UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA:

"DISEÑO DE UN PROTOTIPO PARA LA DETECCIÓN DEL PERIODO DE FERTILIDAD (CELO) ENFOCADO A GANADO VACUNO, USANDO TECNOLOGÍA LPWAN PARA LA EMPRESA MELUA CIA. LTDA."

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN TELECOMUNICACIONES

AUTOR: GÓMEZ SÁNCHEZ ALEXANDER ROBERTO

DIRECTOR: MSC. LUIS EDILBERTO SUÁREZ ZAMBRANO

Ibarra-Ecuador

2025

UNIVERSIDAD TECNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA
AUTORIZACIÓN DE USO Y PUBLICACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

IDENTIFICACIÓN DE LA OBRA

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

DATOS DEL CONTACTO			
CÉDULA DE IDENTIDAD	1003945910		
APELLIDOS Y NOMBRES	Gómez Sánchez Alexander Roberto		
DIRECCIÓN	Cotacachi, Imantag-Colimbuela		
E-MAIL	argomezs@utn.edu.ec		
TELÉFONO FIJO	x	**TELÉFONO MÓVIL**	0939294693

DATOS DE LA OBRA	
TÍTULO	"Diseño de un prototipo para la detección del periodo de fertilidad (celo) enfocado a ganado vacuno, usando tecnología LPWAN para la empresa MELUA CIA. LTDA."
AUTOR	Gómez Sánchez Alexander Roberto
FECHA	24-02-2025
PROGRAMA	Pregrado
TÍTULO	Ingeniero en Telecomunicaciones
DIRECTOR	Msc. Luis Edilberto Suárez Zambrano

CONSTANCIAS

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

Ibarra, a los 25 días del mes de febrero de 2025

EL AUTOR:

Gómez Sánchez Alexander Roberto

CERTIFICACIÓN DEL DIRECTOR DE TRABAJO DE INTEGRACIÓN CURRICULAR

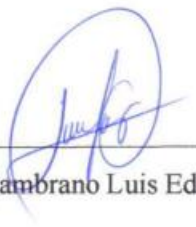
Ibarra, 25 de febrero de 2025

ING. SUÁREZ ZAMBRANO LUIS EDILBERTO, MSC

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

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



Ing. Suárez Zambrano Luis Edilberto, Msc

C.C 100230429-1

DEDICATORIA.

Esta tesis está dedicada, en primer lugar, a mis padres, Luis Gómez y María Sánchez, cuyo apoyo incondicional, sacrificio y esfuerzo han sido fundamentales para que pudiera alcanzar mis metas. Este logro es, en gran parte, suyo.

A mis hermanos, por ser mi fuente constante de inspiración y motivación para seguir adelante en cada paso de este camino.

A mi novia, quien ha estado a mi lado durante todo este proceso académico, apoyándome tanto en los momentos de éxito como en los de dificultad, y brindándome sus sabios consejos para lograr esta meta.

Finalmente, dedico este trabajo a mi hija, quien está por llegar, y quien ha sido mi principal fuente de inspiración y motivación para continuar con mis estudios. Te lo dedico con todo mi amor y gratitud.

Alexander Roberto Gómez Sánchez

AGRADECIMIENTOS

En primer lugar, agradezco a Dios por guiarme por el buen camino, otorgándome sabiduría, salud y fortaleza para alcanzar esta meta.

A mis padres, Luis Gómez y María Sánchez, por todo el sacrificio realizado, el tiempo que me han dedicado y los valores que me han transmitido, los cuales me han permitido ser la persona que soy hoy.

A mis hermanos, Steev y Edison, quienes han estado a mi lado durante todo este proceso, brindándome su apoyo incondicional y ayudándome en cada etapa de este camino.

A mi novia, Fátima Fernández, por su constante apoyo y por ofrecerme sabios consejos que me han ayudado a superar los momentos más difíciles de este proceso.

A mis amigos, Dylan, Jácome y David, por hacer que mi estancia en la universidad fuera más amena y por siempre estar presentes en los momentos de alegría y de desafío.

Finalmente, expreso mi más sincero agradecimiento al Msc. Luis Suárez, director de mi trabajo de titulación, por brindarme su valioso tiempo, orientarme y compartir sus conocimientos, lo que me permitió superar los desafíos presentados durante el desarrollo de este proyecto.

Alexander Roberto Gómez Sánchez

INDICE DE CONTENIDO

IDENTIFICACIÓN DE LA OBRA 2

DEDICATORIA. 5

AGRADECIMIENTOS 6

INDICE DE CONTENIDO 7

INDICE DE FIGURAS 14

INDICE DE TABLAS 18

RESUMEN 19

ABSTRACT 20

CAPITULO I: ANTECEDENTES 21

1.1 Tema 21

1.2 Problema 21

1.3 Objetivo 22

1.3.1 Objetivo General. 22

1.3.2 Objetivos Específicos. 22

1.4 Alcance 23

1.5 Justificación. 25

CAPITULO II: MARCO TEORICO 28

2.1 La ganadería en el Ecuador. 28

2.2 Producción de leche en el Ecuador. 29

2.3 Ganado bovino. 30
2.4 Signos de celo. 30
2.4.1 Incremento de la actividad. 31
2.4.2 Mugido. 32
2.4.3 Monta. 32
2.5 Métodos de detección de celo. 33
2.5.1 Método del parche. 34
2.5.2 Método visual. 34
2.5.3 Método tecnológico. 35
2.6 Internet de las cosas “IoT”. 35
2.6.1 Arquitectura IoT. 36
2.6.2 Ecosistema IoT. 36
2.7 Sistema embebido. 37
2.7.1 Componentes de hardware. 37
2.7.2 Componentes de Software. 38
2.7.3 Placas electrónicas de desarrollo. 38
2.7.4 Sensores 39
2.8 Tecnología LPWAN. 39
2.8.1 LoRa. 40
2.8.1.1 Arquitectura LoRa. 40

2.8.2 Sigfox. 41
2.8.2.1 Arquitectura Sigfox. 41
2.8.3 NB-IoT 42
2.8.3.1 Arquitectura NB-IoT. 43
2.9 Metodología de investigación. 44
2.9.1 Requerimientos. 45
2.9.2 Diseño. 45
2.9.3 Implementación. 46
2.9.4 Verificación. 46
2.9.5 Mantenimiento. 46
CAPITULO III: DISENO E IMPLEMENTACION 47
3.1 Situación actual. 47
3.1.1 Hacienda La María. 48
3.1.2 Ubicación Geográfica. 49
3.2 FASE 1: Especificación de requerimientos. 51
3.2.1 Nomenclatura de requerimientos. 51
3.2.2 Stakeholders. 53
3.2.3 Requerimientos de los Stakeholders. 53
3.2.4 Requerimientos del sistema. 55
3.2.5 Requerimientos de arquitectura. 57

3.3 FASE 2: Diseño 59
3.3.1 Arquitectura del sistema. 59
3.3.2 Elección de componentes de hardware y software. 60
3.3.2.1 Procesador. 61
3.3.2.2 Selección de sensor acelerómetro. 63
3.3.2.3 Selección de tecnología LPWAN. 65
3.3.2.4 Selección de modulo Lora. 66
3.3.2.5 Selección del sistema embebido 68
3.3.2.6 Selección de sistema operativo del Gateway 69
3.3.2.7 Selección de software de procesamiento de datos 70
3.3.2.8 Selección de software de almacenamiento de datos 71
3.3.2.9 Software de visualización de datos 72
3.3.3 Calculo para la fuente de alimentación. 72
3.3.3.1 Consumo total del sistema 73
3.3.3.2 Tiempo de funcionamiento de la batería 75
3.3.3.3 Cálculo del tiempo de carga de la batería 75
3.3.4 Diagrama de bloques del sistema 76
3.3.5 Diseño del nodo central 77
3.3.5.1 Diagrama nodo central 78
3.3.5.2 Diagrama de bloques del nodo central. 78

3.3.5.3 Diagrama de flujo del nodo central 80
3.3.6 Diseño del nodo cliente 81
3.3.6.1 Diagrama nodo cliente 81
3.3.6.2 Diagrama de bloques del nodo cliente. 82
3.3.6.3 Diagrama de flujo del nodo cliente. 83
3.3.6.4 Diseño de pcb del nodo cliente 84
3.3.7 Cálculo de comunicación LoRa. 87
3.3.7.1 Pérdidas de propagación en espacio libre 87
3.3.7.2 Cálculo de potencia de recepción 88
3.3.7.3 Relación Señal-Ruido 89
3.3.7.4 Capacidad del canal 90
3.4 FASE 3: Implementación. 91
3.4.1 Instalación y configuración del servidor. 91
3.4.2 Implementación del nodo central 95
3.4.2.1 Instalación y configuración del servicio MQTT 95
3.4.2.2 Instalación y configuración de NODE-RED 97
3.4.2.3 Instalación y configuración de MySQL. 101
3.4.2.4 Programación del nodo central 102
3.4.3 Implementación del nodo cliente 108
3.4.3.1 Construcción de la placa para el nodo cliente 109

3.4.3.2 Programación del nodo cliente..... 113
3.4.4 Procesamiento de datos..... 119
3.4.4.1 Diagrama de bloques en node red..... 119
3.4.4.2 Lectura y envío de datos a la base de datos. 120
3.4.4.3 Bloque de alerta de detección de celo..... 124
3.4.4.4 Creación de datasets..... 125
3.4.4.5 Entrenamiento del modelo de predicción 127
CAPITULO IV: ANALISIS DE RESULTADOS..... 133
3.5 FASE 4: Verificación..... 133
3.5.1 Funcionamiento de los servicios..... 133
3.5.2 Análisis del espectro de la comunicación LoRa. 135
3.5.3 Análisis del protocolo MQTT..... 136
3.5.4 Datos de animal sin presencia de celo. 139
3.5.5 Monitore de animal con presencia de celo..... 140
3.5.6 Monitoreo y alerta de la batería del nodo cliente..... 141
3.5.7 Evaluación del modelo de predicción..... 142
3.6 Análisis de resultados 144
CONCLUSIONES..... 150
RECOMENDACIONES..... 152
REFERENCIAS..... 153

ANEXOS 162

INDICE DE FIGURAS

Figura 1. Arquitectura IoT propuesta. 24

Figura 2. Aceptación de monta 33

Figura 3. Arquitectura LoRa. 41

Figura 4. Arquitectura de una red Sigfox. 42

Figura 5. Arquitectura NB-IoT. 44

Figura 6. Ubicación geográfica de la hacienda La María. 49

Figura 7. Distancia entre el nodo central y el nodo cliente. 51

Figura 8. Arquitectura general del proyecto. 60

Figura 9. Procesador ESP32. 62

Figura 10. Acelerómetro ADXL 345. 64

Figura 11. Microcontrolador esp32 LoRa v2. 69

Figura 12. Diagrama de bloques del sistema. 77

Figura 13. Diagrama del nodo central. 78

Figura 14. Diagrama de bloques del nodo central. 79

Figura 15. Diagrama de flujo del nodo central. 80

Figura 16. Diagrama nodo cliente. 82

Figura 17. Diagrama de bloques del nodo cliente. 83

Figura 18. Diagrama de flujo del nodo cliente. 84

Figura 19. Esquemático del nodo cliente. 85

Figura 20. Diseño del circuito impreso del nodo cliente. 86

Figura 21. Circuito en 3D del nodo cliente. 86

Figura 22. Configuración del servidor en AWS. 92

Figura 23. Resumen de las configuraciones del servidor. 93

Figura 24. Creación y configuración de la IP elástica. 94

Figura 25. Configuraciones del firewall de AWS..... 95

Figura 26. Instalación del servidor mqtt. 96

Figura 27. Configuración del servidor mqtt..... 97

Figura 28. Instalación del servidor Node-Red 98

Figura 29. Configuración del servidor Node-red. 99

Figura 30. Estado del servicio de node-red..... 100

Figura 31. Interfaz gráfica de node-red. 101

Figura 32. Instalación de My SQL..... 101

Figura 33. Creación de usuario en My SQL. 102

Figura 34. Librerías del nodo central. 103

Figura 35. Parámetros de la comunicación LoRa_Nodo_Central. 103

Figura 36. Credenciales wifi y mqtt..... 104

Figura 37. Función de reconexión mqtt. 105

Figura 38. Configuración del dispositivo LoRa..... 106

Figura 39. Función para la conexión WiFi y MQTT. 107

Figura 40. Funciones para el envío y recepción de datos del nodo central. 108

Figura 41. Circuito impreso en la baquelita..... 110

Figura 42. Eliminación del excedente de cobre. 110

Figura 43. Colocación de componentes en la placa. 111

Figura 44. Collar del prototipo. 112

Figura 45. Collar colocado en la vaca..... 113

Figura 46. Librerías del nodo cliente. 114

Figura 47. Parámetros LoRa para el nodo cliente. 114

Figura 48. Función de lectura de voltaje de la batería externa. 115

Figura 49. Inicialización de procesos del sistema. 117

Figura 50. Lectura y normalización de datos del sensor acelerómetro. 118

Figura 51. Preparación y envió de datos. 119

Figura 52. Bloques de envío de información a la base de datos. 120

Figura 53. Configuración del bloque mqtt. 121

Figura 54. Propiedades del bloque mqtt. 121

Figura 55. Configuración del bloque MySQL de node red. 122

Figura 56. Lectura y envío de información a la base de datos. 123

Figura 57. Código de alerta de desconexión. 123

Figura 58. Lectura del voltaje de la batería. 124

Figura 59. Diagrama de alerta de celo. 124

Figura 60. Código de alerta de celo. 125

Figura 61. Código para la conexión a la base de datos. 126

Figura 62. Código para el etiquetado de datos. 127

Figura 63. Vinculación de Drive con Google Colab. 128

Figura 64. Librerías usadas para el entrenamiento del modelo. 129

Figura 65. Creación de dataframe. 130

Figura 66. Normalización de datos del dataframe. 131

Figura 67. Separación de datos para test y train. 131

Figura 68. Creación y exportación de modelo de predicción. 132

Figura 69. Estado del servicio MQTT. 134

Figura 70. Estado del servicio de NODE-RED. 134

Figura 71. Estado del servicio de MY SQL. 135

Figura 72. Estado del servicio de GRAFANA. 135

Figura 73. Espectro de comunicación LoRa. 136

Figura 74. Paquetes MQTT analizados con wireshark. 137

Figura 75. Cabecera MQTT mensaje tipo "ping request". 137

Figura 76. Cabecera MQTT mensaje tipo "ping response". 138

Figura 77. Cabecera MQTT mensaje tipo "publish message". 139

Figura 78. Monitoreo de vaca sin presencia de celo. 140

Figura 79. Monitoreo de vaca con presencia de celo. 141

Figura 80. Monitoreo del estado de la batería. 142

Figura 81. Reporte del modelo de clasificación. 143

Figura 82. Matriz de confusión. 144

Figura 83. Comparación de resultados de detección de celo. 147

Figura 84. Alerta de detección de celo. 148

Figura 85. Alerta de batería baja. 148

INDICE DE TABLAS

Tabla 1. Nomenclatura usada para el establecimiento de requerimientos. 52

Tabla 2. Prioridad propuesta para los requerimientos..... 52

Tabla 3. Stakeholders (partes involucradas). 53

Tabla 4. Requerimientos de Stakeholders. (StSR)..... 54

Tabla 5. Requerimientos del sistema. (SySR)..... 55

Tabla 6. Requerimientos de hardware. (SRSH)..... 57

Tabla 7. Selección de sistema embebido para el nodo móvil. 61

Tabla 8. Características del microcontrolador ESP32 62

Tabla 9. Selección del sensor acelerómetro para el prototipo de detección de celo. 63

Tabla 10. Características del acelerómetro ADXL345. 64

Tabla 11. Elección de tecnología LPWAN. 66

Tabla 12. Elección de módulo LoRa. 67

Tabla 13. Características del módulo LoRa. 67

Tabla 14. Características del sistema embebido. 69

Tabla 15. Elección de hardware para el gateway. 70

Tabla 16. Elección de base de datos. 71

Tabla 17. Elección del software de visualización. 72

Tabla 18. Consumo eléctrico del nodo cliente. 73

Tabla 19. Control de celo hacienda Melua. 145

Tabla 20. Detección de celo mediante el prototipo electrónico. 146

RESUMEN

El presente trabajo de titulación aborda la problemática existente en la Hacienda La María, ubicada en la ciudad de Cotacachi, cuya principal actividad económica es la producción de leche. En esta hacienda, la detección del celo no se supervisa de manera óptima, lo que genera pérdidas económicas considerables. Por esta razón, se propone el diseño y desarrollo de un prototipo electrónico para la detección del celo, el cual monitorizará a los animales de manera continua, garantizando un control más efectivo del ciclo reproductivo.

Este sistema no solo optimiza la detección del celo, sino que también contribuye al bienestar del animal, ya que elimina la necesidad de que los trabajadores intervengan directamente, minimizando las molestias para el ganado. La detección del celo se llevará a cabo mediante el uso de tecnología avanzada, implementando un diseño electrónico y un protocolo de comunicación de largo alcance, como LPWAN (Low Power Wide Area Network).

Además, se utilizarán modelos de *machine learning* para asegurar que la detección del celo sea precisa y eficiente, permitiendo un control más efectivo de la reproducción del hato ganadero de la hacienda. Este prototipo proporcionará una solución innovadora y tecnológica para mejorar los procesos reproductivos en la ganadería.

ABSTRACT

This thesis addresses the issue present at Hacienda La María, located in the city of Cotacachi, whose main economic activity is milk production. In this hacienda, the detection of estrus (heat) is not optimally supervised, leading to significant economic losses. Therefore, the proposal is to design and develop an electronic prototype for estrus detection, which will continuously monitor the animals, ensuring more effective control of the reproductive cycle.

This system not only optimizes estrus detection but also contributes to the animals' welfare by eliminating the need for workers to directly intervene, minimizing disturbances to the livestock. Estrus detection will be carried out using advanced technology, incorporating an electronic design and a long-range communication protocol, such as LPWAN (Low Power Wide Area Network).

Additionally, machine learning models will be utilized to ensure precise and efficient estrus detection, enabling better control over the reproduction of the herd at the hacienda. This prototype will provide an innovative and technological solution to improve reproductive processes in livestock farming.

CAPITULO I: ANTECEDENTES

1.1 Tema

Diseño de un prototipo para la detección del periodo de fertilidad (celo) enfocado al ganado bovino, usando tecnología LPWAN para la empresa MELUA CIA. LTDA.

1.2 Problema

En el caso del ganado bovino, es fundamental contar con un adecuado manejo y supervisión de la reproducción en el hato ganadero para mejorar la producción lechera. La detección del periodo de fertilidad es un aspecto importante a considerar para mejorar el manejo reproductivo (Eerdenburg, 2018). Sin embargo, este tema a menudo no recibe la importancia necesaria. Estudios realizados por Frank Eedenburg indican que la tasa de detección del celo es inferior al 50%. Además, se ha demostrado que más del 70% de los celos ocurren entre las 6 p.m. y las 6 a.m., lo cual dificulta su detección por parte de los trabajadores, ya que ocurre fuera de su horario laboral. Como resultado, se generan pérdidas económicas en la hacienda y se afectan las tasas de producción de leche.

Actualmente, en la hacienda LA MARIA, que pertenece a la empresa MELUA CIA LTDA, se utiliza la inseminación artificial como método de manejo reproductivo para todas las hembras bovinas en el hato ganadero. Sin embargo, la falta de detección de celos dentro del hato genera considerables pérdidas económicas en la ganadería lechera y cárnica (Senger, 1994).

El método de detección del periodo de fertilidad en la hacienda se realiza de manera visual, lo que implica una observación visual. En la mayoría de los casos, los celos no son detectados debido a factores como: poco tiempo dedicado a la detección visual, mala interpretación de los eventos que marcan el inicio del periodo de fertilidad, gran cantidad de ganado vacuno, mayor porcentaje de celos que ocurren fuera del horario de trabajo.

Por lo tanto, es importante desarrollar un método tecnológico más eficiente para la detección del periodo de fertilidad (celo), que permita recopilar datos como el movimiento o el ángulo de inclinación de las vacas, que son los síntomas más comunes durante el celo. Estos datos se enviarán de forma inalámbrica desde el nodo sensor al nodo base, donde serán procesados para enviar una alerta al teléfono del mayordomo o encargado de las vacas.

1.3 Objetivo

1.3.1 Objetivo General.

Desarrollar un prototipo electrónico de monitoreo del comportamiento que presenta el ganado bovino en estado de celo, usando tecnología LPWAN, para disminuir la perdida en la detección de los periodos de celo en la hacienda LA MARIA.

1.3.2 Objetivos Específicos.

- Realizar una revisión teórica de los parámetros que están involucrados en la propuesta de diseño del prototipo mediante una consulta bibliográfica, con el fin tener un mejor concepto de los temas que componen el desarrollo del prototipo.
- Recopilar información sobre el estado actual de la hacienda, centrándose en los porcentajes de periodos de fertilidad perdidos, mediante la realización de encuestas al personal encargado de la detección del celo.
- Diseñar el prototipo del sistema de detección del celo que permita enviar alertas cuando empiece el periodo de fertilidad de la vaca, con el fin de brindar una pronta atención y realizar la inseminación artificial.
- Realizar pruebas que demuestren la funcionalidad del sistema y comparar los resultados obtenidos del prototipo con los registros propios de la hacienda.

1.4 Alcance

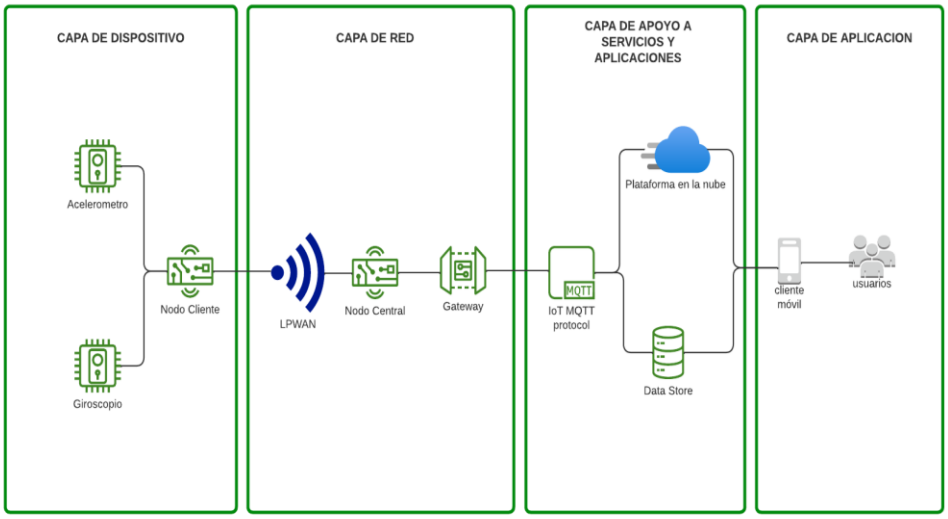
El objetivo de este proyecto es diseñar un prototipo electrónico para mejorar la detección del celo en el hato ganadero de la hacienda LA MARIA, perteneciente a la empresa MELUA CIA. LTDA. El diseño del prototipo se basa en una arquitectura de cuatro capas. En la capa de dispositivos, se encuentran los componentes fundamentales para la detección del periodo de fertilidad en el ganado bovino. Estos dispositivos incluyen sensores especializados y una placa de desarrollo, encargada de recibir y procesar los datos provenientes de los sensores.

Los sensores están diseñados para capturar y medir variables claves relacionadas con el celo en las vacas, como cambios en la temperatura corporal, comportamiento y niveles hormonales. Estos sensores se colocan estratégicamente en el ganado bovino para recopilar información precisa y confiable sobre su estado reproductivo. En la capa de red, se emplea la tecnología LPWAN (Low-Power Wide Area Network) para establecer la comunicación entre los dispositivos y transmitir los datos recolectados por los sensores. En esta capa, se utilizan dispositivos y módulos compatibles con LPWAN, como gateways y nodos, que actúan como puntos de acceso y enlaces de comunicación para los sensores. Estos dispositivos se encargan de recopilar los datos generados por los sensores y enviarlos a través de la red LPWAN a la capa superior para su procesamiento y análisis. En la capa de apoyo a servicios y aplicaciones, se utilizan plataformas tanto locales como en la nube para brindar soporte y funcionalidades adicionales al sistema. Estas plataformas actúan como intermediarios entre la capa de red y la capa de aplicación, facilitando la gestión de datos, el almacenamiento y el procesamiento de información. Además, proporcionan herramientas y servicios adicionales que mejoran la eficiencia y la usabilidad del prototipo de detección del periodo de fertilidad del ganado bovino. En la capa aplicación, se implementa la lógica y las funcionalidades necesarias para detectar el

periodo de celo en las vacas. En caso de detectar el celo, se genera una alerta que puede ser enviada al vaquero a través de diferentes medios, como mensajes de texto, notificaciones en una aplicación móvil o correos electrónicos., conforme a lo descrito en la recomendación ITU-T Y.2060. La figura 1 proporciona una representación visual detallada de cada una de las capas propuestas para la implementación del prototipo del sistema.

Figura 1.

Arquitectura IoT propuesta.



Nota. Arquitectura basada en la recomendación de la ITU-T Y.2060.

El desarrollo, diseño y funcionalidad del prototipo se realizará utilizando la metodología cascada, que implica seguir una secuencia de pasos para asegurar una planificación y desarrollo eficiente del prototipo, así como una evaluación rigurosa de su funcionalidad y rendimiento.

En primer lugar, se obtendrá información sobre el estado actual de la hacienda con relación al porcentaje de preñes de las vacas. Esto permitirá tener una visión clara de la situación actual y establecer una línea base para evaluar la efectividad del prototipo.

A continuación, se realizará un análisis exhaustivo de los parámetros que influyen en la detección del celo de las vacas. Se investigarán factores como el comportamiento de las vacas en celo, los síntomas físicos que presentan y otros indicadores relevantes. Esta información será crucial para definir los requisitos del prototipo.

Luego, se analizarán y evaluarán los diferentes sensores y placas de desarrollo disponibles en el mercado. Se investigarán las características técnicas de cada componente y se seleccionarán aquellos que mejor se adapten a las necesidades del prototipo, considerando factores como la precisión de detección, la duración de la batería y la capacidad de transmisión de datos.

Con base en los requisitos recopilados y la elección de los componentes, se procederá al diseño del prototipo. Se definirá la arquitectura del sistema, la disposición de los sensores, la comunicación inalámbrica y otros aspectos técnicos relevantes.

Una vez diseñado el prototipo, se realizarán pruebas en un entorno controlado para evaluar su funcionalidad. Se verificará que cumpla con los requisitos establecidos y se registrarán y analizarán los resultados obtenidos.

Como etapa final, se comprobará la funcionalidad del hardware y software realizando pruebas en el 10% de las vacas que presenten el celo durante el mes. Posteriormente, se compararán los resultados obtenidos del prototipo con los resultados proporcionados por la hacienda. El objetivo es evidenciar que los resultados del prototipo son más eficientes que el método visual utilizado actualmente en la hacienda.

1.5 Justificación.

Internet de las cosas (IoT) representa la evolución del internet, según Quiñonez Muñoz (2019) es la interrelación inteligente entre objetos que pueden acceder a información, capturar,

almacenar y procesar datos para interactuar con las personas, o con otros sistemas para crear ambientes inteligentes. En este contexto, Salazar & Silvestre (2016) expresan que también puede ser una plataforma en donde se comunican electrónicamente y comparten información varios dispositivos. IoT se refiere a los objetos cotidianos que tienen una particularidad ya que cuentan con sensores, circuitos integrados y conectividad que les permite recolectar e intercambiar datos entre ellos, a estos objetos también se los llama dispositivos inteligentes porque deben ser capaces de realizar monitoreo, control, optimización y automatización.

Mediante la implementación del internet de las cosas (IoT), se busca el incremento de ganado lechero ya que según Osvaldo Barsky (1980) la incorporación de ganado puro, de gran pedigrí ha permitido un aumento sustancial en la producción de leche. Es por ello por lo que la producción de leche en el ganado está estrechamente relacionada con la detección temprana del celo, ya que así tendremos un aumento en la tasa de preñes lo cual se traduce en un aumento en la cantidad de vacas gestantes y a su vez en un mayor número de lactancia en el hato ganadero.

Los seres humanos y especialmente los profesionales del área pecuaria deben propender a evitar el sufrimiento innecesario de los animales destinados a producir productos para la alimentación humana, en países más desarrollados éste es un punto sobre el cual el consumidor ejerce mucha presión, exigiendo que se contemple dentro de la producción y comercialización aspectos relativos al bienestar animal, constituyendo un atributo que se conoce como calidad ética (Agrocalidad, 2020).

La implementación del prototipo está basada en el Articulo 25. Del bienestar animal, en donde se hace referencia a las condiciones de la infraestructura, transporte, alimentación y manejo de los animales, de tal modo que garanticen su comportamiento normal y su calidad de vida (Agrocalidad, 2012).

Una detección precisa del celo permite evitar situaciones de estrés y molestias innecesarias para las vacas, como intentos repetidos de inseminación cuando no están en celo. Al minimizar los tiempos de espera y asegurar una reproducción eficiente, se contribuye al bienestar animal y se promueve un manejo más adecuado de los animales.

CAPITULO II: MARCO TEORICO

En el presente capítulo, se realiza una revisión teórica acerca de los temas planteados. En donde se dará a conocer de manera detallada las características del ganado bovino, sus diferentes tipos y comportamiento que presenta al momento de iniciar el estado de celo. Además, se hablará acerca de los componentes electrónicos, tecnología LPWAN y placas de desarrollo, los cuales se integrarán para crear un sistema, diseñado para proporcionar alertas precisas de detección de celo al vaquero.

2.1 La ganadería en el Ecuador.

En Ecuador, la ganadería desempeña un papel económico crucial. Durante el período de 1985 a 2005, el sector agropecuario contribuyó en promedio con un 13% a la economía nacional. En 2008, la participación de este sector en el Producto Interno Bruto (PIB) fue del 10.7%, colocándolo en el segundo lugar después de la producción petrolera. (MAE, 2023)

La ganadería es una actividad económica importante en Ecuador, tanto en la producción de carne bovina como en la producción de leche. Según el MAGAP, Ecuador es autosuficiente para cubrir la demanda nacional de carne bovina, y produce alrededor de 6,15 millones de litros diarios de leche cruda, lo que representa una fuente de ingresos para casi 1,2 millones de personas y la industria láctea representa alrededor del 4% del PIB Agroalimentario del país.

El ganado bovino, en su mayoría mestizo, exhibe características distintivas derivadas de tres categorías de cruces:

Primero, se destaca la mezcla de razas criollas con aquellas específicamente adaptadas para la producción de carne, tales como el Brahmán, Santa Gertrudis y Charoláis. Esta combinación busca potenciar las cualidades de ambas razas, aprovechando la resistencia y

adaptabilidad de las criollas junto con las características de carne deseables de las razas especializadas en este aspecto.

En segundo lugar, se pueden observar cruces entre razas criollas y aquellas de doble propósito, como el Normando y el Pardo Suizo. Esta categoría busca alcanzar un equilibrio entre la producción de carne y leche, aprovechando las ventajas de ambas finalidades en un solo individuo.

Por último, se encuentra la combinación de razas criollas con aquellas especializadas en la producción de leche, como el Holstein, que generalmente se encuentran en climas fríos. Este cruce se enfoca en obtener individuos capaces de resistir condiciones adversas mientras mantienen altos niveles de producción láctea.

Estas prácticas de cruce, como señala Zambrano (2015), evidencian la diversidad y adaptabilidad del hato bovino, destacando la importancia de seleccionar cuidadosamente las razas para lograr los objetivos deseados en términos de producción y adaptabilidad a diferentes entornos.

2.2 Producción de leche en el Ecuador.

La llegada inicial de animales desde Europa, que incluía vacas destinadas tanto a la producción de leche como de carne, ocurrió en el segundo viaje de Cristóbal Colón en 1493. Estos animales inicialmente llegaron a Centroamérica y gradualmente se expandieron por el resto del continente. En Ecuador, este ganado llegó en el año de 1537 cuando el conquistador Sebastián de Benalcázar llevó animales desde Nicaragua para establecer producciones de leche y carne cerca de Quito y Guayaquil.

En 2017, la producción diaria de leche en Ecuador alcanzó los 5,135,405 litros. De este volumen, el 76.25% correspondió a la región de la Sierra, seguido por la Costa con el 19.66% y

el Oriente con el 4.05%. Durante ese año, se ordeñaron diariamente 856,164 vacas, de las cuales 550,596 estaban en la región de la Sierra, con una producción promedio diaria de 7.11 litros por vaca; 256,803 estaban en la Costa, con una producción promedio de 3.93 litros por vaca; y 48,515 se encontraban en la región del Oriente, con una producción promedio de 4.29 litros por vaca (INEC, 2017).

2.3 Ganado bovino.

Ganado bovino, o comúnmente conocido como ganado vacuno, son animales criados en la mayor parte por los humanos, con la finalidad de proveer materia prima tales como: carne, leche, cuero, entre otros subproductos.

Actualmente en el Ecuador, la ganadería contribuye a la economía del sector agropecuario del país, esto mediante el abastecimiento de productos cárnicos y lácteos, mismos que generan fuentes de trabajo en diferentes zonas geográficas del Ecuador (Falquez, 2017) .

Según (Armando, 2006) “El ganado bovino es considerado el pilar fundamental de la producción pecuaria en todas o casi todas las áreas del planeta gracias a sus peculiaridades en el tubo digestivo que les permiten transformar las materias vegetales en proteínas de alto valor biológico además de otras producciones importantes.”

2.4 Signos de celo.

El proceso de identificación de los signos de celo se puede realizar de dos maneras: primero, la observación visual de los cambios en el comportamiento de la vaca; segundo, mediante el uso de dispositivos electrónicos, a través de la cual se monitorean los cambios internos de la vaca.

Las vacas pueden presentar tres niveles de comportamiento de celo, niveles que son particularmente importantes para el apareamiento exitoso con los machos. Estos son la atracción,

la proceptividad y la receptividad. La atracción se mide por el grado en que una vaca evoca respuestas sexuales al macho. La respuesta depende tanto del olor que produce y de su sensibilidad, como de su receptividad, finalmente la receptividad es la predisposición de la hembra a aceptar el cortejo y la copula del macho. (Sepúlveda y Rodero, 2003, p. 500)

2.4.1 Incremento de la actividad.

Las vacas en estado de celo tienden a presentar un comportamiento sexual que las distingue del resto de vacas, al ser un grupo sexualmente activo, y se puede definir como característica distintiva la intranquilidad y el nerviosismo. Durante este comportamiento, la vaca se observa inquieta, el pastoreo y la alimentación en la mayoría de las ocasiones se ve interrumpido, debido al aumento de desplazamiento de la vaca (García, 2002).

La utilización de podómetros emerge como una herramienta valiosa para la verificación y cuantificación de los cambios en la actividad locomotora del ganado bovino durante el período per estral. Según Lehrer(1992), se observa un aumento significativo en la movilidad, que se manifiesta de manera notable de 8 a 16 horas antes del estro. Sin embargo, es crucial considerar que, según Varner(1994), el pico de actividad ocurre aproximadamente 4 horas antes del inicio del celo.

Es relevante destacar que, al igual que otros signos de estro, el patrón de movimiento registrado también está vinculado con el historial reproductivo y el estado de lactancia de la vaca. Se ha observado que el movimiento aumenta con el número de partos de la vaca, pero disminuye en presencia de la lactancia. Este hallazgo subraya la necesidad de una comprensión completa de los factores que influyen en la actividad locomotora para implementar estrategias de reproducción efectivas y adaptadas a las características individuales de cada animal.

2.4.2 Mugido.

El mugido de las vacas en estado de celo es un comportamiento importante para los encargados de la vigilancia y el control de las vacas, también conocidos como vaqueros, puesto que es un indicador clave de su estado reproductivo. Asimismo, el mugido es una forma de comunicación que usan las vacas para atraer a los machos y se distingue durante este estado porque pueden mugir con más frecuencia y a un tono más agudo, no obstante, este síntoma en algunos casos es causado por diferentes factores. Como indicó Pueyo (2017), el mugido de las vacas es uno de los cambios de comportamiento que pueden indicar que una vaca está en celo, pero no es un indicador preciso del celo, pues las vacas pueden mugir por diferentes razones, como hambre, sed, estrés o dolor.

2.4.3 Monta.

Típicamente, la vaca que se encuentra en estado de celo suele intentar montar a otras vacas que se encuentran en el mismo establo, siendo este un factor difícil de interpretar por el hecho de que al inicio del celo las vacas se montan unas a otras, siendo difícil para el observador identificar cuál vaca del hato se encuentra en celo. La monta es cuando una vaca en particular se queda inmóvil y acepta la monta de las demás, se puede decir que está en celo (Sepúlveda y Rodero, 2003).

Una vaca que permanece inmóvil cuando es montada es una señal primaria, clara e inequívoca de que está en celo. En la figura 2 podemos ver un ejemplo de monta entre dos vacas. A este comportamiento se lo conoce como la “monta estática” la cual indica la máxima intensidad sexual del ciclo estral. (Guáqueta, 2009).

Figura 2.

Aceptación de monta

Nota. Ilustración del proceso de aceptación de monta entre un toro y una vaca, tomado de “Manejo productivo”, por Casamú, s.f., (https://www.casamu.com.ar/manejo-productivo/)

2.5 Métodos de detección de celo.

Para hablar de los métodos de detección del celo en las vacas, es importante tener en cuenta que la detección adecuada del celo es esencial para lograr concepciones en el menor tiempo posible y mejorar la eficiencia reproductiva del hato. Existen varios métodos para detectar el celo en las vacas, desde la observación del comportamiento hasta el uso de dispositivos electrónicos y marcadores.

Cada método tiene sus ventajas y desventajas, y es importante elegir el método que mejor se adapte a las necesidades y recursos de cada productor. En general, la detección del celo es una tarea importante, que requiere atención y dedicación por parte del productor para lograr una buena eficiencia reproductiva en su hato.

2.5.1 Método del parche.

En la actualidad, existen variedades de detectores de actividad de monta, la cual consiste en parches que cambian de colores al realizar una frotación entre la vaca que está en celo y una compañera que se encuentra en el hato. Los parches son dispositivos que se colocan en la base de la cola y se activan cuando la vaca en celo es montada; estos dispositivos consisten “en una cápsula colocada en un parche de tela; con presión prolongada se libera un tinte o se produce una reacción química que hace que la capsula cambie de color” (Perry, 2021, p. 4).

2.5.2 Método visual.

Dentro del hato ganadero, el vaquero debe estar preparado para identificar ciertos comportamientos visuales que indican que una vaca está en celo; uno de los comportamientos a tomar en cuenta es el nerviosismo o la inquietud del animal en cuestión. Aquí se puede identificar por un andar excesivo y por los mugidos que la vaca produce, lo que es un signo de que le vaca está en búsqueda de un toro para la monta.

Otro punto para tomar en cuenta es la observación de un flujo vaginal transparente, dado que la vaca al estar en celo descarga una secreción espesa y transparente del cuello uterino. La secreción del mucus transparente cervical en forma de un hilo colgado, el cual tiene como origen la vulva o como pegotes en la cola y la cadera, que son una buena señal de que la vaca está en celo.

Igualmente, como comportamiento de celo se tiene la reunión de una parte del hato ganadero, pues el ganado que está en celo naturalmente busca otros animales en celo y forman un pequeño grupo, conocido como grupo sexualmente activo. En este punto, las vacas de este grupo suelen tener contacto entre sí, cada vez que un grupo pequeño de animales se reúne deben vigilarse buscando vacas que estén en celo.

2.5.3 Método tecnológico.

El tipo más reciente de detector de actividad de monta es un detector electrónico, pero esto conlleva una inversión económica mayor. Como beneficio de la implementación de dispositivos electrónicos se tiene una detección del celo más precisa. En este aspecto, los dispositivos electrónicos buscan censar las actividades mencionadas, como el incremento de actividad o la aceptación de la monta por parte de otras vacas. Uno de los dispositivos más usados es el podómetro, el cual permite monitorear el incremento de la actividad, por medio de sensores que permiten medir la distancia que se ha desplazado un animal.

El uso del podómetro permite contar el número de pasos que da la vaca. “Cuando está en celo, tiende a caminar más. No obstante, se recomienda combinar este parámetro con otros para lograr mejores resultados en la tasa de detección” (Agrovet Market, 2019, párr. 3).

2.6 Internet de las cosas “IoT”.

El concepto de Internet de las Cosas (IoT) fue introducido en 1999, por el innovador británico Kevin Ashton, cuyo propósito era describir un sistema en el que los objetos del entorno físico pudieran establecer conexiones a internet mediante el uso de sensores. Ashton acuñó este término para destacar la capacidad de conectar a la red mundial las etiquetas de identificación por radiofrecuencia (RFID) que se empleaban en las cadenas de suministro empresariales, para permitir el seguimiento y el conteo de productos sin requerir intervención humana. (Gallardo,2023)

En la actualidad, el concepto de IoT se ha generalizado para describir situaciones en las cuales la conectividad a internet y la capacidad de procesamiento se extienden a una amplia gama de objetos, dispositivos, sensores y artículos cotidianos. El IoT representa una transformación radical en la calidad de vida de las personas, abriendo una amplia gama de

nuevas oportunidades de acceso a datos y servicios en campos como la educación, la seguridad, la atención médica, el transporte, etc.

2.6.1 Arquitectura IoT.

La arquitectura IoT se basa en una jerarquía de capas que abarcan desde el nivel de los dispositivos individuales hasta la nube, pasando por redes de comunicación intermedias; cada capa cumple un papel fundamental en la transmisión de datos y la ejecución de aplicaciones. De la misma forma, se integra con protocolos de comunicación, servicios de nube y sistemas de análisis de datos para habilitar una amplia gama de aplicaciones IoT, desde el monitoreo de la salud hasta la gestión de ciudades inteligentes.

Hoy en día, cualquier dispositivo con capacidad para conectarse a otro es considerado IoT. Esta industria representa a nivel mundial un mercado de $742 billones de dólares y se estima en más de 10 billones el número de dispositivos IoT activos (Fernández, 2022).

2.6.2 Ecosistema IoT.

Se conoce como sistema IoT a un conjunto de diferentes elementos, en donde se tiene los actores o tecnología que están inmersos en el sistema IoT enfocados a un dominio en particular.

Según Memmet Yildhiz (2020), un ecosistema IoT típico se define como una solución integral que consta de sistemas simples o complejos que se integran por componentes y subcomponentes para llevar a cabo tareas simples o múltiples, trabajando de manera coherente y en armonía. En esta perspectiva, el concepto de ecosistema se asemeja a los mecanismos metafóricos de supervivencia y requisitos de integración similares a los que están presentes en los ecosistemas biológicos.

2.7 Sistema embebido.

Un sistema embebido se define como un sistema de computación especializado que tiene el propósito de ejecutar una función o un conjunto de funciones específicas dentro de un sistema mayor. Este tipo de sistema integra componentes de hardware y software que han sido optimizados para trabajar conjuntamente, asegurando un rendimiento eficiente y confiable en las tareas asignadas. A diferencia de los sistemas de uso general, los sistemas embebidos están destinados a tareas particulares y suelen estar sujetos a estrictas limitaciones en cuanto a recursos, como energía, memoria y capacidad de procesamiento.

2.7.1 Componentes de hardware.

Los componentes de hardware son los elementos físicos que constituyen el sistema, permitiendo su funcionamiento y la interacción con el entorno. Estos componentes son esenciales para la implementación y prueba de las funcionalidades del prototipo antes de su producción en masa. En un sistema electrónico, los componentes de hardware son las piezas fundamentales para que el sistema funcione de manera óptima. A continuación, se describen los principales componentes de hardware que se utilizarán para el desarrollo del prototipo:

- Microcontrolador
- Microprocesador
- Memoria
- Sensores
- Interfaces de comunicación
- circuitos de alimentación

2.7.2 Componentes de Software.

Los componentes de software son los elementos intangibles que proporcionan las instrucciones y la lógica necesarias para que el sistema funcione correctamente. En un sistema electrónico, los componentes de software son fundamentales para el control y la operación eficiente del hardware. A continuación, se describen los principales componentes de software que se utilizarán para el desarrollo del prototipo:

- Firmware
- Sistema Operativo
- Interfaces de Usuario
- Herramientas de Desarrollo

2.7.3 Placas electrónicas de desarrollo.

Las placas electrónicas de desarrollo son dispositivos compuestos por un microcontrolador programable al que se le dan instrucciones para realizar tareas específicas. Estas placas disponen de puntos de entrada y salida analógicos o digitales que permiten la integración de componentes externos. En la actualidad, las placas de desarrollo han adquirido gran relevancia en el desarrollo de proyectos de IoT. La elección de la placa de desarrollo más adecuada depende de las necesidades y requisitos del sistema IoT. (Arrow Electronics, 2018)

A continuación, se presentan algunas de las placas de desarrollo más reconocidas en el ámbito de Internet de las Cosas.

- ARDUINO
- ESP32
- RASPBERRY PI
- INTEL GALILEO

- NODE MCU

2.7.4 Sensores

Los sensores son dispositivos que se utilizan para medir magnitudes físicas, químicas o biológicas, para posteriormente transformarlas en señales eléctricas que serán procesadas por un sistema electrónico. Sus aplicaciones albergan un gran campo, pues actualmente todo proceso se ve en la necesidad de ser censado, para así poder tratar los datos y automatizar los procesos y actuar en tiempo real al presentarse algún inconveniente. (Guerra, 2023)

Los sensores tienen una gran importancia en el ámbito del IoT, ya que permiten recopilar y transmitir datos, para posteriormente ser procesados y darles una aplicación. En el caso de la ganadería los sensores permiten detectar el comportamiento del ganado vacuno, para así determinar ciertos aspectos, en donde uno de ellos es la detección del celo.

2.8 Tecnología LPWAN.

Los avances de la tecnología y con ello la necesidad de comunicar diferentes dispositivos que se encuentran a largas distancias han contribuido al desarrollo de soluciones IoT dirigidos a diversas industrias, aquí aparece el termino LPWAN, la cual abarca varias tecnologías orientadas a conectar dispositivos de bajo consumo a escenarios IoT. (Roque, 2020).

Cabe aclarar que el término LPWAN (red de área amplia de bajo consumo de energía) es relativamente nuevo, tanto así que para 2013 ni siquiera existía. Este concepto engloba las nuevas tecnologías con licencia y sin licencia que surgieron en ese momento, diseñadas específicamente para conectar dispositivos que transmiten datos a velocidades bastante bajas.

Las LPWAN representan una evolución clara en las redes de comunicación dirigidas a dispositivos de IoT. Estas redes se crearon con el propósito de conectar una gran cantidad de dispositivos, superando el alcance de las tecnologías convencionales y utilizando la menor

cantidad de recursos posible. Sin embargo, esto se logra a expensas de la velocidad de transmisión, lo que las hace ideales para interconectar sensores y dispositivos que requieren una transmisión de datos a baja velocidad, pero no son adecuadas para la transmisión de contenido de voz, audio o video. (Buitrago et al., 2019, p. 6)

A continuación, se da a conocer las soluciones más apropiadas para el desarrollo del prototipo planteado.

2.8.1 LoRa.

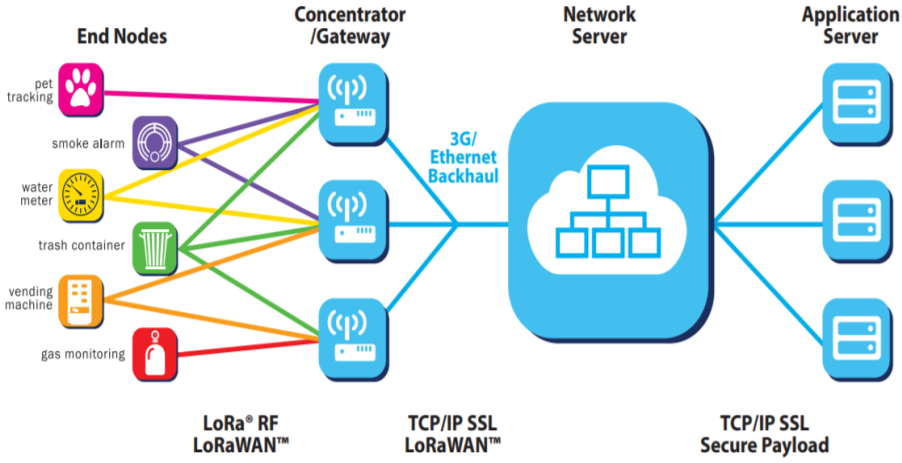
La tecnología LoRa está diseñada para proveer conectividad a larga distancia, permitiendo ser implementada en dispositivos que son alimentados con baterías, esto gracias al poco consumo de energía que tiene este tipo de soluciones. Además, usa técnicas de modulación por espectro extendido, siendo este tipo de modulación la responsable de brindar las características antes mencionadas, así también nos brinda una tolerancia alta a interferencias, haciendo de esta tecnología muy eficiente para soluciones IoT. (Alliance, 2020)

2.8.1.1 Arquitectura LoRa.

LoRa define una arquitectura de red que utiliza la topología de tipo estrella permitida por la capa física LoRa, además de una red en estrella con un servidor de red (Network Server) que recibe los datos de varios concentradores (también conocidos como Gateways en el contexto de LoRaWAN). En la figura 3 se observa más a fondo los componentes que definen una arquitectura LoRa en donde los dispositivos no están asociados a un concentrador/Gateway en particular, sino que sus mensajes pueden ser recibidos por cualquier Gateway de la red que se encarga de enviarlos al servidor de red central (Modernell, 2020).

Figura 3.

Arquitectura LoRa.



Nota. Se muestra la iteración de componentes en la arquitectura LoRa. Recuperado de https://productos-iot.com/lorawan-3/

2.8.2 Sigfox.

Sigfox es una tecnología de comunicación inalámbrica de baja potencia y área amplia (LPWAN) utilizada para transmitir datos a largas distancias con un bajo consumo de energía. Esta tecnología es independiente de las redes de telefonía y se usa para comunicaciones de baja velocidad que permiten la transmisión de pequeñas cantidades de datos; es ideal para aplicaciones de IoT que requieren conexiones a larga distancia y sensores con bajo consumo de energía. La tecnología Sigfox es ampliamente utilizada en diferentes campos, como la electrónica, la robótica y la medicina, y se pueden encontrar tutoriales y proyectos en línea para utilizarla en diferentes aplicaciones (Sigfox, 2021).

2.8.2.1 Arquitectura Sigfox.

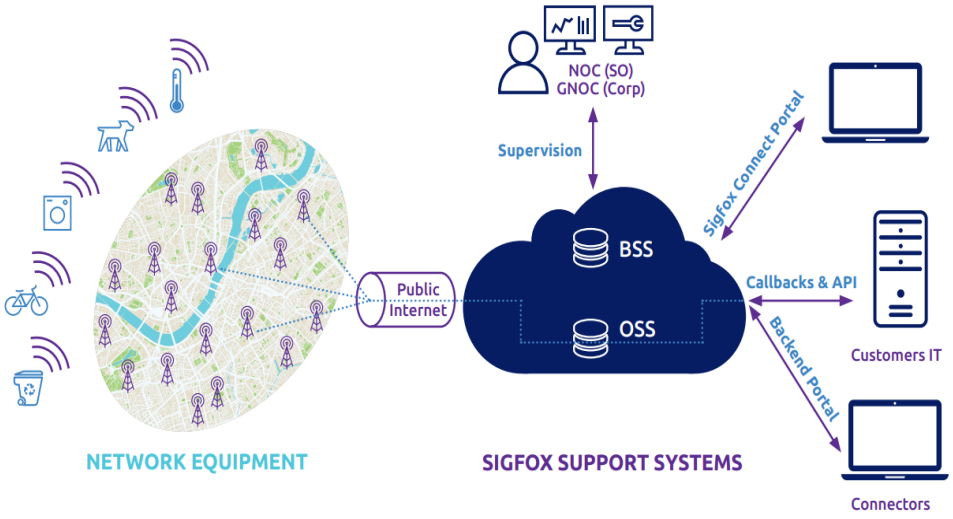
La red Sigfox tiene una arquitectura horizontal y delgada compuesta por dos capas principales. En la figura 4 se muestra los componentes de la arquitectura Sigfox en donde la capa

de Equipo de Red consiste esencialmente en estaciones base (y otros elementos, como antenas) encargadas de recibir mensajes de los dispositivos y transferirlos a los sistemas de soporte de Sigfox.

El Sistema de Soporte Sigfox es la segunda capa que constituye la red central encargada de procesar los mensajes y enviarlos, la cual proporciona también el punto de entrada a los diferentes actores del ecosistema (Sigfox, operadores, canales y clientes finales de Sigfox), a fin de interactuar con el sistema a través de interfaces de atención web o API (Universidad de la Empresa, 2019).

Figura 4.

Arquitectura de una red Sigfox.



Nota. Tomado de "Cómo funciona la red LPWAN Sigfox", por Descifrando Blockchain, 2018, (https://blockchainadministration.blogspot.com/2018/12/como-funciona-la-red-lpwan-sigfox.html)

2.8.3 NB-IoT

NarrowBand-Internet of Things (NB-IoT) es un sistema basado en estándares de tecnología de área amplia de baja potencia (LPWAN), desarrollado para habilitar una amplia

gama de dispositivos y servicios IoT. NB-IoT mejora significativamente la eficiencia energética de los dispositivos de los usuarios, la capacidad del sistema y la eficiencia del espectro, especialmente en áreas de cobertura profunda. Lo anterior permite una duración de batería de más de 10 años para una variedad de casos de uso (IPLOOK, 2022).

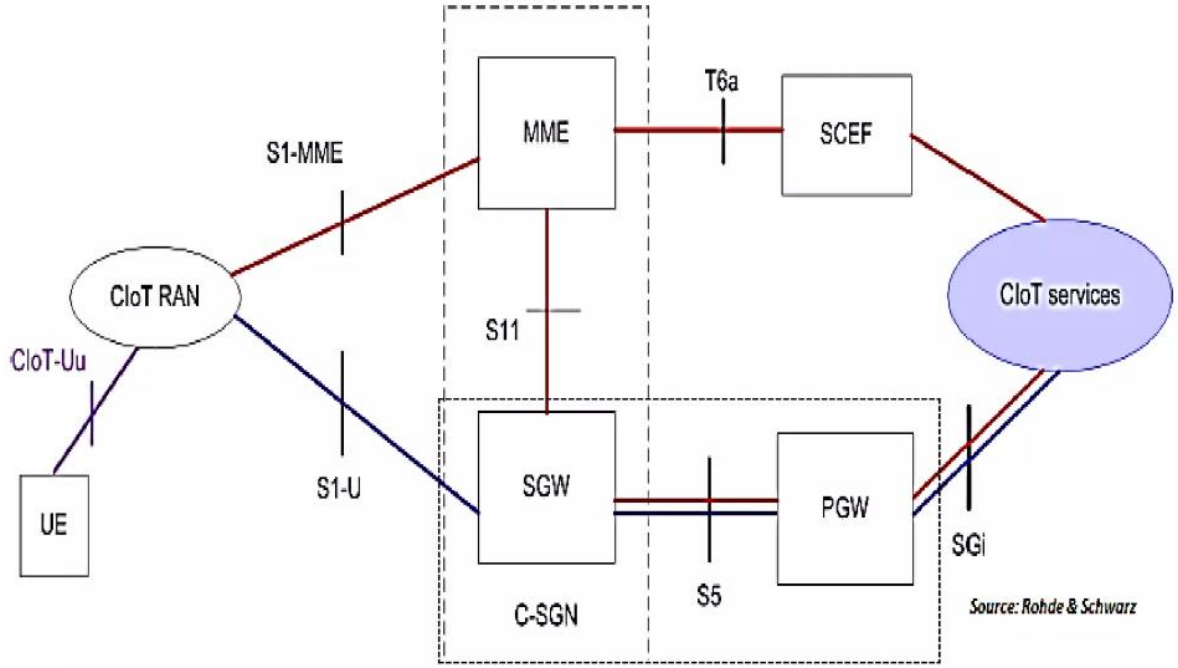
2.8.3.1 Arquitectura NB-IoT.

3GPP ha publicado especificaciones para una solución de banda estrecha para IoT celular conocida como LTE-NB o NB-IoT, que utiliza un ancho de banda de 200 KHz. Esta tecnología se ha lanzado en las versiones Rel.13, Rel.14 y Rel.15. Su enfoque principal es mejorar la cobertura en interiores, reducir costos, prolongar la duración de la batería y admitir un mayor número de dispositivos conectados.

A continuación, en la figura 5 podemos ver los componentes que conforman la arquitectura NB-IoT, en donde los dispositivos NB-IoT transmiten datos a través de la estación base eNodeB, utilizando un ancho de banda estrecho que permite una comunicación eficiente en términos de energía. Los datos se enrutan a través de la red celular utilizando los componentes SGW y PGW, y finalmente se entregan a las aplicaciones y servicios finales. La red gestiona la movilidad y la sesión de los dispositivos a través del MME.

Figura 5.

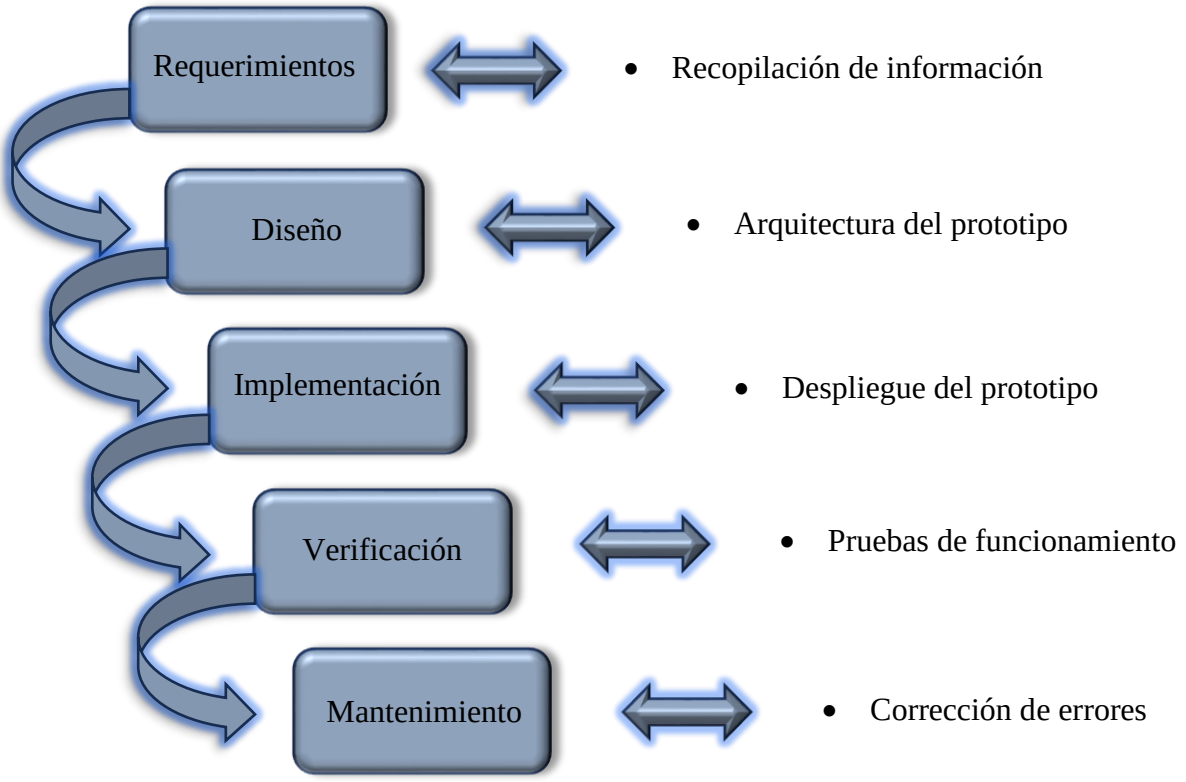
Arquitectura NB-IoT.



Nota. Tomado de "NB-IoT Architecture | LTE-NB IoT Architecture", por RF Wireless World, 2012, (https://www.rfwireless-world.com/Terminology/LTE-NB-IoT-Architecture.html)

2.9 Metodología de investigación.

Según (Coelho, 2020), se define a la metodología de investigación como "un conjunto de procedimientos y técnicas que se aplican de manera ordenada y sistemática en la realización de un estudio", para el desarrollo del presente proyecto se usara la metodología en cascada la cual se caracteriza por tener un proceso secuencial de desarrollo que se divide en diferentes etapas que deben ejecutarse una tras otra. El nombre de esta metodología se debe a que las diferentes fases que componen el proyecto deben colocarse una encima de otra siguiendo un orden concreto y estricto de arriba hacia abajo.



2.9.1 Requerimientos.

Esta es la primera fase usada en este tipo de metodología, aquí se trabaja con las partes interesadas, ya sea la organización o personas, los cuales serán los encargados de brindar información clave para el desarrollo del proyecto. Este primer paso en crucial, debido a que cada fase trabaja con información del paso anterior, es decir los demás pasos dependen del primer paso ya que aquí levantamos toda la información necesaria para poder ejecutar de manera correcta las siguientes fases.

2.9.2 Diseño.

En esta fase se procede a diseñar el prototipo como tal, en donde se separa en dos partes siendo una la parte de elección de hardware y otra la parte de programación o componentes de software, además aquí se hará uso de los requisitos recopilados en la fase de requerimientos y se los organizará para poder complementarlos en un solo sistema.

2.9.3 Implementación.

Esta es la fase en donde todo el sistema entra en acción, es decir se realiza la construcción física del prototipo electrónico, en donde se realiza la soldadura de componentes, y carga de software. Además, en este punto se procede a desplegarlo en el área de pruebas teniendo así listo para evidenciar su funcionamiento.

2.9.4 Verificación.

Una vez listo el prototipo se procede a realizar las pruebas de funcionamiento, en donde se verificará la funcionalidad del prototipo, evidenciando la correcta toma de datos y de las aplicaciones propuestas, las mismas que deben de cumplir con los requerimientos presentados.

2.9.5 Mantenimiento.

Una vez que el prototipo ha sido implementado en el campo, se procederá a realizar correcciones a problemas que se puedan presentar, siendo esta etapa la que estará presente en toda la vida útil del prototipo.

CAPITULO III: DISENO E IMPLEMENTACION

En el presente capítulo, se describe el desarrollo del prototipo, siguiendo la metodología en cascada. La primera fase involucra la recopilación de requisitos del sistema, donde se obtiene información sobre la situación actual de la hacienda y los requerimientos necesarios para el desarrollo del prototipo. Seguidamente, se procede al diseño del prototipo, en el cual se seleccionan los mejores componentes electrónicos, placas de desarrollo, plataforma de visualización, entre otros.

Posteriormente, se realiza la implementación del prototipo, la cual implica la captura y procesamiento de datos provenientes de los sensores, para finalmente darles una aplicación de acuerdo con las necesidades de la hacienda.

3.1 Situación actual.

Para iniciar el desarrollo del presente prototipo, es fundamental comprender la situación actual y la importancia de la detección de celo en el país. Esto implica considerar varios factores, como la identificación del celo, el tratamiento posterior a la identificación (inseminación artificial), el índice de preñez, la gestión adecuada de la producción de ganado y la sostenibilidad económica de las haciendas en relación con la producción de leche.

Lamentablemente, en la actualidad en Ecuador, no se lleva a cabo una gestión reproductiva adecuada en las haciendas. Esto se debe, en parte, a la falta de información y capacitación para los vaqueros responsables de la identificación del celo. Como señala Robin Salverson (2021), "Para lograr un programa exitoso de inseminación artificial, es crucial alcanzar un alto porcentaje de preñez, y eso depende en gran medida de una identificación precisa de los

celos". Sin embargo, en Ecuador, según estudios realizados por Tomas Diaz (2013), la identificación del celo no supera el 50%.

Como resultado, el índice de preñez es tan bajo como el 15% en ejemplares bovinos que han pasado por un proceso de inseminación artificial, generando una situación económica deficiente en las haciendas ganaderas.

3.1.1 Hacienda La María.

En la hacienda La María, ubicada en la ciudad de Cotacachi en la comunidad de Colimbuela, cuya principal actividad económica es la producción de leche, la detección de celo se convierte en un elemento crucial para mejorar la gestión de la reproducción y, por ende, aumentar la producción lechera.

La hacienda alberga un total de 500 cabezas de ganado, compuestas solo por vacas, sin la presencia de toros. La cantidad significativa de vacas representa un desafío para la detección del celo, agravado por el hecho de que solo una persona es responsable de la identificación y la comunicación para la realización de la inseminación artificial. Otro desafío es que el período de celo ocurre principalmente durante la noche.

Para el control del celo, se realiza cada 15 días, donde el veterinario inspecciona y registra manualmente la cantidad de vacas en las que se pasó por alto la identificación del celo. La hacienda obtiene esta información de manera retroactiva, lo que dificulta cualquier corrección. Por lo tanto, mediante la implementación del prototipo, se busca proporcionar alertas en tiempo real sobre las vacas en celo, permitiendo una inseminación artificial oportuna y asegurando la preñez de la vaca. Este enfoque busca aumentar la sostenibilidad económica de la hacienda al mejorar la eficiencia reproductiva y, por ende, la producción de leche.

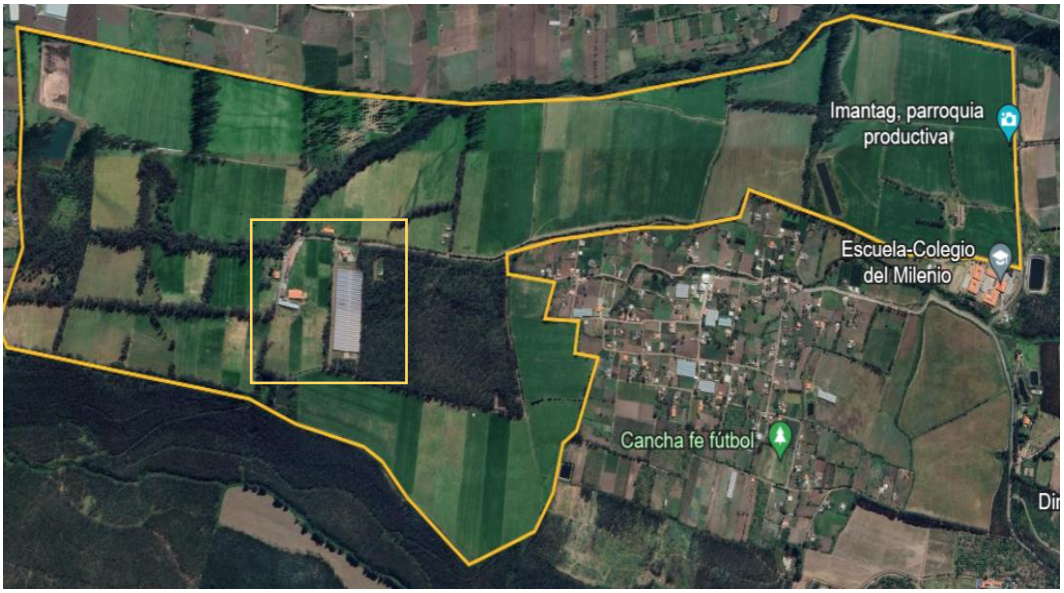
3.1.2 Ubicación Geográfica.

La Hacienda La María, situada en la ciudad de Cotacachi, en la parroquia de Imantag, comunidad de Colimbuela, abarca actualmente un área de 218 hectáreas divididas en secciones conocidas como potreros. Estos potreros, alternados estratégicamente, sirven para proporcionar pasto durante todo el año para el ganado.

El área incluye un punto central que alberga los equipos de ordeño y ordenadores que contienen información detallada sobre cada una de las vacas, como se ilustra en la figura 6 la extensión del terreno es extensa y la vigilancia nocturna de todas las vacas por parte del vaquero se vuelve complicada. Por esta razón, se ha establecido un punto central cercano al cuarto de ordeño, donde las vacas lecheras descansan durante la noche.

Figura 6.

Ubicación geográfica de la hacienda La María.



Nota. Se muestra el área total de la hacienda y el lugar donde se realizará el monitoreo de las vacas. Fuente: realizado por el autor.

El punto central donde descansan las vascas durante las noches es punto clave para la detección del celo. Estudios llevados a cabo por Eerdenburg indican que el período de mayor actividad de celo se registra entre las 6 p.m. y las 6 a.m. Por esta razón, este lugar será el punto de monitoreo para identificar cualquier señal de celo en las vacas.

La figura 7 revela que la ubicación de descanso de las vacas se encuentra a una distancia de 181.19 metros desde el nodo central de recepción de datos. La selección de las herramientas y tecnologías que posibiliten la transmisión de información a estas distancias se abordará en la fase 1, centrada en los requisitos del sistema.

Figura 7.

Distancia entre el nodo central y el nodo cliente.

Nota. Se indica gráficamente la distancia en la que se comunicaran los nodos que componen el prototipo. Fuente: Elaborado por el autor.

3.2 FASE 1: Especificación de requerimientos.

En la primera fase de la metodología planteada se realiza el análisis para determinar los requisitos y requerimientos para el desarrollo del prototipo, en este apartado se realizará un análisis investigativo el cual nos permitirá conocer el estado actual de la hacienda y de la problemática que existe en el proceso de detección de celo en la hacienda la María.

3.2.1 Nomenclatura de requerimientos.

Los requerimientos del sistema se fundamentarán en el estándar IEEE 29148:2011, que hace referencia a los requisitos necesarios para el funcionamiento óptimo del sistema, teniendo en cuenta los interesados clave. Estos requerimientos se clasifican en requerimientos de los Stakeholders, requerimientos del sistema y requerimientos de arquitectura.

Una vez que se hayan aclarado los requerimientos a tomar en cuenta, se procederá a establecer una nomenclatura específica para ellos. La Tabla 1 presenta las nomenclaturas utilizadas para la especificación de requerimientos.

Tabla 1.

Nomenclatura usada para el establecimiento de requerimientos.

Abreviación	Descripción
StSR	Requerimientos de los Stakeholders
SySR	Requerimientos del sistema
SRSH	Requerimientos de arquitectura

Para facilitar la interpretación de los resultados, el estándar permite asignar prioridades a cada uno de los requerimientos. Estos se clasificarán de acuerdo con su importancia, utilizando criterios como Alta, Media y Baja. La Tabla 2 detalla cada una de las prioridades que se utilizarán en el establecimiento de requerimientos.

Tabla 2.

Prioridad propuesta para los requerimientos.

Prioridad	Descripción
Alta	Se refiere a un requisito crítico que debe estar presente en el desarrollo del sistema.
Media	Este tipo de requisito tiene el potencial de influir en la decisión final del sistema, aunque es posible obviarlo en situaciones imprevistas.

Baja	Omitir este tipo de requisitos no genera un impacto significativo en la decisión final del sistema.

3.2.2 Stakeholders.

Los stakeholders, identificados según la nomenclatura como StSR, constituyen el conjunto de partes interesadas en el proyecto. La Tabla 3 proporciona una descripción detallada de los stakeholders y el papel específico que cada uno desempeña en el desarrollo del sistema, esto con el objetivo de lograr un trabajo eficiente.

Tabla 3.

Stakeholders (partes involucradas).

STAKEHOLDERS "StSR"	
Involucrados	**Función**
Ing. Walter Espín	Encargado de administrar la hacienda La María.
Sr. Luis Gomez	Trabajador encargado de la detección del celo.
Doc. Genaro Nieto	Profesional encargado.
MSc. Luis Suárez	Director del trabajo de titulación
Ing. Edgar Jaramillo	Asesor del trabajo de titulación
Sr. Alexander Gomez	Desarrollador del proyecto

3.2.3 Requerimientos de los Stakeholders.

La recopilación de los requisitos de los stakeholders fue llevada a cabo a través de encuestas, proporcionando así una visión más clara de los desafíos que enfrenta la hacienda. En

la Tabla 4 se presentan los requisitos recopilados mediante las encuestas realizadas, y estos valores se extraen del Anexo 1.

Tabla 4.

Requerimientos de Stakeholders. (StSR)

StSR				
#	REQUERIMIENTOS DE STAKEHOLDERS	Prioridad		
		Alta	Media	Baja
StSR1	El ganado debe encontrarse en un lugar específico dentro de la hacienda la María	x		
StSR2	El sistema debe comunicarse inalámbricamente	x		
StSR3	El nodo cliente debe monitorear el comportamiento de la vaca cuando entra en estado de celo	x		
StSR4	El nodo cliente debe ser un dispositivo móvil	x		
StSR5	El nodo centrar debe enviar alertas mediante telegram o whatsapp.	x		
StSR6	Los datos deben visualizarse en un interfaz grafica		x	
StSR7	El nodo cliente se debe ubicar en una posición estratégica, manteniéndose al lado de la vaca en todo momento.	x		
StSR8	El nodo cliente debe estar posicionado de tal manera que no interfiera con las actividades cotidianas de la vaca.	x		

StSR9	El nodo central debe ser capaz de albergar servidores necesarios para el desarrollo del prototipo		x	
	REQUERIMIENTOS DE USUARIO			
StSR10	Todos los trabajadores deber recibir las alertas de la detección del celo proporcionado por el prototipo.			x
StSR11	Las alertas deben manejar un lenguaje técnico.		x	
StSR12	El sistema debe identificar mediante un ID a cada una de las vacas.		x	
StSR13	El sistema debe tener una implementación fácil.	x		
StSR14	Se debe realizar mantenimientos al sistema		x	

3.2.4 Requerimientos del sistema.

Para garantizar un funcionamiento óptimo del sistema y cumplir con su propósito, es necesario definir requisitos que se derivan de las necesidades identificadas por los Stakeholders. Este apartado busca describir las características y funcionalidades esenciales que el sistema debe poseer. La Tabla 5 presenta estos requisitos, abarcando aspectos relacionados con la funcionalidad, uso, estado, entre otros.

Tabla 5.

Requerimientos del sistema. (SySR)

SySR				
#	**REQUERIMIENTOS DEL SISTEMA**	**Prioridad**		
		Alta	Media	Baja
	Requerimientos de interfaz			

SySR1	Los nodos deben contar con pines de entrada y salida de datos.	x	
SySR2	La arquitectura debe estar basada en diagramas permitiendo detectar fallas en el caso de haberlas.	x	
SySR3	El módulo de comunicación inalámbrico debe ser capaz de comunicarse con varios nodos a la vez.	x	
	Requerimientos de uso		
SySR4	El sistema debe estar diseñado para operar dentro del rango de frecuencias de uso libre permitido por la ARCOTEL	x	
SySR5	Los trabajadores deben manipular el nodo cliente		x
SySR6	El nodo central debe conectarse a varios nodos cliente	x	
SySR7	El prototipo debe proporcionar un historial de datos acerca del comportamiento de la vaca los cuales permitan la detección del celo.	x	
SySR8	El prototipo debe demostrar una alta precisión en la identificación del celo, con un mínimo de falsos positivos	x	
	Requerimientos de performance		
SySR9	El prototipo debe enviar alertas cuando detecte el celo en tiempo real.	x	
SySR10	El prototipo debe estar diseñado de tal manera que permita la escalabilidad en un futuro.	x	

SySR11	La funcionalidad del prototipo debe garantizar disponibilidad y confiabilidad.	x		
	Requerimientos energéticos			
SySR12	Los nodos deben estar diseñados de manera que nos garantice un consumo energético mínimo.			x
SySR13	Los nodos tienen que estar diseñados para poder usar baterías recargables.	x		
SySr14	Se debe especificar la duración de las baterías bajo condiciones de operación normal.		x	

3.2.5 Requerimientos de arquitectura.

En esta sección, se presentan los requisitos de arquitectura (SRSH) que describen criterios de diseño, software, hardware y eléctricos. Esta información es crucial para la selección precisa de los componentes de hardware y software, en la Tabla 6 se presentan las variables a considerar en los requerimientos de arquitectura del presente proyecto.

Tabla 6.

Requerimientos de hardware. (SRSH)

SRSH				
#	**REQUERIMIENTOS DE ARQUITECTURA**	**Prioridad**		
		Alta	Media	Baja
	Requerimientos de hardware			
SRSH1	Los componentes electrónicos deben ser compatibles y escalables entre sí.	x		

SRSH2	El sistema embebido debe emplear lenguajes de programación de uso libre.	x	
SRSH3	El sistema embebido debe tener conectividad WiFi.	x	
SRSH4	Los sensores deben ser calibrados, permitiendo así una mejor lectura de los datos.	x	
SRSH5	Los módulos LPWAN deben comunicarse dentro del mismo rango de frecuencia	x	
SRSH6	La comunicación inalámbrica LPWAN debe ser de larga distancia	x	
SRSH7	Los nodos deben estar protegidos de los factores externos tales como el polvo, lluvia, entre otros.		x
SRSH8	El microcontrolador debe poseer puertos analógicos y digitales	x	
	Requerimientos de software		
SRSH9	El IDE y lenguaje de programación debe ser compatible con la placa de desarrollo	x	
SRSH10	Disponer de librerías que permitan la simplificación del código de programación	x	
SRSH11	Para el diseño de la interfaz y aplicaciones se requiere plataformas de código abierto	x	
SRSH12	El sistema hará uso de servicios en la nube		x
SRSH13	El nodo central deberá receptar la señal de varios nodos cliente	x	

Requerimientos de diseño			
SRSH14	La placa de cada uno de los nodos debe tener un tamaño compacto.		x
SRSH15	El nodo central debe estar en un lugar adecuado para la recepción de la comunicación inalámbrica.	x	
SRSH16	El nodo cliente debe ir en un collar para colocarlo en el cuello de la vaca.	x	

3.3 FASE 2: Diseño

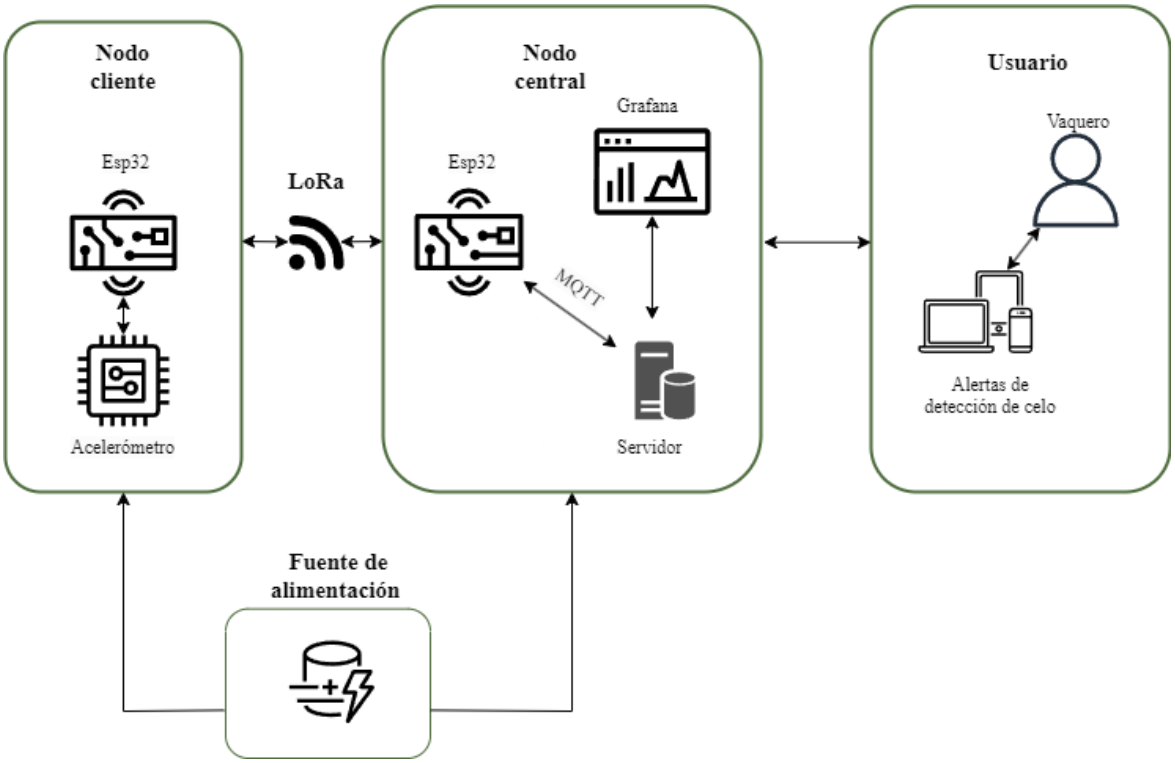
Siguiendo la metodología en cascada, se procede a la fase de diseño del sistema, en la cual se presentará la arquitectura del sistema junto con la elección de los componentes de hardware y software. Estos componentes deben ser compatibles entre sí para llevar a cabo la tarea establecida en el proyecto. Además, se detallarán cada uno de los nodos, tanto del nodo cliente como del nodo central. En esta etapa, se explicará la composición de cada nodo, abordando aspectos como su lógica de programación y su diagramado lógico.

3.3.1 Arquitectura del sistema.

Para el desarrollo del presente proyecto en donde se realizara la implementación de un prototipo que nos permita la detección del celo se usara la arquitectura que se muestra en la figura 8, en donde se tiene la parte de los nodos clientes la cual está compuesta de sensores y de un sistema embebido que enviara los datos recolectados mediante el protocolo LoRa, así también tenemos la parte del nodo central el cual es el encargado de receptar los mensajes LoRa para posteriormente enviarlos mediante el protocolo mqtt a una base de datos y procesarlos, finalmente se tiene la aplicación del proyecto en donde se enviaran alertas al momento de que el prototipo detecte el celo.

Figura 8.

Arquitectura general del proyecto.



Nota. Arquitectura general del proyecto, en donde se evidencia arquitecturas del nodo cliente, nodo central y usuario.

Fuente: Realizado por el autor.

3.3.2 Elección de componentes de hardware y software.

La selección de los componentes de hardware y software que integrarán el sistema se lleva a cabo mediante una tabla comparativa, la cual sigue los requerimientos presentados en la fase 1. En esta sección, cada componente, tanto de hardware como de software, es evaluado y se asigna una calificación, donde 1 indica que cumple con los requisitos y 0 señala que no los cumple. Al final, la elección del componente se basará en aquel que obtenga la calificación más alta en la tabla comparativa.

A continuación, se detalla por separado los componentes de cada nodo para profundizar en la elección realizada y exponer las características específicas de los elementos seleccionados.

3.3.2.1 Procesador.

El microcontrolador será el núcleo del sistema embebido mismo que será encargado de realizar las tareas previamente programadas, necesarias para la detección del celo, en la tabla 7 se indica algunas de las placas de desarrollo tomadas en cuenta, en donde se las calificará de acuerdo con los requerimientos establecidos en la tabla 6.

Tabla 7.

Selección de sistema embebido para el nodo móvil.

Hardware	REQUERIMIENTOS					Valoración
	SRSH2	SRSH3	SRSH8	SRSH14	SRSH17	
esp32	1	1	1	1	1	5
ATmega328p	1	0	1	0	0	2
RP2040	1	0	1	1	1	4
STM32	0	0	1	1	1	3

Cumple: 1

No cumple: 0

Finalizada la evaluación para la selección del procesador, se determina que el esp32 satisface todos los requisitos establecidos. Este es el encargado del controlar los sensores, módulos inalámbricos, y de procesar la información obtenida de los sensores, en la figura 9 se aprecia el procesador escogido para el desarrollo del prototipo.

Figura 9.

Procesador ESP32.

Nota. Procesador que será usado en el sistema embebido. Recuperado de: https://www.mouser.es/new/espressif/espressif-esp32-soc/

Así también en la Tabla 8, se presentan las características principales de la placa de desarrollo seleccionada.

Tabla 8.

Características del microcontrolador ESP32

	ESP32
Características	• Cuenta con una CPU Tensilica Xtensa LX6, que puede ser dual-core o single-core • Velocidad de reloj de hasta 240 MHz • Ultra-low power • Memoria disponible de 520 KB • Dependiendo del módulo, puede integrar hasta 4 MB de Flash (como el ESP32-WROOM-32). • Compatible con 802.11 b/g/n (2.4 GHz).

- Múltiples modos de energía para optimizar el consumo, incluyendo modos de sueño profundo para aplicaciones que requieren batería de larga duración.

3.3.2.2 **Selección de sensor acelerómetro.**

En el desarrollo del diseño del prototipo, es crucial seleccionar un sensor adecuado para medir los parámetros que faciliten la detección del celo. Se ha determinado que el uso de un acelerómetro es esencial, ya que este permitirá monitorear eficazmente el incremento de actividad de la vaca, un indicador clave de celo. La Tabla 9 presenta una comparativa de varios acelerómetros disponibles en el mercado, evaluados y valorados según los criterios específicos establecidos para el desarrollo del prototipo.

Tabla 9.
Selección del sensor acelerómetro para el prototipo de detección de celo.

Hardware	REQUERIMIENTOS				Valoración
	SySR10	SRSH10	SySR12	SRSH14	
MPU-6050	1	1	1	0	3
ADXL345	1	1	1	1	4
MMA8452Q	1	1	1	0	3
BMA400	1	0	1	1	3

Cumple: 1

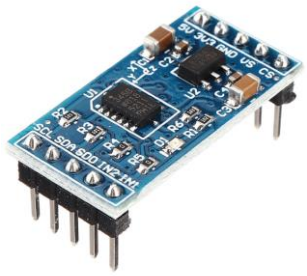
No cumple: 0

Una vez completada la evaluación basada en los parámetros definidos, se ha determinado que el acelerómetro ADXL345 satisface todos los criterios necesarios para el desarrollo del proyecto.

Este sensor es el encargado del monitoreo del comportamiento físico de la vaca al momento de entrar en estado de celo, permitiendo capturar y analizar patrones de movimiento característico asociados con el celo, en la figura 10 se puede observar una representación gráfica del sensor usado en el desarrollo del prototipo.

Figura 10.

Acelerómetro ADXL 345.



Nota. Sensor acelerómetro que será usado para la recolección de variación del movimiento presente en la vaca. Recuperado de: https://roboticsec.com/producto/sensor-acelerometro-adxl345/

En la tabla 10, se describen las características específicas de este sensor, proporcionando una comprensión detallada de su funcionamiento. Esta información será de gran utilidad para facilitar la programación y la integración con la placa de desarrollo seleccionada previamente.

Tabla 10.

Características del acelerómetro ADXL345.

Acelerómetro ADXL345	
Características	• Ofrece una alta resolución de medición con 13 bits, lo que permite detectar cambios menores en la aceleración.

- Capaz de medir la aceleración en rangos de ±2g, ±4g, ±8g, y ±16g, lo que lo hace versátil para diversas aplicaciones, desde la detección de movimientos sutiles hasta el monitoreo de actividades dinámicas.
- Diseñado para ser energéticamente eficiente, opera en un rango de voltaje de alimentación de 2.0V a 3.6V.
- Soporta interfaces digitales SPI (3 y 4 cables) e I2C, permitiendo una fácil integración con la mayoría de los microcontroladores y sistemas de desarrollo.
- Incluye modos de operación de bajo consumo de energía y modos de suspensión para aumentar la eficiencia energética.
- Posee capacidad de filtrado digital integrado, lo que reduce el ruido de la señal y mejora la precisión de la medición.
- Su pequeño tamaño (3x5x1 mm) lo hace ideal para aplicaciones donde el espacio es limitado.

3.3.2.3 **Selección de tecnología LPWAN.**

Debido a las conficiones del área geográfica y la ubicación de las vacas, es indispensable contar con una comunicación inalámbrica entre los nodos cliente y el nodo central, en donde la solución de comunicación inalámbrica que mejor se adapta en este tipo de proyectos es la tecnología LPWAN, siendo esta tecnología ideal debido a su eficiencia energética y al rango de cobertura que nos proporciona.

En la tabla 11 se presentan tres opciones de tecnologías LPWAN las cuales serán evaluadas en función de los requerimientos previamente establecidos. Este análisis permitirá seleccionar la tecnología más adecuada para ser implementada en el desarrollo del proyecto.

Tabla 11.

Elección de tecnología LPWAN.

LPWAN	REQUERIMIENTOS						Valoración
	SySR3	SySR4	SySR10	SySR11	SySR12	SRSH6	
LoRa	1	1	1	1	1	1	6
Sigfox	1	1	0	0	1	1	4
NB-IoT	1	0	1	1	0	0	3

Cumple: 1

No cumple: 0

Tras evaluar cada una de las tecnologías, se identifica a LoRa como la alternativa más adecuada para la comunicación en el prototipo electrónico de detección de celo. Su habilidad para alcanzar hasta 20 kilómetros bajo condiciones óptimas proporciona una cobertura extensa, esencial para abarcar la amplia zona geográfica donde se sitúa el ganado. Además, LoRa destaca por su mínimo consumo energético, un aspecto crucial al seleccionar la tecnología de comunicación inalámbrica para este proyecto.

3.3.2.4 **Selección de modulo Lora.**

Una vez definida la tecnología de comunicación inalámbrica LPWAN a usarse en el desarrollo del proyecto, se procede a analizar los diferentes módulos LoRa disponibles en el

mercado, en la tabla 12 se puede visualizar los diferentes módulos y las valoraciones de cada uno en donde se elegirá al que tenga una mejor valoración.

Tabla 12.

Elección de módulo LoRa.

LPWAN	REQUERIMIENTOS						Valoración
	SRSH1	SySR4	SRSH5	SRSH6	SRSH10	SRSH14	
SX1278	1	1	1	1	1	1	6
SX1262	1	1	1	1	1	1	6
SX1276	1	1	1	1	1	1	6

Cumple: 1

No cumple: 0

Tras concluir la evaluación de los distintos módulos LoRa, se observa que todos satisfacen los criterios preestablecidos, brindando así la flexibilidad para seleccionar cualquiera de ellos. En este contexto específico, se ha decidido optar por el módulo SX1278, una elección influenciada principalmente por su disponibilidad en el mercado ecuatoriano y su costo competitivo. A continuación, se detallarán las características técnicas del módulo SX1278, proporcionando información valiosa que contribuirá al desarrollo del proyecto.

Tabla 13.

Características del módulo LoRa.

Módulo LoRa SX1278

Características	• Frecuencia de operación 137 MHz a 525 MHz. • Potencia de Salida hasta de +20 dBm. • Sensibilidad hasta de -148 dBm. • Ancho de banda del canal configurable desde los 7.8 kHz hasta 500 kHz. • Modos de bajo consumo de energía disponibles, incluyendo un modo de sueño profundo que reduce significativamente el consumo de energía cuando el dispositivo no está activo. • Posee capacidad de filtrado digital integrado, lo que reduce el ruido de la señal y mejora la precisión de la medición.

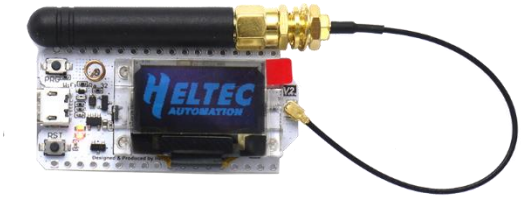
3.3.2.5 Selección del sistema embebido

El sistema embebido se encargará de recopilar los datos del sensor y enviarlos mediante la tecnología seleccionada hacia el nodo Gateway. Tras un análisis previo, se determinó que tanto el microcontrolador como la tecnología inalámbrica más adecuada para el prototipo son el ESP32 y el módulo LoRa SX1278.

Durante la búsqueda en el mercado, se identificó que ambos componentes están integrados en un solo sistema embebido, el ESP32 LoRa V2, el cual cumple con las características requeridas, así también este modelo se encuentra disponible en el país de Ecuador, por esta razón, se eligió este sistema para el desarrollo del prototipo. En la figura 11 se puede apreciar el microcontrolador elegido para el desarrollo del prototipo.

Figura 11.

Microcontrolador esp32 LoRa v2.



Nota. Sistema embebido que contiene el un chip LoRa incluido. Recuperado de: https://megatronica.cc/producto/lora32-433mhz-esp32/

A continuación, se presentan las características del sistema embebido esp32 LoRa v2.

Tabla 14.

Características del sistema embebido.

Esp 32 LoRa v2	
Características	• Dual-core Xtensa LX6 de 32 bits. • Velocidad de reloj hasta 240 MHz. • Wi-Fi 802.11 b/g/n (2.4 GHz). • Rango de 433 MHz a 915 MHz en el módulo LoRa, dependiendo de la región. • Soporta velocidades entre 0.3 kbps y 37.5 kbps, optimizado para transmisión de datos a larga distancia. • Rango de hasta 10 km en condiciones ideales.

3.3.2.6 Selección de sistema operativo del Gateway

Una de las partes más importantes dentro del sistema es el Gateway, el cual se encarga de recibir los valores de los nodos cliente para posteriormente almacenarlos, procesarlos, visualizarlos y

aplicarles una función dentro del sistema. Este componente será seleccionado mediante una ponderación que tomará en cuenta los requerimientos establecidos previamente.

Tabla 15.

Elección de hardware para el gateway.

HARDWARE GATEWAY	REQUERIMIENTOS					Valoración
	StSR5	StSR6	StSR9	SySR10	SRSH3	
Windows server	0	0	0	1	0	1
Ubuntu Sever	1	1	1	1	1	5
Raspberry Pi OS	1	0	1	1	0	3
Cumple: 1						
No cumple: 0						

Tras completar la evaluación, se concluye que Ubuntu server cumple con todos los requisitos necesarios para el desarrollo del proyecto. Debido al bajo consumo de recursos de cómputo y empleabilidad del software esta distribución es adecuada para el desarrollo del proyecto.

3.3.2.7 Selección de software de procesamiento de datos

El procesamiento de los datos se trata de la recolección, envió de datos desde el nodo cliente hacia el nodo receptor, de la misma forma el nodo central es el encargado de recibir los datos para posteriormente enviarlos a la base de datos, este proceso se lo realiza mediante programación, anteriormente se eligió el hardware del prototipo en donde la placa de desarrollo seleccionada ya cuenta con su entorno de programación, este entorno es el IDE de Arduino, el cual la estructura del código está bajo el lenguaje de programación C.

3.3.2.8 Selección de software de almacenamiento de datos

La elección del software de almacenamiento de datos desempeña un papel fundamental en el proceso de detección del celo, ya que permite registrar y gestionar la información obtenida para su análisis posterior. Para lograrlo, es esencial seleccionar una base de datos sólida que ofrezca tanto escalabilidad como facilidad en la ejecución de consultas complejas. A continuación, se evaluarán distintas opciones de bases de datos para identificar aquella que mejor cumpla con los requisitos previamente establecidos y se ajuste a las necesidades del sistema.

Tabla 16.

Elección de base de datos.

SOFTWARE BDD	REQUERIMIENTOS					Valoración
	SySR2	SySR7	SySR10	SySR11	SRSH2	
My SQL	1	1	1	1	1	5
Influxdb	1	1	1	1	1	5
SQL Server	1	1	1	1	0	4
Cumple: 1						
No cumple: 0						

Basado en tabla de puntajes de las bases de datos tenemos como resultado que dos de ellas cumplen con la mayoría de los requerimientos, teniendo así la elección de My SQL como base datos para el desarrollo de este prototipo, debido a su simplicidad, escalabilidad y compatibilidad con las diferentes librerías y lenguajes de programación.

3.3.2.9 **Software de visualización de datos**

Para analizar y procesar adecuadamente los datos, es fundamental disponer de una visualización clara y comprensible de los mismos. Por ello, surge la necesidad de seleccionar un software que facilite esta tarea, asegurando que cumpla con los requisitos específicos para el desarrollo del prototipo. A continuación, se presentan las alternativas evaluadas junto con sus respectivos puntajes.

Tabla 17.

Elección del software de visualización.

SOFTWARE BDD	REQUERIMIENTOS					Valoración
	StSR6	SRSH2	SRSH10	SRSH11	SRSH12	
Kibana	1	1	1	1	0	4
Grafana	1	1	1	1	1	5
Tableau	1	0	1	0	0	2

Cumple: 1

No cumple: 0

Para la visualización de los datos obtenido por el prototipo se ha elegido el software de Grafana siendo este el software con el puntaje más alto, este software nos permitirá ver los datos y poder interpretarlos de mejor manera.

3.3.3 Calculo para la fuente de alimentación.

La fuente de alimentación para los nodos cliente desempeña un papel fundamental en este contexto, dado que el prototipo está diseñado para recolectar datos en tiempo real a través de un sensor acelerómetro durante la noche, y se debe colocar en un collar, lo cual exige que el nodo

sea móvil. Dada esta movilidad, es imprescindible que cuente con una fuente de alimentación externa, concretamente una batería. Idealmente, esta batería debería ser de litio, por sus capacidades de recarga.

3.3.3.1 Consumo total del sistema

Al seleccionar la capacidad adecuada de la batería para los nodos cliente, es vital tener en cuenta el consumo de energía de cada uno de los componentes que integran el nodo. La Tabla 15 proporciona un desglose detallado de los consumos eléctricos de los componentes individuales, así como del consumo total del nodo, facilitando así una elección informada sobre la capacidad de la batería necesaria para satisfacer los requisitos de energía del nodo de manera eficiente.

Tabla 18.

Consumo eléctrico del nodo cliente.

Componente	**Consumo Estado Activo (mA)**	**Consumo Estado Sleep (mA)**
Esp32	80 mA	0.8 mA
Acelerómetro ADXL345	140 µA	20 µA
Módulo LoRa SX178	130 mA	0.2 µA
Total	210.14 mA	80.0202 mA

Tras determinar los valores de corriente requeridos para el funcionamiento del prototipo, se ha concluido que el nodo cliente opera con un consumo activo de 210.14 mA para cada uno de sus componentes. Para calcular el valor de consumo de la batería se utilizará la ecuación 1.

$$Consumo\ Total = \frac{(T_{cn} * I_{cn}) + (T_{cd} * I_{cd})}{T_{cn} + T_{cd}} \qquad (1)$$

Donde:

T_{cn}: Tiempo de consumo en estado activo

I_{cn}: Corriente usada en estado activo

T_{cd}: Tiempo de consume en estado sleep

I_{cd}: Corriente usada en estado sleep

Teniendo los parámetros establecidos se procede a indicar los valores en cada una de las variables las mismas que tienen que ver con el tiempo de consumo en estado activo siendo este el tiempo el que estará el nodo enviando la información, el estado sleep es el tiempo que se encuentra en reposo antes de ser retirado de la vaca.

A continuación, se realiza el cálculo del consumo total de energía, teniendo en cuenta el consumo de cada uno de los componentes electrónicos, así como del tiempo total del funcionamiento.

Datos:

T_{cn}: 7 horas

I_{cn}: 210.14 mA

T_{cd}: 17 horas

I_{cd}: 80.02 mA

Una vez obtenidos los datos, se procede a sustituirlos en la fórmula correspondiente con el propósito de determinar el consumo de energía, lo que permite seleccionar la batería adecuada para la alimentación del sistema embebido.

$$Consumo\ Total = \frac{(7h * 210.14\ mA) + (17h * 80.02\ mA)}{7h + 17\ h}$$

$$Consumo\ Total = \frac{(1470.98hmA) + (1360.34hmA)}{24h}$$

$$Consumo\ Total = \frac{2831.32mA}{24}$$

$$Consumo\ Total = 117.97\ mA$$

3.3.3.2 Tiempo de funcionamiento de la batería

Tras realizar el cálculo correspondiente, se determinó que el sistema tiene un consumo total de 117.94 mA. Basándose en este dato, se identificó que en el mercado ecuatoriano está disponible una batería de 3000 mAh. Con esta información, se procede a calcular el tiempo de autonomía del sistema utilizando la ecuación 2:

$$Tiempo(horas) = \frac{Capacidad\ de\ bateria\ (mAh)}{Consumo\ del\ sistema\ (mA)} \qquad (2)$$

Datos utilizados:

- Capacidad de batería (mAh) = 3000 mAh
- Consumo del sistema (mA) = 117.97 mA

Al reemplazar estos valores en la ecuación, se obtiene:

$$Tiempo\ (horas) = \frac{Capacidad\ de\ bateria\ (mAh)}{Consumo\ del\ sistema(mA)}$$

$$Tiempo(horas) = \frac{3000\ mAh}{117.97\ mA}$$

$$Tiempo(horas) = 25.42\ horas$$

El cálculo establece que el sistema tiene una autonomía de 25.42 horas para un uso continuo. Sin embargo, dado que el sistema recopila datos únicamente durante la noche (de 8:00 p.m. a 3:00 a.m., es decir, **7** horas diarias), se concluye que la batería puede soportar aproximadamente 3 días de operación antes de necesitar recarga.

3.3.3.3 Cálculo del tiempo de carga de la batería

Es crucial garantizar la recarga de la batería, ya que una vez agotada, el sistema dejará de funcionar. Para este caso, se utilizará un módulo de carga con un voltaje de 4.2 V y una corriente de carga de 1 A. El tiempo necesario para recargar la batería se calcula mediante la ecuación 3:

$$Tiempo\ de\ carga = \frac{Capacidad\ de\ la\ bateria\ (mAh)}{Corriente\ de\ carga\ (mA)} \quad (3)$$

Datos utilizados:

- Capacidad de la batería = 3000 mAh
- Corriente de carga = 1A.

Ahora procedemos a reemplazar los valores en la ecuación 3, teniendo como resultado lo siguiente:

$$Tiempo\ de\ carga = \frac{Capacidad\ de\ la\ bateria\ (mAh)}{Corriente\ de\ carga\ (mA)}$$

$$Tiempo\ de\ carga = \frac{3000\ mAh}{1000\ mA}$$

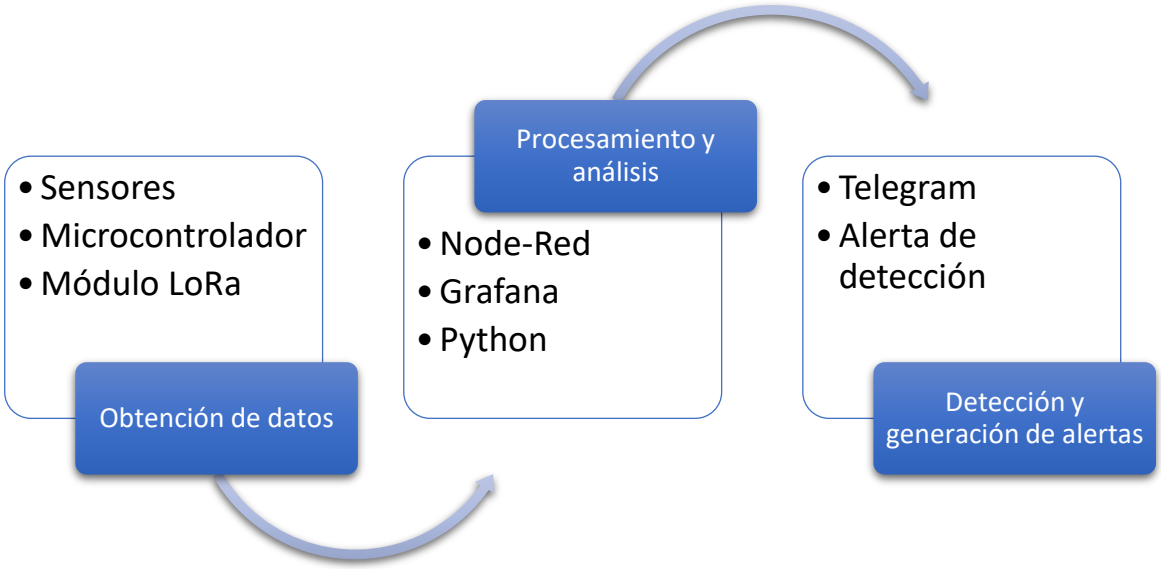
$$Tiempo\ de\ carga = 3\ h$$

3.3.4 Diagrama de bloques del sistema

La funcionalidad del prototipo de detección de celo se basa en la figura 12, en la que se detalla cada uno de los bloques que componen el sistema que interactúan entre sí para poder tomar los datos de los sensores, enviarlos mediante el medio inalámbrico para posteriormente ser procesados y así poder enviar alerta cuando el celo sea detectado o no.

Figura 12.

Diagrama de bloques del sistema.



Nota. Diagrama de bloques que indica los paso que realiza el sistema para la detección del celo. Fuente: Elaborado por el autor.

3.3.5 Diseño del nodo central

El nodo central se encarga de recibir la información enviada por el nodo cliente. Este componente es esencial para el funcionamiento del sistema, ya que centraliza los datos recolectados para su posterior análisis.

La infraestructura del nodo central incluye:

- Recepción de datos: Interacción con el nodo cliente a través de protocolos de comunicación definidos.
- Procesamiento local: Validación y almacenamiento temporal de datos.
- Transmisión a servidores: Envío de la información a una base de datos o sistemas de análisis.

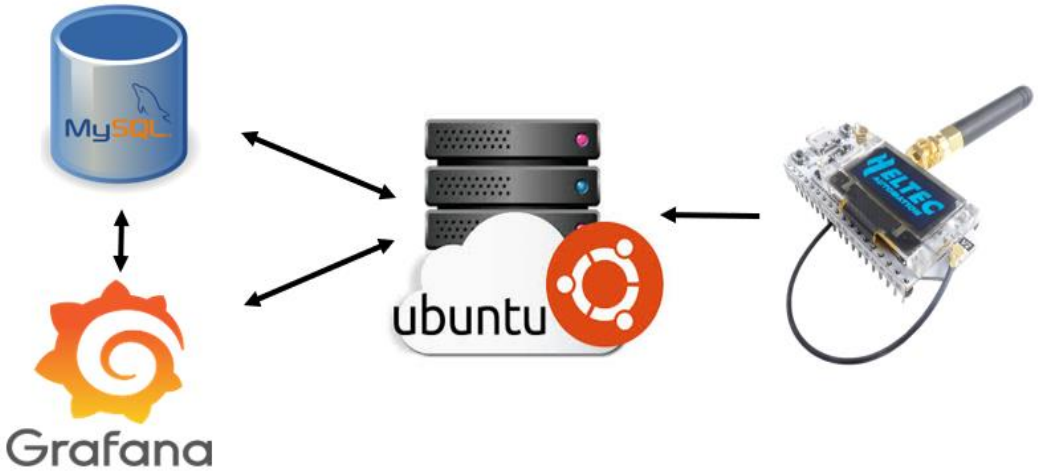
3.3.5.1 Diagrama nodo central

El sistema incluye un **nodo central**, cuya función principal es recibir la información enviada por el nodo cliente. Este nodo central analiza los datos recopilados para determinar si se detecta la presencia de celo en la vaca.

En la **Figura 13**, se presenta el diagrama correspondiente al nodo central, compuesto por un microcontrolador **ESP32 LoRa V2**, encargado de recibir y transmitir la información. Posteriormente, los datos son enviados a un servidor en la nube, donde se realiza el análisis y procesamiento. Además, el sistema incluye un módulo dedicado a la visualización de los datos y la generación de alertas, garantizando una respuesta eficiente y clara ante los resultados obtenidos.

Figura 13.

Diagrama del nodo central.



Nota. Se representa la comunicación entre los componentes que componen el nodo central. Fuente: Elaborado por el autor.

3.3.5.2 Diagrama de bloques del nodo central.

Dentro del diseño del prototipo, se incluye el diagrama de bloques, mostrado en la figura 14. Este diagrama detalla cada una de las etapas que debe cumplir el nodo central para garantizar

el correcto funcionamiento del sistema. Cada bloque representa una función esencial en el desempeño general del prototipo.

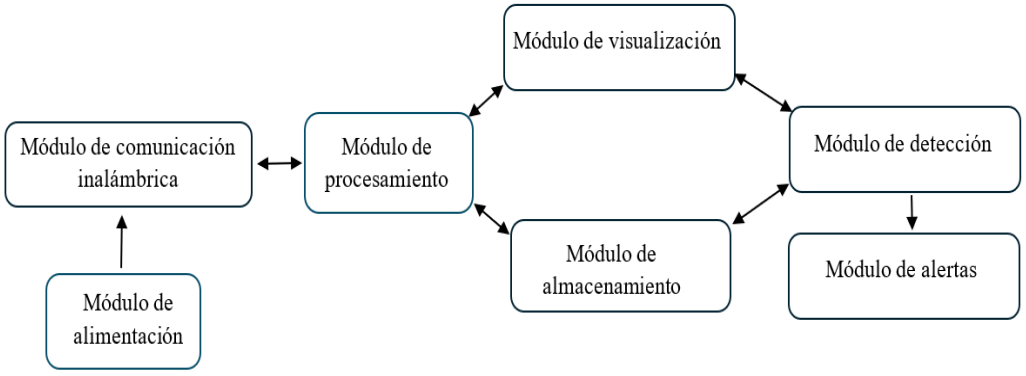
El primer bloque corresponde al sistema de alimentación, encargado de suministrar energía al nodo central. A continuación, se encuentra el módulo inalámbrico, cuya función es recibir los datos enviados por el nodo cliente utilizando la tecnología LoRa. Posteriormente, el módulo de procesamiento se encarga de trabajar con los datos recibidos, otorgándoles una funcionalidad específica según los objetivos del sistema.

Después del procesamiento, los datos pasan a los bloques de visualización y almacenamiento. En esta etapa, mediante consultas SQL, es posible observar el comportamiento individual de cada vaca y almacenar dicha información en una base de datos para su análisis posterior.

Finalmente, el sistema incluye los bloques de detección y alertas. En el bloque de detección, se clasifica si una vaca está en celo o no, mientras que en el bloque de alertas se envían notificaciones mediante Telegram al encargado del manejo del ganado, proporcionando información clave de manera oportuna y eficiente.

Figura 14.

Diagrama de bloques del nodo central.



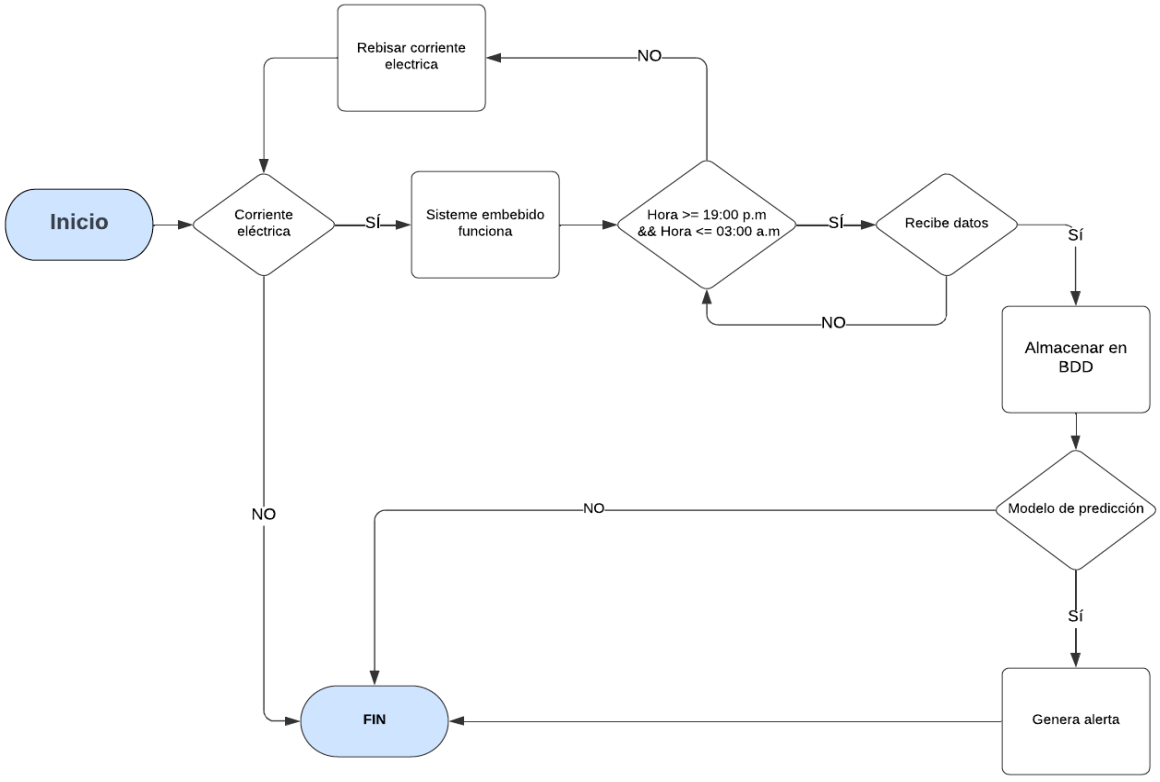
Nota. Representación de la interconexión de los módulos que componen el nodo central. Fuente: Elaborado por el autor.

3.3.5.3 Diagrama de flujo del nodo central

El funcionamiento del prototipo no solo se basa en un diagrama de bloques correctamente diseñado, que representa la parte física del sistema, sino que también requiere la consideración del diagrama de flujo. Este último proporciona información más detallada sobre el funcionamiento interno del prototipo, especificando con precisión la programación contenida en el nodo central. En la figura 15 se ilustra la lógica que guiará el funcionamiento del nodo central, permitiendo comprender de manera clara su estructura y operación.

Figura 15.

Diagrama de flujo del nodo central.



Nota. Se presenta el diagrama de flujo del nodo central, en donde podemos ver el funcionamiento lógico del nodo central. Fuente: Elaborado por el autor.

3.3.6 Diseño del nodo cliente

El sistema está compuesto por diversas partes, entre las cuales se encuentra el nodo cliente, el cual corresponde a un dispositivo móvil que acompañará a la vaca durante todo el proceso de recolección de datos. Debido a esta funcionalidad, es fundamental considerar el diseño de una carcasa que proporcione protección frente a factores climáticos y del entorno. Además, este dispositivo deberá ser alimentado por una fuente de energía externa; como se mencionó en secciones anteriores, dicha fuente será una batería.

La relevancia del nodo cliente radica en su capacidad para recopilar datos en tiempo real y transmitir esta información al nodo central. Cabe destacar que este nodo no estará diseñado para procesar ni tratar los datos, ya que su única función será capturar la información en bruto y enviarla.

En las siguientes secciones se describe el diseño del nodo cliente, incluyendo su diagrama físico y lógico.

3.3.6.1 Diagrama nodo cliente

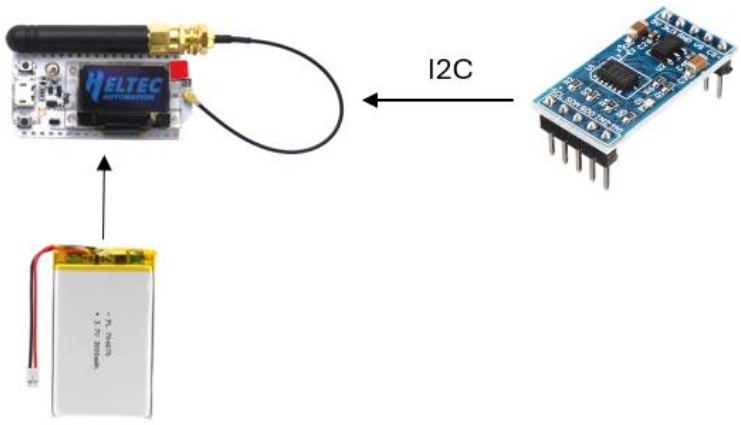
Durante el desarrollo del prototipo, es esencial considerar el diagrama físico de los componentes que integran el nodo cliente. Esto permite comprender el comportamiento y la interacción entre los elementos que conforman dicho nodo.

En la figura 16 se presenta una representación gráfica de los componentes del nodo cliente. Entre ellos, se destaca el sensor acelerómetro ADXL345, encargado de recolectar datos relacionados con variaciones de cambios físicos. Este sensor se comunica con el microcontrolador ESP32 LoRa v2 a través del protocolo I2C. Como se mencionó previamente, el nodo cliente debe ser un dispositivo móvil, razón por la cual cuenta con una fuente de alimentación externa, en este caso, una batería LiPo de 3000 mAh.

En conjunto, todos estos componentes conforman el nodo cliente, el cual desempeña un papel clave en la captura y transmisión de datos.

Figura 16.

Diagrama nodo cliente.



Fuente: Elaborado por el autor.

3.3.6.2 Diagrama de bloques del nodo cliente.

De manera similar al nodo central, el nodo cliente cuenta con un diagrama de bloques, representado en la figura 17. Este diagrama es notablemente más sencillo, ya que el nodo cliente se limita a la recolección de datos provenientes del acelerómetro ADXL345 y a su envío al nodo central. En esta etapa, no se lleva a cabo el procesamiento ni la limpieza de los datos, lo que justifica su diseño más simple.

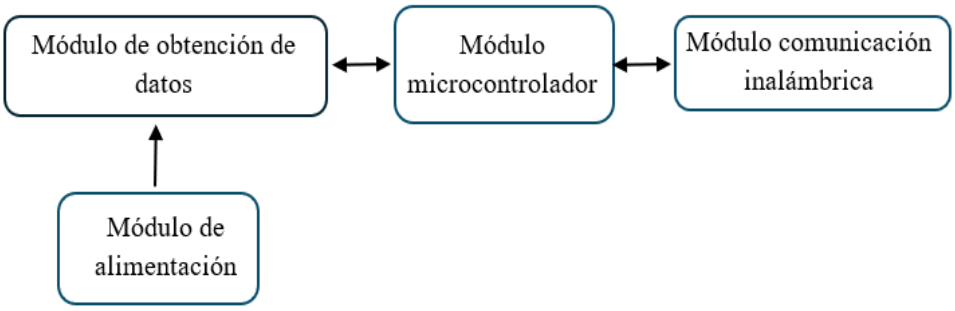
El primer componente es el bloque de alimentación, que debe estar compuesto por baterías, dado que el nodo cliente está diseñado como un dispositivo móvil. A continuación, se encuentra el módulo de obtención de datos, donde interviene el acelerómetro ADXL345 como el sensor encargado de medir las variaciones físicas. Todo el sistema es gestionado por el microcontrolador ESP32 LoRa v2, que se encarga de coordinar las funciones del nodo. Por

último, se incluye el módulo de comunicación inalámbrico, el cual se utiliza para transmitir la información recolectada hacia el nodo central mediante la tecnología LoRa.

Este diseño enfocado en la simplicidad y la eficiencia permite que el nodo cliente cumpla con su papel principal dentro del sistema sin sobrecargarlo con funciones adicionales.

Figura 17.

Diagrama de bloques del nodo cliente.



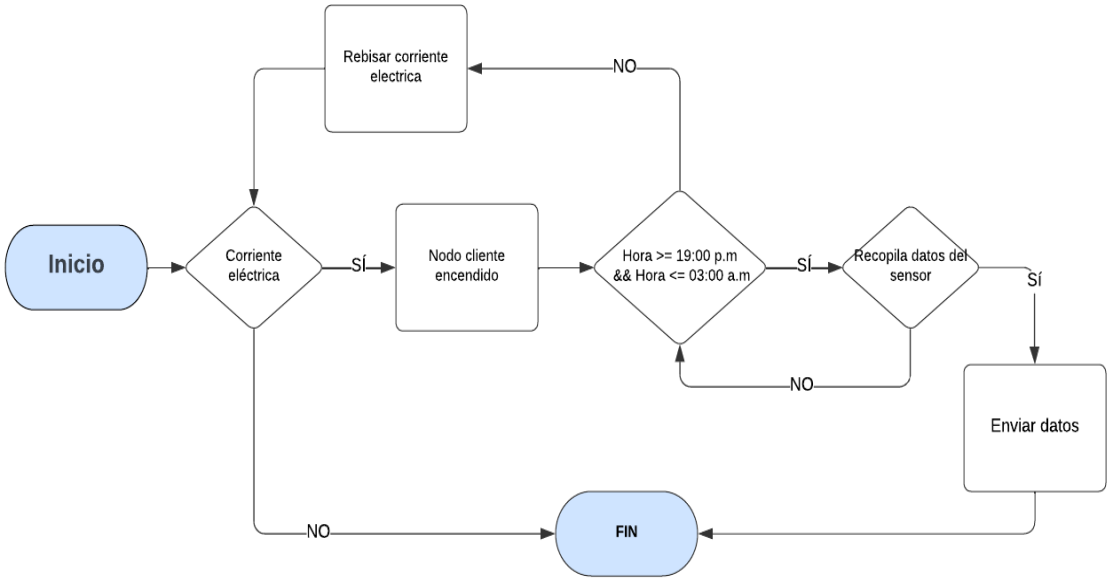
Nota. Representación de la interconexión de los módulos que componen el nodo cliente. Fuente: Elaborado por el autor.

3.3.6.3 Diagrama de flujo del nodo cliente.

De manera similar al nodo central, el nodo cliente también cuenta con su propio diagrama de flujo, como se muestra en la figura 18. Este diagrama es más sencillo, ya que el nodo cliente tiene una funcionalidad limitada: únicamente se encarga de enviar los datos del sensor en los horarios previamente establecidos. Fuera de esta tarea específica, el nodo cliente no desempeña ninguna otra función.

Figura 18.

Diagrama de flujo del nodo cliente.



Nota. Se presenta el diagrama de flujo del nodo cliente, en donde podemos ver el funcionamiento lógico del nodo central.

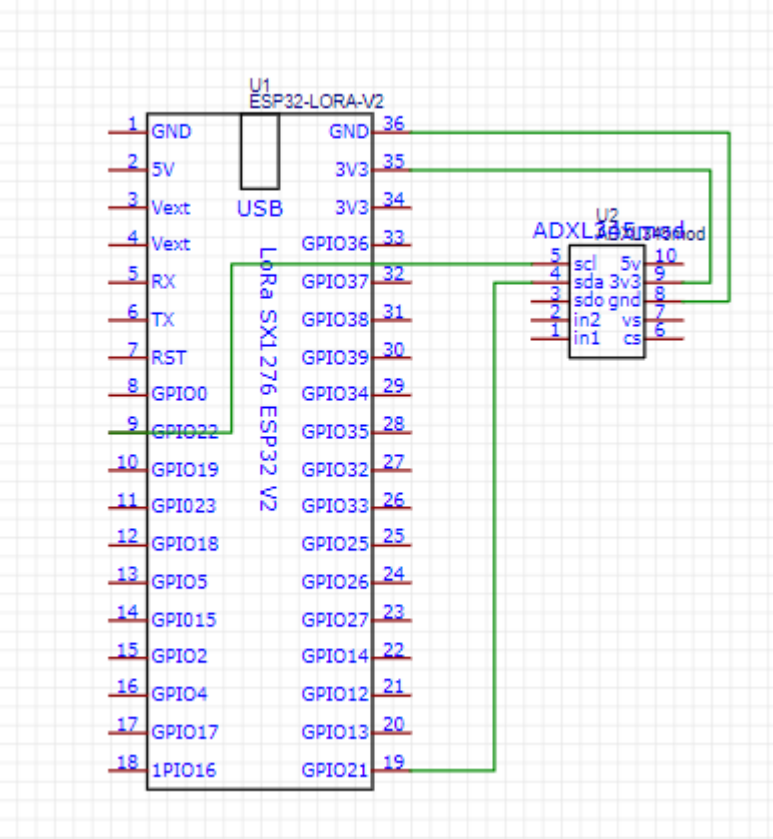
Fuente: Elaborado por el autor.

3.3.6.4 Diseño de pcb del nodo cliente

Otro aspecto crucial en el desarrollo es el diseño de la PCB para el nodo cliente, ya que esta será la base que integrará los componentes principales del nodo. En este caso, los elementos utilizados son únicamente el microcontrolador y el sensor encargado de recopilar los datos. Para el diseño de la placa, se dispone de una variedad de herramientas de software; sin embargo, en esta ocasión se empleó EASY EDA. Este software ofrece una amplia biblioteca de componentes, lo que facilita significativamente el proceso de diseño de la PCB. En la figura 19 se presenta el diagrama esquemático del circuito correspondiente al nodo cliente, que constituye la base para la implementación del diseño.

Figura 19.

Esquemático del nodo cliente.

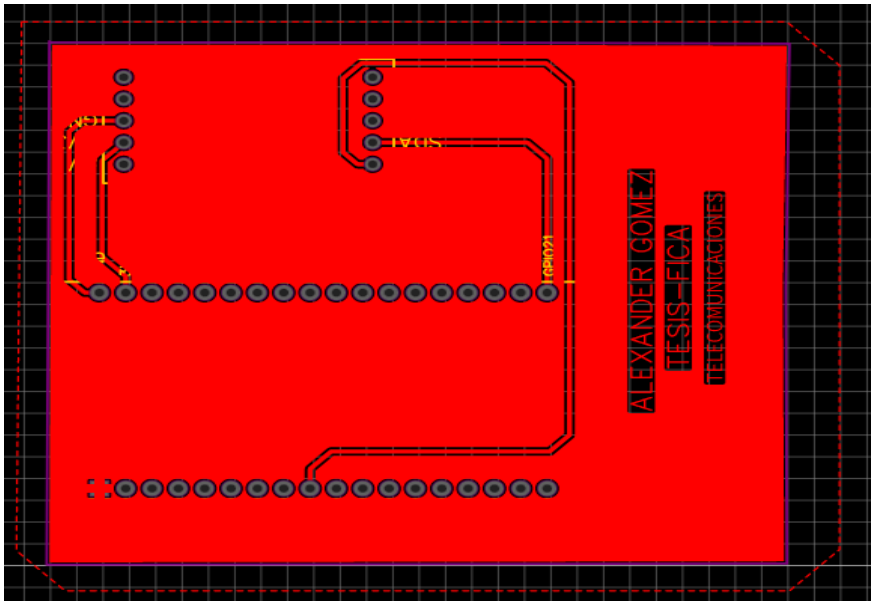


Nota. Esquema de conexión de componentes del nodo cliente. Fuente: Elaborado por el autor.

Una de las principales ventajas del software es su capacidad para generar un diseño de circuito impreso (PCB) que puede ser implementado en una baquelita, como se muestra en la figura 20. Adicionalmente, el software incluye la funcionalidad de generar archivos Gerber, ilustrados en la figura 21. Estos archivos permiten exportar el diseño y enviarlo para la fabricación de una placa con acabados más profesionales, garantizando mayor precisión y calidad en el producto final.

Figura 20.

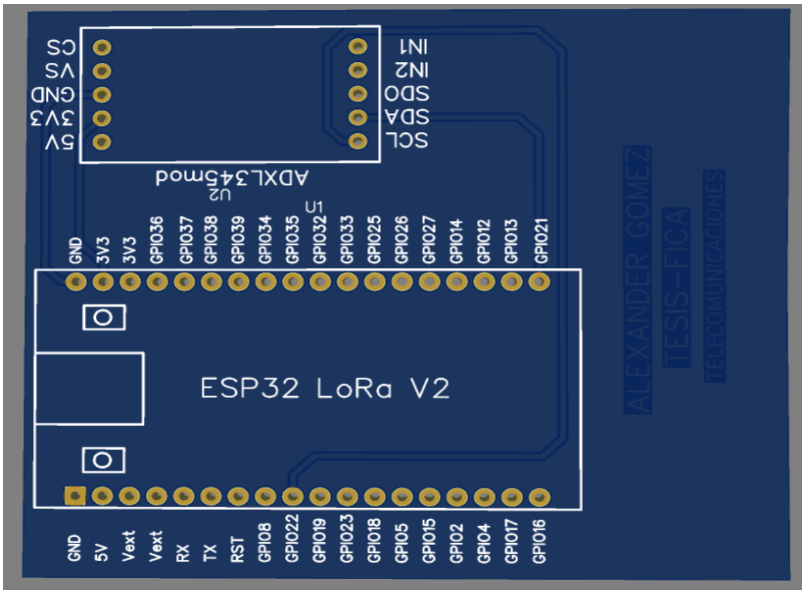
Diseño del circuito impreso del nodo cliente.



Nota. Puesta de cobre en el software de diseño del circuito impreso. Fuente: Elaborado por el autor

Figura 21.

Circuito en 3D del nodo cliente.



Nota. Diseño final en 3D del prototipo propuesto. Fuente: Elaborado por el autor.

3.3.7 Cálculo de comunicación LoRa.

La comunicación del prototipo se lleva a cabo utilizando la tecnología LoRa, lo que requiere un análisis detallado de parámetros clave como la pérdida de trayectoria y el presupuesto de enlace, entre otros. Por esta razón, en este apartado se pone énfasis en los cálculos necesarios para garantizar una comunicación efectiva entre los nodos. A continuación, se presentan los cálculos realizados durante el desarrollo del prototipo para la detección de celo.

3.3.7.1 Pérdidas de propagación en espacio libre

Las pérdidas de propagación en un enlace inalámbrico permiten estimar la atenuación que experimenta una señal al propagarse a través del aire. Este cálculo se lleva a cabo utilizando la ecuación 2, que se presenta a continuación.

$$L_p = 20log_{10}(d) + 20log_{10}(f) - 147.55 \qquad (2)$$

Donde:

d = distancia entre los nodos

f = frecuencia

Para este caso, se considera una distancia de 187 metros entre el nodo central y el nodo cliente, y una frecuencia de 915 MHz, que corresponde a una de las frecuencias de operación de la banda ISM. Sustituyendo estos valores en la ecuación de pérdidas por propagación en espacio libre, se obtiene lo siguiente:

$$L_p = 20log_{10}(187) + 20log_{10}(915x10^6) - 147.55$$

$$L_p = 45.4dB + 179.2\ dB - 147.55$$

$$L_p = 77.05\ dB$$

3.3.7.2 Cálculo de potencia de recepción

Este cálculo permite estimar la potencia que será recibida en los nodos, tomando como base la ecuación 3, la cual se presenta de la siguiente manera:

$$P_r = P_t + G_t + G_r - L_p \qquad (3)$$

Donde:

P_t = Potencia de transmisión

G_t = Ganancia de transmisión

G_r = Ganancia de recepción

L_p = Perdidas de propagación

En este ejemplo, se utilizan las características del módulo ESP32 LoRa V2, que ofrece una potencia de transmisión de 19 dBm. Las antenas integradas en el dispositivo poseen una ganancia de 2 dBi. La pérdida por propagación se considera a partir del cálculo realizado previamente, donde se obtuvo un valor de 77.05 dB. Con estos datos recopilados, se procede a realizar el cálculo, obteniendo como resultado lo siguiente:

$$P_r = 19\ dBm + 2dBi + 2dBi - 77.05\ dB$$

$$P_r = -53.05\ dBm$$

Se estima que la potencia de recepción será de -53.05 dBm. Según la ficha técnica del ESP32 LoRa V2, este dispositivo posee una sensibilidad de -135 dBm. A partir de esta información, se concluye que la comunicación entre los nodos no presentará inconvenientes relacionados con la potencia, ya que la señal recibida se encuentra muy por encima del umbral de sensibilidad del equipo.

3.3.7.3 Relación Señal-Ruido

Otro parámetro importante que considerar es la relación señal-ruido, conocida como SNR por sus siglas en inglés. Este valor es fundamental para evaluar la calidad y fiabilidad de la transmisión de datos. Su cálculo se realiza utilizando la ecuación 4, que se expresa de la siguiente manera:

$$SNR = \frac{P_r}{N} \qquad (4)$$

Donde:

P_r = es la potencia de recepción

N = Potencia del ruido

En este caso, el primer paso consiste en convertir la potencia de recepción de dBm a la unidad de medida en watts. Para realizar esta conversión, se emplea la ecuación 5, obteniendo como resultado lo siguiente:

$$P_r = 1W * 10^{\frac{P_{(dBm)}}{10}} \qquad (5)$$

En la sección anterior, se determinó que la potencia de recepción es de -53.05 dBm. Este valor se sustituye en la fórmula, quedando de la siguiente manera:

$$P_r = 1W * 10^{\frac{-53.05\ dBm}{10}}$$

$$P_r = 5x10^{-9} watts$$

Ahora que se tiene el valor de P_r debemos encontrar el valor de *N* para ello usamos la ecuación 6.

$$N = k * T * B \qquad (6)$$

Donde:

k = Constante de Boltzman

T = Temperatura estándar

B = Ancho de banda

Para este cálculo, se utilizan valores predefinidos para la constante de Boltzmann y la temperatura estándar, que son1.38 x 10^{-23} J/K y 290 K respectivamente. Además, el ancho de banda se toma como 125, según las especificaciones técnicas del ESP32 LoRa V2. Sustituyendo estos valores en la ecuación 6, se obtiene el siguiente resultado:

$$N = (1.38 * 10^{-23}) * 290 * 125$$

$$N = 5{,}0025x10^{-15}Watts$$

Una vez obtenidos los datos necesarios para el cálculo del SNR procedemos a reemplazar los datos en la ecuación 4 teniendo como resultado lo siguiente:

$$SNR = \frac{5x10^{-9}watts}{5{,}0025x10^{-15}watts}$$

$$SNR = 1.0x10^{9}$$

$$SNR(dB) = 10log_{10}(1.0x10^{9})$$

$$SNR(dB) = 90\ dB$$

El resultado obtenido es un valor de 90 dB para la relación señal-ruido (SNR), lo que indica que la comunicación es de buena calidad y presenta una probabilidad muy baja de errores en la transmisión de datos. Esto se debe a que la señal recibida es significativamente más fuerte que el ruido presente en el entorno.

3.3.7.4 Capacidad del canal

Finalmente, se procede a calcular la capacidad del canal, con el objetivo de determinar la cantidad de datos que pueden ser transmitidos en un tiempo dado. Este cálculo se realiza utilizando la ecuación 7, que se representa de la siguiente manera:

$$C = B * log_2(1 + SNR) \qquad (7)$$

Como se observa, se dispone de los valores calculados previamente. A continuación, se procede a sustituir estos valores en la ecuación 7, obteniendo como resultado lo siguiente:

$$C = 125\ Hz * log_2(1 + 1.0x10^9)$$

$$C = 125\ Hz * 29.9$$

$$C = 3737500\ bps\ => 3.74\ Mbps$$

3.4 FASE 3: Implementación.

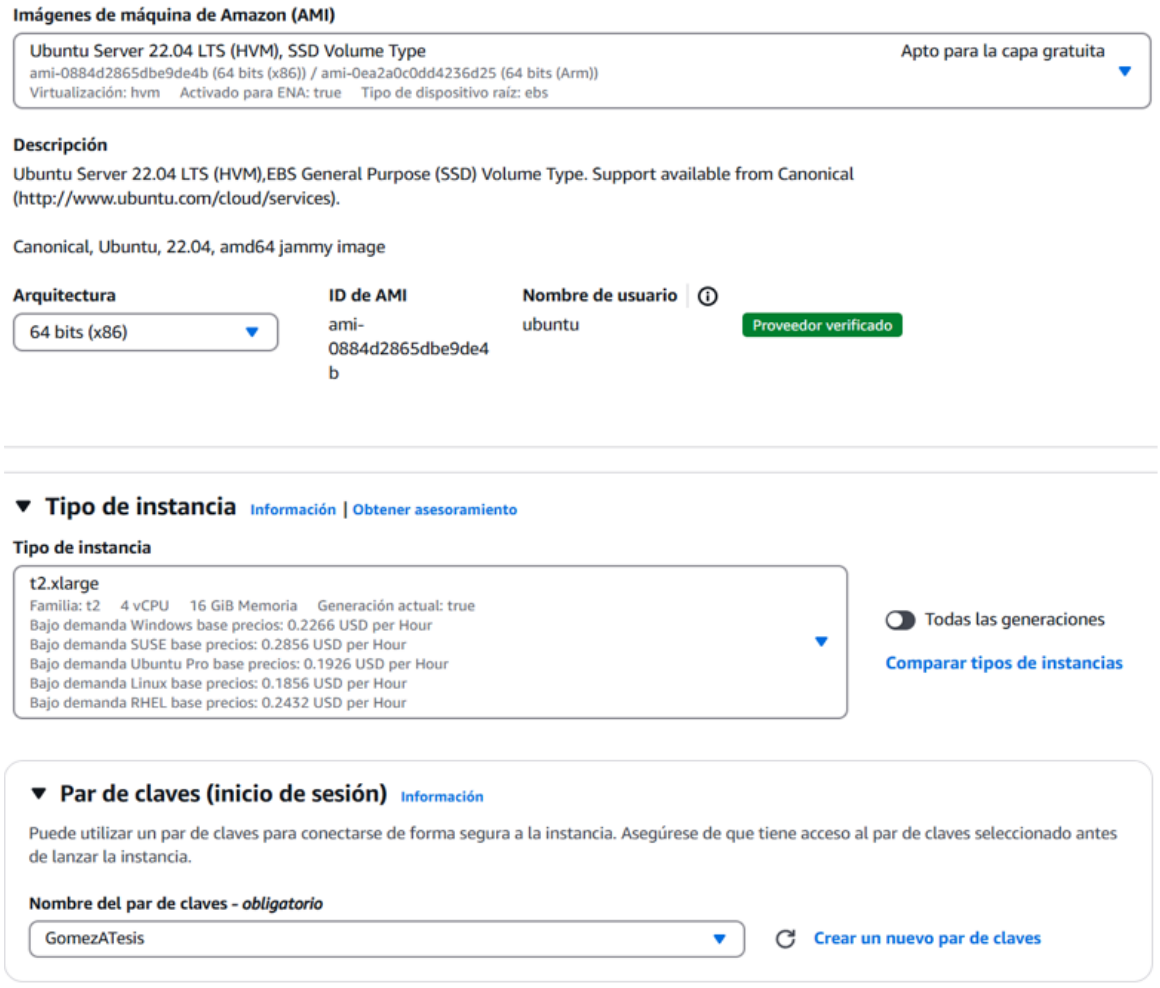
En la tercera fase del desarrollo del prototipo tenemos la implementación del prototipo, la cual está dividida en dos partes siendo estas la implementación del nodo central y la implementación del nodo cliente, la configuración de estas dos conforman el prototipo electrónico de detección de celo, a continuación, se detalla la implementación de cada uno de los nodos.

3.4.1 Instalación y configuración del servidor.

Para la configuración del servidor que alojará varios servicios necesarios para el desarrollo del proyecto, se utilizará la plataforma AWS. El primer paso consiste en crear una instancia, en la cual se debe seleccionar la distribución del sistema operativo, que en este caso será Ubuntu. En la figura 22 se muestran las configuraciones adicionales necesarias para la correcta creación de la instancia. Estas incluyen la elección de la imagen del sistema operativo, las características de procesamiento, como la CPU y la memoria RAM, y la creación de un par de claves. Estas claves permitirán la autenticación para acceder de forma remota al servidor.

Figura 22.

Configuración del servidor en AWS.

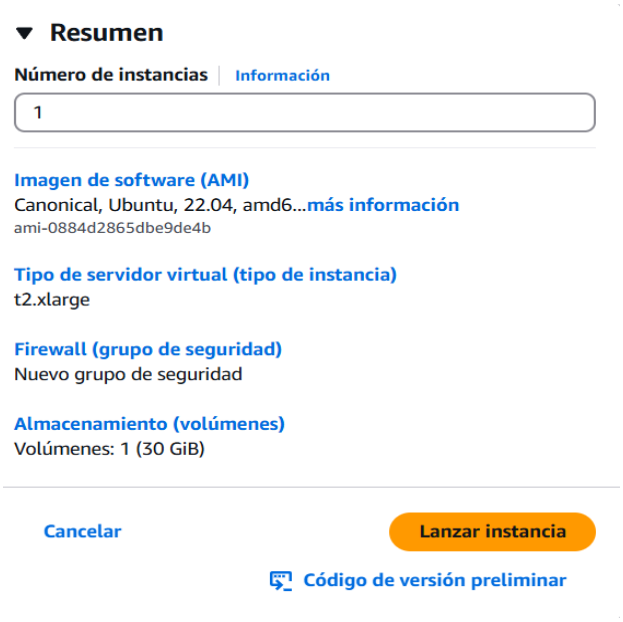


Nota. Configuración necesaria para la creación del servidor que estará alojado en la nube de AWS. Fuente: elaborado por el autor.

Una vez finalizadas las configuraciones iniciales, se mostrará una pantalla con el resumen de las configuraciones realizadas. En esta sección, se verificará que todos los parámetros sean correctos. Posteriormente, se seleccionará la opción de lanzar la instancia, completando así el proceso de creación del servidor.

Figura 23.

Resumen de las configuraciones del servidor.

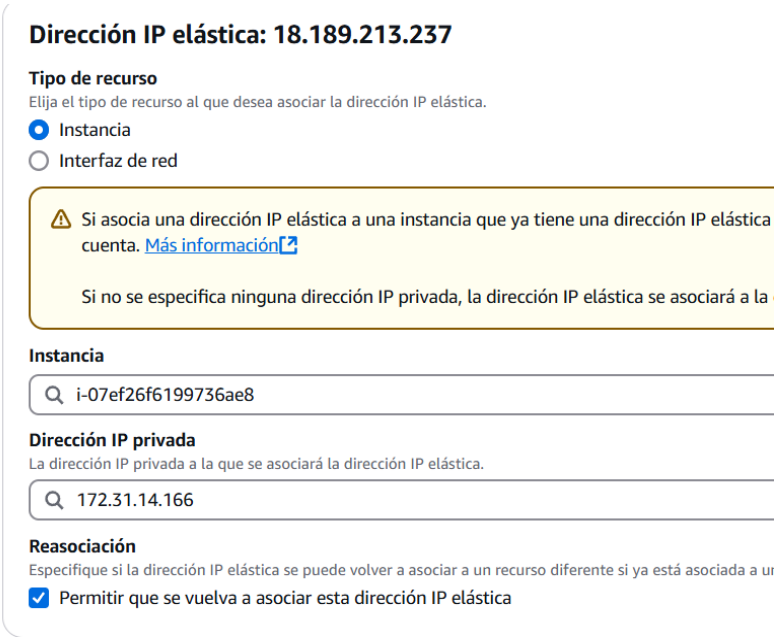


Fuente: elaborado por el autor.

Una vez finalizada la creación del servidor, se procederá a realizar configuraciones adicionales, como la asignación de una IP elástica. Esto se debe a que AWS asigna una IP pública de manera predeterminada, pero si el servidor se reinicia, esta IP cambia, lo que podría generar problemas en la transmisión de información. En el nodo central se debe especificar la IP del servidor, y si esta cambia, el nodo no podrá comunicarse correctamente con el servidor. Para evitar este inconveniente, se crea una IP elástica, la cual se mantendrá activa y fija, incluso en caso de reinicio del servidor. En la figura 24 se muestra el proceso de creación y asignación de la IP elástica a la instancia previamente creada.

Figura 24.

Creación y configuración de la IP elástica.



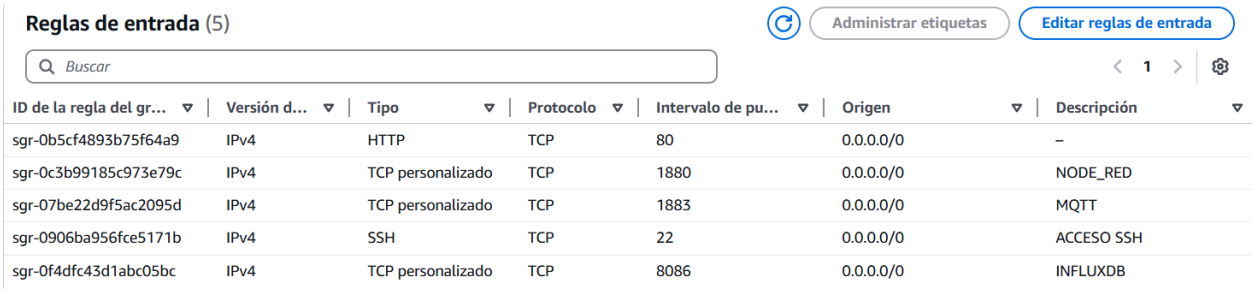
Fuente: Elaborado por el autor.

A continuación, se accede al apartado de *firewall* para crear las reglas de entrada necesarias. Estas reglas son esenciales, ya que cada servicio creado utiliza un puerto específico. Si las reglas no están configuradas, no será posible acceder a los servicios implementados.

En la figura 25 se muestra la creación de las reglas de entrada para los diferentes puertos, que incluyen: el puerto 22 para *SSH*, el puerto 80 para *HTTP*, el puerto 1883 para *MQTT*, el puerto 1880 para *Node-RED* y el puerto 8086 para *InfluxDB*. El origen se configura como *0.0.0.0/0*, permitiendo el acceso desde cualquier ubicación.

Figura 25.

Configuraciones del firewall de AWS.



ID de la regla del gr...	Versión d...	Tipo	Protocolo	Intervalo de pu...	Origen	Descripción
sgr-0b5cf4893b75f64a9	IPv4	HTTP	TCP	80	0.0.0.0/0	–
sgr-0c3b99185c973e79c	IPv4	TCP personalizado	TCP	1880	0.0.0.0/0	NODE_RED
sgr-07be22d9f5ac2095d	IPv4	TCP personalizado	TCP	1883	0.0.0.0/0	MQTT
sgr-0906ba956fce5171b	IPv4	SSH	TCP	22	0.0.0.0/0	ACCESO SSH
sgr-0f4dfc43d1abc05bc	IPv4	TCP personalizado	TCP	8086	0.0.0.0/0	INFLUXDB

Nota. Creación de reglas de firewall para la comunicación del nodo con el servidor alojado e AWS. Fuente: Elaborado por el autor.

3.4.2 Implementación del nodo central

Como primer paso se realiza la implementación del nodo central, ya que en primer lugar se debe tener los diferentes servidores funcionales para así poder realizar las pruebas de envío de la información del nodo cliente, a continuación, se detalla la instalación, configuración de los diferentes servidores necesarios para almacenar y procesar los datos obtenidos por el nodo cliente.

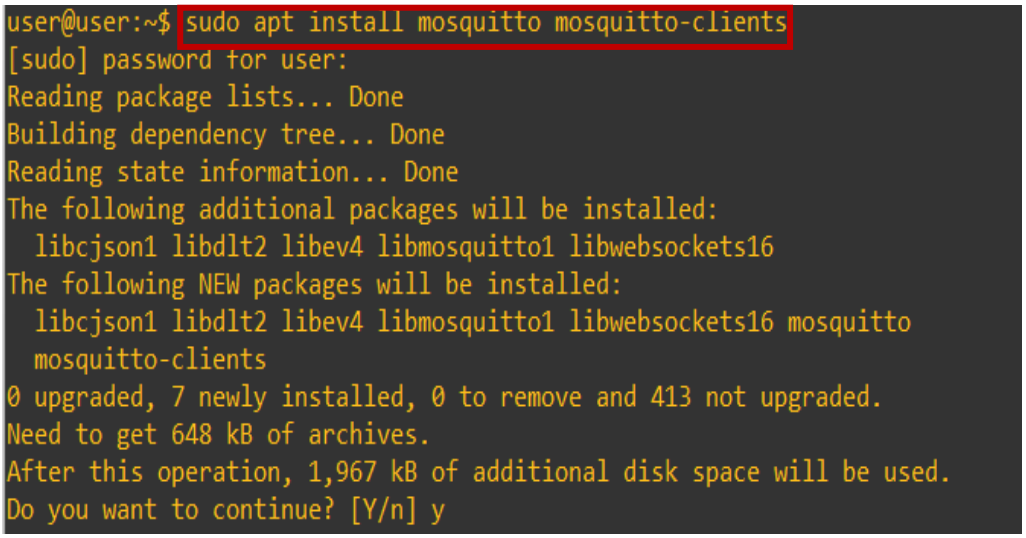
3.4.2.1 Instalación y configuración del servicio MQTT

La comunicación entre los nodos es un aspecto crucial en el desarrollo del prototipo. Por esta razón, se llevará a cabo la instalación y configuración de un servidor MQTT, con el objetivo de optimizar los recursos de cada nodo, considerando que los microprocesadores empleados cuentan con capacidades limitadas.

Para implementar el servidor MQTT, se utiliza un equipo con una distribución Linux. En este servidor se instalará Mosquitto como broker de comunicación MQTT. Para ello, se debe ejecutar el comando que se puede apreciar en la

Figura 26.

Instalación del servidor mqtt.



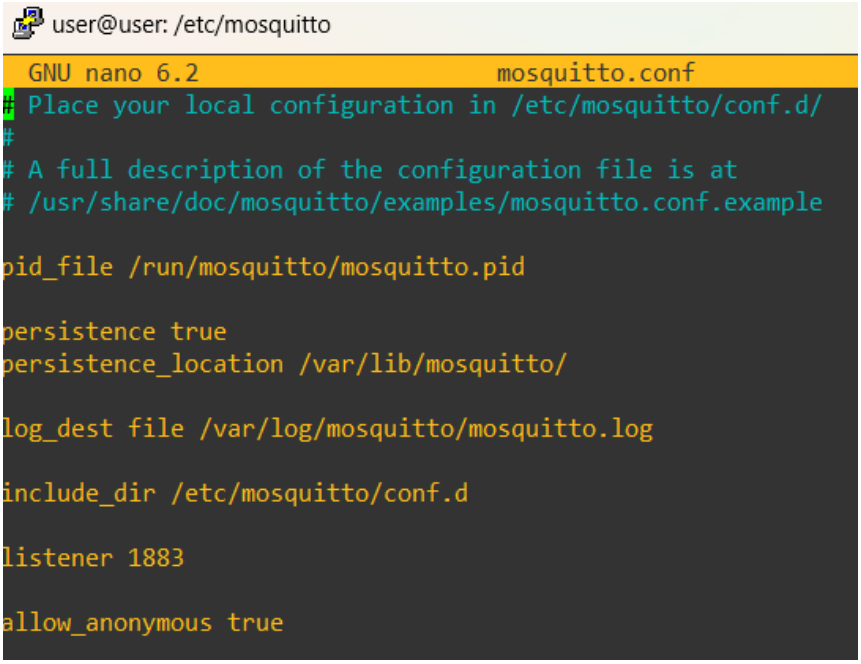
Fuente: Elaborador por el autor.

Una vez completada la instalación del servidor MQTT, se procede a realizar las configuraciones necesarias para permitir el envío de datos desde el nodo central hacia el servidor. Para ello, se accede al directorio **/etc/mosquitto** y se edita el archivo **mosquitto.conf** utilizando un editor de texto.

En este archivo, se debe especificar el puerto de escucha del broker y habilitar las conexiones sin autenticación. Después de realizar estos ajustes, el archivo de configuración debe quedar como se muestra en la figura 27.

Figura 27.

Configuración del servidor mqtt.



Nota. Configuración del archivo mosquitto.conf, en donde se acepta la comunicación mqtt sin necesidad de autentificación. Fuente: elaborado por el autor.

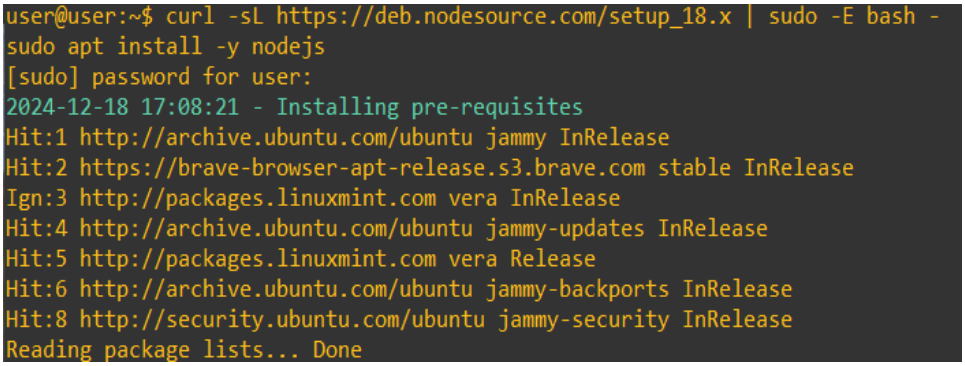
3.4.2.2 Instalación y configuración de NODE-RED

Para el manejo y procesamiento de los datos, se empleará el servidor **Node-RED**, el cual facilita la manipulación de los datos y los transforma en un formato legible para su almacenamiento en la base de datos.

Para lograr esto, se instalan los prerrequisitos necesarios para su correcto funcionamiento. La figura 28 muestra los comandos utilizados durante este proceso de instalación y configuración.

Figura 28.

Instalación del servidor Node-Red



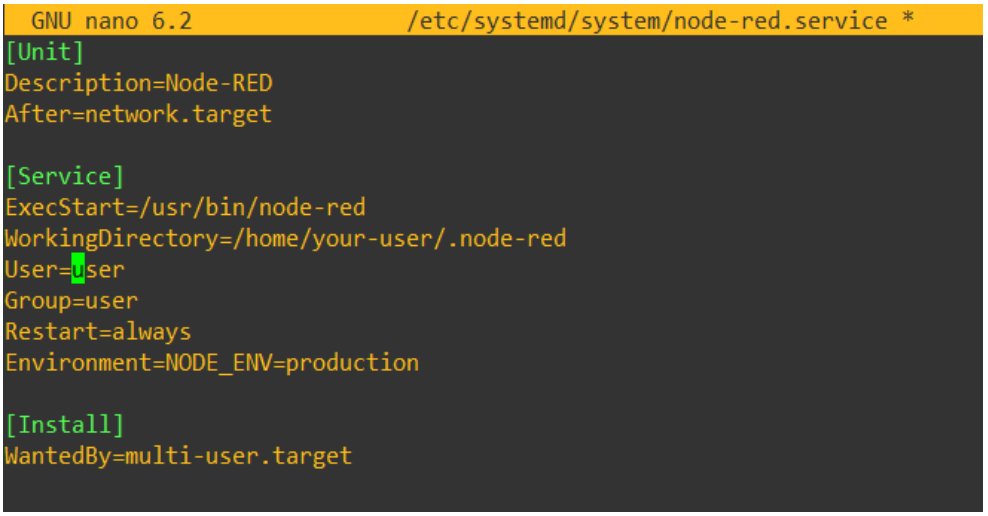
Fuente: Elaborado por el autor.

A continuación, se realizará la configuración para que el servidor Node-RED se inicie automáticamente junto con el sistema operativo, evitando así la necesidad de activar manualmente el servicio. Para ello, se accede al directorio **/etc/systemd/system/** y se crea un archivo denominado node-red. service.

Una vez creado el archivo, se procede a editarlo para configurar el arranque automático, definir la política de reinicio en caso de fallo y especificar las variables de entorno que se pasan al servicio al momento de su inicio. Todos estos detalles se pueden observar en la figura 29.

Figura 29.

Configuración del servidor Node-red.

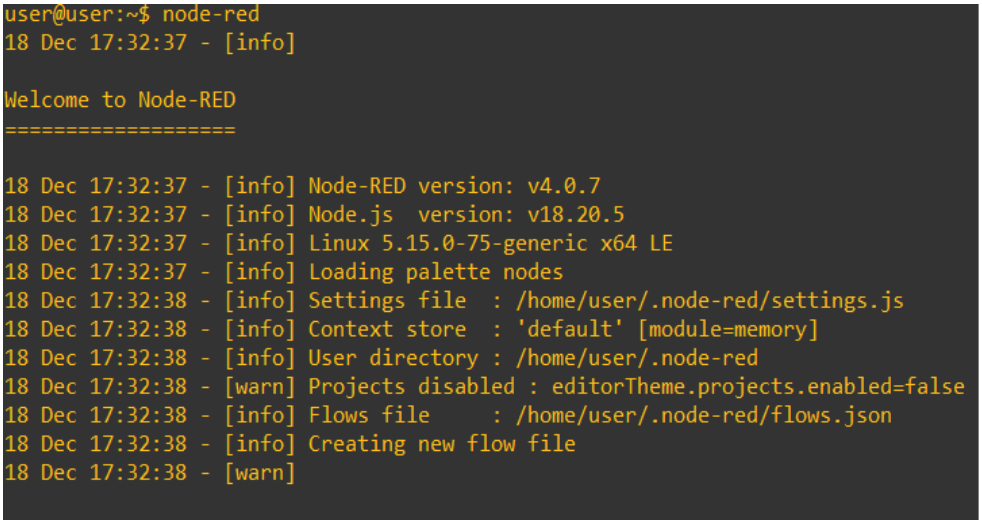


Nota. Archivo de configuración para el arranque del servicio con el sistema de Linux. Fuente: Elaborado por el autor.

Finalmente, se procede a verificar el estado del servidor de Node-RED utilizando el comando correspondiente: node-red. El resultado obtenido se puede observar en la figura 30, donde se muestra la versión del servidor que se está ejecutando; en este caso, corresponde a la versión 4.0.7. En caso de que surja algún problema al intentar iniciar el servidor, en esta ventana se podrá identificar el error específico que está ocurriendo.

Figura 30.

Estado del servicio de node-red

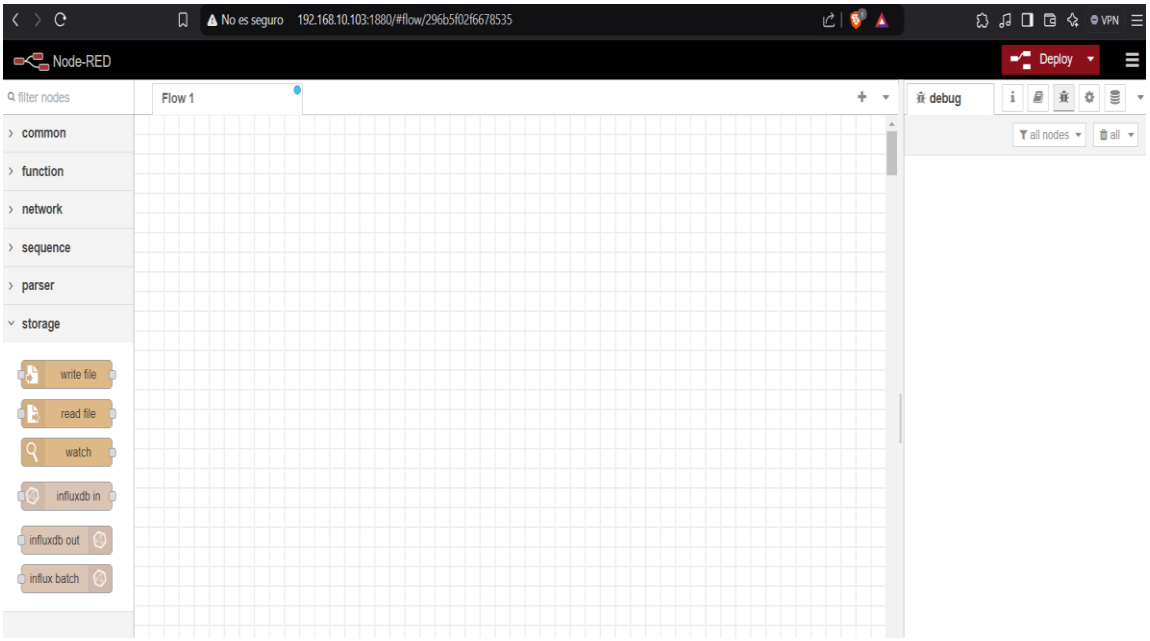


Fuente: Elaborado por el autor.

Una vez verificado el correcto funcionamiento del servidor, se procede a comprobar su operatividad mediante la interfaz web. Para ello, se debe acceder a la siguiente URL: http://dirección-ip:puerto. Si la configuración se ha realizado de manera correcta, se debería visualizar una interfaz similar a la mostrada en la figura 31. En dicha interfaz, se puede observar en el lado izquierdo los nodos necesarios para la implementación del prototipo. Por su parte, en el lado derecho se encuentra la consola de salida y el botón *Deploy*, el cual permite guardar y aplicar los cambios realizados en el proyecto.

Figura 31.

Interfaz gráfica de node-red.



Nota. Panel principal del servicio de node-red. Fuente: Elaborado por el autor.

3.4.2.3 Instalación y configuración de MySQL.

Para almacenar los datos enviados por el sensor, se utilizará la base de datos que obtuvo la mejor calificación en la fase de selección de componentes. Para llevar a cabo la instalación, primero se realizará una actualización mediante el comando apt update. A continuación, se procederá a instalar MySQL; el comando utilizado puede observarse en la figura 32, donde se inicia automáticamente la descarga de los paquetes necesarios.

Figura 32.

Instalación de My SQL.

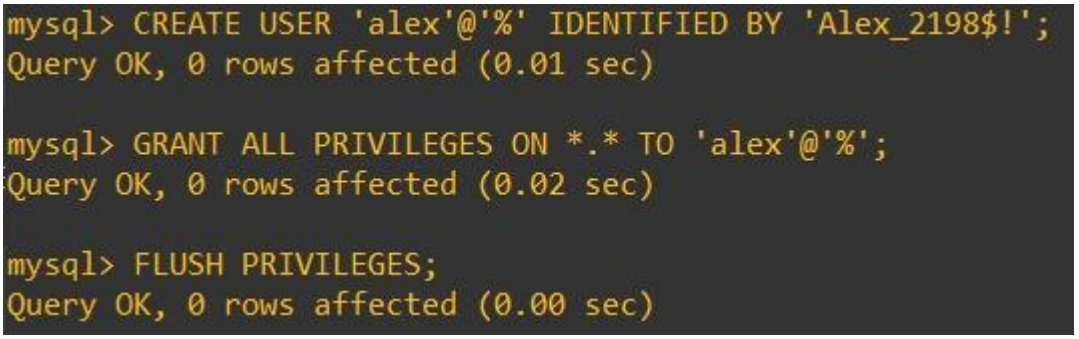
```
ubuntu@ip-172-31-14-166:~$ sudo apt install mysql-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Fuente: Elaborador por el autor.

Una vez finalizada la instalación, se procederá a acceder a la base de datos ingresando como root mediante el comando sudo mysql. Una vez dentro de la base de datos, se creará un usuario con su respectiva contraseña, a continuación, se le asignarán los privilegios que tendrá este usuario. En esta ocasión, se le otorgarán todos los privilegios. Para finalizar, se utilizará el comando FLUSH PRIVILEGES, lo que garantiza que la creación del usuario y la asignación de permisos se guarden correctamente.

Figura 33.

Creación de usuario en My SQL.



Fuente: Elaborador por el autor.

3.4.2.4 Programación del nodo central

Para la programación del nodo central, es necesario realizar la instalación de las librerías requeridas. En este caso, las librerías necesarias se detallan en la figura 34. Las dos primeras librerías son esenciales para establecer la comunicación mediante el protocolo inalámbrico LoRa. La tercera librería se utiliza para gestionar la conexión Wi-Fi del nodo central. Finalmente, se incluye una librería que permite interactuar con el servidor MQTT, facilitando la publicación y suscripción de mensajes en el sistema.

Figura 34.

Librerías del nodo central.

```
#include "LoRaWan_APP.h"
#include "Arduino.h"
#include <WiFi.h>
#include <PubSubClient.h>
```

Fuente: Elaborador por el autor.

Luego tenemos el bloque de código que configura un sistema de comunicación LoRa. Se establece una frecuencia de radio de 915 MHz, un ancho de banda de 125 kHz, un factor de expansión de 7, una tasa de codificación de 4/5 y una longitud del preámbulo de 8. La longitud del payload se define como variable, y la inversión IQ se encuentra desactivada. Además, se configura un tiempo de espera (timeout) para la recepción de datos de 1000 ms, junto con un buffer de recepción de 64 bytes, donde los datos se almacenan en un array denominado rxpacket. Estos parámetros están diseñados para optimizar la transmisión y recepción de datos, ajustándose a las especificaciones del protocolo LoRa.

Figura 35.

Parámetros de la comunicación LoRa_Nodo_Central.

```
#define RF_FREQUENCY 915000000
#define LORA_BANDWIDTH 0
#define LORA_SPREADING_FACTOR 7
#define LORA_CODINGRATE 1
#define LORA_PREAMBLE_LENGTH 8
#define LORA_FIX_LENGTH_PAYLOAD_ON false
#define LORA_IQ_INVERSION_ON false
#define RX_TIMEOUT_VALUE 1000
#define BUFFER_SIZE 64
```

Fuente: Elaborador por el autor.

En la figura 36 se aprecia como se establece la configuración para la conexión WiFi y MQTT. Primero, se define el nombre (SSID) y la contraseña de la red WiFi a la que el dispositivo se conectará. Posteriormente, se especifica la dirección del servidor MQTT, el puerto correspondiente y el tópico al que se enviarán los mensajes. Para gestionar la conexión WiFi, se utiliza un objeto WiFiClient llamado espClient, y para la conexión MQTT, se emplea un objeto PubSubClient denominado client, al cual se le pasa espClient como argumento. Este enfoque permite gestionar de manera eficiente ambas conexiones de red, asegurando una comunicación estable entre el nodo y el servidor.

Figura 36.

Credenciales wifi y mqtt.

```
const char* ssid = "Omega_Gomez";
const char* password = "1711419075";
const char* mqtt_server = "192.168.10.103";
const int mqtt_port = 1883;
const char* mqtt_topic = "test";
WiFiClient espClient;
PubSubClient client(espClient);
```

Fuente: Elaborador por el autor.

Una de las funciones implementadas en el código es la función reconnect () la cual la podemos apreciar en la figura 37, cuyo propósito es mantener la conexión con el servidor MQTT. Esta función emplea un bucle while para verificar constantemente si el cliente MQTT está conectado. Si no lo está, se imprime un mensaje en el monitor serial indicando que se está intentando establecer la conexión. En caso de que la conexión sea exitosa, se imprime el mensaje "Conectado al servidor MQTT". Si la conexión falla, se muestra un mensaje de error con el código de estado del cliente y se intenta reconectar nuevamente después de 5 segundos, utilizando delay (5000). Esta función garantiza que el dispositivo permanezca siempre conectado

al servidor MQTT, reiniciando el proceso de reconexión en caso de ser necesario, lo que asegura la estabilidad de la comunicación.

Figura 37.

Función de reconexión mqtt.

```
void reconnect() {
  while (!client.connected()) {
    Serial.print("Conectando al servidor MQTT...");
    if (client.connect("LoRaReceiver")) {
      Serial.println("Conectado al servidor MQTT");
    } else {
      Serial.print("Error, rc=");
      Serial.print(client.state());
      Serial.println(" intentando nuevamente en 5 segundos...");
      delay(5000);
    }
  }
}
```

Fuente: Elaborador por el autor.

Dentro de la función setup la cual se puede apreciar en la figura 38, se inicializa y configura el dispositivo LoRa. El proceso comienza con la configuración de la comunicación serial a 9600 baudios y la especificación de la placa y el tipo de reloj del microcontrolador (MCU). A continuación, se asignan los manejadores de eventos a RadioEvents para gestionar los distintos estados de la comunicación, como la recepción exitosa (OnRxDone), el tiempo de espera (OnRxTimeout), y los errores de recepción (OnRxError). Luego, se inicializa el radio utilizando el método Radio.Init(&RadioEvents), se establece la frecuencia del canal con Radio.SetChannel, y se configuran los parámetros de recepción específicos para LoRa, como el ancho de banda, el factor de expansión, la tasa de codificación, entre otros. Finalmente, se inicia la recepción con Radio.Rx(RX_TIMEOUT_VALUE) y se imprime un mensaje en el monitor

serial indicando que el receptor LoRa está listo y esperando paquetes. Este proceso asegura que el dispositivo esté correctamente configurado para recibir datos a través de LoRa.

Figura 38.

Configuración del dispositivo LoRa.

```
void setup() {
  Serial.begin(9600);
  Mcu.begin(HELTEC_BOARD, SLOW_CLK_TPYE);
  RadioEvents.RxDone = OnRxDone;
  RadioEvents.RxTimeout = OnRxTimeout;
  RadioEvents.RxError = OnRxError;
  Radio.Init(&RadioEvents);
  Radio.SetChannel(RF_FREQUENCY);
  Radio.SetRxConfig(MODEM_LORA, LORA_BANDWIDTH, LORA_SPREADING_FACTOR, LORA_CODINGRATE,
                    0, LORA_PREAMBLE_LENGTH, 0,
                    LORA_FIX_LENGTH_PAYLOAD_ON, 0, true, 0, 0,
                    LORA_IQ_INVERSION_ON, true);
  Radio.Rx(RX_TIMEOUT_VALUE);
  Serial.println("Receptor LoRa iniciado. Esperando paquetes...");
```

Fuente: Elaborador por el autor.

Dentro de la misma función setup (), se configura la conexión a una red WiFi y a un servidor MQTT. En la figura 39, se pueden observar las configuraciones de programación realizadas. El proceso comienza con el intento de conexión a la red WiFi utilizando el SSID y la contraseña previamente especificados. Mientras no se establezca la conexión (es decir, mientras WiFi.status()!= WL_CONNECTED), el dispositivo se retrasa durante un segundo (delay(1000)) y muestra el mensaje "Conectando al WiFi..." en el monitor serial. Una vez que la conexión WiFi se haya establecido correctamente, se imprime el mensaje "Conexión WiFi establecida." en el monitor serial. A continuación, se configura el cliente MQTT, especificando la dirección IP y el puerto del servidor MQTT. Seguido de esto, se llama a la función reconnect () para garantizar que el cliente MQTT se conecte correctamente y mantenga la conexión activa. Este enfoque

asegura que tanto la conexión WiFi como la conexión MQTT se establezcan de manera confiable en el inicio del dispositivo.

Figura 39.

Función para la conexión WiFi y MQTT.

```
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.println("Conectando al WiFi...");
}
Serial.println("Conexión WiFi establecida.");
client.setServer(mqtt_server, mqtt_port);
reconnect();
```

Fuente: Elaborador por el autor.

Finalmente, se encuentran varias funciones que se pueden observar en la figura 40. Comenzando con la función loop (), esta asegura que el cliente MQTT permanezca conectado, llamando a reconnect() en caso de que no lo esté. Además, gestiona la comunicación mediante client. loop() y Radio.IrqProcess().

La función OnRxDone () se ejecuta cuando se recibe un paquete, copiando los datos al buffer rxpacket y mostrando el paquete recibido, junto con los valores de RSSI y SNR, en el monitor serial. Si el cliente MQTT está conectado, publica los datos recibidos en el tópico especificado.

La función OnRxTimeout () se llama cuando se agota el tiempo de recepción, imprimiendo un mensaje en el monitor serial y reiniciando la recepción de radio.

Por último, la función OnRxError () se ejecuta cuando ocurre un error en la recepción, imprimiendo un mensaje de error y reiniciando la recepción.

Estas funciones aseguran una correcta gestión de la comunicación LoRa y MQTT, así como la gestión de errores y tiempos de espera durante el proceso de recepción.

Figura 40.

Funciones para el envío y recepción de datos del nodo central.

```
void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  Radio.IrqProcess();
}
void OnRxDone(uint8_t *payload, uint16_t size, int16_t rssi, int8_t snr) {
  memcpy(rxpacket, payload, size);
  rxpacket[size] = '\0';
  Serial.printf("\nPaquete recibido: \"%s\"\n", rxpacket);
  Serial.printf("RSSI: %d dBm, SNR: %d dB\n", rssi, snr);
  if (client.connected()) {
    client.publish(mqtt_topic, rxpacket);
    Serial.println("Mensaje enviado a MQTT.");
  } else {
    Serial.println("No conectado a MQTT.");
  }
  Radio.Rx(RX_TIMEOUT_VALUE);
}
void OnRxTimeout(void) {
  Serial.println("Tiempo de recepción agotado.");
  Radio.Rx(RX_TIMEOUT_VALUE);
}
void OnRxError(void) {
  Serial.println("Error en la recepción del paquete.");
  Radio.Rx(RX_TIMEOUT_VALUE);
}
```

Fuente: Elaborador por el autor.

3.4.3 Implementación del nodo cliente

En el contexto del monitoreo de la actividad de la vaca, el nodo cliente desempeña un papel crucial, ya que es el dispositivo encargado de captar las lecturas provenientes de los

sensores y enviar los datos utilizando tecnología de comunicación LoRa. Este nodo debe ser capaz de realizar la adquisición de datos de manera eficiente y confiable, considerando factores como la precisión en las lecturas y la autonomía de la energía, para que la transmisión de información hacia el nodo central se realice sin interrupciones.

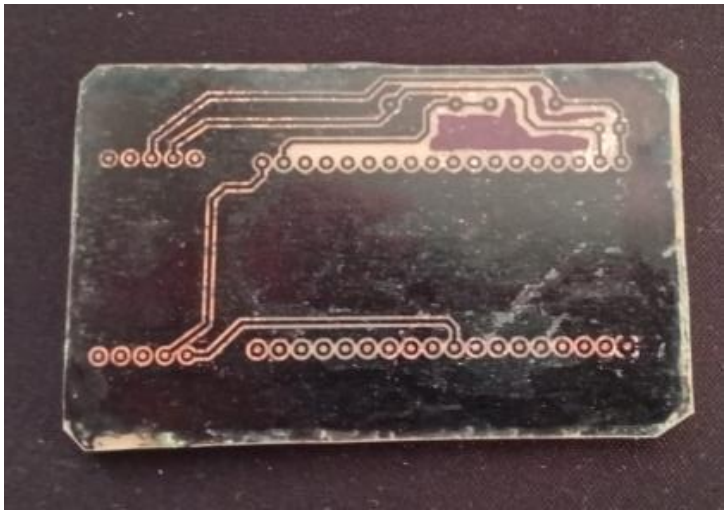
La correcta implementación del nodo cliente es esencial para asegurar que los datos recolectados sean fiables y se transmitan sin fallos al sistema centralizado. Esta transmisión precisa y continua es fundamental para que el análisis y la visualización de la información en el servidor sean efectivos, lo que permitirá tomar decisiones informadas sobre la salud, el comportamiento y el estado general de las vacas.

3.4.3.1 Construcción de la placa para el nodo cliente

Para la construcción del nodo cliente se utilizará una baquelita, sobre la cual se pegará el diseño de la PCB previamente impreso. El diseño se transferirá a la placa de cobre utilizando una plancha de calor para que quede correctamente plasmado. En caso de errores o áreas donde el diseño no se haya transferido de manera adecuada, se utilizará un marcador permanente para corregir y resaltar las secciones necesarias. En la figura 41 se muestra el diseño del circuito impreso ya transferido a la baquelita.

Figura 41.

Circuito impreso en la baquelita.

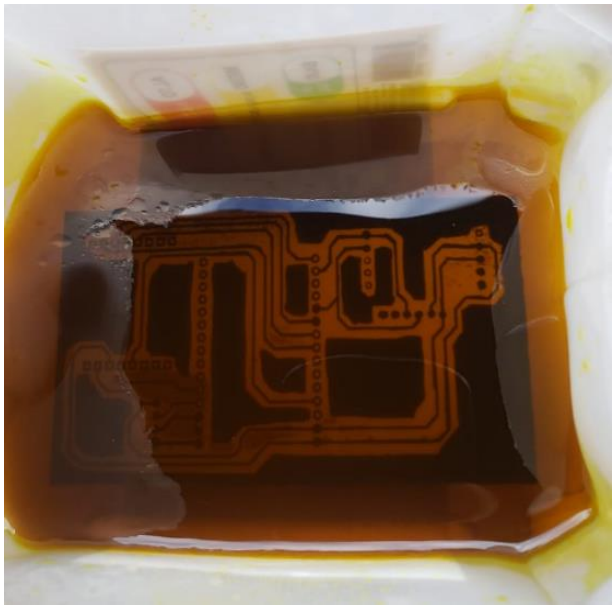


Fuente: Elaborador por el autor.

Seguido vamos a quitar el excedente de cobre para ello usamos acido férrico, el cual se deberá colocar en un recipiente de plástico para que se elimine todo el cobre restante.

Figura 42.

Eliminación del excedente de cobre.

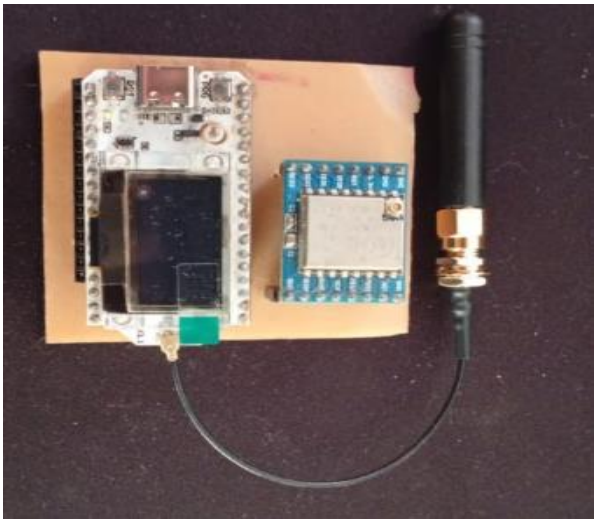


Fuente: Elaborador por el autor.

Una vez quitado el excedente del cobre se procede a soldar los componentes en la placa, para ello nos guiaremos en el diseño de la PCB mostrado anteriormente.

Figura 43.

Colocación de componentes en la placa.



Fuente: Elaborador por el autor.

Finalmente se indica el collar que será usado para colocarlo en la vaca para realizar las pruebas de detección de celo, la misma que está diseñada para soportar los factores ambientales y durabilidad de los componentes.

Figura 44.

Collar del prototipo.

Fuente: Elaborador por el autor.

Para realizar las pruebas del prototipo, se selecciona una vaca y se coloca en una manga, lo que permite acercarse de manera segura para colocar el collar en la posición adecuada. En la figura 45 se observa al trabajador de la hacienda colocando el collar en el cuello de la vaca, dejando así el prototipo listo para llevar a cabo el monitoreo y la detección del celo.

Figura 45.

Collar colocado en la vaca.

Fuente: Elaborador por el autor.

3.4.3.2 Programación del nodo cliente

El código presentado en la figura 46 utiliza varias bibliotecas esenciales para implementar funciones de comunicación. La biblioteca Wire facilita la comunicación I2C, que es crucial para la interacción entre el microcontrolador y dispositivos periféricos como sensores y módulos. Además, se emplea la biblioteca LoRaWAN, que permite gestionar la comunicación a largo alcance utilizando la tecnología LoRa, y las funciones básicas propias de la programación en Arduino. Con estas bibliotecas, el proyecto adquiere la capacidad de manejar tanto la comunicación de corto alcance (I2C) como la de largo alcance (LoRa), lo que es fundamental para el funcionamiento del sistema en diferentes contextos de conectividad.

Figura 46.

Librerías del nodo cliente.

```
#include <Wire.h>
#include "LoRaWan_APP.h"
#include "Arduino.h"
```

Fuente: Elaborador por el autor.

En un sistema LoRa, se configuran varios parámetros clave para garantizar la transmisión eficiente. La frecuencia de radio determina el canal de comunicación, mientras que la potencia de salida regula la cobertura. El ancho de banda afecta la velocidad de transmisión, y el factor de expansión influye en la sensibilidad y el alcance, sacrificando la velocidad de datos. La tasa de corrección de errores mejora la fiabilidad, especialmente en entornos ruidosos.

Además, la longitud del preámbulo es importante para la sincronización de la señal, y el tamaño del buffer de recepción determina cuántos datos se pueden almacenar temporalmente. Estas configuraciones se pueden apreciar en la figura 47, en donde se debe tener en cuenta proporcionar los mismos valores que del nodo central esto con la finalidad de que puedan entablar una comunicación entre los nodos.

Figura 47.

Parámetros LoRa para el nodo cliente.

```
#define RF_FREQUENCY 915000000
#define TX_OUTPUT_POWER 5
#define LORA_BANDWIDTH 0
#define LORA_SPREADING_FACTOR 7
#define LORA_CODINGRATE 1
#define LORA_PREAMBLE_LENGTH 8
#define LORA_FIX_LENGTH_PAYLOAD_ON false
#define LORA_IQ_INVERSION_ON false
#define BUFFER_SIZE 64
```

Fuente: Elaborador por el autor.

En el desarrollo del prototipo, uno de los parámetros cruciales en el nodo cliente es la disponibilidad del dispositivo, lo cual se garantiza mediante la implementación de una función para monitorear el voltaje de la batería. Esta función hace uso del pin analógico 34 del ESP32 para obtener la lectura del voltaje, el cual se transforma en un porcentaje que refleja el estado de carga de la batería.

El proceso comienza con la captura del valor analógico a través de la función analogRead (34), que devuelve un valor en el rango de 0 a 4095, según la resolución del ADC del ESP32.

A continuación, este valor se convierte en un voltaje al dividirlo por 4095 y multiplicarlo por 3.3, que es el voltaje de referencia del microcontrolador. Debido a que se utiliza un divisor de voltaje en el circuito, se aplica un factor adicional de 2 para obtener el valor de la batería.

El siguiente paso es mapear este voltaje, que varía entre 3.0V y 4.2V, a un porcentaje en el rango de 0% a 100% mediante la función map (voltage * 100, 600, 840, 0, 100), donde los valores 600 y 840 representan los valores de voltaje en milivoltios correspondientes a 3.0V y 4.2V, respectivamente. Finalmente, el porcentaje se ajusta para asegurarse de que no se sobrepase los límites de 0% o 100% antes de devolver el valor.

Figura 48.

Función de lectura de voltaje de la batería externa.

```
int getBatteryPercentage() {
  int raw = analogRead(34);
  float voltage = (raw / 4095.0) * 3.3 * 2;
  int percentage = map(voltage * 100, 600, 840, 0, 100);
  if (percentage > 100) percentage = 100;
  if (percentage < 0) percentage = 0;
  return percentage;
}
```

Fuente: Elaborador por el autor.

Una vez declaradas las bibliotecas y variables necesarias, se procede a inicializar la configuración del sistema. En primer lugar, se establece la comunicación I2C, permitiendo la interacción con los componentes conectados. Luego, se inicia la comunicación con el sensor acelerómetro, enviando comandos a los registros 0x2D para activar el sensor y al 0x31 para configurar el rango de datos.

Posteriormente, se establece la comunicación serial a una velocidad de 9600 baudios y se inicializa el microcontrolador. A continuación, se definen los eventos para la transmisión de datos (OnTxDone) y el manejo de tiempos de espera (OnTxTimeout).

Finalmente, se configura el módulo LoRa, ajustando parámetros esenciales como el canal de frecuencia, la potencia de salida, el ancho de banda, el factor de expansión, la tasa de codificación, la longitud del preámbulo, la longitud del payload y la configuración de inversión IQ.

Figura 49.

Inicialización de procesos del sistema.

```
void setup() {
  VextON();
  delay(100);
  Wire1.begin(sdaPin, sclPin, 400000);
  Wire1.beginTransmission(ADXL345);
  Wire1.write(0x2D);
  Wire1.write(0x08);
  Wire1.endTransmission(true);
  Wire1.beginTransmission(ADXL345);
  Wire1.write(0x31);
  Wire1.write(0x00);
  Wire1.endTransmission(true);
  Serial.begin(9600);
  Mcu.begin(HELTEC_BOARD, SLOW_CLK_TPYE);
  RadioEvents.TxDone = OnTxDone;
  RadioEvents.TxTimeout = OnTxTimeout;
  Radio.Init(&RadioEvents);
  Radio.SetChannel(RF_FREQUENCY);
  Radio.SetTxConfig(MODEM_LORA, TX_OUTPUT_POWER, 0, LORA_BANDWIDTH,
                    LORA_SPREADING_FACTOR, LORA_CODINGRATE,
                    LORA_PREAMBLE_LENGTH, LORA_FIX_LENGTH_PAYLOAD_ON,
                    true, 0, 0, LORA_IQ_INVERSION_ON, 3000);
}
```

Fuente: Elaborador por el autor.

El sensor utilizado será un acelerómetro, el cual proporcionará la variación del movimiento a lo largo de los ejes X, Y y Z. Estos datos deberán ser normalizados para facilitar su manejo en etapas posteriores del proceso. La normalización de los valores asegura que las mediciones sean consistentes y comparables, independientemente de la magnitud de las variaciones. Este proceso se ilustra claramente en la figura 50, donde se muestra cómo se ajustan los datos para su posterior análisis.

Figura 50.

Lectura y normalización de datos del sensor acelerómetro.

```
Wire1.beginTransmission(ADXL345);
Wire1.write(0x32);
Wire1.endTransmission(false);
Wire1.requestFrom(ADXL345, 6, true);
if (Wire1.available() == 6) {
  AcX = Wire1.read() | (Wire1.read() << 8);
  AcY = Wire1.read() | (Wire1.read() << 8);
  AcZ = Wire1.read() | (Wire1.read() << 8);

  if (AcX > 32767) AcX -= 65536;
  if (AcY > 32767) AcY -= 65536;
  if (AcZ > 32767) AcZ -= 65536;

  float normalizedAcX = AcX / 256.0;
  float normalizedAcY = AcY / 256.0;
  float normalizedAcZ = AcZ / 256.0;
```

Fuente: Elaborador por el autor.

Finalmente, se procede a preparar y enviar los datos a través del módulo LoRa. Utilizando la función sprintf, se crea un mensaje formateado que incluye las lecturas normalizadas de los ejes X, Y y Z, así como el porcentaje de batería restante. Este mensaje se almacena en la variable txpacket. Luego, se imprime en el monitor serial el contenido del mensaje y su longitud mediante Serial.printf, lo que permite verificar los datos antes de enviarlos.

Una vez que la información está lista, se envía a través del módulo LoRa utilizando Radio.Send. Además, se establece la variable lora_idle en false, lo que indica que el proceso de envío está en curso y que el sistema no está disponible para otras tareas hasta que el envío finalice.

Figura 51.

Preparación y envió de datos.

```
sprintf(txpacket, "X:%.2f Y:%.2f Z:%.2f Battery:%d%%", normalizedAcX, normalizedAcY, normalizedAcZ, batteryPercentage);
Serial.printf("Enviando paquete: \"%s\", longitud: %d\n", txpacket, strlen(txpacket));
Radio.Send((uint8_t *)txpacket, strlen(txpacket));
lora_idle = false;
```

Fuente: Elaborador por el autor.

3.4.4 Procesamiento de datos

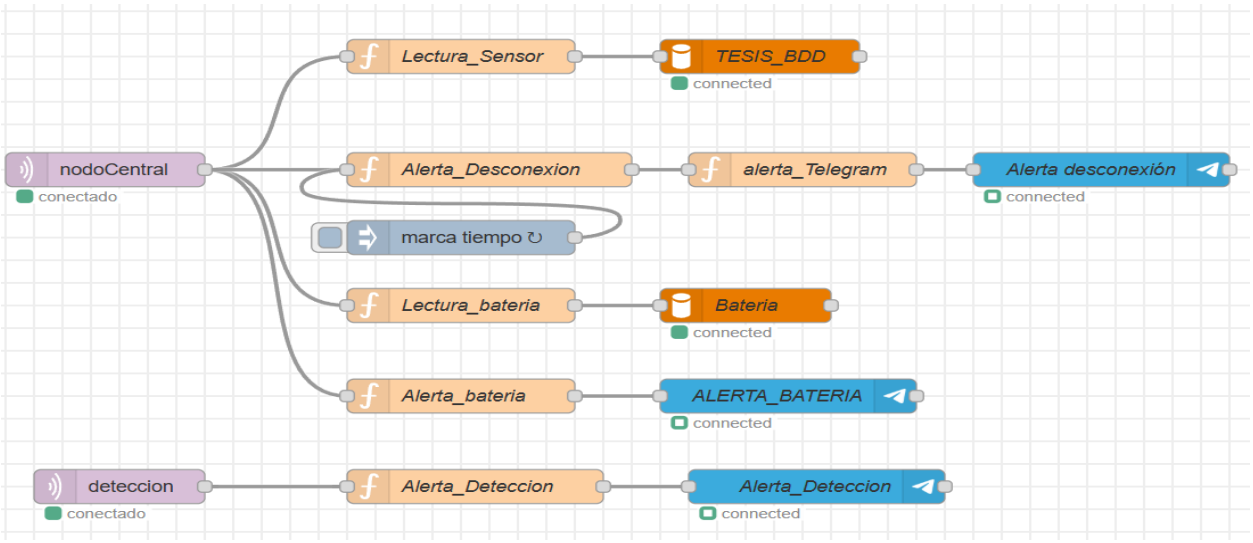
Una vez completado el despliegue de los servidores y la programación de cada uno de los nodos, se procede al procesamiento de los datos. Para ello, se emplearán los softwares previamente mencionados, los cuales permitirán recolectar la información y enviarla a la base de datos. Posteriormente, se generarán los dataframes necesarios para llevar a cabo la detección del celo.

3.4.4.1 Diagrama de bloques en node red

El software utilizado para el tratamiento y manipulación de los datos es Node-RED. En la figura se pueden observar diferentes bloques, entre los cuales se destaca el bloque MQTT, que se encarga de leer los datos del servidor MQTT utilizando el tópico 'nodoCentral'. A continuación, se encuentran los bloques de función, que son responsables de leer los datos y enviarlos a la base de datos. Además, se incluyen funciones que generan alertas de desconexión y de batería descargada.

Figura 52.

Bloques de envío de información a la base de datos.



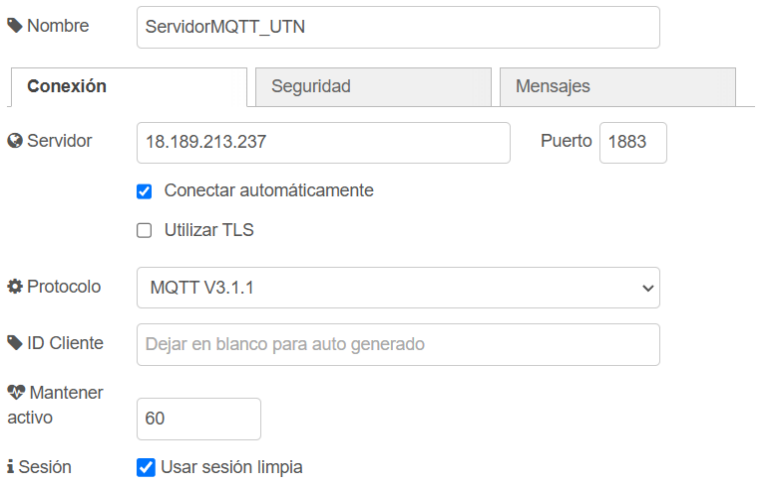
Fuente: Elaborador por el autor.

3.4.4.2 Lectura y envío de datos a la base de datos.

Para realizar la lectura de los datos capturados mediante MQTT, primero se debe configurar el bloque correspondiente. En la figura 53 se muestran las configuraciones realizadas, donde es necesario especificar la dirección IP del servidor y el puerto utilizado para la comunicación. Los demás parámetros se mantienen con sus valores predeterminados.

Figura 53.

Configuración del bloque mqtt.

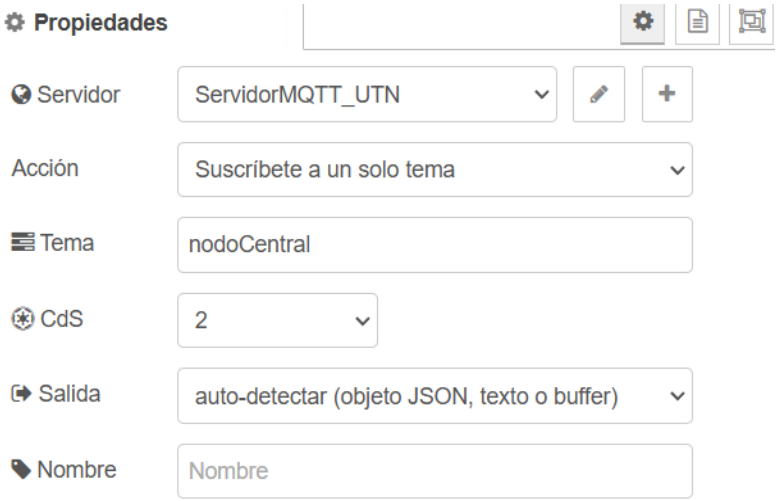


Fuente: Elaborador por el autor.

Una vez realizada la configuración correspondiente del servidor, se accede a las propiedades, donde se debe especificar el topic al que se suscribirá para obtener los datos deseados. En este caso, el software se refiere al topic como 'tema', por lo que se debe indicar el nombre del tema utilizado.

Figura 54.

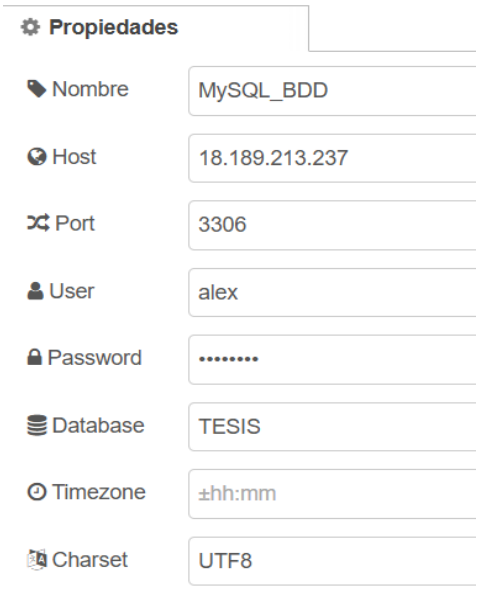
Propiedades del bloque mqtt.



Asimismo, se deberá realizar la configuración del bloque de la base de datos. En este paso, será necesario indicar la dirección IP del servidor donde está alojada la base de datos, así como el puerto correspondiente, que en el caso de MySQL es el 3306. Para que el bloque pueda comunicarse correctamente con la base de datos, se deberán proporcionar el usuario, la contraseña y el nombre de la base de datos, los cuales fueron creados en los apartados anteriores.

Figura 55.

Configuración del bloque MySQL de node red.



Fuente: Elaborador por el autor.

Una vez que los servidores estén configurados, se procederá al código de las funciones. El primer bloque se encarga de leer los datos de los ejes del sensor y convertirlos en números flotantes. Luego, estos datos se enviarán a la base de datos; para ello, se utilizará lenguaje SQL, el cual será almacenado en el mensaje para que así llegue a la base de datos y se ejecute la inserción de la información.

Figura 56.

Lectura y envío de información a la base de datos.

```
var regex = /X:([\-0-9.]+) Y:([\-0-9.]+) Z:([\-0-9.]+) Voltage:([\-0-9.]+)/
var match = msg.payload.match(regex);
if (match) {
    var eje_x = parseFloat(match[1]);
    var eje_y = parseFloat(match[2]);
    var eje_z = parseFloat(match[3]);
    // Crear la consulta SQL para insertar en la base de datos
    msg.topic = `INSERT INTO acelerometro (eje_x, eje_y, eje_z, timestamp)
                 VALUES (${eje_x}, ${eje_y}, ${eje_z}, NOW())`;
} else {
    msg.topic = "ERROR: No match found";
    node.warn("ERROR: No match found");
}
return msg;
```

Fuente: Elaborador por el autor.

Para contar con un mejor control del monitoreo, se desarrolla un código que permite detectar fallas en la recepción de la información, lo que a su vez facilita la recepción de alertas para posteriormente verificar el error y resolver el problema. Por esta razón, se implementa una función sencilla que comprueba si existe un mensaje en la entrada de la función. Esta verificación se realiza cada 2 minutos para evitar la saturación con mensajes de alerta.

Figura 57.

Código de alerta de desconexión.

```
1  var lastMessageTime = context.get('lastMessageTime') || 0;
2  var now = new Date().getTime();
3  if (msg.payload) {
4      context.set('lastMessageTime', now);
5      return null;
6  }
7  if (now - lastMessageTime > 120000) {
8      context.set('lastMessageTime', now);
9      return { payload: "Conexión perdida con el nodo central" };
10 }
11 return null;
```

Fuente: Elaborador por el autor.

Finalmente, se dispone de una función que permite monitorear los datos de la batería, lo cual es otro de los pilares fundamentales del nodo cliente. Si se presentan problemas de alimentación, el nodo no podrá enviar la información al nodo central. Por esta razón, se implementa un código que toma el valor del voltaje y lo convierte en porcentaje, para posteriormente enviarlo a la base de datos.

Figura 58.

Lectura del voltaje de la batería.

```
var payload = msg.payload;
var match = payload.match(/Voltage:([\d.]+)/);
if (match) {
    var voltaje = parseFloat(match[1]);
    var porcentaje = Math.max(0, Math.min(100, ((voltaje - 2) / (4 - 2)) *
    msg.topic = "INSERT INTO bateria (voltaje) VALUES (" + porcentaje + ")"
    return msg;
}
return null;
```

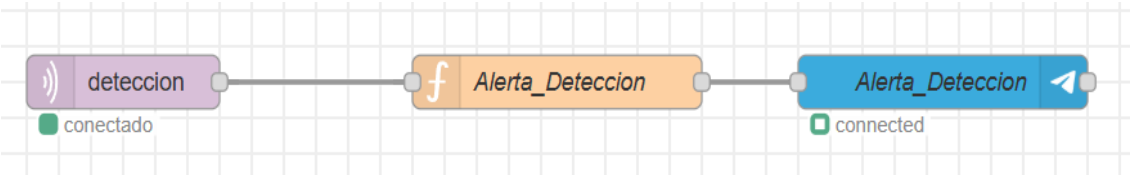
Fuente: Elaborador por el autor.

3.4.4.3 Bloque de alerta de detección de celo.

Para el envío de alertas cuando se detecte celo, se realizará el diagrama que se muestra en la figura 59. En este diagrama, se contempla la recepción del mensaje a través del bloque MQTT. Posteriormente, ese dato ingresará al bloque de función, donde será analizado y se enviará un mensaje de alerta mediante el bloque de Telegram.

Figura 59.

Diagrama de alerta de celo.



Fuente: Elaborador por el autor.

En el bloque de función se lleva a cabo la lectura del valor recibido del bloque MQTT, donde se presentan dos valores: 1 y 0. El valor 1 indica que la vaca está en celo, mientras que el 0 señala que no se ha presentado celo. A través de un condicional, se procederá al envío del correspondiente mensaje que informará al trabajador si se ha detectado celo o no.

Figura 60.

Código de alerta de celo.

```
if (msg.payload == 1) {
    msg.payload = {
        chatId: "7580921330",
        type: "message",
        content: "¡Celo detectado!"
    };
    return msg;
}
else if (msg.payload == 0) {
    msg.payload = {
        chatId: "7580921330",
        type: "message",
        content: "Vaca no presenta celo"
    };
    return msg;
} else {
    return null;
}
```

Fuente: Elaborador por el autor.

3.4.4.4 Creación de datasets.

El conjunto de datos (dataset) es una parte fundamental para el modelo de predicción, ya que de este se obtendrán los datos necesarios para el entrenamiento del modelo. Para ello, lo primero que se debe hacer es establecer la conexión con la base de datos donde están almacenados los datos de los ejes del acelerómetro. En la figura 61 se presenta el código

utilizado para la conexión a la base de datos, donde será necesario indicar la dirección IP del servidor que aloja la base de datos, así como todas las credenciales requeridas para establecer dicha conexión.

Una vez conectados a la base de datos, se procederá a realizar la consulta para extraer los datos necesarios, que en este caso son los valores de los ejes del acelerómetro.

Figura 61.

Código para la conexión a la base de datos.

```
import mysql.connector
import pandas as pd
def conectar_db():
    return mysql.connector.connect(
        host='18.189.213.237',
        user='alex',
        password='Alex_2198$!',
        database='TESIS'
    )
def leer_datos():
    try:
        conexion = conectar_db()
        query = "SELECT id, eje_x, eje_y, eje_z, timestamp FROM acelerometro"
        df = pd.read_sql(query, conexion)
        return df
    except mysql.connector.Error as err:
        print(f"Error al leer los datos: {err}")
        return pd.DataFrame()
    finally:
        conexion.close()
```

Fuente: Elaborador por el autor.

A continuación, se creará una función que permitirá manipular las fechas de los datos que se desean procesar, para posteriormente etiquetarlos. En este caso, se utilizarán las etiquetas 1 para indicar los datos de la vaca que estuvo en celo y el valor 0 para señalar a la vaca que no presentó síntomas de celo. Finalmente, los datos etiquetados se exportarán a un archivo CSV para su uso en el entrenamiento del modelo de predicción.

Figura 62.

Código para el etiquetado de datos.

```
def etiquetar_fechas(row):
    fecha_inicio_celo = pd.to_datetime('2025-01-24 19:00:00')
    fecha_fin_celo = pd.to_datetime('2025-01-25 02:00:00')
    fecha_inicio_no_celo = pd.to_datetime('2025-01-25 19:00:00')
    fecha_fin_no_celo = pd.to_datetime('2025-01-26 02:00:00')
    if fecha_inicio_celo <= row['timestamp'] <= fecha_fin_celo:
        return 1
    elif fecha_inicio_no_celo <= row['timestamp'] <= fecha_fin_no_celo:
        return 0
    else:
        return None
df = leer_datos()
df['timestamp'] = pd.to_datetime(df['timestamp'])
df['etiqueta'] = df.apply(etiquetar_fechas, axis=1)
df = df.dropna(subset=['etiqueta'])
df.to_csv('datos_etiquetados.csv', index=False)
print("Archivo CSV con los datos etiquetados ha sido generado correctamente.")
```

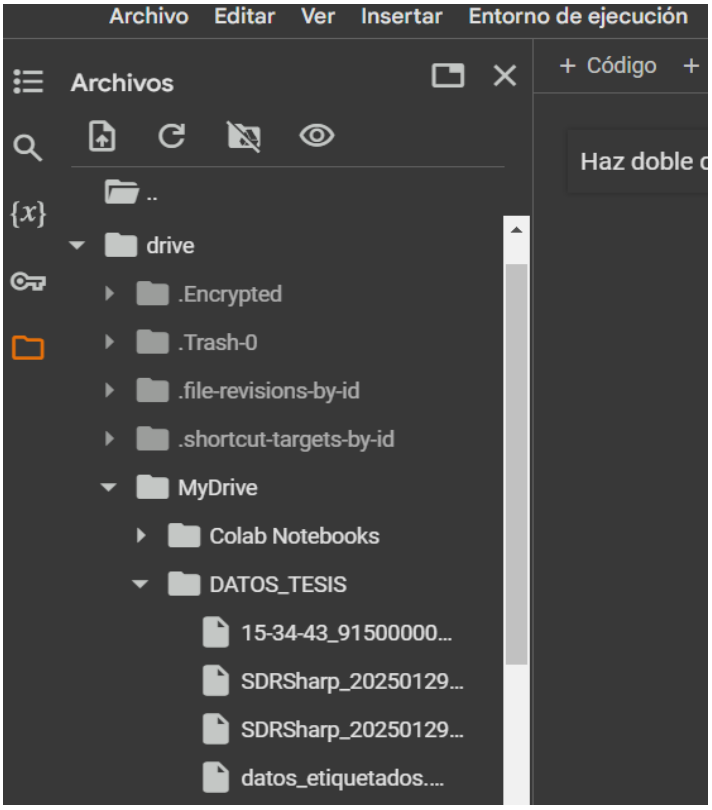
Fuente: Elaborador por el autor.

3.4.4.5 Entrenamiento del modelo de predicción

Para llevar a cabo el entrenamiento del modelo de predicción, se utilizará Google Colab, lo que permitirá trabajar en la nube y no consumir los recursos de la computadora local. Primero, será necesario iniciar sesión y crear un nuevo proyecto. Para acceder a los conjuntos de datos previamente creados, es fundamental asegurarse de que estén almacenados en Google Drive y de que la plataforma esté conectada a la cuenta de Gmail, tal como se ilustra en la figura 63.

Figura 63.

Vinculación de Drive con Google Colab.

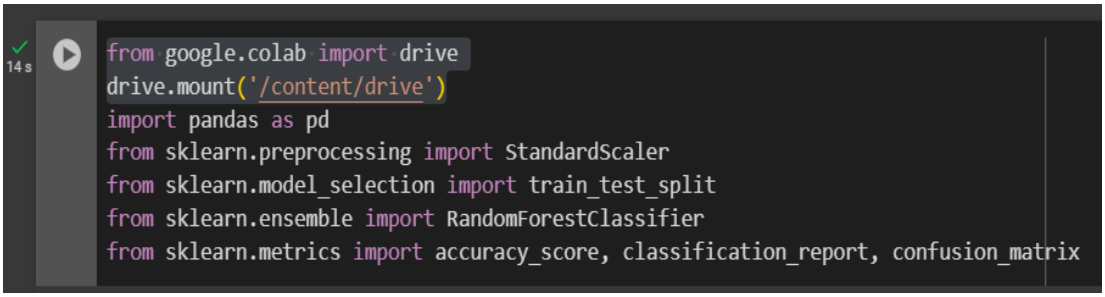


Fuente: Elaborador por el autor.

Lo primero que se debe hacer es montar Google Drive en el entorno de Google Colab. A continuación, se indicarán las librerías con las que se trabajará; en este caso, se utilizará la biblioteca Scikit-learn, que proporcionará las herramientas necesarias para entrenar el modelo de predicción. A continuación, se detallan las librerías utilizadas en el entrenamiento del modelo.

Figura 64.

Librerías usadas para el entrenamiento del modelo.



Fuente: Elaborador por el autor.

A continuación, se creará una variable denominada file_path, que contendrá la ruta del archivo CSV almacenado en Google Drive. Posteriormente, se utilizará pd. read_csv(file_path) para cargar el archivo en un DataFrame llamado df, lo que permitirá la manipulación y análisis de los datos en formato tabular. Finalmente, se empleará print(df) para mostrar el contenido completo del DataFrame en la salida de la consola. En la figura 65 se puede observar la ejecución del fragmento de código, que muestra la visualización del contenido del DataFrame creado.

Figura 65.

Creación de dataframe.

```
# Lextura del archivo csv
file_path = '/content/drive/MyDrive/DATOS_TESIS/datos_etiquetados.csv'
df = pd.read_csv(file_path)
print(df)
```

```
            id  eje_x  eje_y  eje_z            timestamp  etiqueta
0       102309   0.51   0.22  -0.55  2025-01-24 19:00:00         1
1       102310   0.47   0.00   0.15  2025-01-24 19:00:00         1
2       102311  -0.41  -0.18   0.63  2025-01-24 19:00:01         1
3       102312   0.50   0.21   0.48  2025-01-24 19:00:01         1
4       102313   0.72   0.70   0.52  2025-01-24 19:00:02         1
...        ...    ...    ...    ...                  ...       ...
100797  203106  -0.24  -0.24   0.20  2025-01-26 01:59:58         0
100798  203107   0.10  -0.06  -0.08  2025-01-26 01:59:58         0
100799  203108  -0.27   0.02  -0.14  2025-01-26 01:59:59         0
100800  203109  -0.39  -0.06   0.22  2025-01-26 01:59:59         0
100801  203110  -0.37   0.24   0.16  2025-01-26 02:00:00         0

[100802 rows x 6 columns]
```

Fuente: Elaborador por el autor.

Una vez preparado el DataFrame, se procederá a manipularlo con el objetivo de alistar los datos para el entrenamiento del modelo. Para ello, se separarán los valores de cada uno de los ejes en la variable X, mientras que las etiquetas, que son 1 y 0, se almacenarán en la variable Y. A continuación, se realizará la normalización de los datos con el fin de mejorar el desempeño del modelo de predicción.

Figura 66.

Normalización de datos del dataframe.

```
X = df[['eje_x', 'eje_y', 'eje_z']]
y = df['etiqueta']
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
pd.DataFrame(X_scaled, columns=['eje_x', 'eje_y', 'eje_z']).head()
```

	eje_x	eje_y	eje_z
0	1.391238	0.601234	-1.501821
1	1.282113	-0.001505	0.414020
2	-1.118636	-0.494655	1.727740
3	1.363956	0.573837	1.317203
4	1.964143	1.916300	1.426680

Fuente: Elaborador por el autor.

El siguiente paso consiste en dividir los datos en dos conjuntos: uno para entrenamiento y otro para pruebas. En la figura 67 se puede observar la programación de este proceso, donde se debe especificar el porcentaje de datos destinados para cada conjunto. En este caso, se utilizará el 20% para pruebas y el 80% para entrenamiento, lo que se indica en la parte donde se establece test_size = 0.2, representando así el 20% de los datos para pruebas. Por último, se indicará la cantidad de datos utilizados para el entrenamiento y para las pruebas.

Figura 67.

Separación de datos para test y train.

```
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
print(f"Conjunto de entrenamiento: {X_train.shape}")
print(f"Conjunto de prueba: {X_test.shape}")
```

```
Conjunto de entrenamiento: (80641, 3)
Conjunto de prueba: (20161, 3)
```

Fuente: Elaborador por el autor.

Una vez que se han separado los datos, se procederá a crear el modelo, utilizando el algoritmo de Random Forest, debido a su capacidad para manejar grandes volúmenes de datos y su precisión en las predicciones, además de su habilidad para reducir la probabilidad de sobreajuste. En la figura 68 se puede observar el código que permite la creación de este tipo de modelo, donde se indica que se utilizarán 100 árboles de decisión. Mediante el método. fit, se entrena el modelo utilizando los datos de entrenamiento. Después de entrenar el modelo, se emplearán los datos de prueba X_test para realizar las predicciones. Finalmente, se guardará el modelo para su uso posterior en la detección de celo en tiempo real.

Figura 68.

Creación y exportación de modelo de predicción.

```
modelo = RandomForestClassifier(n_estimators=100, random_state=42)
modelo.fit(X_train, y_train)
y_pred = modelo.predict(X_test)

MODEL_PATH = '/content/drive/MyDrive/DATOS_TESIS/modelo_deteccion.pkl'
SCALER_PATH = '/content/drive/MyDrive/DATOS_TESIS/scaler.pkl'

joblib.dump(modelo, MODEL_PATH)
joblib.dump(scaler, SCALER_PATH)
```

Fuente: Elaborador por el autor.

CAPITULO IV: ANALISIS DE RESULTADOS

3.5 FASE 4: Verificación

La fase de verificación de resultados es fundamental para asegurar que el prototipo no solo funcione en un entorno de laboratorio, sino que también sea completamente operativo y preciso en un contexto de uso real. Por esta razón, en esta sección se revisan los resultados obtenidos del prototipo una vez implementado en el animal, es decir, en un entorno de operación real. A continuación, se presenta un análisis detallado de los resultados obtenidos.

3.5.1 Funcionamiento de los servicios

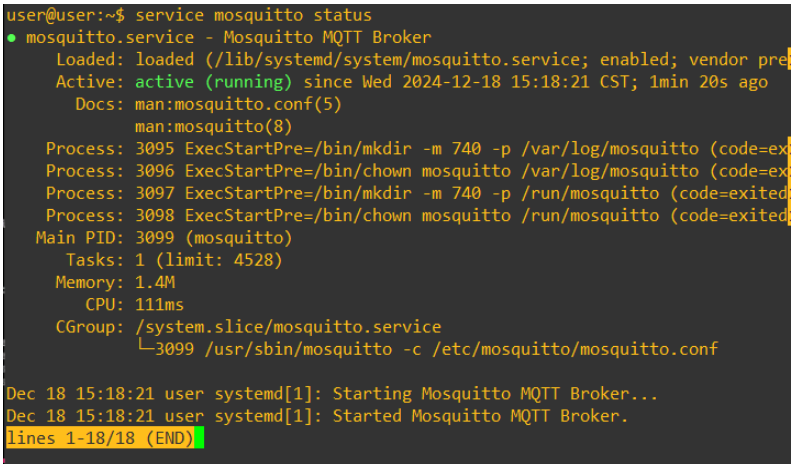
Para garantizar un funcionamiento óptimo del prototipo, es fundamental verificar que los servicios estén operando correctamente. Para ello, se debe acceder a la Shell y utilizar los siguientes comandos:

- node-red
- service mosquitto status
- sudo systemctl status MySQL
- sudo systemctl status grafana-server

Una vez ingresados estos comandos, se mostrará el estado de cada uno de los servicios. Si todo está configurado de manera adecuada, deberíamos observar en pantalla información similar a la que indican las figuras a continuación, confirmando que todos los servicios se han iniciado y están corriendo exitosamente.

Figura 69.

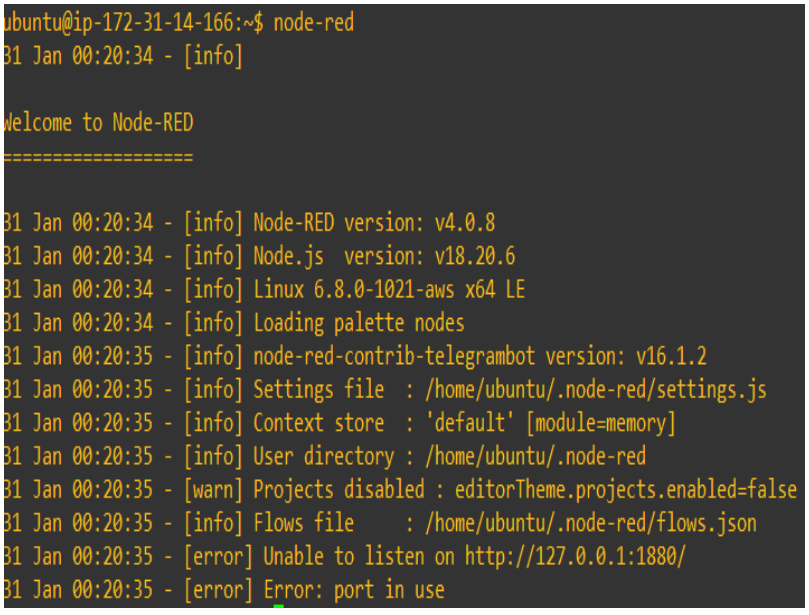
Estado del servicio MQTT.



Fuente: Elaborador por el autor.

Figura 70.

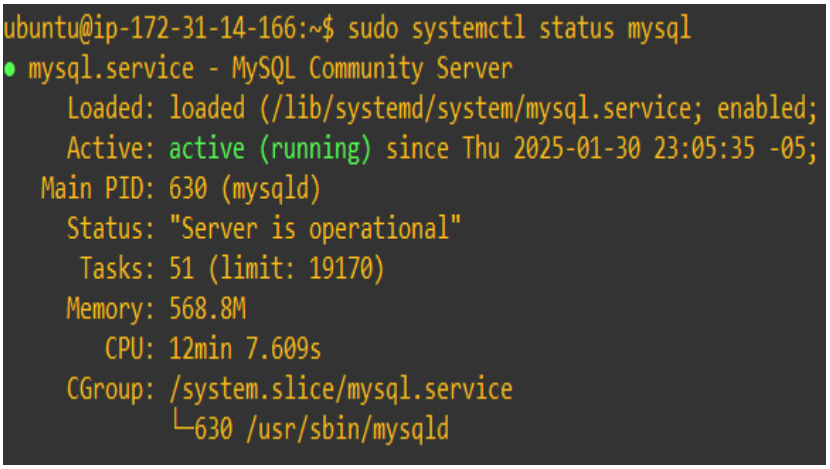
Estado del servicio de NODE-RED.



Fuente: Elaborador por el autor.

Figura 71.

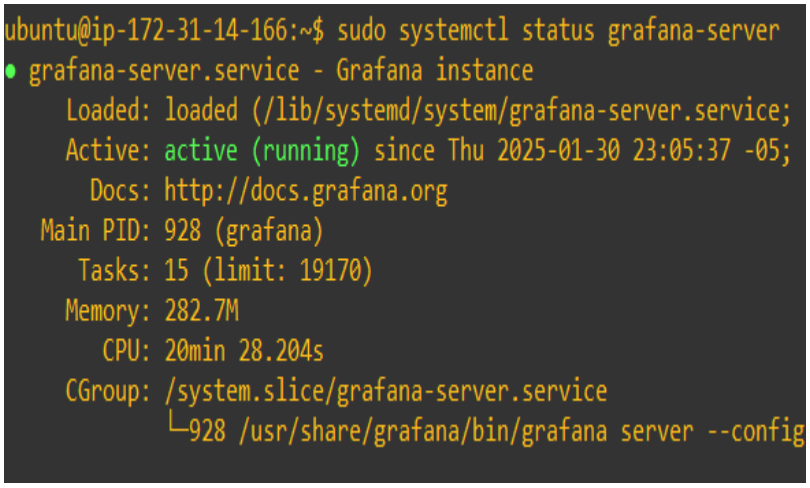
Estado del servicio de MY SQL.



Fuente: Elaborador por el autor.

Figura 72.

Estado del servicio de GRAFANA.



Fuente: Elaborador por el autor.

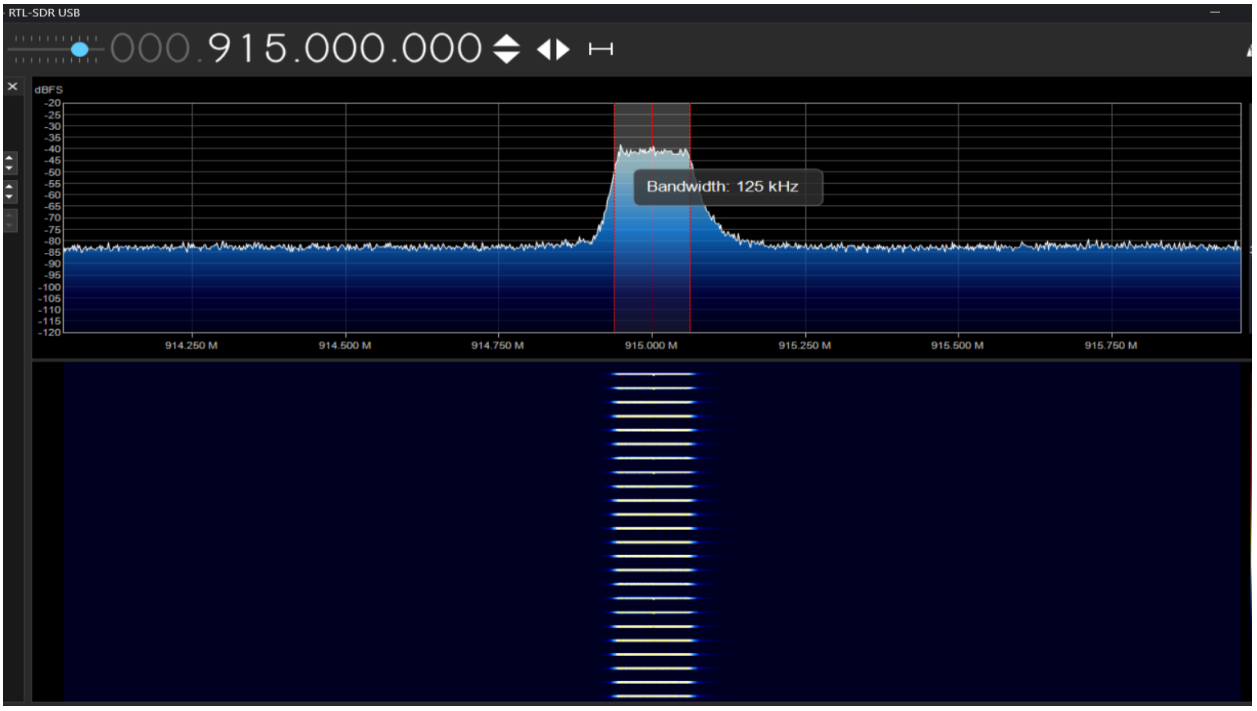
3.5.2 Análisis del espectro de la comunicación LoRa.

Para comprobar que los nodos se comunican mediante la tecnología inalámbrica LoRa, se realizó una captura de los paquetes utilizando un SDR. En este proceso, es necesario especificar la frecuencia utilizada, que en este caso es de 915 MHz. Para verificar el uso de la tecnología

LoRa, se analizará el espectro de la comunicación, el cual presenta un patrón distintivo en el que se pueden observar líneas diagonales que indican la transmisión y recepción de paquetes.

Figura 73.

Espectro de comunicación LoRa.



Fuente: Elaborador por el autor.

3.5.3 Análisis del protocolo MQTT.

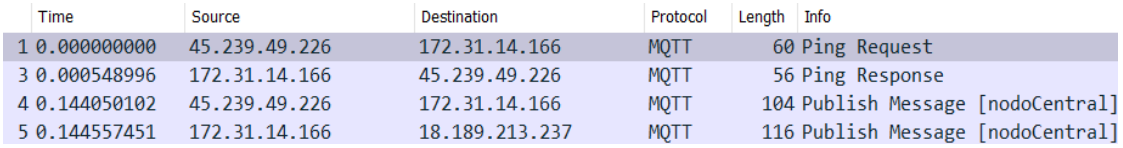
En la verificación de la funcionalidad del prototipo, es fundamental asegurarse de que se esté utilizando el protocolo MQTT. Para ello, se hará uso de la herramienta tcpdump para realizar la captura de paquetes, y para visualizar los paquetes capturados se empleará la herramienta Wireshark. Una vez capturados los paquetes, procederemos a analizarlos. En la figura 74 se pueden observar los paquetes capturados, donde se puede apreciar la comunicación entre el nodo central y el servidor que alberga los servicios.

En esta visualización, se destaca la comunicación entre la dirección IP pública del ESP32 y la dirección IP privada del servidor de AWS. Esto se debe a que en AWS contamos con una IP

elástica, que realiza un redireccionamiento automático; por ello, las dos partes no se comunican directamente a través de sus IP públicas. Una vez que el paquete llega a la IP privada del servidor, este lo envía a la IP pública del servidor, donde están alojados los servicios.

Figura 74.

Paquetes MQTT analizados con wireshark.



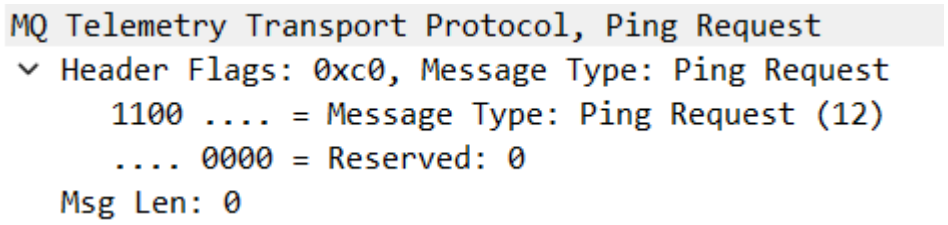
	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	45.239.49.226	172.31.14.166	MQTT	60	Ping Request
3	0.000548996	172.31.14.166	45.239.49.226	MQTT	56	Ping Response
4	0.144050102	45.239.49.226	172.31.14.166	MQTT	104	Publish Message [nodoCentral]
5	0.144557451	172.31.14.166	18.189.213.237	MQTT	116	Publish Message [nodoCentral]

Fuente: Elaborador por el autor.

Dentro de los mensajes que se pueden encontrar en el protocolo MQTT, se encuentra el "ping request", el cual es enviado por el cliente hacia el bróker con el fin de verificar si este se encuentra activo. En la figura 75 se analiza la cabecera del protocolo MQTT, donde se observa un valor de 0xc0, que es el valor hexadecimal, y en binario sería 1100. Este valor nos indica el tipo de mensaje, que en este caso corresponde al mensaje de "ping request".

Figura 75.

Cabecera MQTT mensaje tipo "ping request".



```
MQ Telemetry Transport Protocol, Ping Request
 ⌄ Header Flags: 0xc0, Message Type: Ping Request
      1100 .... = Message Type: Ping Request (12)
      .... 0000 = Reserved: 0
   Msg Len: 0
```

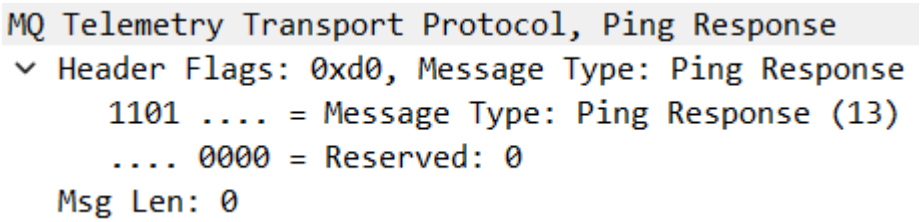
Fuente: Elaborador por el autor.

En respuesta al paquete de "ping request", se encuentra el "ping response". En la figura 76 se presenta la cabecera del paquete de "ping response", el cual está identificado por el valor 0xd0 en hexadecimal y 1101 en binario. Este paquete se genera como respuesta al paquete

enviado por el cliente, con el objetivo de indicar que el bróker está activo. De esta manera, se permite continuar con la conexión y el envío de paquetes desde el cliente hacia el bróker.

Figura 76.

Cabecera MQTT mensaje tipo "ping response".

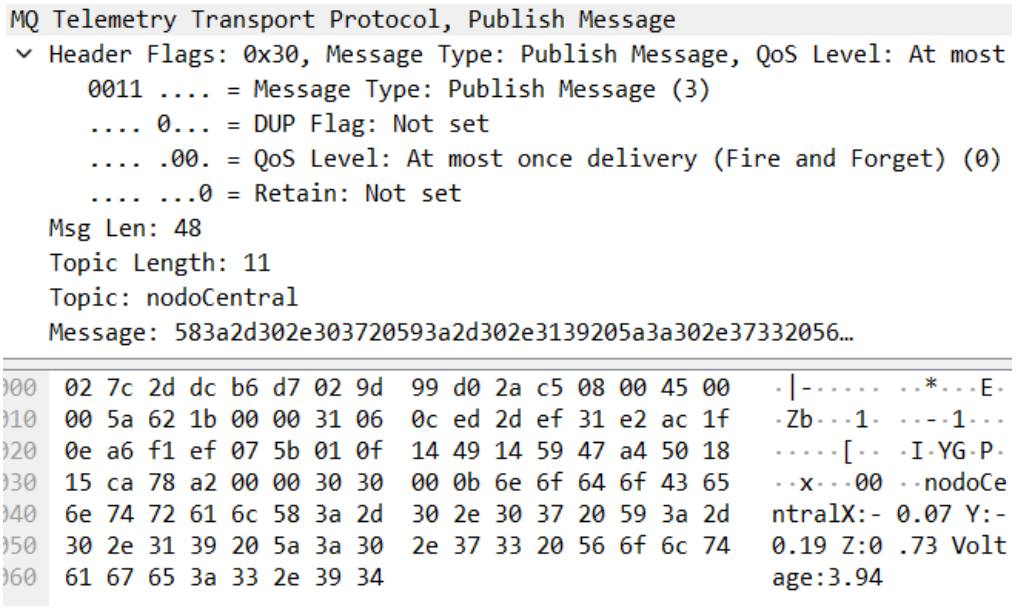


Fuente: Elaborador por el autor.

Finalmente, se encuentra el paquete de "publish message", el cual se genera una vez que se ha confirmado la comunicación entre el cliente y el bróker. Este paquete se identifica con el valor 0x30 en hexadecimal y 0011 en binario, indicando que se trata de un mensaje de tipo publicación. En la figura 77 se puede observar la cabecera del paquete, donde, a diferencia de los paquetes analizados anteriormente, se incluye el mensaje enviado, en el cual se puede apreciar la información de los ejes y del voltaje de la batería. También se puede identificar el tópico al que se están enviando los mensajes, que en este caso es el tópico "nodoCentral".

Figura 77.

Cabecera MQTT mensaje tipo "publish message".



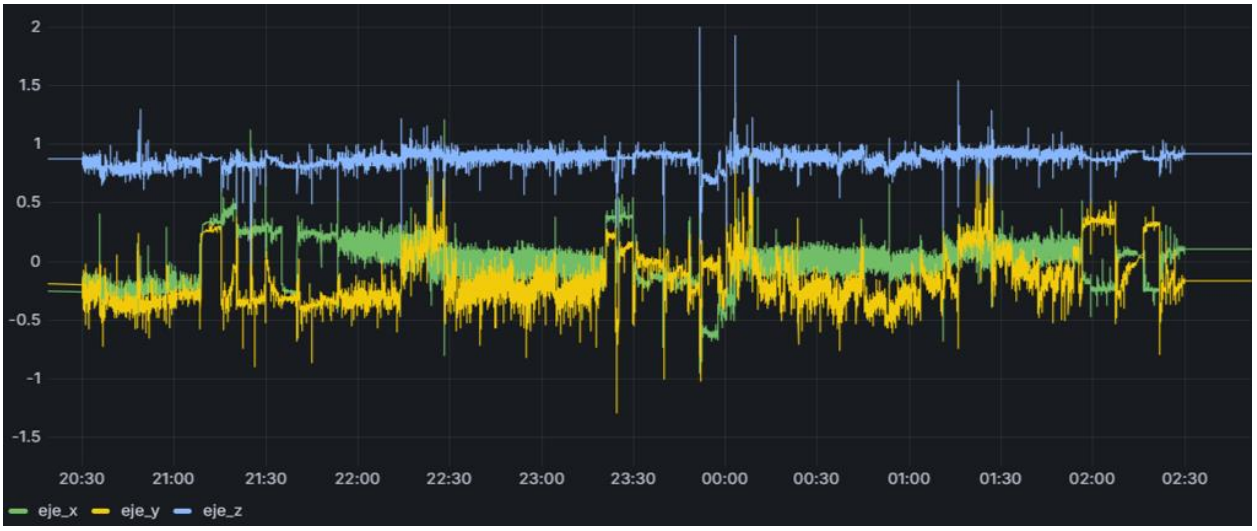
Fuente: Elaborador por el autor.

3.5.4 Datos de animal sin presencia de celo.

Dentro del desarrollo del prototipo, es necesario recopilar datos tanto de las vacas que no presentan celo como de aquellas que sí lo tienen. Este proceso tiene como objetivo la creación de sus respectivos DataFrames, que serán fundamentales para trabajar con los datos obtenidos posteriormente. En la figura 78 se puede observar el monitoreo de una vaca que no muestra signos de celo, la cual presenta una variación de movimientos moderados. Además, se puede apreciar que los datos están normalizados, lo que facilita su manejo y, en consecuencia, contribuye a una detección más precisa del celo en etapas posteriores.

Figura 78.

Monitoreo de vaca sin presencia de celo.



Fuente: Elaborador por el autor.

3.5.5 Monitore de animal con presencia de celo.

Por otro lado, los datos obtenidos de una vaca que presenta celo muestran una mayor alteración en los movimientos, lo cual es resultado del incremento de actividad causado por los cambios hormonales característicos del celo. Esta información será fundamental para realizar comparaciones entre una vaca que está en celo y una que no lo está, lo que permitirá detectar el celo mediante el uso del prototipo propuesto.

Figura 79.

Monitoreo de vaca con presencia de celo.

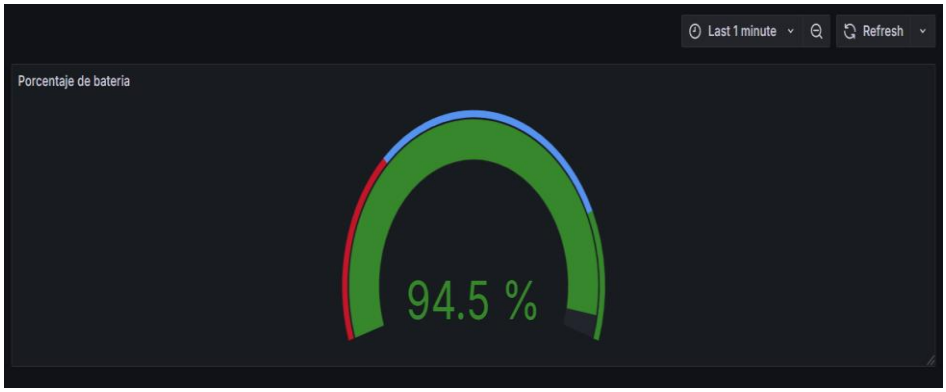
Fuente: Elaborador por el autor.

3.5.6 Monitoreo y alerta de la batería del nodo cliente

Una de las partes fundamentales es el monitoreo de la batería del nodo cliente, ya que es crucial conocer si el nodo puede cumplir con la actividad propuesta sin que esta se vea interrumpida por una disminución en la energía de la fuente de alimentación. En este caso, se puede observar el monitoreo en tiempo real del estado de la batería a través de grafana, lo que permite garantizar que el sistema se mantenga operativo durante el tiempo requerido.

Figura 80.

Monitoreo del estado de la batería.



Fuente: Elaborador por el autor.

3.5.7 Evaluación del modelo de predicción

El modelo de clasificación evaluado alcanzó una precisión global (accuracy) del 92.68%, lo que indica que, en promedio, clasifica correctamente aproximadamente el 93% de los casos. Además, el informe de clasificación presenta métricas clave para cada clase, que en este contexto son "0" y "1", las cuales representan dos estados distintos en la detección de celo en vacas. En relación a la clase "0" (sin celo), se encuentra que el modelo logra una precisión del 88%, lo que significa que de todas las predicciones realizadas como "0", el 88% fueron correctas. A su vez, su recall del 98% sugiere que el modelo detecta casi todos los casos reales de esta clase, y el f1-score de 0.93 indica un buen equilibrio entre precisión y recall.

Por otro lado, para la clase "1" (celo detectado), la precisión asciende al 98%, lo que significa que casi todas las veces que el modelo predijo "1", fue acertado. No obstante, su recall es del 87%, lo que implica que hay un 13% de casos reales de la clase "1" que no fueron detectados por el modelo; sin embargo, el f1-score de 0.92 refleja un rendimiento equilibrado entre precisión y recall.

Asimismo, el promedio macro de estas métricas es del 93%, lo que sugiere que el modelo trata ambas clases de forma equilibrada, y el promedio ponderado también se sitúa en el 93%, lo que indica que, al considerar la cantidad de ejemplos en cada clase, el modelo mantiene un desempeño sólido y consistente en general. En conclusión, el modelo presenta un alto rendimiento en la clasificación, aunque muestra una ligera tendencia a favorecer la clase "0" en términos de recall.

Figura 81.

Reporte del modelo de clasificación.

```
Accuracy: 0.9268389464808293
Reporte de clasificación:
              precision    recall  f1-score   support

           0       0.88      0.98      0.93     10013
           1       0.98      0.87      0.92     10148

    accuracy                           0.93     20161
   macro avg       0.93      0.93      0.93     20161
weighted avg       0.93      0.93      0.93     20161
```

Fuente: Elaborador por el autor.

La matriz de confusión del modelo Random Forest revela que, en los casos en los que la vaca no presenta celo (clase 0), se clasificaron correctamente 9,814 como verdaderos negativos y 199 fueron erróneamente clasificadas como celo también conocido como falsos positivos. Por otro lado, para los casos en los que la vaca sí presenta celo (clase 1), se identificaron correctamente 8,872 como verdaderos positivos, mientras que 1,276 fueron incorrectamente clasificadas como no celo es decir como falsos negativos.

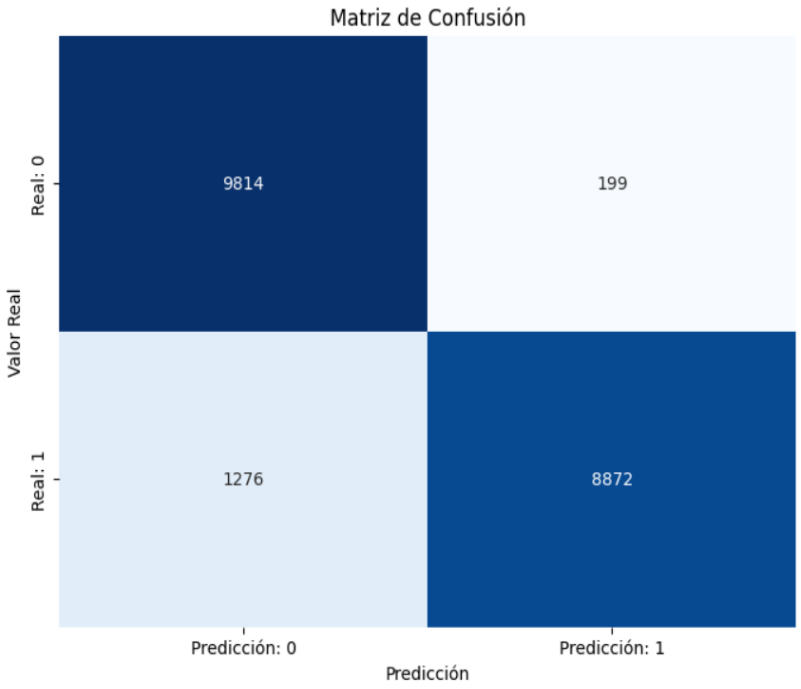
El modelo exhibe una precisión global del 92.6%, lo que indica que clasifica correctamente la mayoría de los casos. En términos de precisión, cuando el modelo predice que una vaca está en celo, es correcto en el 97.8% de las ocasiones, y su recall del 87.4% muestra

que identifica la mayoría de los casos de celo, aunque aún hay algunos que se pierden. El F1-score del 92.3% confirma un equilibrio entre precisión y recall.

Sin embargo, los 1,276 falsos negativos podrían representar un problema si la detección del celo es prioritaria. Por lo tanto, se podrían ajustar los parámetros del modelo o el umbral de decisión para mejorar la detección sin incrementar significativamente los falsos positivos.

Figura 82.

Matriz de confusión.



Nota. Representación gráfica de la matriz de confusión, donde se indica la eficacia del modelo de detección. Elaboración propia.

3.6 Análisis de resultados

Con la implementación del prototipo, se logró evidenciar una correcta detección de celo, gracias al trabajo conjunto del nodo cliente y el nodo central. Este sistema monitorea la actividad de la vaca mediante el uso de un sensor acelerómetro, el cual registra los datos de los ejes X, Y y Z del movimiento de la vaca, permitiendo detectar anomalías en su comportamiento. En

particular, se busca identificar un aumento en la actividad de la vaca durante las horas de la noche en caso de que presente celo.

En el Anexo 2 se presenta la información proporcionada por la hacienda, donde se lleva un registro de las detecciones de celo realizadas durante el último mes. En la Tabla 19 se muestra la extracción de datos del control de celo de la hacienda MELUA, con una muestra de 13 vacas. De estas, 9 fueron detectadas en celo y posteriormente inseminadas.

Los resultados evidencian que el método de detección visual no es completamente eficiente, ya que 4 vacas no fueron identificadas en celo. Esto se debe a la gran cantidad de animales que maneja la hacienda, lo que dificulta al trabajador observar a todas las vacas de manera continua. Además, este método implica una pérdida de tiempo, ya que el monitoreo visual requiere una supervisión constante, lo que limita la eficiencia del proceso.

Tabla 19.

Control de celo hacienda Melua.

No	Estado	Fecha inseminación	Observación
6625	Celo detectado	05-02-2025	Método visual
6641	Celo detectado	03-02-2025	Método visual
6860	Celo detectado	05-02-2025	Método visual
1447	Celo perdido	-	Método visual
7003	Celo detectado	03-02-2025	Método visual
7077	Celo detectado	05-02-2025	Método visual
2925	Celo perdido	-	Método visual
8082	Celo detectado	03-02-2025	Método visual
8246	Celo detectado	02-02-2025	Método visual

7316	Cele perdido	-	Método visual
7100	Celo detectado	04-02-2025	Método visual
7096	Celo perdido	-	Método visual
7278	Celo detectado	01-02-2025	Método visual

Nota. Tabulación de datos obtenidos de detección mediante el uso visual. Elaboración propia.

Por otra parte, en la Tabla 20 se presentan los resultados de detección de celo utilizando el prototipo electrónico. En este caso, se trabajó con una muestra de 6 vacas, todas las cuales fueron detectadas en celo, alcanzando hasta el momento una eficacia del 100% en la detección.

Además, el uso del prototipo representa un ahorro significativo de tiempo, ya que elimina la necesidad de contar con un trabajador monitoreando visualmente a las vacas de forma constante. Esto permite una mejor gestión del personal, optimizando sus labores y permitiéndoles desempeñar otras actividades dentro de la hacienda.

Tabla 20.

Detección de celo mediante el prototipo electrónico.

No	Estado	Fecha Inseminación	Observación
3925	Celo detectado	24-01-2025	Prototipo usado
3262	Celo detectado	25-01-2025	Prototipo usado
6260	Celo detectado	30-01-2025	Prototipo usado
7127	Celo detectado	31-01-2025	Prototipo usado
7148	Celo detectado	07-02-2025	Prototipo usado
7049	Celo detectado	08-02-2025	Prototipo usado

Nota. Tabulación de datos obtenidos de detección mediante el uso del prototipo. Elaboración propia.

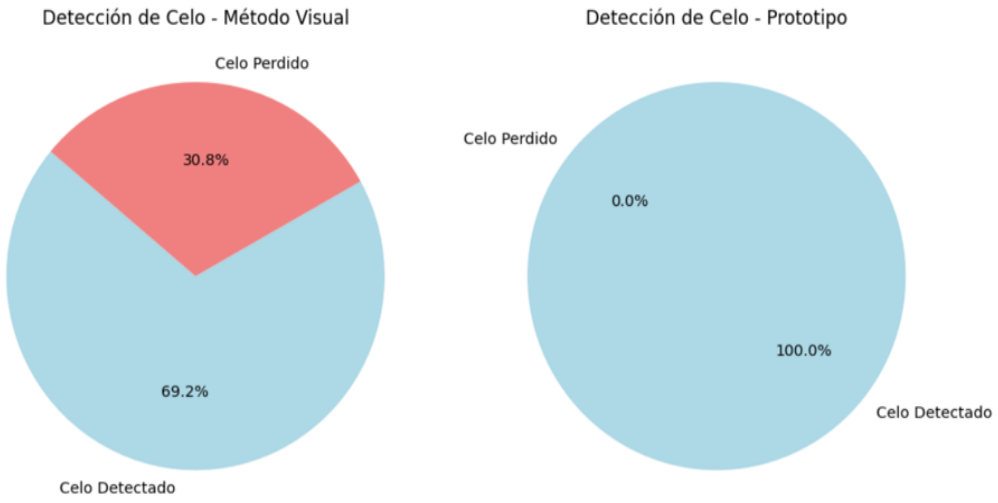
Como resultado final, se realizó una comparación entre ambos métodos: el método visual, tradicionalmente utilizado en la hacienda, y el método basado en el prototipo electrónico. En la

Figura 83 se presenta esta comparativa, donde se observa que el método visual tiene una tasa de falla del 30% en la detección de celo. En contraste, el prototipo electrónico logró una tasa de detección del 100%, sin pérdidas en la identificación del celo.

Estos resultados evidencian la superioridad del prototipo en términos de precisión y eficacia, destacando su potencial para optimizar el proceso de detección de celo en la hacienda.

Figura 83.

Comparación de resultados de detección de celo.



Nota. Porcentaje de celos detectados de los dos métodos. Elaboración propia.

Para el adecuado tratamiento de las vacas en celo detectadas, se implementa un sistema de generación de alertas, que se activa una vez que se realiza la detección mediante el modelo previamente entrenado. Este modelo permite leer los datos enviados por el sensor a la base de datos, donde se toman los valores de las últimas 6 horas para analizarlos e introducirlos en el modelo de predicción.

Una vez realizada la predicción, se envía un valor de 1 en caso de haber detectado celo y un valor de 0 en caso de no detectarlo. Tras el envío de este valor, el software de Node-RED

registra ese dato. A través de una condición, se genera una alerta que será enviada mediante Telegram al usuario final.

Figura 84.

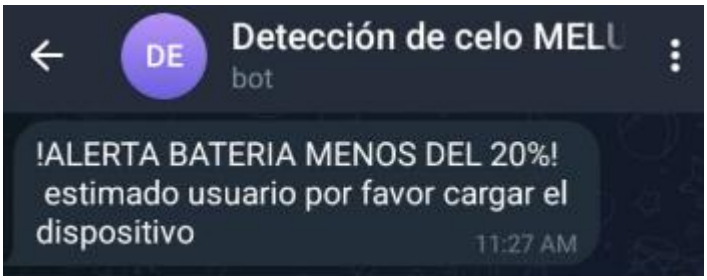
Alerta de detección de celo.

Nota. Mensaje de detección enviado al usuario. Elaboración propia

Asimismo, se implementa la generación de alertas para el control de la batería. Mediante el monitoreo constante del prototipo, se pueden generar alertas que indican cuando la batería se está agotando. De igual manera, esta alerta es enviada mediante Telegram al usuario final, quien deberá proceder a realizar la carga correspondiente. De este modo, se garantiza la disponibilidad del prototipo, asegurando que el monitoreo del animal se lleve a cabo durante todo el tiempo necesario para cumplir con el objetivo de la detección del celo.

Figura 85.

Alerta de batería baja.



Nota. Mensaje de alerta enviado al usuario. Elaboración propia

Finalmente, una vez generada la alerta, el encargado de atender a las vacas en celo procede a realizar la inseminación artificial. Este profesional verifica el comportamiento de la

vaca de manera visual durante aproximadamente 3 horas. Posteriormente, separa a la vaca que está siendo monitoreada y la lleva al lugar donde se llevará a cabo la inseminación, concluyendo así todo el proceso de monitoreo y tratamiento del animal

CONCLUSIONES

La revisión teórica sobre los parámetros fisiológicos y conductuales relacionados con el periodo de celo en el ganado bovino ha sido fundamental para el diseño del prototipo. Esta fase permitió identificar los indicadores clave que deben ser monitoreados, tales como cambios en el comportamiento y actividad física. Además, se comprendió la importancia de elegir la tecnología adecuada, siendo LPWAN (Low Power Wide Area Network) una opción idónea debido a su bajo consumo de energía y amplia cobertura, lo que garantiza la eficiencia del sistema en entornos rurales y alejados.

La recopilación de información en la hacienda LA MARIA sobre los porcentajes de periodos de fertilidad perdidos, mediante encuestas al personal encargado, mostró una importante mejora en los procesos de detección del celo. Las encuestas revelaron que una gran proporción de los períodos fértiles se pasaban por alto debido a la dificultad de observar al ganado de manera continua. Esto destaca la necesidad de implementar un sistema automatizado que permita una detección más temprana y precisa, lo cual es crucial para optimizar la tasa de fertilidad y aumentar la productividad del ganado.

El prototipo diseñado para la detección del celo bovino se basa en sensores que monitorizan el comportamiento y la actividad física de las vacas, integrados con la tecnología LPWAN para el envío de alertas en tiempo real. Este diseño permite una monitoreo constante y remoto, lo que facilita una pronta intervención para la inseminación artificial en el momento óptimo. El sistema propuesto también tiene la capacidad de ser escalable, permitiendo su implementación en distintas áreas dentro de la hacienda.

Las pruebas realizadas sobre el prototipo demostraron que el sistema de detección es efectivo en la identificación del inicio del celo, con una alta precisión al comparar los resultados

obtenidos del prototipo con los registros previos de la hacienda. Esto confirma la funcionalidad del sistema y su capacidad para minimizar los errores humanos en la identificación de los períodos fértiles. Además, la integración de la tecnología LPWAN permitió que las alertas fueran enviadas a largas distancias sin comprometer la eficiencia energética, lo que aumenta la viabilidad de la solución en entornos rurales.

La implementación del prototipo propuesto no solo mejoraría la tasa de detección del celo, sino que también optimizaría el manejo reproductivo del ganado, lo cual resultaría en una mayor tasa de fertilidad y, por ende, una mejora en la productividad del ganado. La automatización de este proceso reduce la dependencia de la observación directa y la intervención humana, permitiendo que los empleados de la hacienda se enfoquen en otras tareas importantes. Además, la disminución de los periodos de fertilidad perdidos se traduciría en una mejora en la eficiencia operativa y económica de la hacienda.

RECOMENDACIONES

Se recomienda continuar con la investigación y el desarrollo de sensores de mayor precisión y menor costo, que puedan detectar de manera más exacta los parámetros fisiológicos y comportamentales asociados al celo. Además, explorar tecnologías complementarias como el uso de cámaras térmicas o sistemas de visión artificial podría mejorar aún más la precisión en la detección del celo, brindando una mayor confiabilidad al sistema.

Aunque el prototipo automatiza en gran parte la detección del celo, es fundamental que el personal encargado del manejo y monitoreo del ganado reciba capacitación continua sobre el uso del sistema y la interpretación de las alertas. Esto garantizará una integración efectiva de la tecnología en los procesos de trabajo y permitirá que el personal pueda intervenir de manera correcta y eficiente en el momento adecuado.

Para asegurar la continuidad y el buen funcionamiento del prototipo a largo plazo, se recomienda establecer un plan de mantenimiento preventivo que incluya revisiones periódicas tanto de los sensores como de la infraestructura tecnológica. Esto minimizará posibles fallos en los equipos, garantizando que el sistema esté operativo en todo momento, y prolongará la vida útil del prototipo.

Una vez validado el prototipo en la hacienda LA MARIA, sería conveniente evaluar su implementación en otras fincas ganaderas con características similares. Esto permitirá obtener más datos y experiencias que puedan servir para ajustar y mejorar el sistema, así como comprobar su escalabilidad y adaptabilidad a diferentes tipos de ganado y condiciones de campo. La implementación a gran escala podría beneficiar a otros ganaderos y contribuir al desarrollo de la industria ganadera en general.

REFERENCIAS

330ohms. (2 de julio de 2021). *RP2040: El microcontrolador más potente en su gama.* https://blog.330ohms.com/2021/07/02/rp2040-el-microcontrolador-mas-potente-en-su-gama/

Agronet. (13 de julio de 2018). *Observe estos signos para detectar el celo en bovinos.* https://www.agronet.gov.co/Noticias/Paginas/Observe-estos-signos-para-detectar-el-celo-en-bovinos.aspx

Agrovet Market. (23 de septiembre de 2019). *Métodos que se utilizan para detectar el celo en la ganadería.* https://www.agrovetmarket.com/noticias-salud-animal/detalle/metodos-que-se-utilizan-para-detectar-el-celo-en-la-ganaderia

Álvarez, M., Guzmán, J., & González, V. M. (2006). *Aprendizaje por búsqueda: de la información al conocimiento.* Centro Universitario de la Costa - Universidad de Guadalajara.

G. Roque and V. S. Padilla, "LPWAN Based IoT Surveillance System for Outdoor Fire Detection," in IEEE Access, vol. 8, pp. 114900-114909, 2020, doi: 10.1109/ACCESS.2020.3003848.

Amazon. (2023a). *sb components Chip microcontrolador IC RP2040 Raspberry Pi RP2040 diseñado por Raspberry Pi (10 piezas).* https://www.amazon.com/-/es/components-microcontrolador-RP2040-Raspberry-dise%C3%B1ado/dp/B09K417B76

Amazon. (2023b). *ESP32-DevKitC-VE Tabla de desarrollo.* https://www.amazon.com/-/es/ESP32-DevKitC-VE-Tabla-de-desarrollo/dp/B087TNPQCV

Amazon. (2023c). *Vis Viva Raspberry Pi Pico + Raspberry Pi - Calcomanía con logotipo (4 unidades, regular).* https://www.amazon.com/-/es/Vis-Viva-Raspberry-Pico-Calcoman%C3%ADa/dp/B0BHMBXRCK?th=1

Analog Device. (2022). *Data Sheet ADXL 345.* Analog Device.

Arca Electrónica. (2023). *Chip de Microcontrolador para Raspberry Pi Pico RP2040.* https://www.arcaelectronica.com/products/chip-de-microcontrolador-para-raspberry-pi-pico-rp2040?variant=43863946690787

AWS Amazon. (2023a). *Características de AWS IoT Core.* https://aws.amazon.com/es/iot-core/features/

AWS Amazon. (2023b). *AWS IoT Core.* https://aws.amazon.com/es/iot-core/

AWS Amazon. (2023c). *Preguntas frecuentes acerca AWS IoT Core.* https://aws.amazon.com/es/iot-core/faqs/

Azure Microsoft. (2023). *Azure IoT Hub.* https://azure.microsoft.com/en-us/products/iot-hub

Barsky, O., Barril, A., Cosse, G., Morandi, J., & Vinueza, H. (1980). *El Proceso de Transformación de la Producción Lechera Serrana y el Aparato de Generación - Transferencia en Ecuador.* Flacso.

Bavera, G. A. (2016). *Sitio argentino de producción animal.* https://www.produccion-animal.com.ar/produccion_bovina_de_leche/00-principal_produccion_leche.htm

Bermeo, E., Zabala, C. M., & Arroyo, M. G. (s.f.). *Comparación de la eficacia de la detección de estros mediante la aplicación de dos métodos detección visual vs. utilización de parches.* Vademécum Veterinario.

Buitrago, L. M., Manrique, M. Á., & Hernández, J. (2019). *Redes LoRaWAN.revisión de componentes funcionales en aplicaciones ioT [trabajo final de grado].* Universidad Distrital Francisco José de Caldas. http://hdl.handle.net/11349/22411

Cabezas, A., & Delgado, E. (2013). Google Scholar e índice h en biomedicina: la popularización de la evaluación bibliométrica. *Medicina Intensiva, 37*(5), 343-354. https://doi.org/10.1016/j.medin.2013.01.008

Carvajal, A., Martínez, M. E., & Tapia, M. (2020). El ciclo estral en la hembra bovina y su importancia productiva. *Informativo INIA Remehue*(246), 1-4. https://hdl.handle.net/20.500.14001/4022

Casamú. (s.f.). *Manejo productivo.* https://www.casamu.com.ar/manejo-productivo/

Cevallos, O. F. (2017). *Caracterización morfométrica y molecular del bovino criollo en la provincia de Maqnabí (Ecuador) [tesis de grado].* Universidad de Córdoba. https://dialnet.unirioja.es/servlet/tesis?codigo=114959

Cruz, A. (2006). Principales factores que afectan la prolificidad del ganado vacuno en Latinoamérica. *REDVET. Revista Eléctronica de Veterinaria, 7*(10), 1-11. https://www.redalyc.org/articulo.oa?id=63617167006

Delgado, A. F. (2020). *Evaluación de la sincronización de celo en vacas y vaconas Braham en la Hacienda Don Manuel [tesis de grado].* Universidad Agraria del Ecuador.

Descifrando Blockchain. (15 de diciembre de 2018). *Cómo funciona la red LPWAN Sigfox.* https://blockchainadministration.blogspot.com/2018/12/como-funciona-la-red-lpwan-sigfox.html

Envira. (28 de enero de 2019). *Tecnología IoT en la ganadería.* https://enviraiot.es/tecnologia-en-la-ganaderia/

Ferretrónica. (2023). *Acelerómetro Análogo de 3 ejes MMA7361.* https://ferretronica.com/products/acelerometro-analogo-de-3-ejes-mma7361

Findchips. (s.f.). *MMA7361 MMA7361 MMA7361 MMA7361 Triaxial Triaxial Triaxial Triaxial Acceleration Acceleration Acceleration Acceleration Sensor.* https://datasheetspdf.com/pdf-file/900699/ETC/MMA7361/1

García, A. V. (2002). Detección de celos: interpretación de los síntomas. *Revista Plan Agropecuario*(50), 31-32.

González, K. (24 de octubre de 2018). *El ciclo estral de la vaca.* https://zoovetesmipasion.com/ganaderia/reproduccion-bovina/el-ciclo-estral-de-la-vaca

Google Cloud. (2023). *Soluciones de Google Cloud IoT.* https://cloud.google.com/solutions/iot?hl=es-419

GSL Industrias. (14 de julio de 2021). *Qué es un microcontrolador.* https://industriasgsl.com/blogs/automatizacion/que_es_un_microcontrolador

Guáqueta, H. (2009). Ciclo estral: fisiología básica y estrategias para mejorar la detección de celos. *Revista de la Facultad de Medicina Veterinaria y de Zootecnia, 56*, 163-183. https://repositorio.unal.edu.co/handle/unal/26847

Guerra, J. (2023a). *ESP32 Wifi y Bluetooth en un solo chip.* https://programarfacil.com/esp8266/esp32/

Guerra, J. (2023b). *Arduino Nano il capo de la familia Arduino Nano.* https://programarfacil.com/blog/arduino-blog/familia-arduino-nano/

Hacienda El Cucharo. (11 de septiembre de 2019). *Ciclo Estral de las Vacas.* https://www.haciendaelcucharo.com/post/ciclo-estral-de-las-vacas

Idrovo, T. L. (2016). *Evaluación de la tasa de preñez en vacas repetidoras con la aplicación de eCG al momento de la inseminación artificial [tesis de pregrado].* Universidad Politécnica Salesiana. https://dspace.ups.edu.ec/handle/123456789/11877

IPLOOK. (2022). *Solución NB-IoT.* https://es.iplook.com/Solucion/solucin_nb_iot.html

IvenSense. (2013). *MPU-6000 and MPU-6050. Product Specification Revision 3.4.* IvenSense.

Laverde, J. A., & Laverde, C. G. (2021). Internet de las cosas aplicado en la agricultura ecuatoriana: Una propuesta para sistemas de riego. *Dilemas Contemporáneos: Educación, Política y Valores, 8*(2), 1-14. https://doi.org/10.46377/dilemas.v8i2.2542

León, J. L. (2013). *Detección de estro por observación visual y el sistema de parche estrus alert en vacas charolais mestizas sincronizadas [tesis de maestría].* Universidad de Cuenca. http://dspace.ucuenca.edu.ec/handle/123456789/3746

León, J., Ramónez, J., Ayala, L., Vázquez, J., Rodas, R., & Pesántez, J. (2017). Comparación de dos métodos de detección de celo en vacas mestizas Charoláis en la Amazonía ecuatoriana. *Moskana, 8*, 81-83. https://publicaciones.ucuenca.edu.ec/ojs/index.php/maskana/article/view/1493

LinkedIn. (2023). *Ubidots.* https://www.linkedin.com/company/ubidots/

MCI Electronics. (s.f.). *Arduino Nano.* https://arduino.cl/arduino-nano/

Microsoft. (6 de agosto de 2023). *Azure IoT Hub concepts overview.* https://learn.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide

Ministerio del Ambiente, Agua y Trancisión Ecológica del Ecuador. (2013). *Implementación de Buenas Prácticas para el Manejo Adaptativo del Sistema Pecuario y la Conservación del Ecosistema Páramo en la Microcuenca de Papallacta. Guía para el manejo sanitario de ganado bovino en la parroquia de Pallacta.* PRAA, Ecopar.

Modernell, P. D. (2020). *Plataforma de pruebas de conformidad LoRaWAN [tesis de maestría]*. Universidad de la República. https://hdl.handle.net/20.500.12008/25761

Naylamp Mechatronics. (2023). *ATMEGA328P DIP.* https://naylampmechatronics.com/microcontroladores/111-atmega328p-dip.html

Naylamp Mechatronics. (2023). *Módulo ADXL345 Acelerómetro Digital.* https://naylampmechatronics.com/sensores-posicion-inerciales-gps/528-modulo-adxl345-acelerometro-digital.html

Nebel, R. L., Dransfield, M. G., Jobst, S. M., & Bame, J. H. (2000). Automated electronic systems for the detection of oestrus and timing of AI in cattle. *Animal Reproduction Science, 60-61*, 713-723. http://dx.doi.org/10.1016/S0378-4320(00)00090-7

Nephos IT. (2023). *¿Cómo trabaja IoT y la Nube?* https://www.nephosit.com/como-trabaja-iot-y-la-nube/

Newark. (2023). *Microchip ATMEGA328P-AU.* https://www.newark.com/es/microchip/atmega328p-au/microcontroller-mcu-8-bit-atmega/dp/68T2935

Opio, C., & Sangoluisa, P. (2022). *Innovaciones en el sector ganadero. Compendio de experiencias en América Latina y el Caribe. 2021.* Organización de las Naciones Unidas para la Alimentación y la Agricultura [FAO].

Ordóñez, M. J. (2021). *Análisis de parámetros reproductivos en hembras bovinas en base a estudios realizados en protocolos de sincronización de celo en la Universidad Agraria del Ecuador [tesis de grado].* Universidad Agraria del Ecuador.

Palomares, R. A., Bosques, J., & Graves, W. M. (2017). *Estrategias de Detección de Celo para Ganado Lechero.* https://extension.uga.edu/publications/detail.html?number=B1212-SP

Pardal, P. (2017). *Redes de Área Extensa para aplicaciones de IoT: modelado de comunicaciones Sigfox [tesis de máster].* Universitat Politècnica de València. https://riunet.upv.es/handle/10251/86052

Perry, G. (2021). *Detección de Celos en el Ganado Bovino.* South Dakota State University Extension.

Pronaca. (2021). *La eficiencia reproductiva del ganado.* https://www.procampo.com.ec/index.php/blog/10-nutricion/116-la-eficiencia-reproductiva-del-ganado

Pueyo, D. (2017). *Efectividad de cuatro métodos para la detección de celo en vacuno de carne [Trabajo fin de carrera].* Escuela Politécnica Superior de Huesca. http://hdl.handle.net/10532/3955

Raspberry Pi. (2022). *Raspberry Pi Pico.* Raspberry Pi Ltd.

Raspberry Pi. (2023). *Raspberry Pi Pico and Pico W.* https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html

Reproducción Veterinaria. (11 de agosto de 2017). *Ciclo estral en la vaca.* https://www.reproduccionveterinaria.com/fisiologia-y-anatomia-obstetrica/fisiologia-obstetrica2/ciclo-estral/ciclo-estral-en-la-vaca/

RF Wireless World. (2012). *NB-IoT Architecture | LTE-NB IoT Architecture.* https://www.rfwireless-world.com/Terminology/LTE-NB-IoT-Architecture.html

Rivas, P. C., Suárez, Á., & Ramírez, E. (2011). Influencia de las hormonas metabólicas y la nutrición en el desarrollo folicular en el ganado bovino: implicaciones prácticas. *Revista de Medicina Veterinaria, 1*(21), 155-173. https://doi.org/10.19052/mv.574

Rodríguez, J. D. (2017). *Dispositivo LoRa de comunicación a largo alcance y bajo consumo energético para aplicaciones del ámbito del desarrollo [tesis de máster].* Universidad Politécnica de Madrid - Universidad Complutense de Madrid. https://oa.upm.es/44890/

Rojas, D. (2021). *Detección de celo con tres métodos en vacas lecheras Holstein [tesis de grado].* Universidad Mayor de San Simón. http://hdl.handle.net/123456789/33611

Sampaulo, P. (2020). *Redes LPWAN: una guía completa de inicio a fin. Nuevas redes para el Internet de las Cosas.* RedGps.

Selk, G. (2004). *Heat Detection Aids for Dairy and Beef A.I.* Oklahoma Cooperative Extension Service.

Sepúlveda, N. &. (2003). *Comportamiento sexual durante el estro en vacas lecheras. Interciencia*. Comportamiento sexual durante el estro en vacas lecheras. Interciencia: http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S0378-18442003000900002&lng=es&tlng=es

Sepúlveda, N., & Rodero, R. (2003). Comportamiento sexual durante el estro en vacas lecheras. *Interciencia, 28*(9), 500-503. https://www.redalyc.org/articulo.oa?id=33908402

Sigfox. (2021). *¿Qué es Sigfox?* https://sigfox.com.py/que-es-sigfox/

Steren. (2023). *Placa de desarrollo NANO.* https://www.steren.com.co/placa-de-desarrollo-nano.html

Suárez, J. (2022). *Estudio del comportamiento de la tecnología LoRa en entornos urbanos [tesis de grado].* Universidad de Cantabria. http://hdl.handle.net/10902/24050

Tesla Electronic. (17 de agosto de 2019). *MPU-6050.2.* https://www.teslaelectronic.com.pe/mpu-6050-2/

Ubidots. (2023). *Platform Ubidots.* https://es.ubidots.com

Universidad de la Empresa. (2019). *Particularidades de la red Sigfox.*

https://ude.edu.uy/particularidades-de-la-red-sigfox/amp/

Zambrano, R. (9 de julio de 2015). *La ganadería bovina.*

https://www.asambleanacional.gob.ec/es/contenido/la-ganaderia-bovina-0

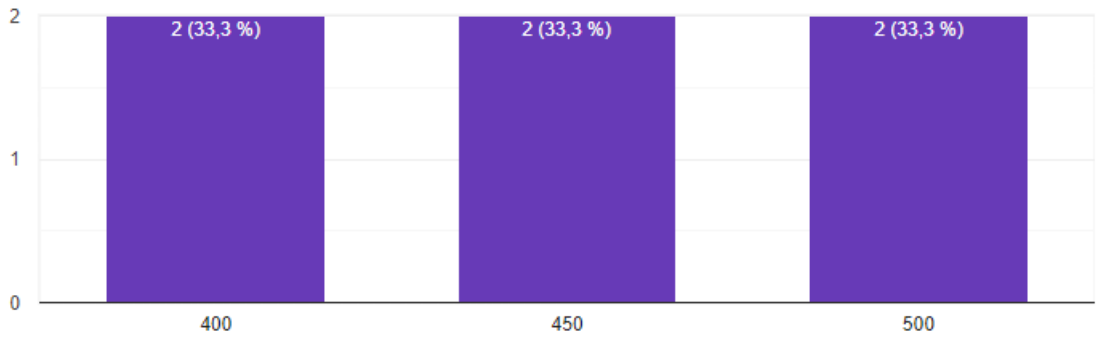
ANEXOS

ANEXO 1.

Actualmente cual es la cantidad de ganado lechero presente en la hacienda La Maria?

Copiar

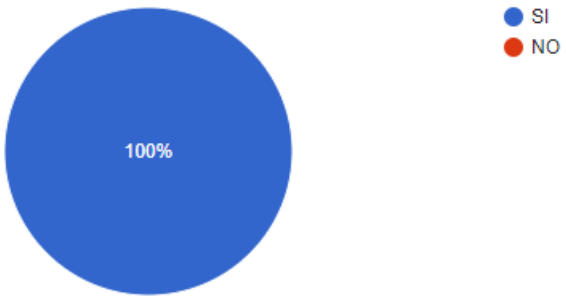
6 respuestas



Considera que la detección de celo resulta mas fácil si todo el ganado se encuentra en un lugar en especifico?

Copiar

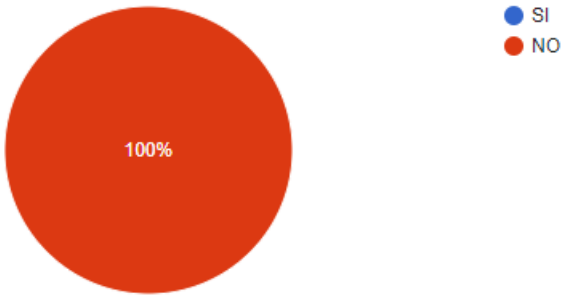
6 respuestas



La detección del celo le resulta fácil, considerando la cantidad de ganado lechero?

Copiar

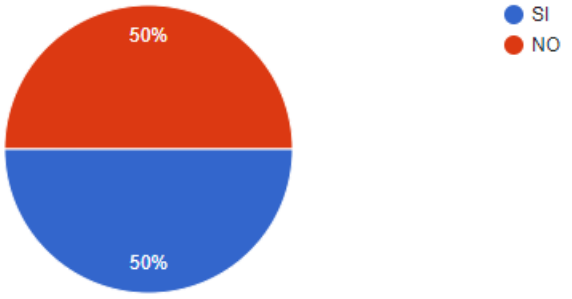
6 respuestas



Conoce aspectos técnicos acerca de las etapas del celo como: tiempo de celo, duración de celo, cambios físicos, entre otros?

Copiar

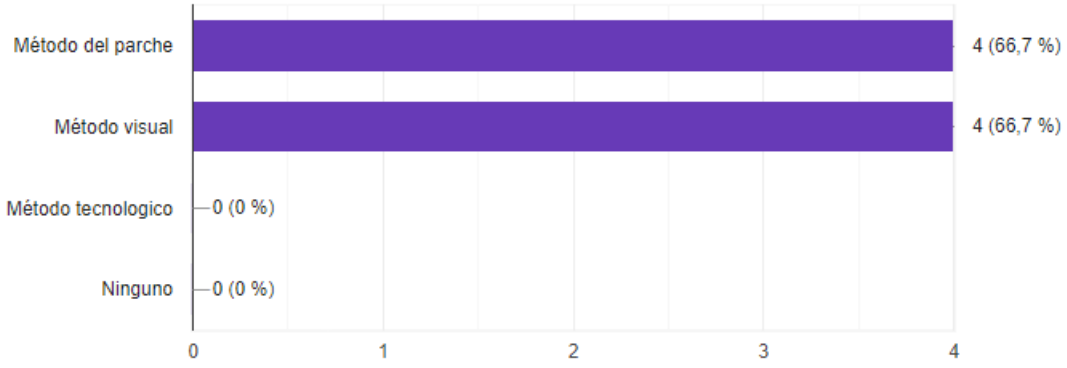
6 respuestas



Cuales métodos de detección de celo usa actualmente la hacienda La Maria?

Copiar

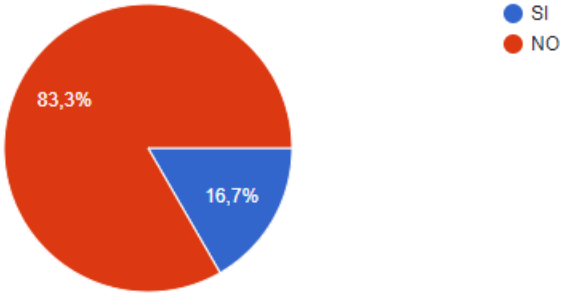
6 respuestas



Considera que los métodos usados actualmente son eficientes al momento de detectar el celo?

Copiar

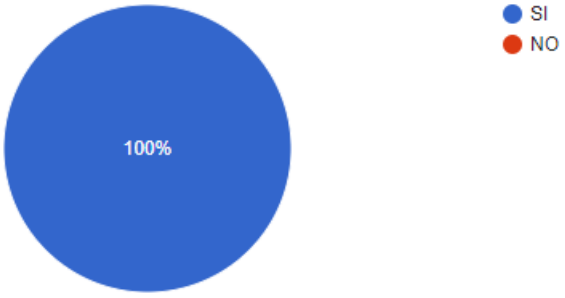
6 respuestas



Cree usted que el uso de la tecnología podría ayudar en la detección del celo ?

Copiar

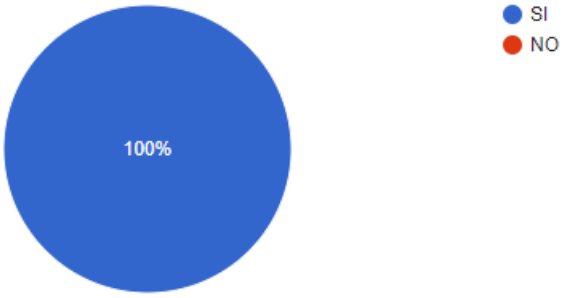
6 respuestas



Estaría dispuesto a implementar un método de detección de celo tecnológico?

Copiar

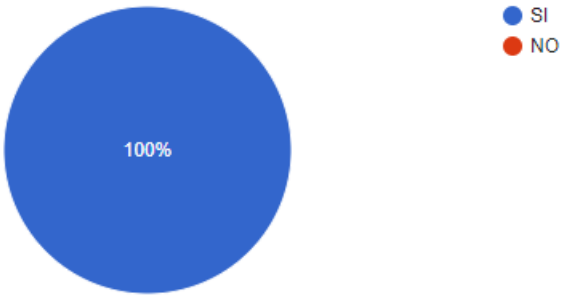
6 respuestas



Considera que el prototipo genere alertas cuando el celo sea detectado?

Copiar

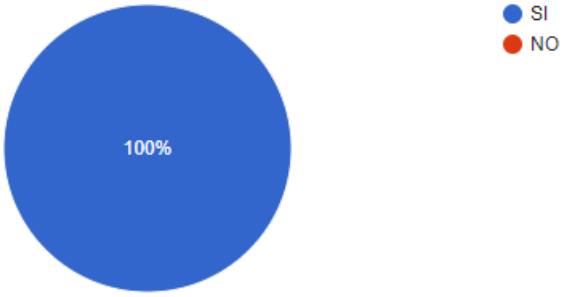
6 respuestas



El nodo debería comunicarse inalámbricamente?

Copiar

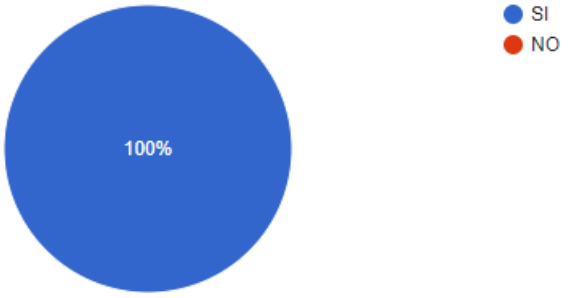
6 respuestas



El nodo debería tener una larga duración de batería?

Copiar

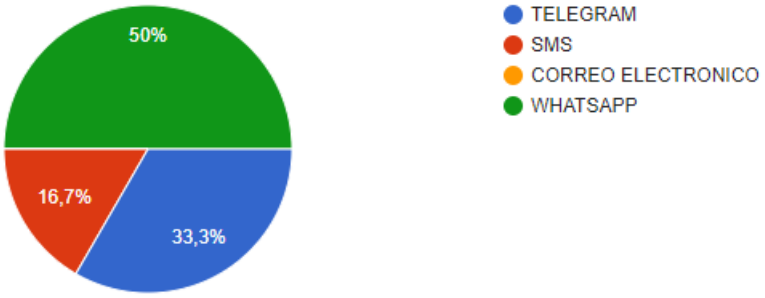
6 respuestas



Por que medio desearía recibir las alertas ?

Copiar

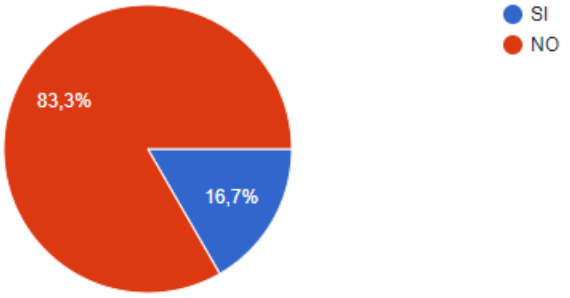
6 respuestas



Considera que los datos obtenidos se muestren en una interfaz grafica?

Copiar

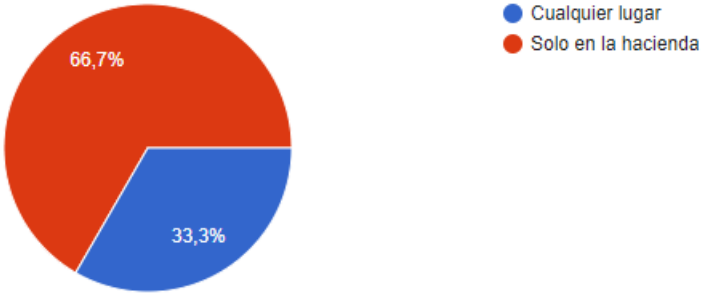
6 respuestas



Cree que los datos obtenidos se deberían visualizar desde cualquier lugar o solo en la hacienda?

Copiar

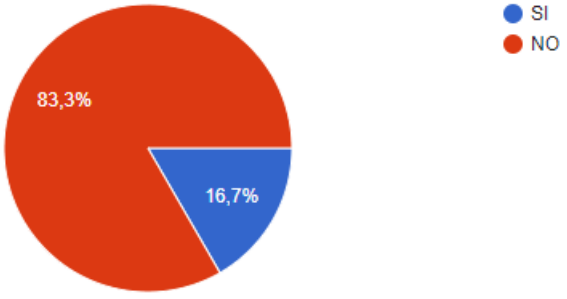
6 respuestas



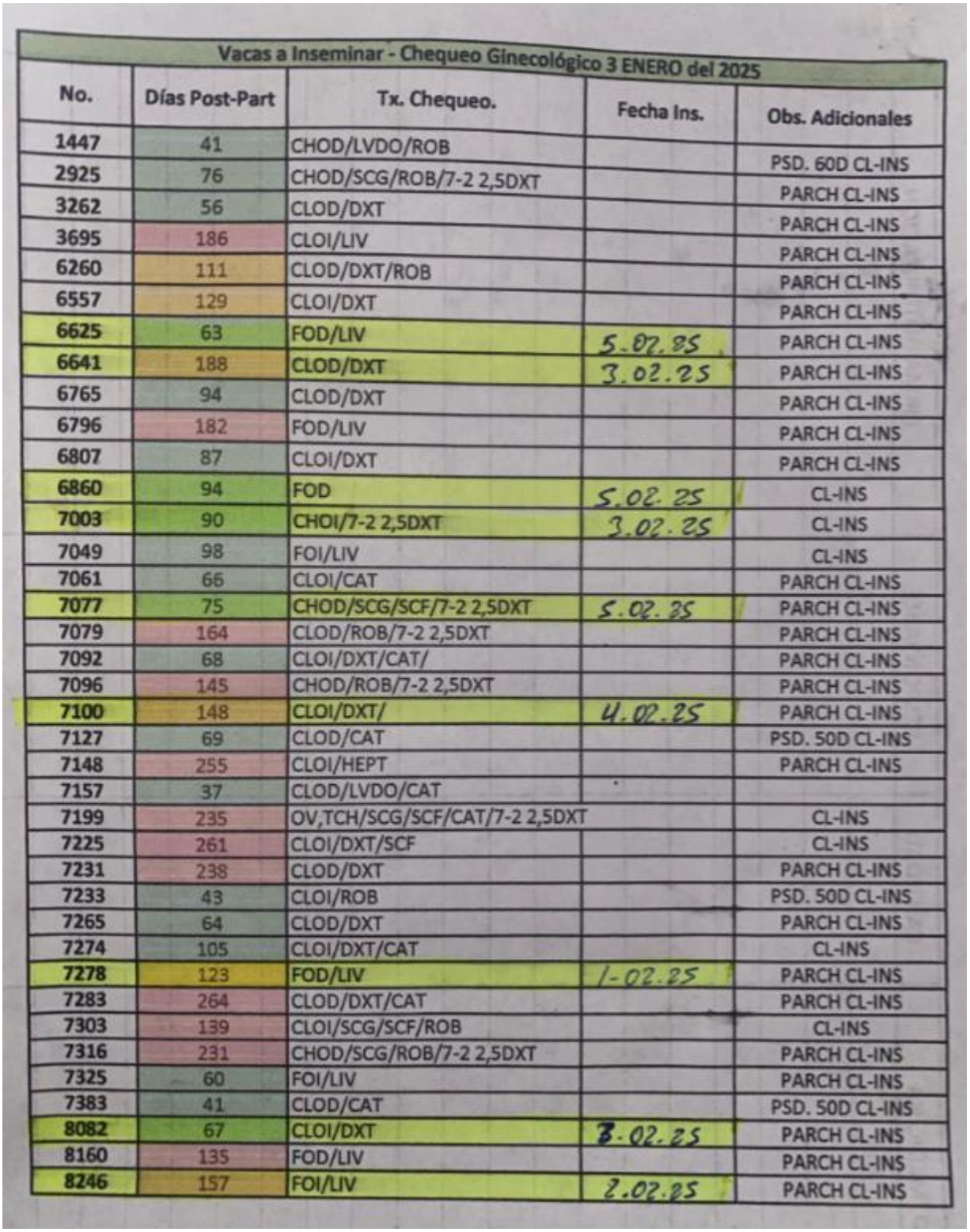
Las alertas se deberían enviar a todos los trabajadores de la hacienda ?

Copiar

6 respuestas



ANEXO 2.



Vacas a Inseminar - Chequeo Ginecológico 3 ENERO del 2025				
No.	Días Post-Part	Tx. Chequeo.	Fecha Ins.	Obs. Adicionales
1447	41	CHOD/LVDO/ROB		PSD. 60D CL-INS
2925	76	CHOD/SCG/ROB/7-2 2,5DXT		PARCH CL-INS
3262	56	CLOD/DXT		PARCH CL-INS
3695	186	CLOI/LIV		PARCH CL-INS
6260	111	CLOD/DXT/ROB		PARCH CL-INS
6557	129	CLOI/DXT		PARCH CL-INS
6625	63	FOD/LIV	5.02.25	PARCH CL-INS
6641	188	CLOD/DXT	3.02.25	PARCH CL-INS
6765	94	CLOD/DXT		PARCH CL-INS
6796	182	FOD/LIV		PARCH CL-INS
6807	87	CLOI/DXT		PARCH CL-INS
6860	94	FOD	5.02.25	CL-INS
7003	90	CHOI/7-2 2,5DXT	3.02.25	CL-INS
7049	98	FOI/LIV		CL-INS
7061	66	CLOI/CAT		PARCH CL-INS
7077	75	CHOD/SCG/SCF/7-2 2,5DXT	5.02.25	PARCH CL-INS
7079	164	CLOD/ROB/7-2 2,5DXT		PARCH CL-INS
7092	68	CLOI/DXT/CAT/		PARCH CL-INS
7096	145	CHOD/ROB/7-2 2,5DXT		PARCH CL-INS
7100	148	CLOI/DXT/	4.02.25	PARCH CL-INS
7127	69	CLOD/CAT		PSD. 50D CL-INS
7148	255	CLOI/HEPT		PARCH CL-INS
7157	37	CLOD/LVDO/CAT		
7199	235	OV,TCH/SCG/SCF/CAT/7-2 2,5DXT		CL-INS
7225	261	CLOI/DXT/SCF		CL-INS
7231	238	CLOD/DXT		PARCH CL-INS
7233	43	CLOI/ROB		PSD. 50D CL-INS
7265	64	CLOD/DXT		PARCH CL-INS
7274	105	CLOI/DXT/CAT		CL-INS
7278	123	FOD/LIV	1-02.25	PARCH CL-INS
7283	264	CLOD/DXT/CAT		PARCH CL-INS
7303	139	CLOI/SCG/SCF/ROB		CL-INS
7316	231	CHOD/SCG/ROB/7-2 2,5DXT		PARCH CL-INS
7325	60	FOI/LIV		PARCH CL-INS
7383	41	CLOD/CAT		PSD. 50D CL-INS
8082	67	CLOI/DXT	3.02.25	PARCH CL-INS
8160	135	FOD/LIV		PARCH CL-INS
8246	157	FOI/LIV	2.02.25	PARCH CL-INS