista que existe entre receptor y emisor. Por cuestiones internas de la hacienda no fue posible instalar el receptor en otra localización reduciéndose así la visibilidad y por consecuencia el rango de cobertura el cual abarca una distancia máxima actualmente de 365 metros.

Por otro lado, al momento de validar muestras de temperatura, se debió tomar muestras a pocos animales, ya que solo se cuenta con un nodo final además que el ganado se coloca en diferentes puntos de la hacienda para llevar a cabo su alimentación y es movido constantemente. Además, para realizar las mediciones se requiere estar en contacto con el animal y esto puede producirle estrés, lo cual no es recomendable puesto que es ganado lechero y puede disminuir la producción.

3.6 COSTO DE IMPLEMENTACIÓN

Finalmente, para cerrar el tercer capítulo, a partir de tabla 3.2 y tabla 3.3 se muestran los costos de los elementos involucrados en el desarrollo de este proyecto. Los costos asociados a los materiales son considerablemente altos debido a que fueron adquiridos por unidad y por parte de diferentes proveedores, de igual manera la implementación de los circuitos impresos fue realizada de manera artesanal, en el caso de requerirse una producción a gran escala los elementos pueden ser adquiridos directamente desde el fabricante en un volumen más alto y con una reducción de costos notable, de manera que el costo de venta final del dispositivo se pueda mantener, incluyendo además el costo de producción y generando utilidad. Cabe recalcar que el costo actual del dispositivo remoto es similar al comercializado por mOOvement el cual es de 65\$ en Australia por lo tanto se debe considerar el costo de importación e impuestos [34].

Por otro lado, en la tabla 3.4 se especifica el costo de ingeniería que corresponde al diseño, construcción y pruebas realizadas en torno al prototipo, este costo se considera solo para la primera etapa de desarrollo y no para la producción en masa del dispositivo. El costo total del proyecto se muestra en la tabla 3.5.

Tabla 3.2. Costo elementos para implementación de dispositivo remoto

Nombre	Especificación	Cant.	C. Unit	Costo
Firebeetle	Tarjeta micro controlada ESP-32, bajo consumo	1	\$ 11.90	\$ 11.90
Chip LoRa	RFN95W 915Mhz LoRa	1	\$ 13.89	\$ 13.89
Conector SMA	Conector SMA para PCB	1	\$ 2.11	\$ 2.11
Antena LoRa	Antena LoRa 2DBI 915Mhz	1	\$ 3.00	\$ 3.00
GPS NEO 6M	Módulo GPS Ublox Neo 6M, 1Hz	1	\$ 12.00	\$ 12.00
VTC 6	Batería Litio 18650 Sony, 3000 mAh, 3.7V	1	\$ 12.00	\$ 12.00
MLX 90614	Sensor infrarrojo, termopila 3.7V	1	\$ 11.50	\$ 11.50
Carcasa	Carcasa plástica Impreso 3D	1	\$ 5.00	\$ 5.00
Collar	Collar sintético y sujetadores	1	\$ 5.00	\$ 5.00
PCB		1	\$ 1.00	\$ 1.00
			Subtotal	\$ 77.40

Tabla 3.3. Costo elementos para implementación del receptor

Nombre	Especificación	Cant.	C. Unit	Costo
Raspberry	Raspberry PI3 B+, 1Gb RAM	1	\$ 69.99	\$ 69.99
Fuente voltaje	Conversor AC-DC, 110Vac - 5Vdc@3A	1	\$ 8.50	\$ 8.50
RFM95W	Chip LoRa 915Mhz LoRa	1	\$ 13.89	\$ 13.89
Conector SMA	Conector SMA para PCB	1	\$ 2.11	\$ 2.11
Cable micro USB	Cable de datos micro USB	1	\$ 2.50	\$ 2.50
DHT22	Sensor temperatura y humedad relativa	1	\$ 6.99	\$ 6.99
Caja plástica	Caja plástica 105 x 104 x 63 mm	1	\$ 4.90	\$ 4.90
Impresión del contenedor DHT22	Estructura sensor DHT22 para exteriores	1	\$ 10.00	\$ 10.00
Extensión Eléctrica	Extensión eléctrica L, F, N 16awg, 15mts	1	\$ 14.22	\$ 14.22
Patch cord conector RJ45	Cable UTP CAT 5e conector RJ45	15	\$ 0.21	\$ 3.15
Prensa Estopa	G11	2	\$ 0.59	\$ 1.18
Antena	915Mhz, 5dbi	1	\$ 10.99	\$ 10.99
Tarjeta microSD	Tarjeta microSD 16gb class 10	1	\$ 6.90	\$ 6.90
PCB		1	\$ 1.00	\$ 1.00
Router	Router TP-LINK	1	\$ 20.00	\$ 20.00
Subtotal				\$176.82

Tabla 3.4. Costo de horas de ingeniería.

Elemento	Especificación	# Horas	Costo Hora	Costo Total
Costo de Ingeniería	DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE MONITOREO DE GEOPOSICIÓN Y TEMPERATURA CORPORAL PARA GANADO VACUNO MEDIANTE DISPOSITIVOS DE COMUNICACIÓN LORA.	400	\$ 6.00	\$ 2400.00

Tabla 3.5. Costo Total del Prototipo.

Elemento	Costo
Costo elementos para implementación del dispositivo remoto	\$ 77.40
Costo elementos para implementación del receptor	\$ 176.82
Costo de horas de ingeniería	\$ 2400.00
Costo Total sin impuestos	\$ 2654.22

4. CONCLUSIONES Y RECOMENDACIONES

En este capítulo son presentadas las conclusiones y recomendaciones referentes a este proyecto de titulación, con énfasis en el diseño, implementación, pruebas y correcciones realizadas sobre los sensores de geo-posición y temperatura corporal, además del hardware y software del prototipo.

4.1. CONCLUSIONES

- A partir de los resultados se puede concluir que el dispositivo cumple con los objetivos y alcances de este proyecto de titulación.
- Después de realizado el análisis bibliográfico en el país no está disponible este tipo de dispositivos para el rastreo de ganado y que empleen esta tecnología. A diferencia de países como Estados Unidos, Australia y Países Bajos en los cuales utilizan este tipo de sensores para realizar un mejor manejo del ganado y en la prevención de robos.
- Para el desarrollo de este prototipo es más deseable el uso de un collar. Existen alternativas que emplean un arete, sin embargo, para la realización de pruebas y debido a la implementación temprana de estos dispositivos, la manipulación constante del dispositivo podría someter al ganado a una situación de estrés, es por esta razón que el collar permite una manipulación más fácil y cómoda para el animal.
- Al realizar pruebas se determinó que, aunque LoRa tiene un consumo de energía menor, esto no contempla a los demás dispositivos que forman parte de una

determinada aplicación por esta razón el dispositivo debe tener periodos de ahorro de energía. Todos los dispositivos comerciales entran en estado de hibernación y mientras mayor sean los periodos el tiempo de duración de batería incrementa.

- Después de tomar datos de geo-posición dentro de la hacienda, se determinó una distancia de cobertura aceptable, sin embargo, menor a la establecida por LoRa. Esta situación se presenta por varios factores, en primer lugar, se empleó como receptor un chip LoRa diseñado para usarse como nodo final, debido a que durante el desarrollo de este proyecto de titulación los concentradores comerciales no se encontraban disponibles. En segundo lugar, está la irregularidad del terreno, en este caso específico las distancias máximas se obtuvieron en terreno plano, a partir de los 355 metros el terreno pasaba a ser una ladera y por tanto se perdía por completo la línea de vista.
- Al validar el sensor de temperatura se observa que las variaciones de temperatura rectal son mínimas, no siendo este el caso para las variaciones de temperatura corporal que tienen un rango mucho más amplio de variación, sin embargo, si una de ellas aumenta la otra también lo hace. Es así como las muestras tomadas de la superficie del cuerpo, aunque no son iguales a las internas, pueden dar información útil sobre una reducción o aumento excesivo en la temperatura del ganado, de esta manera el cuidador después de corroborar mediante el termómetro rectal puede decidir si llamar o no al especialista veterinario.

4.2. RECOMENDACIONES

- El prototipo desarrollado en este proyecto de titulación a nivel de hardware en cuanto al dispositivo remoto puede sumar diferentes mejoras como un acelerómetro para el análisis de movimiento orientado a la activación del dispositivo remoto. El ganado puede pasar varias horas en un mismo lugar alimentándose, de manera que en ciertos casos la actualización de información puede darse con pausas más prolongadas. De esta manera, si se sobrepasa un umbral de aceleración se podría tomar como un desplazamiento y activar el dispositivo para enviar los datos requeridos hacia el receptor.
- Por otra parte, es posible reducir las dimensiones del dispositivo remoto, el tamaño actual del dispositivo está limitado a las dimensiones de los diferentes

módulos que lo componen. Una alternativa es construir una PCB con todos los elementos electrónicos activos y pasivos por separado, de esta manera se puede descartar los elementos incluidos por el fabricante de cada módulo y que no se utilizan en este proyecto.

- A nivel de software, la interfaz al ser creada empleando herramientas de desarrollo web, existe la posibilidad de enviar esta información a través de internet y monitorear las variables fuera de la red local.
- Para enviar datos desde el receptor a través de internet se recomienda cambiar de framework, pues Flask no se recomienda para su uso en producción es decir ya en un dispositivo comercial. Debido a que este framework es compatible con python y conlleva una implementación más amigable fue el framework escogido para el desarrollo de este prototipo.
- El siguiente paso es aumentar el número de dispositivos remotos, de modo que se puede analizar el desplazamiento del ganado a lo largo del día. Así, por ejemplo, si un animal permanece todo el tiempo en un mismo punto podría deberse a una lesión que limita su movimiento.
- Para incrementar la línea de vista se recomienda colocar el dispositivo receptor en una posición más alta, esto puede realizarse mediante una estructura metálica. Debido al costo de implementación este punto no pudo ser llevado a cabo.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] R. G. Velasco, Bolivar, “El robo de ganado deja un perjuicio millonario en el sector,” Guayaquil, 2020.
- [2] Asamblea Nacional del Ecuador, *Código Orgánico Integral Penal*. 2014.
- [3] O. Dieng, B. Diop, O. Thiare, and C. Pham, “A study on IoT solutions for preventing cattle rustling in African context,” *ACM Int. Conf. Proceeding Ser.*, 2017, doi: 10.1145/3018896.3036396.
- [4] R. C. Report, “A CHALLENGING TIME FOR THE COUNTRYSIDE Rural Crime Report,” 2020.
- [5] J. M. Schieltz, S. Okanga, B. F. Allan, and D. I. Rubenstein, “GPS tracking cattle as a monitoring tool for conservation and management,” *African J. Range Forage Sci.*, vol. 34, no. 3, pp. 173–177, 2017, doi: 10.2989/10220119.2017.1387175.
- [6] Libelium, “How a dairy farm increased their milk production 18% with IoT and Machine Learning | Libelium,” 2019. <http://www.libelium.com/how-a-dairy-farm-increased-their-milk-production-18-with-iot-and-machine-learning/>.
- [7] D. W. Bailey, M. G. Trotter, C. W. Knight, and M. G. Thomas, “Use of GPS tracking collars and accelerometers for rangeland livestock production,” *Transl. Anim. Sci.*, vol. 2, no. 1, pp. 81–88, 2018, doi: 10.1093/tas/txx006.
- [8] G. J. Bishop-hurley, D. L. Swain, D. M. Anderson, P. Sikka, C. Crossman, and P. Corke, “Virtual fencing applications : Implementing and testing an automated cattle control system,” *ELSEVIER*, vol. 56, pp. 14–22, 2007, doi: 10.1016/j.compag.2006.12.003.
- [9] Z. Butler, P. Corke, R. Peterson, and D. Rus, “Virtual Fences for Controlling Cows,” 2004, doi: 10.1109/ROBOT.2004.1302415.
- [10] J. Isaksson, “Changes in dairy cows’ temperature,” pp. 1–52, 2017.
- [11] O. S. Oystein V. Sjaastad, Knut Hove, *Physiology of Domestic Animals*, Second. Oslo: Scandinavian Veterinary Press, 2010.
- [12] J. M. Bewley, M. E. Einstein, M. W. Grott, and M. M. Schutz, “Comparison of Reticular and Rectal Core Body Temperatures in Lactating Dairy Cows,” *J. Dairy Sci.*, vol. 91, no. 12, pp. 4661–4672, 2008, doi: 10.3168/jds.2007-0835.
- [13] B. Yadav, G. Singh, and A. Wankar, “The use of infrared skin temperature measurements for monitoring heat stress and welfare of crossbred cattle,” *Indian J. Dairy Sci.*, vol. 70, pp. 1–5, 2017, [Online]. Available: https://www.researchgate.net/publication/312045096_The_use_of_infrared_skin_temperature_measurements_for_monitoring_heat_stress_and_welfare_of_crossbred_cattle.
- [14] J. Keown and R. Grant, “How to reduce heat stress in dairy cattle,” p. 2, 2005, [Online]. Available: <http://extension.unl.edu/publications>.
- [15] R. Avendaño, *Milk Production: An Up-To Date Overview Of Animal Nutrition, Management*

And Health, 1st ed. INTECH, 2012.

- [16] C. S. Minerals, "Heat stress in dairy cows Challenges for the dairy industry in Asia." <https://celticseaminerals.com/heat-stress-dairy-cows/>.
- [17] INEC- ESPAC, "Principales Resultados ESPAC," 2019. [Online]. Available: shorturl.at/lry58.
- [18] INAMHI, *Anuario meteorológico Nº 53-2013*, no. 52. 2017.
- [19] T. R. Wrenn, J. Bitman, and J. F. Sykes, "Body Temperature Variations in Dairy Cattle during the Estrous Cycle and Pregnancy," *J. Dairy Sci.*, vol. 41, no. 8, pp. 1071–1076, 1958, doi: 10.3168/jds.S0022-0302(58)91053-1.
- [20] S. Dayanna, O. Sanabria, K. Yiney, and A. Parra, "Celo En Bovinos Fundaments and Current Methods of Estrus," pp. 1–28.
- [21] N. Sellier, E. Guettier, and C. Staub, "A Review of Methods to Measure Animal Body Temperature in Precision Farming," *Am. J. Agric. Sci. Technol.*, no. January 2014, 2014, doi: 10.7726/ajast.2014.1008.
- [22] E. B, "LoRa Documentation," p. 47, 2019, [Online]. Available: <https://lora.readthedocs.io/en/latest/#range-vs-power>.
- [23] D. H. Kim, E. K. Lee, and J. Kim, "Experiencing LoRa network establishment on a smart energy campus testbed," *Sustain.*, vol. 11, no. 7, 2019, doi: 10.3390/su11071917.
- [24] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent, "LoRaWAN™ Specification," 2016, [Online]. Available: https://lora-alliance.org/wp-content/uploads/2020/11/lorawan1_0_2-20161012_1398_1.pdf.
- [25] V. Fialho and F. Azevedo, "Wireless Communication Based on Chirp Signals for LoRa IoT Devices," *i-ETC ISEL Acad. J. Electron. Telecommun. Comput.*, vol. 4, no. 1, p. ID-6, 2018.
- [26] Semtech, "LoRa and LoRaWAN," *Semtech Tech. Pap.*, no. December 2019, pp. 1–17, 2020, [Online]. Available: <https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/>.
- [27] F. Adelantado, X. Vilajosana, P. Tuset-peiro, B. Martinez, J. Melià-seguí, and T. Watteyne, "Understanding the Limits of LoRaWAN," no. January, 2017.
- [28] LightBug, "Battery life calculator." <https://thelightbug.com/>.
- [29] A. Muaz Abdul Rahman, F. Hafizhelmi Kamaru Zaman, and S. Afzal Che Abdullah, "Performance Analysis of LPWAN Using LoRa Technology for IoT Application," *Int. J. Eng. Technol.*, vol. 7, no. 4.11, p. 252, 2018, doi: 10.14419/ijet.v7i4.11.21387.
- [30] J. S. Andersen and E. Joakim, "Investigating the practical performance of the LoRaWAN technology," p. 61, 2017, doi: <https://www.diva-portal.org/smash/get/diva2:1175407/FULLTEXT01.pdf>.
- [31] F. Turčinović, J. Vuković, S. Božo, and G. Šišul, "Analysis of LoRa Parameters in Real-World Communication," no. December, pp. 8–12, 2020, doi:

10.1109/ELMAR49956.2020.9219028.

- [32] J. CATALANO *et al.*, “RP002-1.0.2 LoRaWAN® Regional Parameters,” 2020, [Online]. Available: https://loro-alliance.org/wp-content/uploads/2020/11/RP_2-1.0.2.pdf.
- [33] L. Casals, B. Mir, R. Vidal, and C. Gomez, “Modeling the Energy Performance of LoRaWAN,” 2017, doi: 10.3390/s17102364.
- [34] MOOvement, “How the moovement system works.” [Online]. Available: https://4tags.com.au/downloads/moovement_flyer.pdf.
- [35] H. Herrmann and H. Bucksch, “SODAQ Cattle Tracker,” 2014. doi: 10.1007/978-3-642-41714-6_131100.
- [36] SODAQ, “SODAQ Solar Powered LoRa Cattle tracker V2.” <https://shop.sodaq.com/sodaq-cowtracker-v2.html>.
- [37] LightBug, “LightBug Zero & Pro User Manual.” p. 90, [Online]. Available: https://thelightbug.com/user_manual.pdf.
- [38] u-blox AG, “NEO-6 u-blox 6 GPS Modules,” 2017. [Online]. Available: [https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_\(GPS.G6-HW-09005\).pdf](https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf).
- [39] u-blox AG, “NEO-7 - Data Sheet Document,” 2014. [Online]. Available: https://www.u-blox.com/sites/default/files/products/documents/NEO-7_DataSheet_%28UBX-13003830%29.pdf.
- [40] u-blox AG, “u-blox M8 concurrent GNSS modules,” 2016. [Online]. Available: https://www.u-blox.com/sites/default/files/NEO-M8-FW3_DataSheet_UBX-15031086.pdf.
- [41] Melexis, “MLX90614 - Single and Dual Zone Infrared Thermometer in TO-39,” 2006. [Online]. Available: https://www.sparkfun.com/datasheets/Sensors/Temperature/MLX90614_rev001.pdf.
- [42] INAMHI, “Red de estaciones automáticas hidrometereológicas,” 2021. <http://186.42.174.236/InamhiEmas/#>.
- [43] Aosong Electronic, “DHT22 Datasheet,” *Tech. Datasheet*, vol. 22, pp. 1–10, 2015, [Online]. Available: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>.
- [44] SEMTECH, “Sx1276/77/78/79 Datasheet,” 2015.
- [45] SEMTECH, “SX1301 Datasheet,” 2017.
- [46] J. & Penn, “User Manual for WisLink-LoRa Concentrator Module Raspberry Pi HAT Edition Table of Contents,” 2019. [Online]. Available: https://cdn.sparkfun.com/assets/f/4/7/9/7/RAK2245_Pi_HAT_User_Manual_V1.2.pdf.
- [47] Hope Microelectronics Co., “Datasheet: RFM95/96/97/98(W) v1.0,” vol. 98, p. 121, 2014, [Online]. Available: http://www.hoperf.com/rf_transceiver/lora/RFM95W.html%5Cnhttp://www.hoperf.com/

upload/rf/RFM95_96_97_98W.pdf.

- [48] RaspberryPi, "Raspberry Pi 3 Model B+ Datasheet," *Datasheet*, p. 5, 2016, [Online]. Available: <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf>.
- [49] R. Pi, "Raspberry Pi 4 Computer Model B," *Raspberrypi.Org*, no. May, pp. 1–6, 2020, [Online]. Available: www.raspberrypi.org.
- [50] E. Upton and G. Halfacree, *Raspberry Pi User Guide*, 4th ed. Willey, 2016.
- [51] Espressif, "ESP32 Series Datasheet," *Espr. Syst.*, pp. 1–61, 2019, [Online]. Available: www.espressif.com.
- [52] Heltec Automation, "Wifi LoRa 32," p. 1, 2020, [Online]. Available: [https://resource.heltec.cn/download/WiFi_LoRa_32/V2/WIFI_LoRa_32_V2\(868-915\).PDF](https://resource.heltec.cn/download/WiFi_LoRa_32/V2/WIFI_LoRa_32_V2(868-915).PDF).
- [53] DFROBOT, "DFR0654." https://wiki.dfrobot.com/FireBeetle_Board_ESP32_E_SKU_DFR0654#target_12.
- [54] R. T. Corporation, "RT9080 LDO Regulator," no. April, pp. 1–14, 2016, [Online]. Available: https://www.richtek.com/assets/product_file/RT9080/DS9080-00.pdf.
- [55] D. Incorporated, "Ap2112 Cmos Ldo Regulator," no. June, pp. 1–18, 2017, [Online]. Available: <https://www.diodes.com/assets/Datasheets/AP2112.pdf>.
- [56] FullWatt, "Baterías de litio falsas," 2020. <http://blog.fullwat.com/baterias-de-litio-falsas-parte-2/>.
- [57] Sony Energy Devices Corporation, "Lithiom Ion Rechargeable Battery Technical Information US18650VTC6." 2015, [Online]. Available: https://www.kronium.cz/uploads/SONY_US18650VTC6.pdf.
- [58] P. Org, "Python beginners guide." <https://wiki.python.org/moin/BeginnersGuide/Overview>.
- [59] P. Basis, "Flask," 2021. <https://pythonbasics.org/what-is-flask-python/>.
- [60] M. Makai, "WSGI Servers." <https://www.fullstackpython.com/wsgi-servers.html>.
- [61] OpenLayers, "Open Layers." <https://openlayers.org/en/latest/doc/tutorials/background.html>.
- [62] M. D. Network, "Javascript." https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript.
- [63] M. D. Network, "Structuring the web with HTML." <https://developer.mozilla.org/en-US/docs/Learn/HTML>.
- [64] M. D. Network, "Learn to style HTML using CSS." <https://developer.mozilla.org/en-US/docs/Learn/CSS>.

- [65] Oracle, "Base de datos definida." <https://www.oracle.com/mx/database/what-is-database/>.
- [66] SQLite, "About SQLite." <https://www.sqlite.org/about.html>.
- [67] SQLite, "Most Widely Deployed and Used Database Engine." <https://www.sqlite.org/mostdeployed.html>.
- [68] N. T. P. A. Corp, "TP4056 1A Standalone Linear Li-Ion Battery Charger with Thermal Regulation in SOP-8," *Data Sheet*, p. 3, 2019.
- [69] ORMER, "DHT22 TEMPERATURE HUMIDITY SHELTER." <https://www.thingiverse.com/thing:3222598>.
- [70] TP-LINK, "TL-WR840N." [Online]. Available: https://static.tp-link.com/res/down/doc/TL-WR840N_2.0.pdf.
- [71] M. Hart, "Tiny GPS," 2013. <https://github.com/mikalhart/TinyGPS>.
- [72] G. Baddeley, "GPS-NMEA sentence information," 2001. <http://aprs.gids.nl/nmea/>.
- [73] S. Mistry, "LoRa." <https://www.arduino.cc/reference/en/libraries/lora/>.
- [74] Lady Ada, "Using Melexis MLX90614 Non-Contact Sensors." <https://learn.adafruit.com/using-melexis-mlx90614-non-contact-sensors/wiring-and-test>.
- [75] LastMinuteEngineers, "Insight Into ESP32 Sleep Modes & Their Power Consumption." <https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/>.
- [76] RaspberryOrg, "Raspberry Pi Documentation," 2022. <https://www.raspberrypi.com/documentation/computers/configuration.html>.
- [77] P. Org, "Miscellaneous operating system interfaces." <https://docs.python.org/3/library/os.html?highlight=os#module-os>.
- [78] Adafruit and B. Rubell, "LoRa and LoRaWAN Radio for Raspberry Pi," 2019. <https://learn.adafruit.com/lora-and-lorawan-radio-for-raspberry-pi/rfm9x-raspberry-pi-setup>.
- [79] Adafruit and Lady Ada, "DHT Humidity Sensing on Raspberry Pi or Beaglebone Black with GDocs Logging," 2012. <https://learn.adafruit.com/dht-humidity-sensing-on-raspberry-pi-with-gdocs-logging/python-setup>.
- [80] RaspberryOrg, "Build a Python Web Server with Flask." <https://projects.raspberrypi.org/en/projects/python-web-server-with-flask/1>.
- [81] P. Org, "DB-API 2.0 interface for SQLite." DB-API 2.0 interface for SQLite.
- [82] P. Org, "Time access and conversions." <https://docs.python.org/3/library/time.html?highlight=time#module-time>.
- [83] Numpy, "NumPy documentation," 2021. <https://numpy.org/doc/stable/>.

- [84] P. Org, “threading — Thread-based parallelism.” <https://docs.python.org/3/library/threading.html#module-threading>.
- [85] P. Org, “multiprocessing — Process based parallelism.” <https://docs.python.org/3/library/multiprocessing.html?highlight=multiprocessing#module-multiprocessing>.
- [86] P. Org, “json — JSON encoder and decoder.” <https://docs.python.org/3/library/json.html>.
- [87] JQuery, “jQuery.ajax().” <https://api.jquery.com/jQuery.ajax/>.
- [88] OpenLayers, “The ol package,” [Online]. Available: <https://openlayers.org/download/>.
- [89] Q. Project, “QGIS Documentation,” 2021. https://docs.qgis.org/3.16/es/docs/user_manual/preamble/foreword.html.
- [90] OpenLayers, “Open Layers Documentation.” <https://openlayers.org/en/latest/apidoc/>.
- [91] A. Anisya and G. Y. Swara, “Implementation of Haversine Formula and Best First Search Method in Searching of Tsunami Evacuation Route,” *IOP Conf. Ser. Earth Environ. Sci.*, vol. 97, no. 1, 2017, doi: 10.1088/1755-1315/97/1/012004.
- [92] RaspberryOrg, “Raspberry Pi Using Linux.” https://www.raspberrypi.com/documentation/computers/using_linux.html.
- [93] Highcharts, “Highcharts Documentation.” <https://www.highcharts.com/docs/index>.
- [94] F. M, “El termómetro infrarrojo como alternativa de medición de temperatura corporal en vacas lecheras,” pp. 1–34, 2020.
- [95] Z. Shuailong, Z. Changfan, J. Xunping, H. Jingshu, and X. Dequan, “Progress on Infrared Imaging Technology in Animal Production: A Review,” *MDPI*, pp. 5–6, 2022, doi: 10.3390/s22030705.
- [96] Z. Gil, K. Adamczyk, P. Zapletal, J. Frelich, M. Šlachta, and A. Andreasik, “Impact of the location of the dairy cows in the barn on their body surface temperature,” *J. Cent. Eur. Agric.*, vol. 14, no. 3, pp. 228–237, 2013, doi: 10.5513/JCEA01/14.3.1310.
- [97] FLUKE, “Manual Usuario 62 max/62 max +,” 2012. <https://www.fluke.com/es-es/producto/medicion-de-temperatura/termometros-infrarrojos/fluke-62-max>.

ANEXOS:

ANEXO A. Manual de Usuario

ANEXO A

MANUAL DE USUARIO

A.1. INTRODUCCIÓN

El robo de ganado es un problema en el país, y una de las principales razones se debe a que las personas a cargo de su cuidado no pueden permanecer todo el tiempo junto al ganado a lo largo del día. Por otro lado, la temperatura de un ser vivo puede variar de acuerdo con su estado de salud o respecto a condiciones ambientales. Es por esta razón que se creó el *prototipo de monitoreo de geo-posición y temperatura de ganado vacuno*, el cual permite conocer a ubicación del ganado en un área del terreno de crianza; además, permite conocer la temperatura de la superficie del cuerpo de la vaca, así como las condiciones ambientales en el terreno. Es así como, esta manera el cuidador puede ir tras una vaca y regresarla a la manada en caso de que esta se encuentre fuera del área predeterminada, de igual forma si observa una temperatura elevada en el animal puede tomar acciones como llamar a un veterinario para evaluar su estado de salud, o incrementar el suministro de agua si la temperatura ambiente y la humedad relativa son muy elevadas.

A.1.1. Objetivo

Este manual tiene como objetivo principal mostrar el funcionamiento del sistema, como poner en marcha el dispositivo remoto y las consideraciones que se deben tener durante su operación, así como la funcionalidad del dispositivo remoto y las opciones de la interfaz de usuario.

A.2. CARACTERÍSTICAS

En este apartado está dividido en dos etapas, en primer lugar, están las partes físicas que componen el prototipo, y en la segunda etapa los programas creados para su utilización en este prototipo.

A.2.1. Hardware

El prototipo tiene dos componentes físicos, el primero es el dispositivo remoto que contiene los sensores de geo-posición y temperatura y el chip LoRa utilizado por la envío de información, forma parte de un collar de modo que puede ser colocado en el cuello del animal. El segundo componente es el receptor que obtiene la información

desde el dispositivo remoto, este lo conforman un chip LoRa, un sensor de temperatura, sensor de humedad relativa y una raspberry Pi 3 B+ que se encarga de gestionar la información, almacenarla y mostrarla en la interfaz la cual puede ser accedida desde un computador a través de un router formando así una red local con topología estrella.

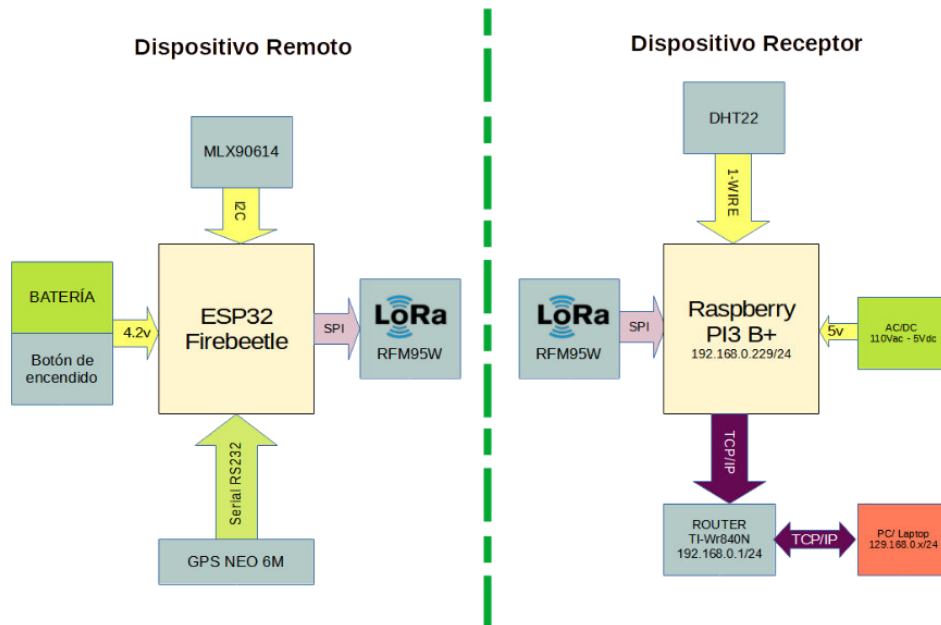


Figura A.1. Esquema general del hardware [Elaboración Propia].

A.2.1.1 Dispositivo remoto

El dispositivo remoto está formado por un GPS neo-6m del fabricante UBLOX, un sensor de temperatura infrarrojo MLX90614 de Melexis, y un transceptor LoRa RFM95W, una tarjeta micro controlada Firebeetle basada en ESP32 el cual envía la información provista por estos sensores mediante el chip LoRa. Todo este conjunto de elementos es alimentado por una batería de Ion Litio 18650 de 3000mAh. En la figura A.2 se observa todos estos componentes de hardware, contenidos en una carcasa de plástico y sujetos a un collar.

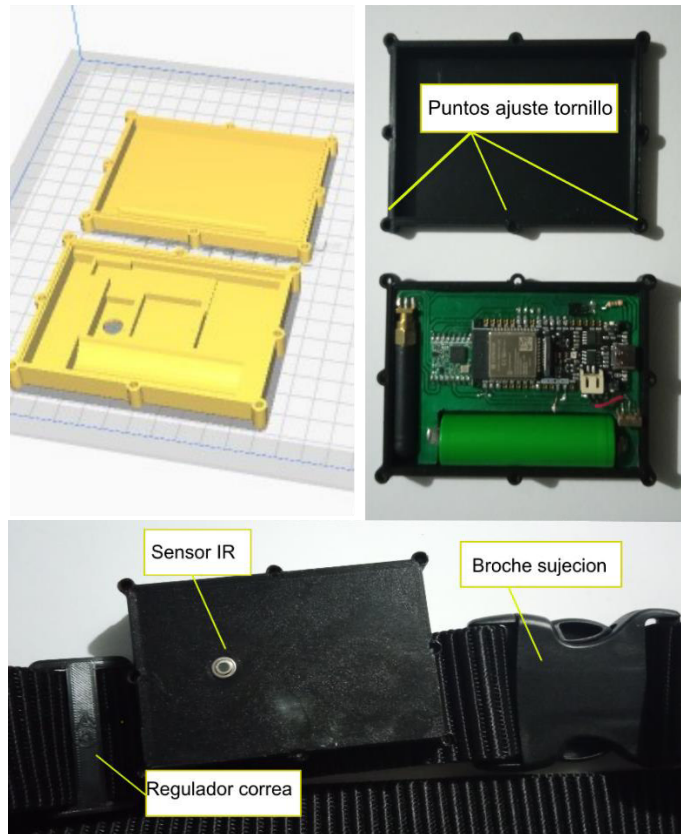


Figura A.2. Estructura para el nodo final [Elaboración Propia].

A.2.1.2 Dispositivo receptor

El dispositivo receptor está contenido en una caja de plástico, y mediante prensa estopa ingresan a la caja los cables de alimentación, ethernet, antena LoRa cable del sensor de temperatura y humedad relativa, de tal manera que el dispositivo pueda permanecer en el exterior sin ser afectado por la lluvia. En su interior se encuentra la Raspberry Pi 3B +, a la cual se conectan el chip LoRa y el sensor de temperatura y humedad relativa. En la figura A.3. se observa el dispositivo receptor.

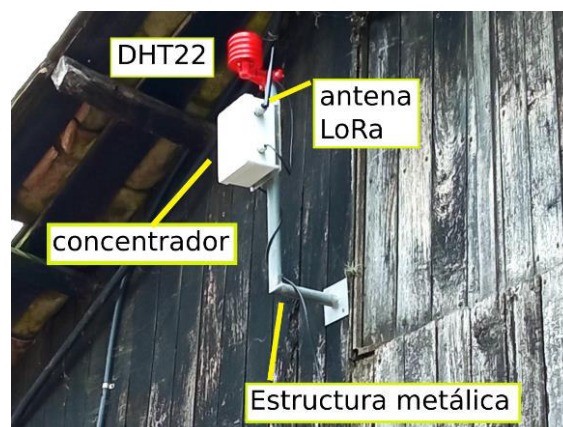


Figura A.3. Dispositivo receptor [Elaboración Propia].

A.2.2. Software

A nivel de software el dispositivo remoto está programado en una variante de C, mediante la IDE de Arduino. En cuanto al dispositivo remoto se emplea un conjunto de los siguientes recursos de programación: *python*, *JavaScript*, *CSS*, *HTML*, *SQLite* y comandos nativos del sistema operativo *Raspbian* para la programación de la ejecución de tareas. El programa principal está desarrollado en Python, este se encarga de ejecutar la lectura de los sensores, el almacenamiento de datos, permitir que el servidor corra en la red local de forma que un computador pueda acceder a la interfaz desde cualquier punto de esta red.

La interfaz de usuario almacenada en un espacio de memoria de la Raspberry emplea JavaScript y su librería JQuery para realizar peticiones al programa principal de esta forma puede obtener los datos contenidos en la raspberry para mostrarlos, permite cambiar parámetros como el área de la cerca virtual, ordenar el reinicio del receptor. Además, permite desplegar un mapa de la zona de la hacienda mediante la herramienta Open Layers que se puede ejecutar sobre dispositivos que emplea HTML5 como la mayoría de los navegadores web.

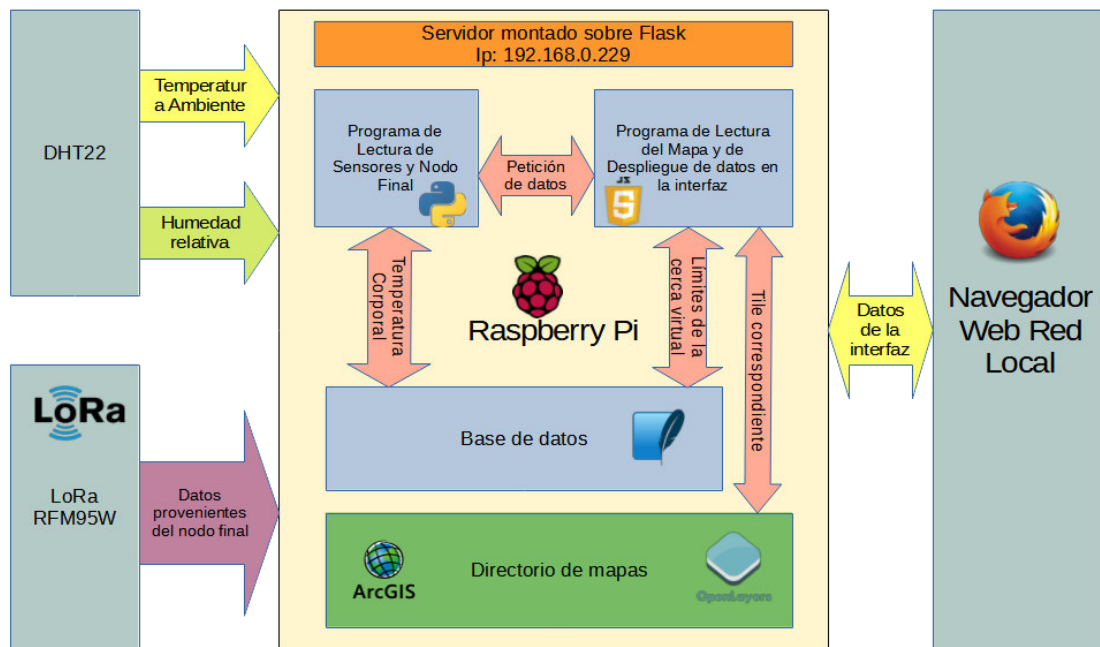


Figura A.4. Arquitectura de software en el receptor [Elaboración Propia].

A.3. GUIA PARA EL USO DEL DISPOSITIVO

A.3.1. Dispositivo Remoto

En primer lugar, se debe encender el dispositivo mediante el interruptor ubicado en el PCB como se observa en la figura. Una vez encendido el dispositivo automáticamente inicia con el proceso de conexión con los satélites, la lectura de los sensores y el envío de datos programado.

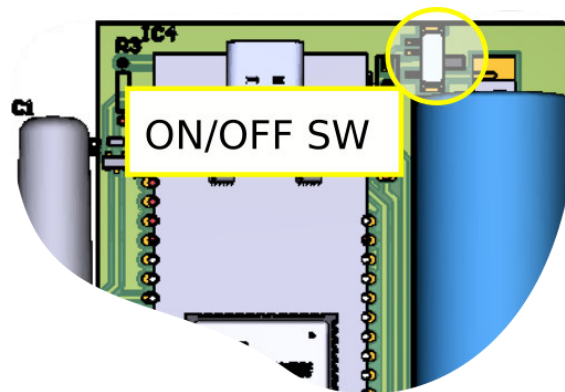


Figura A.5. Encendido y apagado del dispositivo remoto [Elaboración Propia].

Después se colocará la tapa en el dispositivo como se observa en la figura, para fijar los componentes se requiere de cinco tornillos M3 de una pulgada, y 2 tornillos M3 de $\frac{3}{4}$ ".

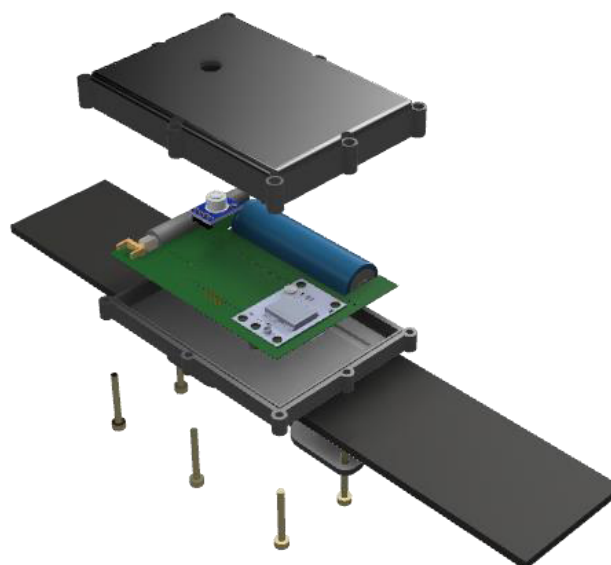


Figura A.5. Ensamblaje del dispositivo remoto o nodo final [Elaboración Propia].

Después de ensamblado el dispositivo, se deberá colocar en el cuello de la vaca, regulando la longitud del collar y cerrando este mediante el broche, observado en la figura.

A.3.3. Interfaz de Usuario

La interfaz de usuario se ejecuta desde el navegador web, la interfaz en esta versión no es responsiva y está diseñada para ejecutarse desde una computadora, siempre que esta esté conectada al mismo router que el dispositivo receptor. Para ejecutarse se requiere que el dispositivo receptor esté conectado a la fuente de alimentación principal. En la barra de búsqueda del navegador se debe colocar la siguiente dirección configurada por defecto.

192.168.0.229:5000

La pantalla principal, figura A.5, contiene una barra de menú desde la cual se permite ordenar el apagado o reinicio del sistema, configurar las características de la cerca virtual y obtener información acerca del prototipo. Por otro lado, contiene dos barras laterales informativas, en la barra izquierda se puede observar datos relacionados con el dispositivo remoto, mientras que en la derecha esta la información del sensor de temperatura y humedad relativa instalado junto con el dispositivo receptor. Finalmente, en la parte central se encuentra el mapa de la zona donde se puede determinar la posición del dispositivo remoto y la cerca virtual.

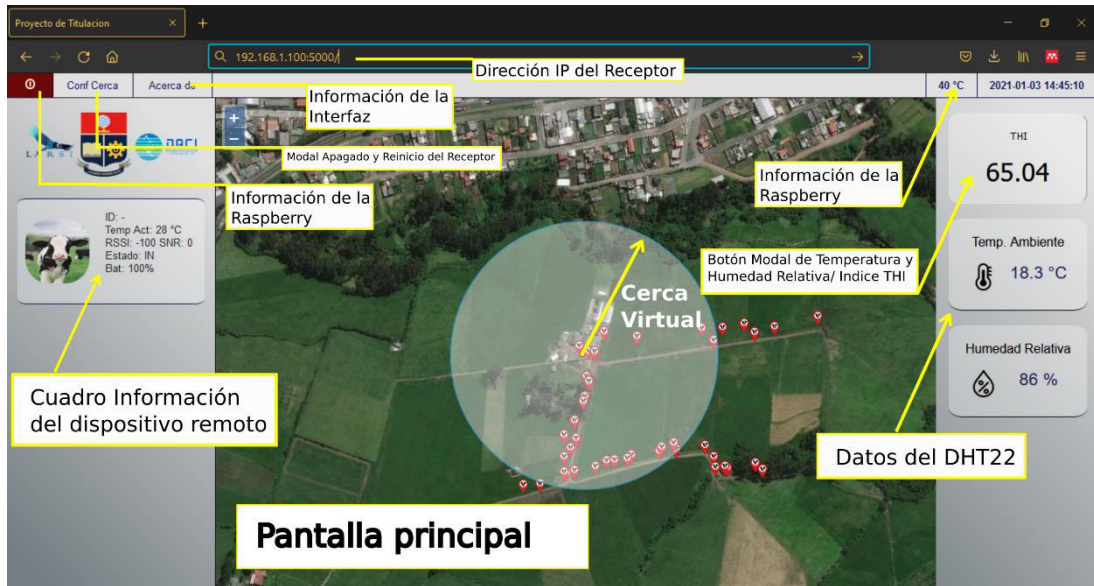


Figura A.5. Interfaz pantalla principal [Elaboración Propia].

Dentro de la barra derecha se encuentran los datos obtenidos desde el sensor de temperatura y humedad relativa, además, el THI calculado a partir estos datos el cual si presenta un resultado superior a 68 significa que el ganado podría sufrir de estrés térmico. El indicador de THI funciona además como un botón que abre un modal que contiene una gráfica de la variación de temperatura y humedad relativa de las últimas 24 horas como se observa en la figura A.6.

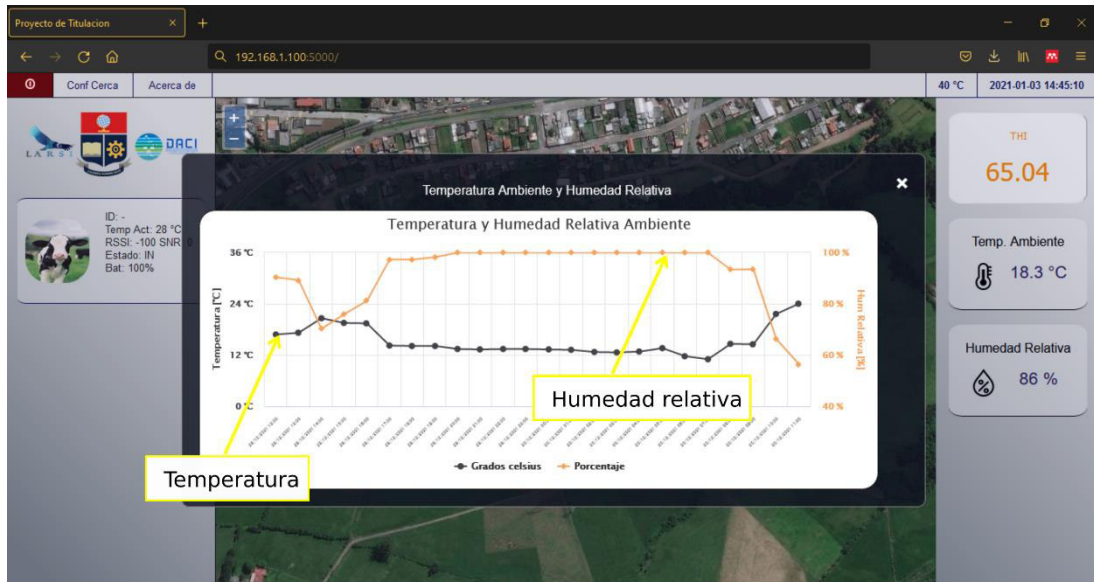


Figura A.6. Modal de gráficas de temperatura y humedad relativa [Elaboración Propia].

Dentro de la barra de menú esta la opción de controlar el apagado y reinicio del dispositivo remoto, como se observa en la figura A.7, para cerrar el menú basta con presionar *NO* en cualquiera de las dos opciones.

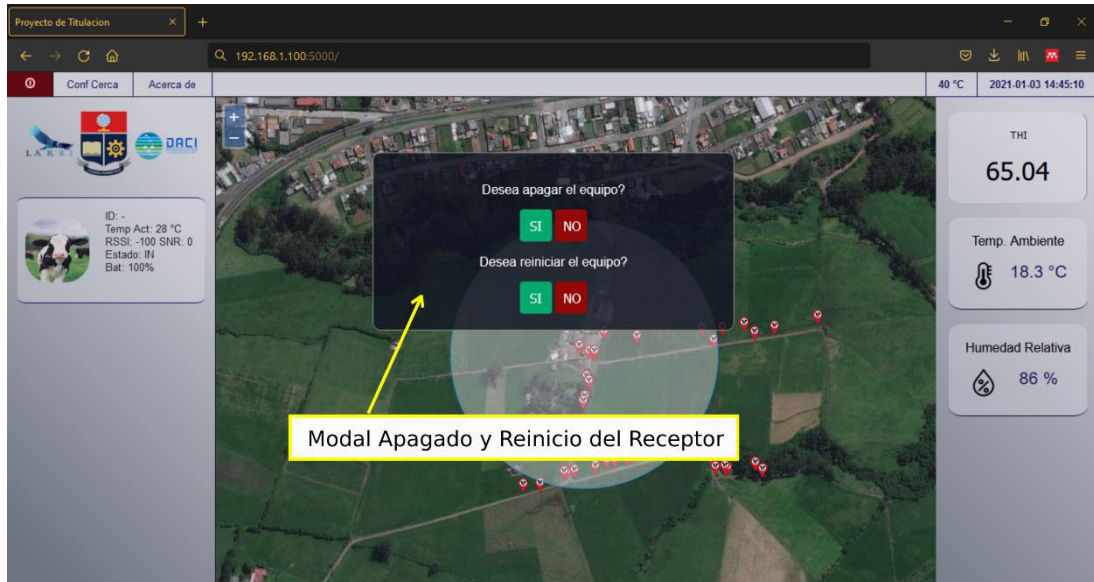


Figura A.7. Modal de apagado y reinicio del sistema [Elaboración Propia].

Otro de los componentes del menú es la opción de configurar los parámetros de la cerca que en este caso es un círculo, figura A.8, desde aquí se puede manipular el centro de la ubicación de la cerca y el radio de esta.

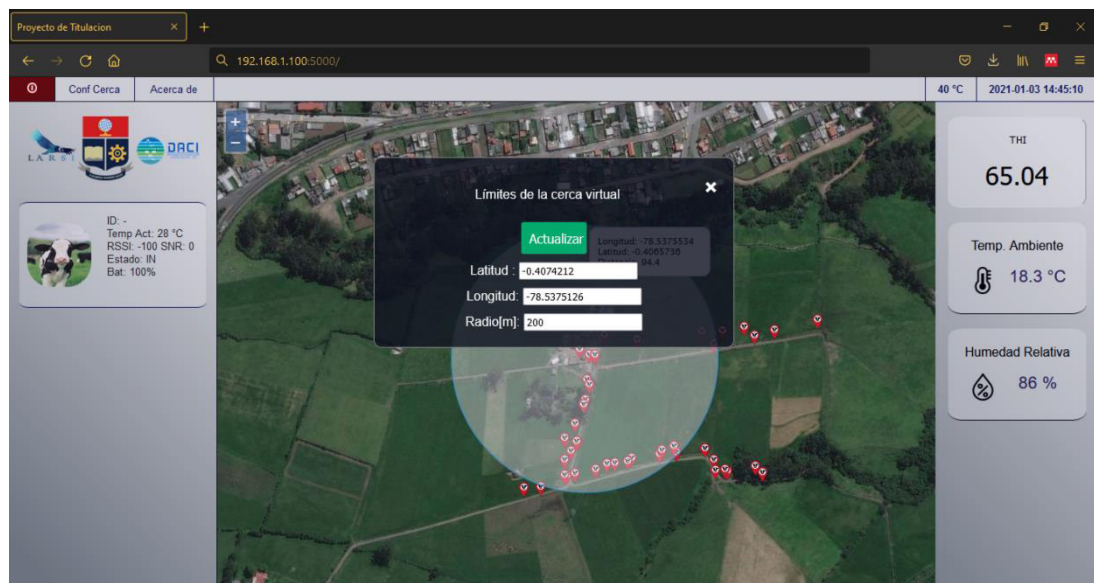


Figura A.8. Modal de configuración de la cerca virtual [Elaboración Propia]

A.4. RECOMENDACIONES DE USO

- Asegúrese que el dispositivo receptor se encuentre conectado a la red eléctrica.
- Asegúrese que el computador en el cual se va a abrir la interfaz esté conectado en la misma red que el dispositivo receptor, recuerde que el dispositivo tiene asignada la dirección IP fija
- Asegúrese que el dispositivo remoto está bien sujeto alrededor del cuello del animal.

ORDEN DE EMPASTADO