



Diseño e implementación de un sistema prototipo para el rastreo y monitoreo del ganado lechero utilizando tecnologías IoT en la ciudad de San Gabriel provincia del Carchi

Bastidas Cevallos, Yomira Marisol y Cortez Ayala, Yuleidy Dayanna

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Trabajo de titulación, previo a la obtención del título de Ingeniero en Electrónica y
Telecomunicaciones

Ing. Sáenz Enderica, Fabián Gustavo Mgs.

28 de junio del 2022



Bastidas_Yomira_Cortez_Yuleidy (2).pdf

Scanned on: 14:2 June 27, 2022 UTC



Overall Similarity Score



Results Found



Total Words in Text

Identical Words	418
Words with Minor Changes	81
Paraphrased Words	460
Omitted Words	0



www.copleaks.com
FABIAN GUSTAVO
SABRE ENDRILCA



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Certificación

Certifico que el trabajo de titulación: **"Diseño e implementación de un sistema prototipo para el rastreo y monitoreo del ganado lechero utilizando tecnologías IoT en la ciudad de San Gabriel provincia del Carchi"** fue realizado por las señoritas **Bastidas Cevallos, Yomira Marisol y Cortez Ayala, Yuleidy Dayanna**; el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Sangolquí, 28 de junio del 2022



Firmado digitalmente por:
**FABIAN GUSTAVO
SAENZ ENDERICA**

Ing. Sáenz Enderica, Fabián Gustavo Msc.

C. C.: 0102343985



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Responsabilidad de Autoría

Nosotras, **Bastidas Cevallos, Yomira Marisol** con cédula de ciudadanía n° 0504091661 y **Cortez Ayala, Yuleidy Dayanna** con cédula de ciudadanía n° 0401523980, declaramos que el contenido, ideas y criterios del trabajo de titulación: **Diseño e implementación de un sistema prototipo para el rastreo y monitoreo del ganado lechero utilizando tecnologías IoT en la ciudad de San Gabriel provincia del Carchi** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 28 de junio del 2022

Bastidas Cevallos, Yomira Marisol

C.C.: 0504091661

Cortez Ayala, Yuleidy Dayanna

C.C.: 0401523980



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Autorización de Publicación

Nosotras **Bastidas Cevallos, Yomira Marisol** con cédula de ciudadanía n° 0504091661 y **Cortez Ayala, Yuleidy Dayanna** con cédula de ciudadanía n° 0401523980, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Diseño e implementación de un sistema prototipo para el rastreo y monitoreo del ganado lechero utilizando tecnologías IoT en la ciudad de San Gabriel provincia del Carchi** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Sangolquí, 28 de junio del 2022

Bastidas Cevallos, Yomira Marisol

C.C.: 0504091661

Cortez Ayala, Yuleidy Dayanna

C.C.: 0401523980

DEDICATORIA

Este trabajo va dedicado en primer lugar a Dios por brindarme salud, sabiduría y fuerza para poder lograr alcanzar esta meta. A mis amados padres Narcisa y Juan que han sido el pilar fundamental en esta difícil etapa, por el amor incondicional que siempre me han brindado, por sus consejos que me han guiado día a día, por su paciencia, sacrificio y esfuerzo para apoyarme a lo largo de mi carrera universitaria, son la fuente de mi motivación para salir adelante.

A mi hermana Fernanda por siempre estar para mí en las buenas y en las malas, especialmente en los momentos más grises, por su cariño y por su empuje para que nunca me dé por vencida; y,

A mi querido ángel, mi abuelita Luz Amelia que, aunque ya no se encuentra junto a nosotros, siempre creyó en mí y me mantenía en sus oraciones, me brindo palabras de aliento que aún resuenan en mis oídos. Ahora solo puedo susurrarte al cielo, lo logre.

A mis demás familiares y amigos que de una u otra forma estuvieron apoyándome en esta etapa, gracias por su apoyo moral.

Yomira Marisol Bastidas Cevallos

Este trabajo va dedicado a Dios, por permitirme estar aquí para cumplir cada uno de mis sueños.

A mi familia por ser el motor principal para terminar con mi carrera profesional y quienes han estado en los triunfos y fracasos a lo largo de mi vida universitaria y que sin dudarlo me brindaron su apoyo incondicional para poder enfrentar obstáculos que se han presentado durante todo este tiempo de estudio.

A mis amigos y amigas que han estado siempre para apoyarme y con los que he compartido buenos y malos momentos que me han permitido aprender más de la vida.

A Lenin por su cariño, amor y apoyo incondicional durante este proceso y por motivarme a ser mejor en mi profesión.

Este logro es para ustedes.

Yuleidy Dayanna Cortez Ayala

AGRADECIMIENTO

En primer lugar, queremos dar gracias al Director de Tesis Ing. Fabián Sáenz, quien por medio de su experiencia y conocimientos nos orientó en cada una de las etapas de este proyecto para que lo culminemos de forma exitosa.

Nuestro agradecimiento a familiares, amigos, compañeros quienes de una u otra forma nos apoyaron y fueron fieles testigos de los esfuerzos y malas noches que pasamos para culminar este trabajo. De forma especial a nuestros padres, que siempre han estado presentes para apoyarnos, motivarnos y que gracias a sus sabios consejos hemos llegado a este punto final de nuestra carrera universitaria.

También agradecer al Ing. Cristhian Moreno quien muy amablemente nos facilitó los medios y herramientas que fueron necesarios para llevar a cabo el proceso de toma de datos, parte final de la investigación.

Por último, agradecemos a La Universidad de las Fuerzas Armadas ESPE, autoridades y maestros que fueron parte esencial de nuestra formación como profesionales y por permitirnos aprender de cada uno de ellos.

Yomira Bastidas y Yuleidy Cortez

Tabla de contenido

Resumen.....	21
Abstract.....	22
Capítulo I.....	23
Definición del Proyecto.....	23
Introducción	23
Antecedentes.....	24
Planteamiento del problema	26
Justificación	27
Alcance del Proyecto	29
Objetivos.....	29
Objetivo General	29
Objetivos Específicos	29
Viabilidad	30
Capítulo II.....	31
Estado del arte	31
Bovinos y su morfología.....	31
Ganado Bovino.....	31
Ganado según el clima.....	31
Producción de leche por ganado vacuno.....	33
Tipo de ganado bovino	33
Clasificación de las razas bovinas	34
Vacas para carne	34
Vacas para producción de leche	35
Valores fisiológicos normales de un bovino.....	35
Temperatura.....	36
Frecuencia cardíaca	36
Frecuencia respiratoria.....	36

Gestación	36
Movimientos rumiales	36
Velocidad de movimiento	37
Desplazamiento.....	37
Indicadores del estado de salud en ganado bovino.....	38
Vocalizaciones	39
Tasas de morbilidad	39
Cambio en el peso, cambio en la condición corporal y producción de leche.....	39
Eficiencia Reproductiva	40
Aspecto físico	40
Enfermedades relacionadas con la variación de temperatura y frecuencia	
cardíaca.....	40
Actinobacilosis.....	40
Carbón sintomático	41
Edema maligno	41
Hemoglobinuria bacilar.....	41
Leptospirosis	41
Pododermatitis	41
Influencia de los cambios climáticos en la producción de leche	42
Relación de la temperatura con la producción y calidad de la leche	42
Ganadería ecuatoriana	43
Sistemas de trazabilidad	44
Beneficios.....	44
Capítulo III.....	45
Marco teórico.....	45
Internet de las Cosas	45
Arquitectura de sistemas de IoT	46
Tecnologías de comunicación para IoT.....	46

GSM.....	47
Sigfox.....	48
LoRa	49
Zigbee	50
Comparación de las tecnologías de comunicación para el IoT.....	52
Tarjetas de Desarrollo.....	52
Comparación de las tarjetas de desarrollo	53
Sensores	54
Sensores de temperatura	54
Comparación de sensores de temperatura.....	55
Sensores de ritmo cardíaco.....	55
Comparación de sensores de ritmo cardíaco	56
Dispositivos de posicionamiento.....	56
Comparación de los dispositivos de posicionamiento.....	57
Fuentes de alimentación	58
Equipos Lora.....	58
Gateway	58
Módulo de comunicación.....	59
Tecnología Cloud Computing.....	60
Proveedores de Cloud Computing	61
AWS.....	61
Google Cloud	62
IBM Cloud	62
Microsoft Azure	62
Comparativa entre Proveedores de servicio Cloud Computing	63
Capítulo IV	65
Metodología	65
Método de investigación	65

Método cualitativo	65
Método cuantitativo	66
Requisitos funcionales	66
Universo y muestras	67
Universo	67
Muestra	68
Procesamiento de la información.....	68
Tecnología a emplearse en el sistema prototipo IoT	68
Arquitectura de la red de LoRa.....	69
Seguridad LoRaWAN	70
Dispositivos seleccionados para la construcción del prototipo IoT	72
Arduino Uno	72
Sensor de temperatura.....	73
Sensor de ritmo cardíaco	73
Sensor GPS	74
Módulo LoRa Shield	75
Gateway LoRa LG01-N	75
Batería LiPo	76
Servidor en la nube seleccionado para el prototipo IoT.....	77
Servidor en Amazon.....	77
Descripción del lugar geográfico.....	78
Extensión de la Hacienda	80
Descripción de la Vaca en la cual se coloca el dispositivo final.....	81
Capítulo V	83
Diseño e Implementación	83
Arquitectura del sistema.....	83
Diseño del nodo cliente.....	85
Diseño del nodo Gateway	87

Diseño del circuito Electrónico	89
Diagrama de conexiones	89
Comparación de sensores seleccionados con equipos patrones	91
Termómetro GP-300 y el sensor de temperatura MLX90614.....	91
Oxímetro Teaisiy C101H1 y el sensor de pulso cardíaco	93
Google Maps y GPS UBLOX NEO 2019	95
Implementación del Hardware y Software del sistema prototipo IoT	98
Diseño en PCB del circuito electrónico	98
Nodo cliente	98
Programación del nodo cliente	99
Sensor de Temperatura.....	99
Sensor de Frecuencia Cardíaca	100
Sensor de Seguridad.....	101
Sensor GPS UBLOX NEO.....	102
Módulo LoRa Shield	103
Diseño 3D de la carcasa del nodo cliente.....	107
Nodo Gateway.....	108
Diseño del servidor en la nube AWS	113
Instalación Node.js	121
Configuración del Bróker para MQTT	122
Instalación de MySQL	125
Creación de la Base de datos.....	127
Desarrollo de la interfaz de visualización en Ubidots	129
Prototipo Final.....	135
Costo de Implementación del prototipo de collar.....	136
Cálculo del área de propagación.....	138
Capítulo VI	141
Pruebas y Resultados	141

Pruebas de alcance	141
Consumo energético.....	146
Pruebas de funcionamiento	149
Validación de la integración de sensores al nodo	150
Validación del sistema de comunicación LoRa	150
Validación de seguridad del prototipo de collar.....	151
Pruebas de conexión Gateway LoRa al Servidor AWS.....	153
Pruebas de conexión Servidor – Ubidots.....	156
Resultados y análisis de la información obtenida.....	156
Visualización de las alarmas en Ubidots.....	167
Representación gráfica en Ubidots.....	168
Capítulo VII	169
Conclusiones, Recomendaciones y Trabajos futuros	169
Conclusiones	169
Recomendaciones	170
Trabajos futuros	171
Referencias	172

Índice de Tablas

Tabla 1 <i>Tecnologías de comunicación para el IoT</i>	52
Tabla 2 <i>Tipos de Tarjetas de Desarrollo en el mercado</i>	53
Tabla 3 <i>Tipos de sensores de temperatura</i>	55
Tabla 4 <i>Tipos de sensores de ritmo cardíaco</i>	56
Tabla 5 <i>Tipos de los sensores de posicionamiento</i>	57
Tabla 6 <i>Tipos de fuentes de alimentación</i>	58
Tabla 7 <i>Gateways LoRa en el mercado</i>	59
Tabla 8 <i>Módulos de comunicación LoRa en el mercado</i>	60
Tabla 9 <i>Proveedores de servicio de Cloud Computing</i>	63
Tabla 10 <i>Distribución de los pines LoRa Shield con Arduino</i>	86
Tabla 11 <i>Pruebas de geolocalización GPS-NEO vs Google Maps</i>	97
Tabla 12 <i>Elementos y costos del prototipo de collar</i>	136
Tabla 13 <i>Coordenadas del Gateway y puntos de cercado del área de prueba</i>	141
Tabla 14 <i>Valores de SNR y RSSI</i>	143
Tabla 15 <i>Consumo teórico de los componentes del nodo</i>	147

Índice de figuras

Figura 1 <i>Ganado lechero de zonas cálidas o tropicales</i>	32
Figura 2 <i>Ganado lechero de zonas frías o templadas</i>	32
Figura 3 <i>Tipos de ganado bovino</i>	33
Figura 4 <i>Arquitectura para IoT</i>	46
Figura 5 <i>Estructura de la red GSM</i>	48
Figura 6 <i>Estructura de la red Sigfox</i>	49
Figura 7 <i>Estructura de la red LoRa</i>	50
Figura 8 <i>Estructura de red en malla de Zigbee</i>	51
Figura 9 <i>Posicionamiento competitivo del proveedor de la nube</i>	61
Figura 10 <i>Arquitectura de LoRa</i>	70
Figura 11 <i>Seguridad de LoRaWAN</i>	71
Figura 12 <i>Arduino Uno</i>	72
Figura 13 <i>Sensor MLX90614</i>	73
Figura 14 <i>Sensor XD-58C</i>	74
Figura 15 <i>Sensor GPS</i>	74
Figura 16 <i>Módulo LoRa Shield</i>	75
Figura 17 <i>Gateway LoRa LG01-N</i>	76
Figura 18 <i>Batería</i>	77
Figura 19 <i>Ubicación de la Hacienda La Morenita</i>	79
Figura 20 <i>Ordeño</i>	79
Figura 21 <i>Diferentes Áreas de la Hacienda</i>	80
Figura 22 <i>Diferentes Áreas de la Hacienda</i>	80
Figura 23 <i>Diferentes Áreas de la Hacienda</i>	81
Figura 24 <i>Extensión donde se realizarán las pruebas</i>	81
Figura 25 <i>Vaca en la cual se colocó el dispositivo</i>	82
Figura 26 <i>Diagrama de bloques de la arquitectura del sistema</i>	83

Figura 27 <i>Módulo Lora Shield con pines para conexión con Arduino</i>	85
Figura 28 <i>Diseño del diagrama de flujo del nodo final</i>	87
Figura 29 <i>Diseño del diagrama de flujo del nodo Gateway</i>	89
Figura 30 <i>Diagrama Esquemático</i>	90
Figura 31 <i>Esquema de conexión física del nodo cliente</i>	91
Figura 32 <i>Termómetro GP-300</i>	92
Figura 33 <i>Curvas de temperatura</i>	93
Figura 34 <i>Diagrama de dispersión de la temperatura</i>	93
Figura 35 <i>Oxímetro Teaisiy C101H1</i>	94
Figura 36 <i>Curvas del pulso cardíaco</i>	94
Figura 37 <i>Diagrama de dispersión del pulso cardíaco</i>	95
Figura 38 <i>Coordenadas obtenidas por el sensor GPS</i>	96
Figura 39 <i>Coordenadas obtenidas por Google Maps</i>	97
Figura 40 <i>Diseño en PCB</i>	98
Figura 41 <i>Configuración del sensor MLX90614</i>	100
Figura 42 <i>Configuración del sensor de frecuencia cardíaca</i>	101
Figura 43 <i>Configuración del sensor de seguridad</i>	102
Figura 44 <i>Código para la configuración del módulo GPS</i>	103
Figura 45 <i>Código para configurar el módulo LoRa</i>	104
Figura 46 <i>Concatenación de cada variable con sus respectivos valores</i>	105
Figura 47 <i>Código para el envío y recepción de datos</i>	106
Figura 48 <i>Diseño de la caja para el nodo cliente</i>	107
Figura 49 <i>Conexión vía Ethernet con conexión directa desde la PC</i>	108
Figura 50 <i>Interfaz de inicio de sesión del Gateway LoRa</i>	109
Figura 51 <i>Configuración del Gateway LoRa a la red WiFi de la hacienda La Morenita</i>	109
Figura 52 <i>Configuración de los parámetros de radio LoRa</i>	110
Figura 53 <i>Configuración de los parámetros de MQTT en el Gateway LoRa</i>	111

Figura 54 Configuración de los parámetros del canal MQTT en el Gateway LoRa	112
Figura 55 Formato del payload LoRa configurado para las variables del prototipo de collar IoT	113
Figura 56 Crear Cuenta en la consola de Amazon	113
Figura 57 Lanzamiento de la instancia	114
Figura 58 Creación de la Instancia-Selección de parámetros	115
Figura 59 Aumento de Volumen de almacenamiento	116
Figura 60 Puertos para el Grupo de Seguridad	117
Figura 61 Creación de las llaves	117
Figura 62 Elección de llaves	118
Figura 63 Colocar la dirección IP	118
Figura 64 Acceso a la instancia por medio de SSH	119
Figura 65 Creación de Dominio	120
Figura 66 Asignación de IP y dominio	120
Figura 67 Carpeta que contiene los certificados SSL	121
Figura 68 Archivos de descarga para instalar Node.js	122
Figura 69 Acceso al bróker por medio de MQTT	125
Figura 70 Instalación de mysql	125
Figura 71 Configuración de credencial para la comunicación en tiempo real	126
Figura 72 Creación de la base de datos	127
Figura 73 Creación de la tabla	128
Figura 74 Tabla con datos	128
Figura 75 Crear política	130
Figura 76 Creación de claves para conexión con Ubidots	131
Figura 77 Creación del complemento en Ubidots	132
Figura 78 Variables asignadas en Ubidots	133
Figura 79 Interfaz de visualización en Ubidots	134
Figura 80 Imagen del prototipo final que se colocó a la vaca	135

Figura 81 <i>Peso del dispositivo final</i>	136
Figura 82 <i>Ubicación del Gateway</i>	141
Figura 83 <i>Gráfica de la ubicación del Gateway y puntos de cercado</i>	142
Figura 84 <i>Diagrama de bloques de la alimentación</i>	146
Figura 85 <i>Consumo de corriente del nodo arrojado por un amperímetro</i>	148
Figura 86 <i>Gráfica del consumo energético del nodo cliente</i>	149
Figura 87 <i>Verificación de la toma de datos por cada uno de los sensores</i>	150
Figura 88 <i>Recepción de datos en el Gateway</i>	151
Figura 89 <i>Visualización de cambios de estado de seguridad por el monitor serie de Arduino</i>	152
Figura 90 <i>Visualización del mensaje de alerta en la plataforma Ubidots</i>	152
Figura 91 <i>Estructura de la conexión entre el gateway y AWS</i>	153
Figura 92 <i>Parámetros de configuración del WebSocket en el bróker</i>	154
Figura 93 <i>Ratificación del cliente publicador en el bróker</i>	154
Figura 94 <i>Llegada de los datos del gateway al bróker</i>	155
Figura 95 <i>Llegada de datos a la base de datos en Amazon</i>	155
Figura 96 <i>Ingreso a la aplicación de Ubidots</i>	156
Figura 97 <i>Disposición del dispositivo collar en la vaca Nua</i>	157
Figura 98 <i>Temperatura superficial de la vaca Nua en la Semana 1</i>	158
Figura 99 <i>Temperatura superficial de la vaca Nua en la Semana 2</i>	159
Figura 100 <i>Ritmo cardíaco de la vaca Nua en la Semana 1</i>	160
Figura 101 <i>Ritmo cardíaco de la vaca Nua en la Semana 2</i>	161
Figura 102 <i>Gráfica de geolocalización de la vaca Nua en la Semana 1</i>	162
Figura 103 <i>Gráfica de geolocalización de la vaca Nua en la Semana 2</i>	162
Figura 104 <i>Gráfica de los datos obtenidos de geolocalización en Google Maps</i> ...	163
Figura 105 <i>Velocidad de la vaca Nua en la Semana 1</i>	164
Figura 106 <i>Velocidad de la vaca Nua en la Semana 2</i>	164
Figura 107 <i>Gráfica superponiendo temperatura vs velocidad</i>	165

Figura 108 <i>Desplazamiento o distancia recorrida de la vaca Nua durante los 15 días de prueba</i>	166
Figura 109 <i>Mensaje de alarma cuando el collar es desconectado</i>	167
Figura 110 <i>Mensaje de alarma cuando se sobrepasan los límites de temperatura</i>	167

Resumen

En la actualidad hablar de IoT es posible gracias a que la ciencia ha evolucionado a pasos agigantados, el objetivo principal de estos avances son el de mejorar la vida del ser humano y que procesos complejos se los pueda realizar de una forma más fácil, automática y que implique menor tiempo, es decir que la respuesta frente a cualquier situación sea ágil. Es así que ahora se puede hablar de la incursión de la tecnología en el sector ganadero con el único afán de mejorar la calidad de vida de los animales de granja, mejorar la calidad de alimentos que provienen de algunos de estos animales y en algunos casos mejorar la calidad de materia prima que también es obtenida de los mismos. En el presente trabajo se ha desarrollado un dispositivo prototipo que permite monitorear y rastrear a vacas lecheras, este dispositivo se encarga de medir variables como son: temperatura, frecuencia cardíaca, posición y velocidad, estos datos que se recolectan son visualizados en una aplicación móvil que es de fácil acceso para el usuario, en esta aplicación se muestran los datos de cada día, y alertas en caso de que el animal haya sobrepasado los límites en temperatura y posición, además se alerta cuando el collar o dispositivo prototipo ha sido retirado del animal, todos estos datos se almacenan para tener un reporte organizado y poder verificar si la vaca está en condiciones normales en su desarrollo o si los cambios tuvieron que ver con alguna alteración en su estado de salud. Además, en este trabajo se concluyó que LoRa es ideal para realizar el envío y recepción de datos, ya que esta tecnología es muy efectiva ya que tiene un largo alcance y además permite enviar datos de manera ilimitada.

Palabras claves: monitoreo, sensores, LoRa, IoT.

Abstract

Currently talking about IoT is possible thanks to the fact that science has evolved by leaps and bounds, the main objective of these advances are to improve the life of the human being and that complex processes can be carried out in an easier, automatic way and that implies less time, that is to say that the response to any situation is agile. Thus, now we can talk about the incursion of technology in the livestock sector with the sole desire to improve the quality of life of farm animals, improve the quality of food that comes from some of these animals and in some cases improve the quality of raw material that is also obtained from them. In the present work a prototype device has been developed that allows to monitor and track dairy cows, this device is responsible for measuring variables such as: temperature, heart rate, position and speed, these data that are collected are visualized in a mobile application that is easily accessible to the user, in this application the data of each day are shown, and alerts in case the animal has exceeded the limits in temperature and position, it is also alerted when the collar or prototype device has been removed from the animal, all this data is stored to have an organized report and to be able to verify if the cow is in normal conditions in its development or if the changes had to do with any alteration in its state of health. Also in this work it was concluded that LoRa is ideal for sending and receiving data, since this technology is very effective since it has a long range and also allows you to send data in an unlimited way.

Key words: monitoring, sensors, LoRa, IoT.

Capítulo I

Definición del Proyecto

Este capítulo expone los antecedentes, el planteamiento del problema, la justificación, el alcance, los objetivos a cumplir, la viabilidad y las variables dependientes e independientes del presente proyecto.

Introducción

En la actualidad se vive en la era de los dispositivos conectados a la Internet. El término IoT ha revolucionado múltiples ámbitos industriales, empresariales y comerciales. Se estima que esta revolución siga creciendo debido a la demanda de comunicación inalámbrica, asimismo la fidelidad y calidad de los datos transmitidos y enviados es crucial para lograr una mayor participación de los usuarios.

San Gabriel perteneciente a la Provincia del Carchi ubicado al norte de Ecuador, es un sector donde la ganadería representa un tema de especial interés, ya que es altamente productor de ganado bovino como de sus derivados (leche, quesos, yogurt, mantequilla, etc), cuyo comercio representa una de las principales fuentes de ingresos del sector. En toda industria se suelen presentar riesgos que representan pérdidas monetarias al sector, entre los principales factores negativos está el robo o hurto del bovino por su elevado costo rodeando los \$500 a \$1000 dependiendo la raza. Uno de los aspectos por lo que ocurre el robo es porque se libera al animal en grandes extensiones de campos para que estos se alimenten lapsos de tiempo sin supervisión lo que es aprovechado por las personas dueño de lo ajeno. Otro factor negativo por el cual se pierden cabezas de ganado es su sistema rudimentario de monitoreo de variables indicadoras de su estado de salud que generalmente lo realizan por inspección visual y al tacto lo que incrementa la toma de decisiones no acertadas.

Al presenciar la afectación del sector ganadero ya que para quienes pertenecen a esta industria a veces representa su único ingreso, se plantea el desarrollo un sistema de rastreo y monitoreo del bovino en tiempo real para atacar esta problemática, con beneficios de uso como la visualización de los datos desde un aplicativo. El desarrollo de esta herramienta busca ayudar a los ganaderos optimizando su trabajo y así evitar las pérdidas de sus animales.

Antecedentes

El Internet de las cosas hace referencia a la interconexión de objetos que se usan diariamente en la red, con el objetivo de adquirir datos que pueden ser usados en diferentes temas como el monitoreo del clima, la detección de niveles de ruido en la ciudad o el nivel de contaminación del aire y además es posible aplicarlo al sector de ganadería, y de manera más específica a las vacas. En varios artículos se ha mencionado que M2M que se refiere a la comunicación Máquina a Máquina se utiliza en vacas no sólo para su monitoreo, sino para medir diferentes variables que permiten indicar el nivel de salud en estos animales y en base a esto realizar estadísticas con el objetivo de anticiparse a posibles enfermedades, algunos sensores pueden calcular la riqueza alimenticia que reciben y la calidad de la leche que producen, entre otros datos (ganadero, 2018). Además, en algunas investigaciones se ha mencionado el monitoreo del ciclo estral o el período en donde la vaca se encuentra en celo.

Monitoreo en la ganadería bovina a través de redes gsm/gprs, es un trabajo final de titulación que ha sido presentado en la UNIVERSIDAD SANTO TOMÁS DE AQUINO el año 2016 en la ciudad de Medellín, en donde el autor hizo uso de la tecnología de comunicación inalámbrica (módulo GSM/GPRS), y diferentes sensores para medir diferentes variables, entre las que se encuentra, la frecuencia cardíaca, la temperatura del animal y la posición, todo esto se realizó en tiempo real, en este caso el módulo GSM/GPRS capturó la información,

misma que fue enviada a través de un SMS mediante el uso de telefonía móvil, con el fin de mejorar el proceso que se lleva a cabo para la administración de ganado en el territorio de Medellín (Soyomar, 2016).

Protocolo LORA para el análisis de medición con GPS y arduino en la industria ganadera del Ecuador: una revisión sistemática, es un trabajo publicado en el 2021 en donde se concluye que para el sector de la ganadería en el Ecuador, es muy útil la implementación de una infraestructura que tenga tecnología LORA, debido a que los enlaces inalámbricos punto a multipunto son muy necesarios cuando se necesita abarcar distancias grandes cuyo acceso es complicado, además de que el tiempo de instalación es reducido y a la vez favorece a los escenarios que tienen la necesidad de flexibilidad de la movilidad mediante los sensores como es el caso de GPS y al momento de implementar esta tecnología es posible determinar que cuenta con la cobertura necesaria para ofrecer una red en el sector ganadero (Pazmiño, 2021).

A pesar de que LoRaWAN no está tan expandido como SigFox, en los últimos años se ha evidenciado el desarrollo de diferentes proyectos en base a este protocolo que se lo ha utilizado para el diseño de aplicaciones IOT.

Otro aspecto importante a mencionar es el robo de ganado ya que en Ecuador ha sido muy notable, especialmente en las provincias de la Sierra, por tanto, los avances tecnológicos que se han dado a nivel de la ganadería son de gran impacto, ya que al tener un localizador o GPS es posible el monitoreo del ganado con esto se puede determinar la posición exacta del animal y así reducir el robo de estos animales, además se puede realizar el monitoreo de otras variables, como por ejemplo, la temperatura, el ritmo cardíaco, mismas que afectan en gran magnitud a la producción ganadera.

Es necesario mencionar que ya existen empresas que brindan este servicio, como es el caso de GPS GANADOS, este se comporta como un pastor virtual, así el ganado equipado con un GPS permite a los dueños saber la ubicación exacta cuando salen a pastar (GPS GANADOS, 2018).

Se contextualiza el proyecto en la actualidad, proporcionando antecedentes sobre investigaciones que se han realizado para monitorear el ganado bovino.

Planteamiento del problema

Actualmente en Ecuador los representantes de CNT, Ministerio de Agricultura y Ganadería y de la Agencia de Regulación y Control Fito y Zoonosanitario (Agrocalidad) han venido analizando la ejecución de técnicas que hagan uso de la tecnología, con el fin de reducir e impedir el abigeato de ganado en el campo y han manifestado la importancia de incorporar en estos sistemas el monitoreo de la salud del ganado para poder generar la trazabilidad de estos animales, lo cual es demandado por los mercados internacionales (MAG, CNT y Agrocalidad analizan sistemas tecnológicos para evitar abigeato, 2020).

Además, se recalca que la identificación adecuada del tipo de tecnología para el campo es crucial ya que los problemas de línea de vista, conectividad son los principales a solventar, por ende, la elección de cual funcionaria mejor para transmitir datos y ubicaciones GPS, controlaría el hurto de ganado y nos brindaría información de su actividad diaria.

Generalmente este tipo de monitoreo lo realiza una persona de manera presencial, no representa mayor dificultad cuando el ganado es poco, pero si se supera cierta cantidad de reses este trabajo se torna dificultoso, con una serie de limitaciones que impiden llevar un monitoreo eficaz y confiable. Mientras que respecto a la localización del mismo es un problema, aún más cuando la zona ganadera es extensa. Lo que disminuye la pronta toma de decisiones para evitar

enfermedades del ganado bovino que repercute en la calidad de los alimentos producidos por la misma por ejemplo carne y leche, también prevenir el abigeato.

Frente a lo expuesto anteriormente, se han planteado las siguientes tres preguntas de acercamiento:

1. ¿Cómo afecta el robo de ganado en el sector económico en Ecuador?
2. ¿Qué tan beneficioso es contar con un dispositivo basado en tecnologías IoT para monitorear la actividad física de ganado bovino para los ganaderos de la ciudad San Gabriel de la provincia del Carchi, Ecuador?
3. ¿Cuáles son las variables fisiológicas específicas que debe censar el dispositivo para constatar la salud de las vacas?

A través de estas preguntas se pretende dar solución a los problemas que se presentan en cuanto al monitoreo de variables respecto al estado de salud del ganado bovino y a su localización en tiempo real a través de App móviles.

Justificación

En el Estado ecuatoriano, el ámbito ganadero cumple un rol sumamente importante para el desarrollo económico, ya que gracias a la ganadería es posible generar nuevos emprendimientos y como consecuencia de esto nuevas fuentes de empleo, por ejemplo de la vaca se pueden obtener diferentes derivados como son: carne, leche, cuero, cerdas, etc. y que al ser procesadas generan una gran variedad de productos óptimos para la venta y consumo. De acuerdo al Banco Central del Ecuador, el sector agrícola, ganadero, cultivo de bosques y la pesca aportan al PIB nacional el 9,63% con \$9.626.014 miles de dólares anuales. De acuerdo al Servicio de Rentas Internas o también conocido como SRI, constan 156.060 números RUC que pertenecen al sector de ganadería, de los cuales, 62.478 estuvieron en funcionamiento en el año 2014 al año 2019 (Mayorga, Freire, Sánchez, & Vayas, 2020).

Además, se debe mencionar que el ganado bovino en el Ecuador en el período entre el 2014 al 2019 cubrió un porcentaje del 66% siendo este sector el que mayor productividad brindó al país, así el 71% de la leche obtenida se destinó a la venta, y con respecto a la cría de estos animales en el año 2016 sumó \$ 583,60 millones, teniendo así una participación de 0.84% en el total del PIB. En el año 2016 se tuvo un incremento de 2% con respecto al año 2015 (CFN, 2017).

Dada la importancia de la ganadería bovina el objetivo de este proyecto se da en torno a las necesidades de los ganaderos como, por ejemplo: reducir el robo de ganado, mejorar y aumentar su productividad económica, vigilar el estado de salud de su ganado es decir realizar el monitoreo del comportamiento que tienen estos animales.

Por lo general los ganaderos supervisan el estado de sus animales de forma rudimentaria, por medio de la observación presencial desde horas de la madrugada hasta avanzadas horas de la noche, esto representa un gran esfuerzo físico. Pero cuando el número de vacas es grande, este método se vuelve complicado, con lo cual puede causar posibles pérdidas de reses e incluso disminuir la cantidad y calidad de la leche que producen.

Con respecto a lo mencionado con anterioridad, se plantea la implementación de un sistema prototipo que permita la adquisición de diferentes variables fisiológicas que se pueden usar para mejorar el sistema de productividad del bovino y conocer su estado de salud, tales variables son la temperatura corporal, ritmo cardiaco, velocidad de movimiento, desplazamiento, además de emitir la ubicación geográfica del ganado en tiempo real mediante el uso de tecnología IoT. Esta información se puede visualizar a través de una aplicación móvil, además se muestran alertas cuando alguna de las variables que se ha medido sobrepasa el rango normal que se ha tomado como base, por ejemplo, en el caso de que la temperatura aumente o disminuya puede significar que el animal tiene problemas de

salud. Adicionalmente, una alerta cuando el ganado esté fuera de una región específica, así el ganadero sabe que su bovino está perdido o es un intento de robo.

Alcance del Proyecto

El proyecto se enfoca en 4 aspectos que se detallan a continuación:

1. Investigación de temas que tienen similitud con la idea de proyecto presentado, para tener una base y mejorar las características del mismo, además de un correcto desarrollo de la documentación, identificación de las variables necesarias que se van a medir y de los elementos que van a formar parte del proyecto para cumplir con el objetivo principal.
2. Encontrar en el mercado los sensores y demás elementos que permitan medir las variables con gran precisión y exactitud, además una batería que sea de carga fácil y que no se desgaste rápidamente.
3. Desarrollo de la lógica de programación y funcionamiento automático del prototipo para el monitoreo y rastreo de ganado bovino.
4. Visualizar los datos por medio de una aplicación móvil, misma que permite verificar el funcionamiento del prototipo.

Objetivos

Objetivo General

Desarrollar un sistema prototipo capaz de monitorear y transmitir información de variables fisiológicas como la posición, velocidad de movimiento, desplazamiento, temperatura corporal y ritmo cardíaco de un animal bovino, con interacción al usuario mediante una aplicación móvil.

Objetivos Específicos

- Realizar un estudio del arte para recopilar información sobre la productividad y especificaciones de salud en bovinos.

- Investigar y seleccionar un servicio de tecnologías IoT que se usará para la aplicación móvil.
- Investigar y seleccionar los sensores y demás elementos electrónicos necesarios para la medición de las variables fisiológicas.
- Diseñar e implementar el sistema electrónico que permita monitorear las variables fisiológicas específicas determinadas para ser visualizadas desde una aplicación móvil.
- Realizar pruebas y mediciones para verificar el correcto funcionamiento del prototipo.

Viabilidad

- a) **Teórico:** hay varias fuentes de investigación que se han analizado, algunas se han realizado en Ecuador y otras se han realizado en Colombia, gracias a la información que se ha logrado encontrar se ha podido definir con mayor exactitud el enfoque del proyecto y en base a lo recopilado se puede mejorar las características y el alcance del prototipo que se desea realizar.
- b) **Recursos Naturales:** El estudio se lo realizará en animales, específicamente en vacas, en la ciudad de San Gabriel situada en la provincia del Carchi. Con la ejecución de esta investigación no se alterará ni causará ningún daño a un individuo, comunidad ni ambiente, más bien tiene la finalidad de conocer la ubicación de estos animales para evitar el robo y los signos vitales que ayudarán a identificar la existencia de alguna enfermedad.
- c) **Recurso Temporal:** El presente trabajo de investigación se lo realizará en un plazo de aproximadamente 8 meses, por la ejecución de todos los procesos de investigación.
- d) **Recursos Financieros:** El proyecto contará con el auspicio de una empresa, la misma que brindará los elementos y dispositivos necesarios para realizar el prototipo mencionado.

Capítulo II

Estado del arte

En el presente capítulo, se desarrollan los temas que tienen relevancia con las especificaciones de ganado bovino. Se detalla la descripción de ganado bovino y su morfología, se describen las razas de vacas bovinas, se mencionan los valores fisiológicos que permiten identificar si el animal está en condiciones normales, se describen los indicadores del estado de salud en bovinos, se explican las enfermedades en relación con la variación de la temperatura y frecuencia cardíaca así también los parámetros que afectan en la productividad de leche. Se presenta un análisis del sector ganadero y los sistemas de trazabilidad.

Bovinos y su morfología

Ganado Bovino

El ganado bovino o también conocido como ganado vacuno hace referencia al conjunto de vacas, toros y bueyes, mismos que han sido domesticados por el hombre para la producción en diferentes ámbitos, como por ejemplo producción de carne, leche e incluso producción de cuero.

Este grupo de mamíferos que se alimentan de hierba generalmente, permite satisfacer ciertas necesidades bien sea alimenticias o económicas, e incluso sus derivados son utilizados para la realización de otros productos que son utilizados por el ser humano.

“Estos animales se caracterizan por ser rumiantes, su cuerpo presenta un gran tamaño, se caracterizan por tener una altura entre 120cm a 150 cm y cuyo peso es de aproximadamente 600kg a 800 kg.” (ConceptoDefinición, 2021).

Ganado según el clima

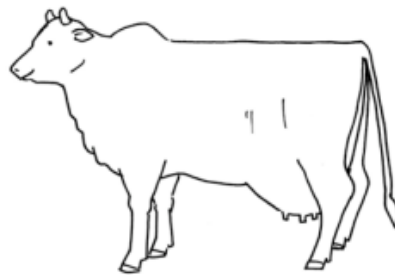
Con respecto al tipo de lugar en el que crece, este animal ha modificado su cuerpo con el fin de adaptarse a las distintas condiciones ambientales.

Un claro ejemplo es la caracterización del cuerpo del ganado vacuno, en caso de vivir en zonas frías o templadas, estos animales se caracterizan por tener una superficie corporal menor a la del ganado vacuno que vive en clima cálido o tropical, además poseen pelo en mayor cantidad con el objetivo de protegerse de temperaturas bajas e incluso evitar la disipación de calor, lo contrario pasa con los del clima cálido que tienen un pelaje corto y mayor pigmentación en la piel lo que facilita la disipación del calor en exceso.

A continuación, se muestra una comparación de la forma del cuerpo del ganado lechero de acuerdo al clima en el que se desarrollan.

Figura 1

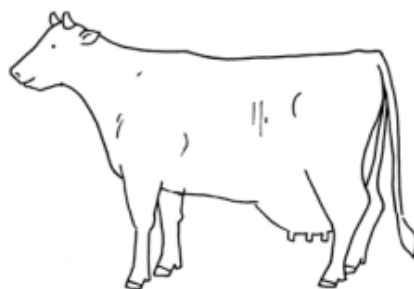
Ganado lechero de zonas cálidas o tropicales.



Nota. El gráfico expone al ganado lechero que se desarrolla en temperaturas abrigadas, como se puede apreciar, estos animales tienen joroba, pecho más amplio, una gran barbilla. Tomado de *Tipos de Ganado Bovino* (p.3), 2006.

Figura 2

Ganado lechero de zonas frías o templadas.



Nota. El gráfico representa al ganado lechero que vive en clima frío o templado, estos animales tienen una gran capacidad de producción lechera y son muy sensibles al calor y a las enfermedades del clima cálido. Tomado de *Tipos de Ganado Bovino* (p.3), 2006.

Producción de leche por ganado vacuno

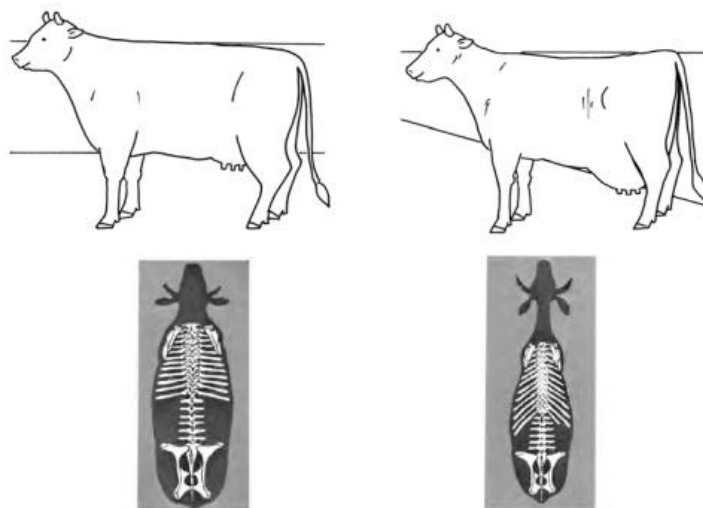
La cantidad de leche que puede producir una vaca, depende del tipo de ganado y del medio ambiente en el que vive, además de que podría producir una cantidad importante de leche siempre y cuando se le de las condiciones y bienestar adecuado.

Tipo de ganado bovino

A estos animales se les puede clasificar en: animales para la producción de leche, carne o en algunos casos para trabajo; el ganado que se usa para la producción de carne, tiene “forma rectangular y posee una mayor complexión por lo que la acumulación de carne es mayor”, mientras que el ganado que se usa para la producción de leche es de “forma triangular, su musculatura es muy poca y lo más esencial en este tipo de animal es que posee ubres grandes” (cf_pierola, 2016).

Figura 3

Tipos de ganado bovino.



Nota. El gráfico representa a los tipos de ganado vacuno, en el lado izquierdo se representa al ganado bovino para producción de carne (cuerpo forma rectangular), en el lado derecho se representa al ganado bovino para producción de leche (cuerpo forma triangular). Tomado de *Tipos de Ganado Bovino* (p.1), 2016.

Clasificación de las razas bovinas

Vacas para carne

Simmental. Esta raza es muy conocida a nivel mundial, tiene su origen en el valle Simme que se encuentra en Suiza. Esta raza es considerada como un animal de doble propósito, es decir que es usado para producción de leche, y para la producción de carne. Es un animal que tiene un gran crecimiento y docilidad. Entre los colores característicos están: rojizo, amarillo, rojo oscuro o negro.

Angus. Este tipo de animal es muy conocido por su rusticidad, tiene una buena productividad y es muy fértil, además el organismo es muy resistente a problemas relacionados con la pigmentación y a enfermedades; entre las cualidades más destacables, esta raza no desarrolla cáncer en ojos, tiene menor riesgo a padecer queratitis infecciosa y necrosis en las patas. Excelente raza para la producción de carne, con tamaño es moderado, cuenta con contornos redondeados y musculosos. Su garganta se caracteriza por no tener la presencia de papada, no acumula grasas debajo de la piel, entre los colores más comunes está el pelaje negro o colorado (Ganadería.com, s.f.).

Charolais. Esta raza tuvo su origen en las regiones Centro Oeste y Sudoeste de Francia, poseen un color blanco o blanco cremoso, es de gran tamaño: los toros adultos tienen un peso de 900 a 1250 kg, mientras que las vacas tienen un peso entre 560 a 950 kg, su pelaje se caracteriza por ser corto en la temporada de verano y es largo cuando es temporada de invierno. Entre las características que más destacan en este animal está el buen desarrollo que presenta la musculatura en sus extremidades. (Ganadería.com, s.f.).

Vacas para producción de leche

Holstein – Friesian. Esta vaca es originaria del norte de Holanda, a pesar de que este animal se ha introducido en climas diferentes, tiene un mejor desarrollo en clima templado, con respecto a la calidad de leche, es la que menos cantidad de sólidos presenta, es la raza con mayor peso, las hembras presentan 650 kilos aproximadamente, su ubre es bien balanceada y fuertemente adherida. El color representativo en su pelaje es el blanco y negro, pero también existe el color blanco y rojo. Esta raza es la mejor para la producción de leche con un promedio de 7, 899Lt. con 3,6% de grasa.

Pardo Suiza: Es originaria de Suiza, es la segunda raza más productora de leche en el mundo, el color de su pelaje es pardo-gris, tiene una baja incidencia en los procesos metabólicos, se adapta a los cambios climáticos extremos, tiene una fortaleza en patas y pezuñas, con respecto al peso, las hembras pueden alcanzar hasta un peso de 700kg y su producción de leche es de 6,779 Lt., el porcentaje de grasa y de proteína es de 4 y 3.5 respectivamente.

Jersey: Su origen proviene de la Isla de Jersey, este animal tiene una fácil adaptación a diferentes climas, incluyendo los tropicales, lo más sobresaliente de este animal, la leche producida por esta raza es rica en sólidos a diferencia de la Holstein. El peso de las hembras puede llegar hasta los 430kg, la producción de leche es de 5,265 Lt., el porcentaje de grasa y proteína es de 4.7 y 3.7 respectivamente (Valerio, s.f.).

Valores fisiológicos normales de un bovino

Las diferentes variables fisiológicas permiten describir los procesos homeostáticos del organismo, cuyo fin es el de conservar un equilibrio del medio interno frente a condiciones del ambiente, estos valores varían de acuerdo al animal” (CONtextoganadero, CONtextoganadero, 2020). A continuación, se muestran los valores normales en un animal bovino.

Temperatura

El horario óptimo para la toma de la temperatura es a las 8 am o a las 5 pm, con esto se evita que evita que la temperatura se vea afectada por los rayos solares, los valores normales están entre 37.7 a 39°C en ganado adulto, mientras que para los becerros los valores normales van desde 38.5 a 39.5 °C.

Si la temperatura supera el límite máximo se conoce como hipertermia, mientras que si el valor de la temperatura se encuentra por debajo del límite inferior se conoce como hipotermia, en cualquiera de estas dos situaciones es muy probable que el animal se encuentre enfermo.

Frecuencia cardíaca

El pulso permite conocer el estado del aparato circulatorio. La palpación en bovinos se realiza en el maxilar externo, las pulsaciones por minuto son de 80-120 en becerros, mientras que en ganado adulto va desde 40 a 80.

Frecuencia respiratoria

Indica el número de respiraciones por minuto en el animal. El animal toma oxígeno del medio y desprende dióxido de carbono, los valores normales son de 10 a 30 en ganado adulto, para los becerros va desde 15 a 40.

Gestación

Las vacas tardan 9 meses en concebir o 280 días.

Movimientos rumiales

Los movimientos rumiales hacen referencia al número de veces que el rumen se mueve, para realizar la mezcla y procesar, por lo general este proceso dura 2 minutos (CONtextogadero, 2019) .

Velocidad de movimiento

La velocidad de los animales es una magnitud en km/h o en m/s que relaciona la distancia recorrida y el periodo de tiempo usado en recorrer dicha distancia, es una variable fácil de determinar, pero son múltiples los factores que afectan su valor. Para los bovinos en pastoreo se ve alterada por el estado de salud del animal, edad y raza. Así también, en parte por las condiciones y tamaño del terreno, de la posición de los bebederos y sombras, de las condiciones climáticas, y de su comportamiento durante la caminata. Las velocidades de vacas lecheras en desplazamiento a ordeño o bebederos varían entre 2 a 5 km/h (Fumero, 2016).

Además, estos animales pueden correr un promedio de 40Km/h en condiciones normales (PRIETO & ROJAS SCARPETTA, 2019).

Desplazamiento

La actividad de los animales de caminar de un lugar a otro se lo conoce como desplazamiento. Aunque las vacas tienen una rutina muy marcada durante el día, cualquier cambio en la rutina puede alterar la producción de leche. Ellas pasan alrededor de 14 horas descansando y el resto del tiempo puede estar entre: 4 horas caminando, 5 horas y media alimentándose y 30 minutos bebiendo agua (Protection, 2016).

Esta actividad puede variar generalmente por la ubicación de ordeño y la disponibilidad de la pastura; el periodo de ovulación también genera cambios puesto que su temperatura disminuye y para estabilizarla aumenta su desplazamiento, esto pasa a menudo en zonas frías o templadas que en zonas cálidas.

Indicadores del estado de salud en ganado bovino

Gran parte de los factores que están vinculados con la producción animal afectan en el bienestar de los mismos, por ejemplo algunos de los factores que tienen un efecto notable sobre el bienestar animal es la alimentación, la sanidad, el lugar en el cual viven, si estos no están en las condiciones que se consideran normales para el ganado puede influir de forma negativa al desarrollo adecuado del animal y provocar enfermedades, estrés agudo o crónico, lo que trae consigo una disminución en la calidad de vida de estos animales, menor eficiencia reproductiva y un deterioro en los productos que proviene del ganado vacuno.

Tales motivos son suficientes para que el dueño de los animales de granja vele por el bienestar de los mismos, para lo cual se debe medir haciendo uso de una serie de indicadores (Armengol Silvia, 2006).

Los indicadores aportan información con respecto al estado de la salud de los animales y el estado emocional que se refleja en el comportamiento animal.

Para lo cual se consideran cuatro criterios siguientes:

- Alimentación
- Alojamiento
- Salud
- Comportamiento y emociones

Sin embargo cada uno de estos criterios puede subdividirse, un claro ejemplo sucede con la alimentación, aquí se incluye la ausencia de hambre y sed prolongada, con respecto al factor salud aquí se debe tomar en cuenta la ausencia de lesiones y enfermedades, además de la presencia de dolor que es producto de la castración, el corte de cola, el descornado, entre otros y para finalizar se hace referencia al comportamiento y las emociones aquí se debe considerar la relación que tiene el animal con su dueño o con la persona que se encarga de su cuidado y además la ausencia o presencia de temor en el animal. En caso de que el animal no

esté en condiciones óptimas para su desarrollo, el potencial productivo va a ser mínimo.

Además de los criterios que se mencionaron anteriormente se debe tomar en cuenta el indicador fisiológico, mismo que hace referencia al estrés o en otras palabras al funcionamiento del organismo animal. El estrés se provoca por la secuencia cronológica de una serie de estímulos de variable magnitud que de manera progresiva producen la transición de un estado de equilibrio a una reacción de alarma y en caso de que no tenga una respuesta adecuada el animal muere.

Los indicadores zootécnicos, se fundamentan en que el disconfort o situaciones de incomodidad afecta el nivel productivo de los animales, por lo tanto, se valora el bienestar en función del nivel sanitario y productivo.

Vocalizaciones

Son cualquier sonido audible que el bovino emite, las vocalizaciones son un indicador efectivo del bienestar animal, en este caso el 99% de las mismas tienen relación con eventos aversivos como el uso de picana eléctrica o intentos fallidos de insensibilización, además el sonido que produce la vaca puede brindar información sobre el estado neurofisiológico del animal, procesos cognitivos y afectivos, siempre que sea interpretado de forma correcta.

Tasas de morbilidad

Las tasas de morbilidad pueden ser indicadores directos o indirectos de bienestar animal del ganado. Al comprender el origen de la enfermedad de estos animales se puede actuar a tiempo en caso de que el bienestar animal no sea adecuado.

Cambio en el peso, cambio en la condición corporal y producción de leche

Los cambios de peso corporal especialmente una pérdida repentina de peso, son indicadores de deficiencia en la sanidad y el bienestar animal, además la

disminución de la producción de leche es un indicador que compromete el bienestar animal.

Eficiencia Reproductiva

Los bajos niveles con respecto a la reproducción en una raza en particular, pueden ser indicadores de que el bienestar del animal no está en sus condiciones normales

Aspecto físico

Los síntomas siguientes pueden indicar problemas en la salud y bienestar del animal:

- presencia de ectoparásitos
- pérdida de pelaje, color o textura anormal del pelo
- inflamaciones, heridas o lesiones
- anomalías en las pezuñas
- posturas anormales como, por ejemplo: arqueamiento dorsal
- deshidratación

Enfermedades relacionadas con la variación de temperatura y frecuencia cardíaca.

Actinobacilosis

Aquí el animal pierde peso de forma progresiva cara y cabeza, presentándose tumefacciones, abscesos y fístulas; especialmente en la lengua, carrillos y piso de la lengua. Los ganglios linfáticos tienden a inflamarse con lo cual se provoca la formación de abscesos, teniendo dificultad para el consumo de los alimentos, masticación y la rumia, además se presenta el incremento en la temperatura, con lo cual se provoca la fiebre.

Carbón sintomático

La enfermedad afecta a los animales menores de tres años, por lo general hay presencia de fiebre, claudicación y en casos extremos el animal no puede levantarse. La zona muscular que ha sido afectada presenta calor y dolor, luego se vuelve fría y el dolor desaparece.

Edema maligno

El animal presenta fiebre, se produce inflamación cerca de la herida con exudado seroso y gas subcutáneo, cuando hay la presencia de gérmenes asociados.

Hemoglobinuria bacilar

El animal presenta hemoglobinuria y anemia. Se da una disminución en el apetito y la secreción láctea disminuye, hay polipnea, taquicardia, en caso de que la temperatura esté demasíadamente baja, es un indicador de que el animal puede morir rápidamente si no es atendido.

Leptospirosis

Esta enfermedad presenta los siguientes síntomas: fiebre, ictericia, hemoglobinuria, la leche se vuelve color ladrillo, en caso de que la vaca esté en período de gestación la vaca puede abortar, y ya en casos extremos hay presencia de hemorragia en las mucosas, insuficiencia renal, y meningitis.

Pododermatitis

El animal al padecer de esta enfermedad presenta fiebre a veces muy tenue, manifiesta gran dolor, en ocasiones se le hace imposible desplazarse de un lugar a otro, se da la presencia de pérdida de peso, su producción reduce notablemente, si es vaca lechera, pero si es un toro semental, esta infección dificulta o impide la monta (Guadalupe Parra, 2013).

Influencia de los cambios climáticos en la producción de leche

Los cambios climáticos afectan en gran magnitud a la producción y a la calidad de la leche.

Además, la reducción de la humedad en el suelo afecta la absorción de nutrientes por parte de la planta y la concentración de los mismos en el forraje, afectando en la alimentación del animal y con esto la calidad de leche se ve perjudicada.

Los bovinos tienen una temperatura corporal que es considerada constante por sus procesos vitales y productivos, por lo tanto, en los procesos metabólicos, se produce calor, que sumado a las condiciones ambientales a las que están expuestos estos animales, pueden generar procesos de estrés calórico que afectan el desempeño productivo y reproductivo del animal, al igual que la calidad de la leche.

Relación de la temperatura con la producción y calidad de la leche

El 10% de la producción de leche varía por los factores climáticos tales como la temperatura. Una mala producción de leche debido al estrés por calor se debe a que el consumo de alimento disminuye, mientras que las necesidades de energía del animal aumentan (Gómez, 2017).

En caso de que la temperatura del medio sobrepase los 27°C para los animales de origen templado y 35°C para los de origen tropical, provoca efectos negativos en los sistemas de termorregulación, como es el caso del aumento de temperatura rectal, falta de apetito, la producción de leche disminuye con un cambio en la composición de la misma y en ganado de carne, y en algunos casos pérdida de peso lo que puede provocar un retraso en el desarrollo del animal.

La producción de leche disminuye entre un 50 y 75% con rangos superiores a 26,5°C en vacas de raza Holstein y superiores a 29,5°C en vacas de raza Jersey y Pardo Suizo. La temperatura crítica a la cual se produce la disminución en la

producción de leche oscila entre 21 y 26,5°C para vacas de raza Holstein y Jersey y entre 29,5 y 32°C para vacas de raza Pardo Suizo (Salvador).

A temperaturas bajas la producción de leche no se ve tan afectada en comparación a temperaturas superiores a 15,5°C, y también se da una disminución en el rendimiento en grasa y con ello se disminuye la proporción de los ácidos grasos de cadena corta.

Ganadería ecuatoriana

La ganadería en Ecuador es una actividad productiva de gran importancia para la economía, ya que a partir de esta se generan alimentos indispensables para el consumo, como la carne, leche y derivados lácteos ricos en proteínas.

La cría de ganado bovino se dividió en dos grandes categorías: la cría de ganado para producción de carne o leche; y, por otro lado, las unidades familiares en las que la tenencia de ganado cumple funciones múltiples.

La ganadería es muy importante dentro del sector agropecuario del Ecuador ya que contribuye a la actividad económica rural con la oferta de productos que provienen de la carne y de la leche, que son parte de la canasta básica y la seguridad alimentaria del país.

Con respecto al número de cabezas de ganado que existen en todo el país, en 2018 hay un registro de 4.10 millones de cabezas de ganado vacuno, el 37% correspondían a la costa. Del total de ganado vacuno, el 55% pertenecen a la raza criolla, 43% pertenecen a las razas Holstein Friessian, Brahaman y otros; una mínima proporción corresponde a razas puras para la línea carne, leche y doble propósito.

El uso de tecnología en este sector es muy importante, ya que genera un gran aporte a la innovación en estos sectores tan estratégicos para el desarrollo

económico del país, sin embargo, el poco apoyo que existe y las pequeñas inversiones en créditos ganaderos, hace que el mejoramiento en la producción ganadera sea escaso.

Sistemas de trazabilidad

La trazabilidad garantiza que los alimentos cumplan con diferentes normas de seguridad y de sanidad, además que sean aptos para como el consumo humano, estos requisitos se encuentran en el Código Sanitario de la Organización Mundial de la Sanidad Animal.

Las personas que se encargan de producir alimentos o cualquier otro producto tienen como base, la trazabilidad, ya que por medio de esta se puede realizar una administración y manejo adecuado de la calidad de dichos productos, mientras que el ente gobernante se encarga de verificar que los productores cumplan con los requisitos y puedan a asegurar a la población de que los alimentos son aptos para el consumo.

Una de las opciones para que la trazabilidad se efectúe de forma adecuada, es hacer uso de la identificación electrónica por radiofrecuencia, aquí se colocan microchips que permiten el control de la vacuna, el estado de gestación, enfermedades, y otros aspectos que el dueño necesita para controlar su ganado, sin embargo en nuestro país se usa el arete, que no permite controlar de forma efectiva al animal y con ello se acarrea un sin número de problemas. (El Productor, 2017).

Beneficios

Los beneficios son:

- Control de ingreso de enfermedades exóticas.
- Certificar la seguridad alimentaria.
- Garantizar la certificación oficial de exportaciones.
- Cumplir con cada una de las nuevas exigencias del mercado.

Capítulo III

Marco teórico

En este capítulo, se presentan varios conceptos que son usados en el proyecto, como el IoT y su arquitectura, se describen de forma general las tecnologías de comunicación, se mencionan las tarjetas de desarrollo, sensores y elementos enfocados a los criterios diseño del prototipo. Se presentan los proveedores de Cloud Computing y se describe el entorno de desarrollo de una App móvil.

Internet de las Cosas

Actualmente el término Internet de las Cosas (IoT) está presente en el día a día, sin embargo, no existe una definición universal aceptada para dar significado a esta palabra. En términos generales, el Internet de las Cosas hace referencia a escenarios en los que la conexión a una red de internet y la capacidad de cómputo se despliegan a objetos, sensores y artículos que habitualmente se usa a diario, con el fin que estos dispositivos conecten e intercambien datos con una mínima participación humana (Rose, Eldridge, & Chapin, 2015).

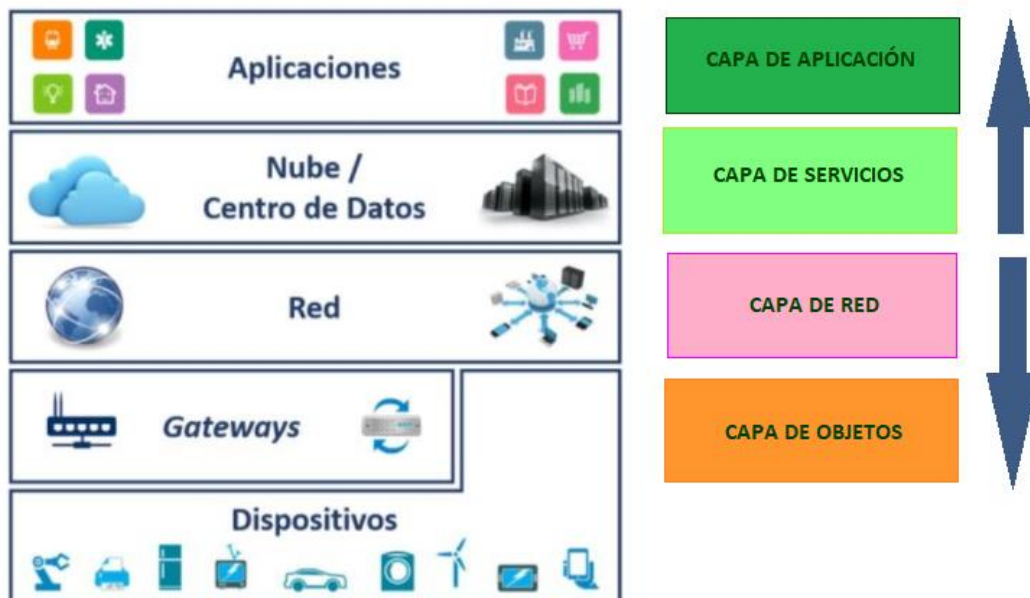
Se trata de una tecnología que engloba la integración de productos comunes, sistemas y sensores en red, que aprovechan los avances tecnológicos como la miniaturización de los elementos electrónicos y las interconexiones de red, abriendo paso a nuevas oportunidades de negocio y capacidades que antes no eran viables. Esta tecnología se ha venido impulsando gracias a seis tendencias tecnológicas: conectividad ubicua, adopción generalizada de redes basadas en el protocolo IP, economías en la capacidad de cómputo, miniaturización, avances en la interpretación de datos y surgimiento de la Cloud Computing. Hay muchos beneficios de esta nueva tecnología, como mejorar la eficiencia y productividad.

Arquitectura de sistemas de IoT

El principal requisito de IoT es que las cosas u objetos físicos estén interconectados en la red. La arquitectura de un sistema de IoT debe asegurar las operaciones de las cosas, permitiendo establecer un puente entre las cosas y el mundo virtual de IoT (Sosa, Tello, & Lara). En base al enfoque de la Arquitectura Orientada a Servicios (SOA) para el desarrollo de sistemas de IoT esta tecnología se conforma de 4 capas, como se muestra en la Figura 4

Figura 4

Arquitectura para IoT.



Nota. La figura representa las 4 capas de la arquitectura basada en SOA para IoT. Tomado de *Propuesta de Arquitectura para Internet de las Cosas* (p.3), Benítez, Calderón y Plasencia, 2016.

Tecnologías de comunicación para IoT

Las redes de comunicaciones han ido evolucionando hacia el sector del IoT, que permiten principalmente conectar dos “generadores de datos”, los cuales no se encuentran cercanos entre sí, que, conectados de alguna forma a Internet, y que al

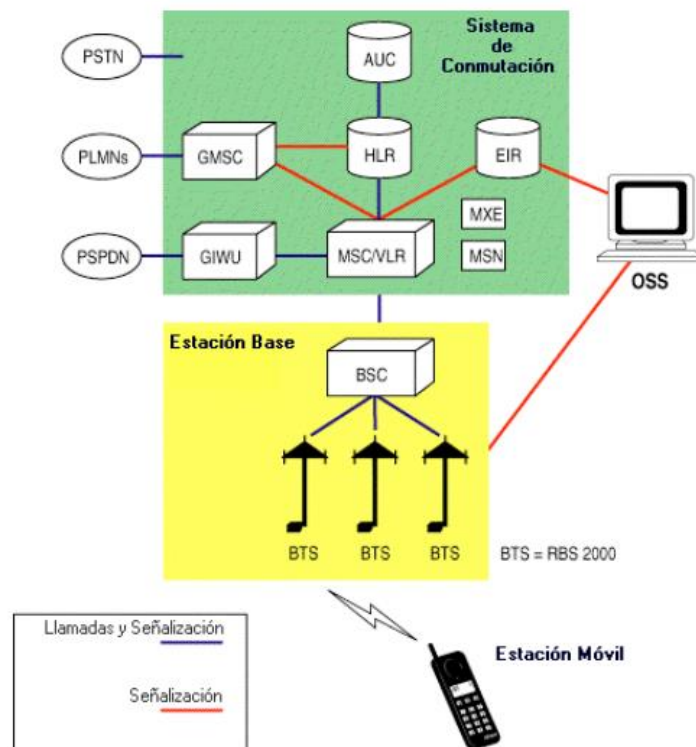
disponer de una red de comunicaciones les permite mandar datos a Internet para su posterior tratamiento y prestación de servicios, estos son los pilares fundamentales para un proyecto de Internet de las Cosas. A continuación, se detallan algunas de las tecnologías de comunicación para IoT:

GSM

Las siglas de GSM corresponden al nombre en inglés del Global System for Mobile Communications (Sistema Global para las Comunicaciones Móviles), que es el sistema de telefonía móvil digital más utilizado en la mayor parte del mundo y el estándar adoptado para teléfonos móviles en Europa. El sistema GSM es el sistema para comunicaciones móviles de segunda generación basado en la tecnología de células de radio, misma que aparece para dar solución a los problemas de los sistemas analógicos. El estándar es abierto, no propietario y poseía alrededor del 80% de los usuarios de teléfonos móviles del mundo en 2004. Está diseñado para transmitir servicios móviles de voz, por lo tanto, se basa en la conmutación de circuitos, lo que le diferencia del sistema General Packet Radio Service (GPRS). Al llevar a cabo la transmisión mediante conmutación de circuitos los recursos quedan ocupados durante toda la comunicación y la tarificación es por tiempo (EcuRed, 2021). GSM opera en cuatro principales bandas GSM-850, GSM-900, GSM-1800 y GSM-1900. GSM 900 MHz y GSM 1800 MHz son utilizados en la mayor parte del mundo, a excepción de Estados Unidos, Canadá y el resto de América Latina, lugares en los que se utilizan las bandas de 850 y 1900 MHz, puesto que en EE. UU, las bandas de 900 y 1800 MHz están ocupadas para uso militar (Hernández, 2011).

Figura 5

Estructura de la red GSM.



Nota. La figura muestra el esquema de funcionamiento de la red GSM. Tomado de *Guía para el desarrollo de aplicaciones GSM basadas en el Sistema de Señalización Nº7*, Lizón, 2013.

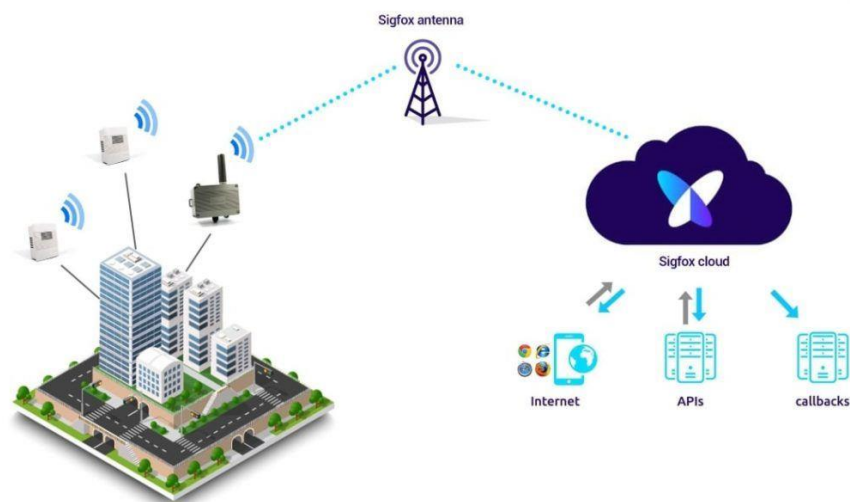
Sigfox

Sigfox es una red de comunicaciones LPWAN (Low-power Wide-area network) o red de bajo consumo y área extensa, enfocada para IoT más extendida a nivel mundial, cuenta con una cobertura cercana al 98% del territorio europeo y americano. Esta red está diseñada para tener un bajo consumo de energía, disminuyendo el costo de dispositivos conectados a la red. Está construida sobre una modulación de ultra narrow band (UNB) diseñada para trabajar con bajas velocidades de transferencias de 10 a 1.000 bits por segundo. La red Sigfox opera en las bandas de 868MHz en Europa y 902MHz en Estados Unidos. Los fabricantes de dispositivos IoT se han adaptado a la tecnología Sigfox, facilitando la subida de

datos a la nube de Sigfox para su posterior manipulación a través de cualquier conexión a Internet al permanecer los datos en los servidores de la compañía. A esto hay que añadir que tienen un despliegue y cobertura casi global, además disponibilidad de soporte por Azure de Microsoft, lo que acelera en gran medida la ejecución de un proyecto de IoT. El bajo costo de esta tecnología, su aceptación por los fabricantes de dispositivos, o el que sea una red bidireccional son otros factores a favor (efor, 2020). La estructura de la red Sigfox se constituye de antenas y estaciones base independientes de los despliegues de telefonía, las cuales se comunican con sensores finales y con el servidor de Sigfox.

Figura 6

Estructura de la red Sigfox.



Nota. La figura muestra el esquema de funcionamiento de la red Sigfox a nivel mundial. Tomado de *Qué es Sigfox*, 2021. Obtenido de:

<https://vicentferrer.com/sigfox/>

LoRa

LoRa es una red LPWAN con un modelo de negocio muy similar a Sigfox pero con una tecnología inalámbrica algo diferente ya que, entre otras cosas, utiliza un espectro de comunicaciones más amplio que SigFox. Además, esta tecnología

está mejor preparada para una comunicación bidireccional en tiempo real con dispositivos de Internet de las Cosas, por este motivo tiene grandes posibilidades de aplicación para Smart Cities. Por otro lado, las especificaciones para los fabricantes que quieran comunicar sus equipos a través de LoRa son más abiertas o menos estrictas que con Sigfox. LoRa responde a la necesidad de alcanzar comunicaciones a largas distancias, y mayor solidez frente a interferencias factores importantes al optar por una tecnología para un proyecto de IoT (efor, 2020).

Figura 7

Estructura de la red LoRa.



Nota. La figura muestra el esquema de funcionamiento de la red LoRa. Tomado de *LPWAN como base de comunicaciones para IoT, 2021*. Obtenido de:

<https://pandorafms.com/blog/es/que-es-lpwan/>

Zigbee

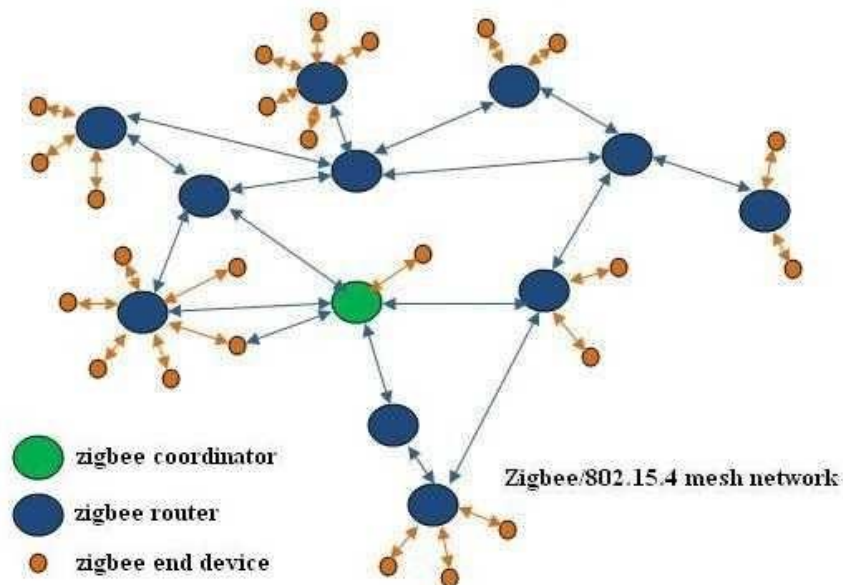
ZigBee es una tecnología inalámbrica, cuya red de comunicaciones es muy utilizada desde hace años en varios sectores como en aplicaciones domóticas e industriales. La actual versión ZigBee PRO cumple con mejoras de en las especificaciones de tasas de envío de datos bajas, pero con un alcance de cobertura próximo a los 100 metros (efor, 2020). Se basan en el estándar LR-WPAN (red de área personal inalámbrica de baja velocidad), es decir, IEEE 802.15.4. Los productos Zigbee tendrán capas de protocolo (PHY, MAC, red, seguridad, aplicación). Las capas de red, seguridad y aplicación están definidas por zigbee

alliance. Opera en varias bandas de frecuencia a saber 868MHz, 915MHz y 2.4 GHz. De estos productos zigbee basados en 2,4 GHz son muy populares en todo el mundo. Según la antena, la potencia de salida del transmisor y la frecuencia, cubrirá entre 10 y 100 metros. Zigbee admite topología de malla. Los productos Zigbee ofrecen aproximadamente 250 Kbps de velocidad de datos. Aproximadamente 65000 dispositivos son compatibles con una red zigbee (World, 2021).

Por otro lado, algunas de las ventajas que se destacan en una comunicación ZigBee para proyectos de IoT son: bajo consumo, seguridad superior al resto de tecnologías, alta escalabilidad, capacidad para soportar un gran número de nodos y robustez.

Figura 8

Estructura de red en malla de Zigbee.



Nota. La figura muestra el enrutamiento o encaminamiento con topología malla formado por tres tipos de dispositivos coordinadores, routers y dispositivos finales.

Tomado de *What is zigbee?*, 2021. Obtenido de: <https://www.rfwireless-world.com/Terminology/what-is-zigbee.html>

Comparación de las tecnologías de comunicación para el IoT

Tabla 1

Tecnologías de comunicación para el IoT.

Tecnología	Lora	SigFox	GSM	Zigbee
Frecuencia	EU: 868 MHz US: 902 – 928 MHz AM: 915 MHz	EU: 433/868 MHz US: 915 MHz	300 – 1200 MHz	2.4 GHz 868/915 MHz
Ancho de banda	250/125 kHz	100 Hz	200 kHz	
Velocidad de datos	50 Kbps	100 bps	20 – 100 Mbps	20 – 250 Kbps
Alcance	2 – 5 km (urbano) 15 – 20 km (rural)	3 – 10 km (urbano) 30 – 50 km (rural)	Depende de la operadora	10 - 100 m
Limitación de mensajes	Ilimitado	140 (UL), 4(DL)	160	N/A
Vida de batería	15 años	20 años	>10 años	100 – 1000+ días
Consumo	Bajo	Bajo	Muy alto	Medio
Disponibilidad Seguridad	Muy bajo Baja	Medio Media	Muy alto Alta	Muy bajo Alta
Topología	Estrella	Estrella	Topología celular	Punto a punto, estrella, malla, árbol, clúster

Nota. Esta tabla presenta una comparativa de las tecnologías de comunicación para IoT que se mencionaron anteriormente. Adapta de varias fuentes bibliográficas.

Tarjetas de Desarrollo

Una Tarjeta de Desarrollo es una placa o circuito que contiene un microcontrolador principal que ejecuta un conjunto de instrucciones de un programa suministrado. En torno a este procesador (unidad principal) se realiza un diseño

electrónico que permite: la programación del componente, suministra el voltaje adecuado para el correcto funcionamiento del controlador y proporciona acceso a las entradas y salidas del microcontrolador para la conexión de sensores y actuadores (Concepción, 2019). Existen diferentes tipos de placas de desarrollo, las más famosas en estos momentos son: Arduino, Raspberry Pi, ESP8266/ESP32 y otras.

Comparación de las tarjetas de desarrollo

Tabla 2

Tipos de Tarjetas de Desarrollo en el mercado.

Parámetros	Arduino Mega 2560	Arduino UNO	Raspberry PI 3 Model B+	NodeMCU ESP8266
Procesador	ATmega2560	ATmega328P	Broadcom BCM2837B0	ESP8266
Voltaje de operación	5V	5V	5V	3.3V
Voltaje de alimentación	7V – 12V	7V – 12V	5V	7V – 12V
Consumo	50mA a 200mA	45mA a 80mA	300mA a 1.34 A	20mA a 240mA
Puertos GPIO	54	14	40	36
Puertos serial	4	1	-	2
Entradas analógicas	16	6	-	15
Salidas analógicas	15	6	-	4
FLASH	256 kB	32 kB	-	4 MB
SRAM	8 kB	2 kB	1 GB	520 kB
Frecuencia de reloj	16 MHz	16 MHz	1.4 GHz	80/160 MHz
Ethernet/ Wi-Fi/ Bluetooth	No	No	Si	Si

Parámetros	Arduino Mega 2560	Arduino UNO	Raspberry PI 3 Model B+	NodeMCU ESP8266
Código abierto	Si	Si	Si	Si
Sistema operativo	No	No	Linux	XTOS
Dimensiones	102 x 53 mm	69 x 53 mm	83 x 56 mm	52 x 31 mm
Costo	\$22	\$30	\$90	\$11

Nota. Esta tabla presenta una comparativa de las características de tarjetas de desarrollo y sus precios en el mercado. Adaptada de la tabla Conclusion of the microcontroller comparison, de DiyIoT, 2020.

Sensores

Un sensor es un dispositivo capaz de detectar el cambio en magnitudes físicas o químicas específicas (temperatura, humedad, presión, etc.) y transformarlas a alguna salida en el otro sistema. Un sensor convierte un fenómeno físico en un valor eléctrico mediante un transductor para luego ser transmitido a una pantalla legible para el ser humano, para su lectura o procesamiento adicional (Smith, 2020).

En el mercado existen diferentes tipos de sensores para medir variables fisiológicas y el comportamiento de animales. A continuación, se presenta una breve descripción de ellos de acuerdo a las variables de interés del proyecto.

Sensores de temperatura

Un sensor de temperatura es un componente electrónico que permite medir las variaciones de temperatura del aire o la temperatura superficie de líquidos y sólidos, y las transforma en una señal eléctrica. También se denominan termosensores, o detectores de calor, sensores de calor o sondas térmicas, que en otras aplicaciones se usan para el control de circuitos (SENSORS, 2021).

Comparación de sensores de temperatura

Tabla 3

Tipos de sensores de temperatura.

Parámetros	MLX90614	DS1922L	DS18B20
Voltaje de operación	3V	3V	3V
Voltaje de alimentación	3.3V – 5V	3V – 5.25V	3.3V – 5.5V
Consumo	2mA	20mA	<5mA
Temperatura de operación	-40°C a 125 ° C	- 10°C a 65°C	-10°C a 85°C
Exterior	Si	Si	Si
Rango	-70°C a 380°C	-40°C a 85°C	-55 a 125°C
Precisión	±0.5°C	±0.5°C	±0.5°C
Dimensiones	16 x 11 mm	17 mm	10 x 1000 mm
Costo	\$15	\$93	\$5

Nota. Esta tabla muestra una comparativa de ciertos parámetros de evaluación entre diferentes sensores de temperatura y sus precios en el mercado. Información adaptada de diferentes sitios web.

Sensores de ritmo cardíaco

Un sensor de pulso o ritmo cardíaco es un dispositivo que detecta la actividad eléctrica del corazón (ECG). Este sensor registra una lectura de la frecuencia y el ritmo del corazón en tiempo real. Al ser de código abierto facilita obtener graficas del ritmo cardíaco mediante Processing si se está empleando una tarjeta Arduino (RAUTEC, 2021).

Comparación de sensores de ritmo cardíaco

Tabla 4

Tipos de sensores de ritmo cardíaco.

Parámetros	XD-58C	MAX30100	MAX30102	SEN-11574
Voltaje de alimentación	3.3V – 5V	1.8V – 3.3V	3.3V – 5V	3.3V – 5V
Consumo	4mA	600µA	600µA	4mA
Temperatura de operación	-40°C a 85 °C	-40°C a 85 °C	-40°C a 85 °C	-40 a 85 °C
Exterior	Si	Si	Si	Si
Dimensiones	16 x 16 mm	14 x 17 mm	0.5 x 8.5 mm	16 x 16 mm
Costo	\$6	\$9	\$11	\$25

Nota. Esta tabla muestra una comparativa de ciertos parámetros de evaluación entre diferentes sensores de ritmo cardíaco y sus precios en el mercado. Información adaptada de diferentes sitios web.

Dispositivos de posicionamiento

El Sistema de Posicionamiento Global (GPS) es un sistema de navegación global por satélite que proporciona información fiable de localización, velocidad y sincronización horaria, en cualquier lugar del mundo. El GPS se compone de tres elementos satélites en el espacio, control terrestre que son las estaciones en la tierra de seguimiento y control, y equipos de usuarios (receptores y transmisores GPS) para sincronizar los datos de los usuarios. Los satélites del GPS transmiten señales que reciben desde el espacio e identifican los receptores del GPS, los cuales posteriormente proporcionan individualmente las coordenadas de latitud, longitud y altitud, así como la hora local precisa (TIMING, 2021).

Comparación de los dispositivos de posicionamiento

Tabla 5

Tipos de los sensores de posicionamiento.

Parámetros	SIM 808	SIM908	Módulo L80 de QUECTEL	GPS NEO-6M
Bandas de Frecuencia	GSM 850, EGSM 900, DCS 1800, PCS 1900	Quad Band 850/900/1800/1900 MHz	1575.42 MHz	1575.42 MHz
Voltaje de Alimentación	3.4V – 4.4V	3.2V – 4.8V	2.8V – 4.3V	3.5V – 5V
Temperatura de operación	40°C a 85 °C	-30°C a 80 °C	-40°C a 85°C	-40°C a 85°C
Consumo	42mA	77mA	7µA-26mA	37mA
Tiempo de primera respuesta de GPS	30 s	30 s	35 s	30 s
Frecuencia de refresco	-	-	1Hz	5Hz
Sensibilidad de Rastreo de GPS	-165 dBm	-160 dBm	-165 dBm	-165 dBm
Velocidad de transmisión	9600 Baudios	9600 Baudios	9600 Baudios	9600 Baudios
Precisión	Media	Media	Alta	Alta
Dimensiones	50 x 50 mm	30 x 30 mm	16 x 16 mm	23 x 30 mm
Antena integrada	Si	Si	Si	Si
Costo	\$45	\$50	\$39	\$15

Nota. Esta tabla muestra una comparativa de ciertos parámetros de evaluación entre diferentes sensores de posicionamiento (GPS) y sus precios en el mercado.

Información adaptada de diferentes sitios web.

Fuentes de alimentación

Una fuente de alimentación es un componente de cualquier dispositivo electrónico, que se encarga de proporcionar la corriente o energía necesaria que el equipo requiera. A continuación, se describe las especificaciones de ciertas fuentes de alimentación.

Tabla 6

Tipos de fuentes de alimentación.

Parámetro	Baterías de litio	Pilas	Panel solar	Baterías LiPo
Capacidad	600 mAh	2.000 mAh y 2.500 mAh	220mA	1000 mAh y 2200 mAh
Alto voltaje	5V, 9V, 12V	1.2V, 1.5V, 9V	5V	3.7, 4.2V, 7.4V, 11.1V
Química	Ion de litio	Níquel Cadmio, Níquel Metal Hidruro.	Monocristal	Litio y polímero
Peso	60g	Varía de acuerdo al amperaje	12g	50g
Costo	\$10	Varía de acuerdo al voltaje	\$8	Varía de acuerdo al voltaje \$8 a \$22

Nota. Esta tabla muestra una comparativa de las especificaciones de evaluación de algunos tipos de fuentes de alimentación y sus precios en el mercado. Información adaptada de diferentes sitios web.

Equipos Lora

Gateway

Un gateway o puerta de enlace es un dispositivo que actúa como puente transparente entre los dispositivos finales y el servidor de red centralizado. En otras palabras, a través de este se establece la comunicación entre dos equipos ya sean

que estos están dentro de una misma red o diferentes (Arduino, 2018). A continuación, se describe las especificaciones de algunos de estos equipos.

Tabla 7

Gateways LoRa en el mercado.

Parámetros	LG01-P	LG01-S	LG02/OLG02	LG01-N
Fabricante	Dragino	Dragino	Dragino	Dragino
Voltaje de operación	9V – 15V	12V	12V	12V
Banda de frecuencia	915MHz	Disponible en frecuencias 433/868/915/920 MHz	915Mhz	915/920MHz
Consumo	0.5A	1A	1A	0.1A
Alcance	5 a 10 km en áreas abiertas, 300 mts en áreas densas	5 a 10 km en áreas abiertas, 300 mts en áreas densas	5 a 10 km en áreas abiertas, >500m en áreas densas	5 a 10 km en áreas abiertas, 300mts en áreas densas
Programación	Arduino	Arduino	Linux	Linux
Código abierto	Si	Si	Si	Si
Costo	\$100	\$100	\$121	\$130

Nota. Esta tabla muestra una comparativa de las especificaciones de evaluación de algunos Gateway para comunicación LoRa y sus precios en el mercado. Información adaptada de diferentes sitios web.

Módulo de comunicación

Un módulo de comunicación es un dispositivo que se encarga de facilitar el envío y recepción de mensajes entre el servidor y los clientes. A continuación, se describe las especificaciones de algunos de estos equipos.

Tabla 8

Módulos de comunicación LoRa en el mercado.

Parámetros	LoRa Shield	LoRa GPS Shield	RHF76-052AM	Ra-01 Sx1276
Voltaje de operación	3.3V – 5V	2.7V – 2.9V	3.7V	1.8V - 3.7V
Banda de frecuencias	915MHz	915MHz	434/470/868 /915MHz	868/915MHz
Consumo	25mA	10.3mA	10mA	1 µA
Sensibilidad	-148dBm	-148dBm	-140dBm	-139dBm
Sistema operativo	Arduino	Arduino	Arduino	Arduino
Costo	\$35	\$40	\$39	\$20

Nota. Esta tabla muestra una comparativa de las especificaciones de evaluación de algunos módulos de comunicación para una red LoRa y sus precios en el mercado.

Información adaptada de diferentes sitios web.

Tecnología Cloud Computing

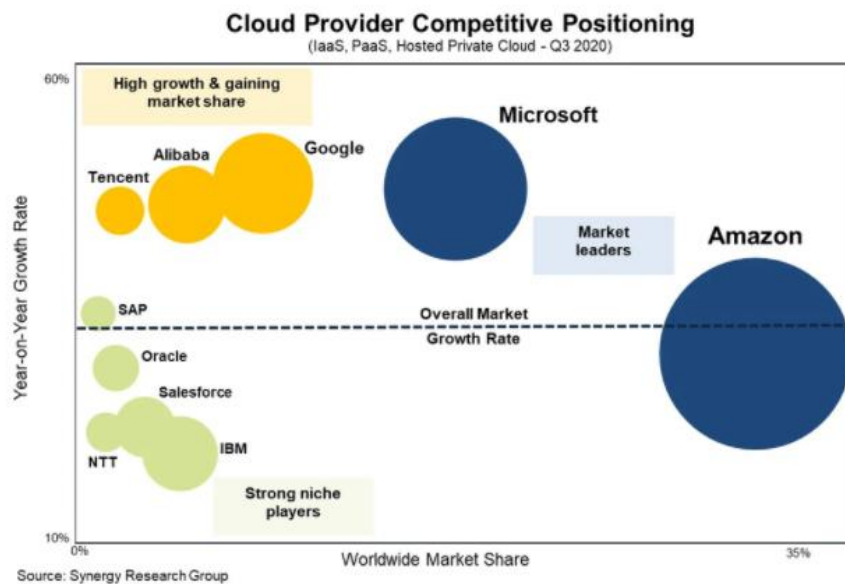
La computación en la nube es una tecnología que permite acceder remotamente a softwares, procesamiento de datos y almacenamiento de archivos por medio de Internet, es decir, abarca todo lo relacionado con los servicios alojados en internet. En la computación en la nube, no se precisa instalar aplicaciones localmente en computadoras. Su crecimiento ha sido exponencial, es por ello que se han desarrollado modelos de implementación o comúnmente conocidos como tipos de computación en la nube que son nube pública, nube privada, nube híbrida y nube comunitaria o multicloud. Así también se han desarrollado modelos de servicios IaaS (Infraestructura como servicio), PaaS (Plataforma como servicio) y SaaS (Software como servicio) (Flores, 2021).

Proveedores de Cloud Computing

La demanda de computación en la nube es cada vez grande, puesto que más empresas están migrando a la nube. Los proveedores de Cloud Computing son responsables del mantenimiento del servidor, incluyendo actualizaciones de seguridad, manejan los problemas de recuperación más rápidamente en comparación con las que no son en la nube públicas, entre otras cosas (salesforce, 2021). A continuación, se describen los principales proveedores de nubes públicas.

Figura 9

Posicionamiento competitivo del proveedor de la nube.



Nota. Esta figura muestra el crecimiento y el posicionamiento competitivo de los proveedores de nube arrojado en el informe de octubre del 2020 de Synergy Research Group. Tomado de *Cloud Computing: Tipos de nubes, servicios y proveedores*, 2021. Obtenido de: <https://openwebinars.net/blog/tipos-de-cloud-computing/>

AWS

Amazon Web Service (AWS) fue el primero en ofrecer a las empresas servicios de infraestructura en la nube, es uno de los pioneros en el mercado y con

mayor crecimiento. En la imagen anterior se puede constatar que sigue liderando. Amazon provee una plataforma fiable en la nube que utilizan miles de empresa en todo el mundo. Tiene a disposición una gran cantidad de servicios y herramientas, y lo más importante, su interfaz es fácil de usar (Flores, 2021). Su oferta de servicios SaaS es algo más limitada que la de otros proveedores de Cloud Computing, sus servicios de IaaS y PaaS son de verdadera calidad (Roca, 2021).

Google Cloud

Google también es un proveedor de nube pública, aunque también cuenta con entornos híbridos y multicloud, ofrece soluciones Cloud Computing a través de su plataforma, Google Cloud Platform (GCP). En los últimos años ha crecido exponencialmente y muchísimas grandes compañías lo utiliza. Se destacan por sus más de 100 productos en servicios PaaS y SaaS (Roca, 2021).

IBM Cloud

IBM es uno de los proveedores de computación en la nube más abierto y seguro para crear aplicaciones web o móviles. Dispone de diversos productos en el catálogo, incluidos todos los tipos de servicios en la nube (IaaS, PaaS, SaaS) tanto en nube pública como en nube híbrida, brindándole una ventaja frente a sus competidores (Roca, 2021).

Microsoft Azure

Microsoft ofrece servicio de Cloud Computing a través de su solución Microsoft Azure. Tardo en integrarse al mercado, pero ha crecido rápidamente y a gran escala. De igual modo que Amazon, es un proveedor de nube pública que ofrece diferentes servicios. Una de sus principales características, es que ofrece los tres tipos de servicios de nube (IaaS, PaaS, SaaS) (Flores, 2021).

Comparativa entre Proveedores de servicio Cloud Computing

Tabla 9

Proveedores de servicio de Cloud Computing.

Proveedor	AWS	Google Cloud	IBM Cloud	Microsoft Azure
Categoría informática	IaaS	PaaS	IaaS	PaaS
Escalabilidad automática	A través de Amazon Cloud Watch	BigTable y GFS	IBM SmartCloud Application Workload Services	Auto scaling application block y Windows Azure Fabric Controller
Soporta sistema operativo Windows	Si, usa Windows Server R2 2003, Windows Server 2008, Windows Server R2 2008, Windows Server 2012	No	Microsoft Windows Server 2003, Microsoft Windows Server 2008	Windows Server 2012 Datacenter, Windows Server 2008 R2 SP1
Soporta sistema operativa Linux	Sí, usa Linux Enterprise Server, Red Hat Enterprise Linux	Sí, pero usan la técnica sandboxing para las aplicaciones y Google provee acceso limitado al sistema operativo, que no es alterado por ningún motivo	Sí, Red Hat Enterprise Linux, SUSE Linux Enterprise Service	Opens USE 12.3, SUSE Linux Enterprise Server 11 Services Pack 2, Ubuntu Server 12.04 LTS, Ubuntu Server 12.10 LTS, Ubuntu Server 13.04 LTS, Open Logic CentOS 6.3, Ubuntu Server 12.10 DAILY
Soporte para lenguajes	C++, C#, Java, Perl, Python, Ruby	Python, Java, Go (experimental)	Java, PHP	Net, Java, Node.js, Python

Proveedor	AWS	Google Cloud	IBM Cloud	Microsoft Azure
Soporte para almacenamiento de datos	Amazon SSS, Amazon Relational DB Service, Amazon SimpleDB, SQL Server Express, SQL Web, SQL Server Standard	Base de datos no relacionales "Big Table", No soporta base de datos relacionales	DB2, Oracle, MS SQL, MySQL, Informix, Sybase	SQL Relacional Almacenes de tablas NoSQL Blob no estructurado
Servidor web	Apache, IIS, otros	Jetty Web Server	WebSphere Application Server V7.0 and V8.0	IIS V7.5
Aplicación basada en web	Si	No	Si	No
Interfaz de control de líneas de comandos	Si	No	No	Si
Soporte técnico gratuito	No	Si	Si	Si
Foros de ayuda	Si	Si	Si	Si
Herramienta de diagnóstico para soporte	Si	No	Si	No

Nota. Esta tabla muestra un análisis comparativo entre los proveedores de servicio de computación en la nube. Adaptado de la tabla de Análisis comparativo entre plataformas Cloud Computing, de Grupo FRACTALIA, 2014.

Capítulo IV

Metodología

En el presente capítulo, se abarca todos los temas que conforman el proceso metodológico del proyecto con la finalidad de evaluar la situación de partida y facilitar el desarrollo de la programación. Primero, se analizarán el método de investigación utilizado. Después, se indicarán los requisitos funcionales de acuerdo al enfoque del dispositivo. A continuación, se presenta la descripción cualitativa y cuantitativa del modelo. Seguido, se describe el universo al que va dirigido y el procesamiento de la información recolectada. Finalmente, se describe la selección de la tecnología inalámbrica, sensores y elementos a utilizarse en el proyecto.

Método de investigación

Para realizar el presente proyecto, se plantea utilizar una metodología mixta (cualitativa y cuantitativa), haciendo uso de las fortalezas de ambos tipos de información, combinándolas y tratando de disminuir sus debilidades para lograr un entendimiento del trabajo de estudio.

Método cualitativo

Por medio del método cualitativo se realiza la recopilación de toda la información necesaria acerca de las enfermedades que afectan al ganado bovino, indicadores del estado de salud, el análisis del sector ganadero en Ecuador. Además, información acerca de tecnologías para el sistema de monitoreo de ganado bovino, las redes de comunicación inalámbrica.

Dentro del desarrollo del trabajo se utilizarán diversos sensores para determinar los valores de las variables fisiológicas de estos animales como sensor de temperatura corporal, sensor de ritmo cardiaco, sensor de desplazamiento y posición, se recopilará la información de estos equipos para realizar su correcta selección.

Método cuantitativo

Este método consiste en la experimentación, por lo tanto, se procede a realizar el diseño del circuito del prototipo de monitoreo y la programación de los sensores. Posteriormente se construirá el prototipo, se realizará la calibración de sensores, el diseño del sistema de comunicación inalámbrica y la inclusión de los sensores al sistema de comunicaciones. Por último, se realiza la recepción y visualización de datos por medio de una aplicación móvil.

Finalmente se ejecutarán pruebas de campo del sistema prototipo de rastreo y monitoreo en ganado bovino en la ciudad de San Gabriel provincia del Carchi y así verificar su correcto funcionamiento.

Requisitos funcionales

Debido al enfoque del proyecto se ha dividido en dos grupos de información:

Se necesita conocer información de las variables médicas del ganado lechero, como son los rangos de temperatura corporal, ritmo o pulso cardíaco, velocidad de movimiento, entre los principales. En los cuales el bovino se encuentra en su estado de salud normal.

Se necesita conocer información de la infraestructura de telecomunicaciones que se diseñará para la red de comunicación, como son los rangos de medición de los sensores y su consumo de energía, si estos equipos son aptos para trabajar en el exterior y la fiabilidad de sus datos. El rango de alcance de los protocolos que se pueden ejecutar en IoT (comparando principalmente las velocidades de transmisión, alcance en metros, tipo de cifrado, etc).

En cuanto a ello el principal problema es que los sensores seleccionados para medir las diferentes variables sean de tipo no invasivo para que no afecte el estado del bovino, así también como el tamaño y peso del dispositivo prototipo final,

ya que al estar diseñado para colocarlo en el animal debe ser de un tamaño y peso adecuado. Puesto que, si el estrés se ve involucrado, las variables cambian.

Por otro lado, el problema en cuanto a la infraestructura de telecomunicaciones es el área de cobertura, porque generalmente el lugar donde suelen estar los animales carece de buena señal lo cual dificulta la recepción de la información y por ende la base datos se vuelve poco fiable. También la veracidad de los datos censados ya que estos se verán afectados por las condiciones climáticas.

Descripción cualitativa del modelo

La información de las variables seleccionadas a ser censadas en el bovino, deben ser confiables y ser las necesarias para realizar el monitoreo del estado de salud del animal. Posteriormente poder realizar un análisis de los datos recolectados para conocer si alguna variable está fuera de los rangos normales ya que esto puede representar una afectación en la salud y en caso de la ubicación geográfica un posible intento de hurto.

Descripción cuantitativa del modelo

Consiste en la evaluación de la precisión de los datos censados para que se puedan tomar como aceptables, de la red de comunicación. Por otra parte, haciendo uso de esta información y del modelo estadístico realizar graficas comparativas para ver la tendencia que toman los datos y emitir comentarios.

Universo y muestras

Universo

Para la investigación que se va a realizar el objeto de estudio corresponde a los animales de granja en este caso vacas lecheras de la región de la Sierra en Ecuador.

Muestra

Para poder recolectar la información necesaria para cumplir con el objetivo de esta investigación, la muestra a censar serán vacas productoras de leche en la ciudad de San Gabriel, cantón Montúfar correspondiente a la provincia del Carchi, por tanto, los datos provienen de variables como la temperatura, la frecuencia cardíaca, desplazamiento, posición y la velocidad de movimiento del bovino.

Es importante mencionar que los datos recolectados de todas estas variables a medir, se analizarán para determinar los cambios que se tuvo en las diferentes variables y con esto verificar si alguna variación dependió del lugar o de otros factores.

Procesamiento de la información

Para el diseño del prototipo se realizarán pruebas de campo con la finalidad de obtener datos que provienen de los sensores con los que cuenta el dispositivo diseñado. Mismo que se ha encargado de medir las variables ya definidas para el ganado bovino productor de leche.

Es necesario mencionar que no se tiene acceso a una base de datos para realizar una comparación con los datos que se van a recopilar con este dispositivo prototipo por lo cual será necesario que la información recopilada sea almacenada para poder realizar un análisis que permita cumplir con el objetivo.

Tecnología a emplearse en el sistema prototipo IoT

El mundo de la IoT presenta diferentes alternativas para conectar y establecer la comunicación entre dispositivos eléctricos al internet. Los principales criterios en elección de tecnología para este proyecto son bajo consumo de energía, bajo costo, dimensiones reducidas de los dispositivos, cobertura semejante a la de las redes metropolitanas, es decir, largo alcance y conexiones bidireccionales

seguras. Lo que nos permite conseguir el objetivo de recibir la información captada por los sensores y enviarlo al servidor de red sin dificultad.

Después de investigar en varias fuentes bibliográficas acerca de las redes de comunicación para el IoT y teniendo en cuenta las características mencionadas en la Tabla 1, se seleccionó la tecnología LoRa ya que es la que mejor se adapta a nuestro escenario de área extendida LPWAN (Low Power Wide Area Network), cubriendo perfectamente la extensión geográfica total de estudio, además cabe mencionar que el lugar de estudio dispone de servicio de internet aventajando el despliegue de la tecnología LoRa. Por otra parte, esta tecnología inalámbrica alarga la vida útil de la batería del dispositivo al ser de bajo consumo, también permite una gran cantidad de dispositivos conectados, trabaja en bandas no licenciadas, es más adecuada para aplicaciones IoT a gran escala que envían solo ráfagas de datos pequeñas y poco frecuentes a un menor costo de hardware. Por último, LoRa es probablemente la mejor opción si necesita una verdadera bidireccionalidad debido al enlace simétrico y adecuada para redes de IoT en las que se requieran sensores que no dispongan de acceso a corriente eléctrica de red, como es el caso.

Arquitectura de la red de LoRa

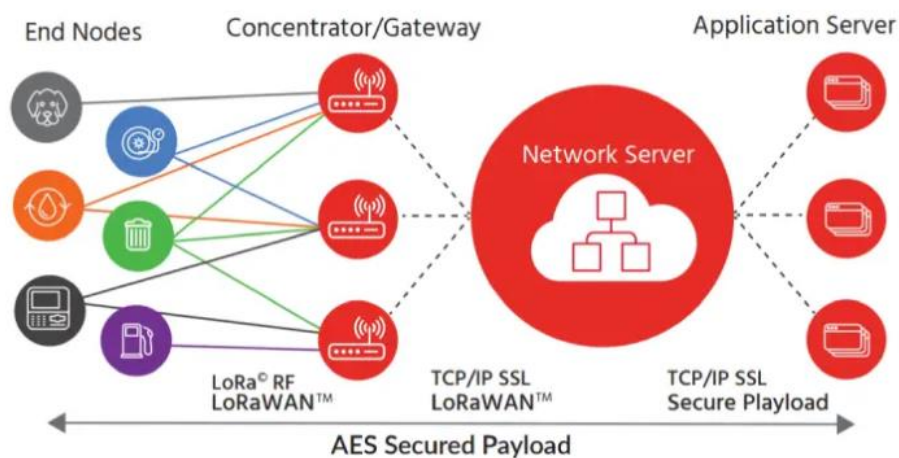
La arquitectura de una red Lora está basada en una topología en estrella y está compuesta por cuatro elementos:

- **Puntos finales:** Son los elementos que recopilan datos de sensores, lo transmiten upstream y downstream, y establecen la comunicación con el servidor por medio del Gateway.
- **LoRa Gateway:** Actúa como un puente transparente para comunicar los datos que se transmiten entre los nodos finales y los servidores upstream.

- **Servidor de red LoRa:** Su función es conectar varias puertas de enlace para recibir los datos enviados de los puntos finales LoRa a través de una conexión TCP/IP segura.
- **Servidor de aplicación:** Analiza los datos recopilados del nodo final o cliente y determina las acciones que ejecutara dicho nodo.

Figura 10

Arquitectura de LoRa.



Nota. Una red LoRa está constituido por cuatro bloques y tiene dos capas de seguridad. Tomado de *LoRa Shield*. Obtenido de:

[https://www.mokolora.com/es/what-is-](https://www.mokolora.com/es/what-is-lora/#:~:text=La%20arquitectura%20de%20red%20de%20Lora%20es%20una,lado%20y%20servidor%20de%20red%20en%20el%20otro.)

[lora/#:~:text=La%20arquitectura%20de%20red%20de%20Lora%20es%20una,lado%20y%20servidor%20de%20red%20en%20el%20otro.](https://www.mokolora.com/es/what-is-lora/#:~:text=La%20arquitectura%20de%20red%20de%20Lora%20es%20una,lado%20y%20servidor%20de%20red%20en%20el%20otro.)

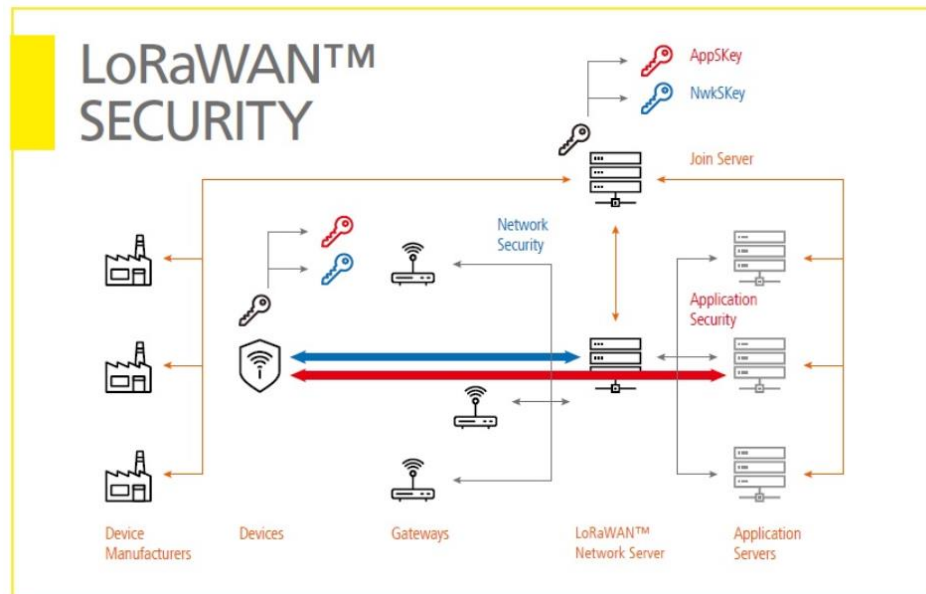
Seguridad LoRaWAN

El desacoplo del backend IP en servidores de red y aplicación motiva a LoRaWAN a emplear dos capas independientes de seguridad con cifrado AES de 128 bits, una para proteger la integridad y encriptación de los metadatos de la capa MAC (clave de sesión de red, o NwkSKey), asegurando la autenticidad del nodo y otra para proteger la encriptación de punto a punto de los payloads de la aplicación

(clave de sesión de aplicación, o AppSKey), asegurando de que el operador tiene permiso para manejar la red.

Figura 11

Seguridad de LoRaWAN.



Nota. El flujo de datos desde un nodo final LoRa a la aplicación tiene cifrado y descifrado al principio y al final de la cadena de datos, para que sólo el nodo final del sensor y la aplicación tengan acceso a los datos de texto sin formato. Obtenido de: <https://humanizationoftechnology.com/seguridad-en-redes-lorawan/revista/iot/01/2019/>

Existen dos métodos para implementar las claves de seguridad:

- **Activación por personalización (ABP):** Aquí los dispositivos finales LoRaWAN suelen ser programados directamente en fábrica con las claves de autenticación de la red LoRaWAN específica.
- **Activación por aire (OTAA):** La activación inalámbrica utiliza un método de autenticación por ID para la aplicación, dispositivo final, y red para así obtener la dirección y claves NwkSKey/AppSKey. Es recomendado utilizar este método porque las claves no están

predeterminadas y pueden ser regeneradas en cada nueva conexión (Pickering, 2017).

Dispositivos seleccionados para la construcción del prototipo IoT

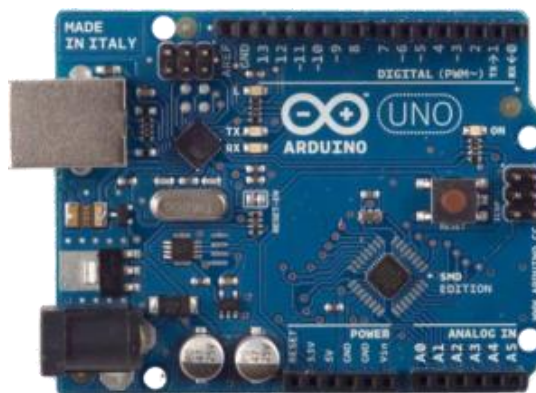
En este apartado se describe las principales características de los dispositivos seleccionados que se han ajustado a los requerimientos del sistema prototipo collar de rastreo y monitoreo para vacas lecheras.

Arduino Uno

La elección de la tarjeta de desarrollo se hizo en base a cuatro criterios: la disponibilidad del dispositivo, la capacidad de almacenamiento, el número de entradas/salidas y su entorno de programación. Teniendo en cuenta las características mencionadas en la Tabla 2, se ha optado por el Arduino Uno que consta de 14 pines de entradas/salidas digitales, 6 de las cuales pueden ser utilizadas como salidas PWM, 6 entradas análogas, 1 puerto serie por hardware (UART), un oscilador de 16 MHz, un conector de alimentación, un conector ICSP, una conexión USB, y un pulsador para el reset. El voltaje de operación es de 5V y es compatible con diferentes shields. Sus dimensiones son: 68.6 x 53.4 mm.

Figura 12

Arduino Uno.



Nota. La gráfica presenta la placa Arduino Uno.

Sensor de temperatura

Para la elección del sensor de temperatura adecuado se ha tenido en cuenta para que proceso se va usar, en este caso es la medición en vacas lecheras. En torno a esto los criterios que se consideraron son la precisión, el rango de medida y el consumo de corriente. Teniendo en cuenta las características mencionadas en la Tabla 3, se ha escogido el sensor infrarrojo de temperatura MLX90614 que permite medir la temperatura de un objeto sin contacto. Es un chip de silicio con una fina membrana micro-mecanizada, diseñada para ser sensible a la radiación infrarroja emitida por un objeto a distancia. El voltaje de operación es de 3V, su precisión es de $\pm 0.5^{\circ}\text{C}$, su rango de temperatura es de -70°C hasta 380°C , su consumo energético es de 25mA y sus dimensiones son 16 x 11 x 6 mm.

Figura 13

Sensor MLX90614.



Nota. La gráfica presenta el sensor infrarrojo de temperatura MLX90614.

Sensor de ritmo cardíaco

Teniendo en cuenta las características mencionadas en la Tabla 4. Para monitorear el ritmo cardíaco en las vacas, se ha optado por el sensor XD-58C, que fue seleccionado por su rango de medida que va desde -40°C hasta 85°C , su consumo de energía es bajo de 4mA. El voltaje de operación es de 5V y sus dimensiones son: 16 x 16 x 3 mm. Es un sensor de tipo plug and play diseñado para Arduino; combina un sensor óptico de frecuencia cardíaca con circuitos de

cancelación de ruido y amplificación, haciendo que sea sencillo y rápido para obtener lecturas de pulso o ritmo cardíaco confiables en tiempo real.

Figura 14

Sensor XD-58C.



Nota. La gráfica presenta el sensor de ritmo cardíaco XD-58C.

Sensor GPS

Teniendo en cuenta las características mencionadas en la Tabla 5. Se opta por el sensor GPS Neo-6M por las siguientes pautas: su bajo consumo de corriente de 10mA, su menor tiempo de primera respuesta o tiempo de inicio de 30s y su mayor frecuencia de refresco de 5Hz.

Figura 15

Sensor GPS.



Nota. La gráfica presenta el sensor GPS Neo-6M.

Módulo LoRa Shield

Teniendo en cuenta las características mencionadas en la Tabla 8, en lo referente al módulo de comunicación LoRa se ha optado por un dispositivo de la gama Dragino. El transceptor está basado en el chip Semtech SX1276/SX1278, permite enviar datos a baja tasa de transferencia y proporciona una comunicación de espectro extendido de largo alcance con alta inmunidad a las interferencias y al tiempo que minimiza el consumo de energía. Puede lograr alta sensibilidad por sobre -148dBm. Su consumo de corriente es 10.3mA y sus dimensiones son: 60 x 53 x 25 mm.

Figura 16

Módulo LoRa Shield.



Nota. Este escudo transceptor de Dragino permite desarrollar diferentes aplicaciones de largo alcance LoRa.

Gateway LoRa LG01-N

Dragino oferta una amplia gama de equipos Gateway, en este caso los criterios de evaluación son: la frecuencia de trabajo, el radio de cobertura y disponibilidad del dispositivo. Teniendo en cuenta las características mencionadas en la Tabla 7, se opta por el Gateway LG01-N de código abierto de un solo canal permite conectar de manera fácil la red inalámbrica de LoRa a través de WiFi,

Ethernet o celular 3G/4G a una red IP, trabaja en una sola frecuencia de 915 MHz, tiene un alcance de 5 a 10 km en áreas abiertas y 300mts en áreas densas. La característica más importante al evaluar la antena de este equipo es la ganancia en la potencia en la señal, ya que, si la potencia recibida es mayor, entonces la capacidad de la antena en el medio será mayor y esta posee una alta sensibilidad de hasta -148dBm.

Figura 17

Gateway LoRa LG01-N.



Nota. Este Gateway del kit Dragino permite enviar datos a largas distancias de transmisión con bajos costos energéticos.

Batería LiPo

La batería de polímero de iones de litio son pilas recargables, están compuestas en ocasiones de múltiples celdas en su interior, cada una de las cuales tiene un voltaje nominal de 3.7V y un voltaje máximo de 4.2V dependiendo de los materiales con los que se ha hecho la batería. Se caracterizan por ser ligeras y por poder almacenar una gran cantidad de energía. En este caso la batería es de 2 celdas posee un voltaje de 7.4V y una tasa de descarga de 2.6A.

Figura 18

Batería.



Nota. La gráfica presenta la batería LiPo marca TCB.

Servidor en la nube seleccionado para el prototipo IoT

Servidor en Amazon

Basados en el análisis comparativo de la Tabla 9 y la investigación bibliográfica realizada se elige Amazon con sus servicios de Amazon Web Services (AWS) ofrece productos globales basados en la nube, incluidas almacenamiento, base de datos, ambientes virtuales, entre otras. AWS proporciona 15 Gb de almacenamiento mensual gratuito durante un año. Además, poseen seguridad en cada capa correspondiente a aplicaciones y dispositivos, tienen un control anticipado como es el cifrado y el control de acceso. Otro de los factores importantes y que dan ventaja a esta plataforma es su escalabilidad, es decir que es adaptable a cantidades numerosas de dispositivos. Además, es necesario mencionar que Amazon tiene un costo económico y entre más usuarios acceden a esta plataforma el costo sigue reduciendo.

Amazon Elastic Compute Cloud (Amazon EC2) permite que las instancias se configuren de acuerdo a las necesidades que requiera dependiendo de la aplicación que se vaya a realizar, por ejemplo, la capacidad de CPU, memoria, almacenamiento y redes (AWS, s.f.).

Como se ha mencionado, Amazon posee una infinidad de tipos de instancias las mismas que se ajustan al tipo de aplicación que se va a realizar, en este caso se

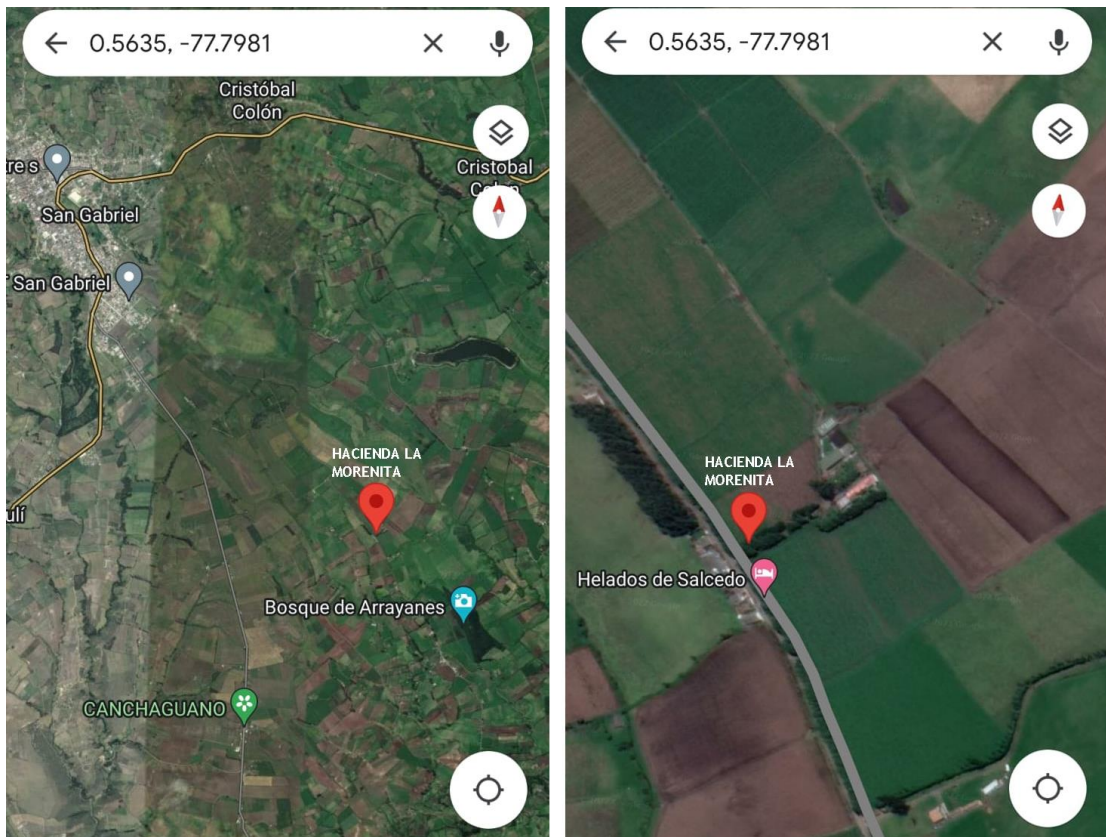
hizo uso de la instancia T2, ya que este tipo de instancia en comparación a las otras, es de uso general de bajo costo y que está incluida en la capa gratuita, además de poseer una estabilidad en recursos informáticos, memoria y red (AWS, s.f.).

Descripción del lugar geográfico

El lugar en el que se realizaron las pruebas y tomas de datos se encuentra en la provincia del Carchi, este lugar queda a 15 minutos de la ciudad de San Gabriel, cuyo nombre que recibe el terreno es “Hacienda La Morenita”, está ubicado en la comunidad de Monteverde, perteneciente al cantón Montúfar, el dueño de este extenso terreno recibe el nombre de Cristhian Moreno, quien muy amablemente concedió el permiso para poder realizar las pruebas, además se puede mencionar que esta hacienda cuenta con 67 vacas lecheras, terneros y caballos, una de las vacas se la utilizó para medir las variables que ya se ha mencionado con anterioridad. El lugar cuenta con una pequeña casa en la que residen los cuidadores de la “Morenita”, un ordeño en el cual se encuentra el dispositivo de internet mismo que sirvió para conectar el Gateway para que se efectuó la comunicación con el nodo final y con la ayuda de Google Maps se puede apreciar en la figura siguiente la ubicación del terreno y en la figura 20 se muestra el ordeño en donde se encuentra el dispositivo de Internet.

Figura 19

Ubicación de la Hacienda La Morenita.



Nota. En el gráfico se puede apreciar la ubicación de la Hacienda La Morenita.

Figura 20

Ordeño.



Nota. Esta imagen evidencia el ordeño en el cual se tiene acceso a Internet para poder realizar la comunicación del Gateway con el nodo final.

Extensión de la Hacienda

Su perfil topográfico, es regular rodeado de un bosque que indica la finalización del terreno. La Hacienda La Morenita cuenta con una extensión de 200 hectáreas de las cuales 2000m son usados para el pastoreo de los animales, en las siguientes imágenes se puede apreciar el terreno el cual es despejado que facilita tener una buena línea de vista al nodo Gateway.

Figura 21

Diferentes Áreas de la Hacienda.



Figura 22

Diferentes Áreas de la Hacienda.



Figura 23

Diferentes Áreas de la Hacienda.



Haciendo uso del software de Google Maps se grafica el área de la hacienda donde los animales pasaran pastando durante el período de pruebas.

Figura 24

Extensión donde se realizarán las pruebas.



Descripción de la Vaca en la cual se coloca el dispositivo final

La vaca que se usó para colocar el nodo final tiene el nombre de NUA, la edad es de 3 años, es un animal cuya raza hace referencia a Holstein Negro, su peso es de aproximadamente 450kg, hace 8 meses tuvo un parto y por causa del

mismo su peso redujo y su altura es de 1.37m. Esta vaca es la que mejor comportamiento tuvo al momento de colocarle el prototipo ya que se intentó en otras vacas, pero no se dejaron colocar el collar. En la figura siguiente se aprecia al animal que se usó para la realización de las pruebas.

Figura 25

Vaca en la cual se colocó el dispositivo.



Nota. Esta imagen evidencia a la vaca de nombre NUA que se usó para colocar el dispositivo final y poder realizar la toma de datos.

Capítulo V

Diseño e Implementación

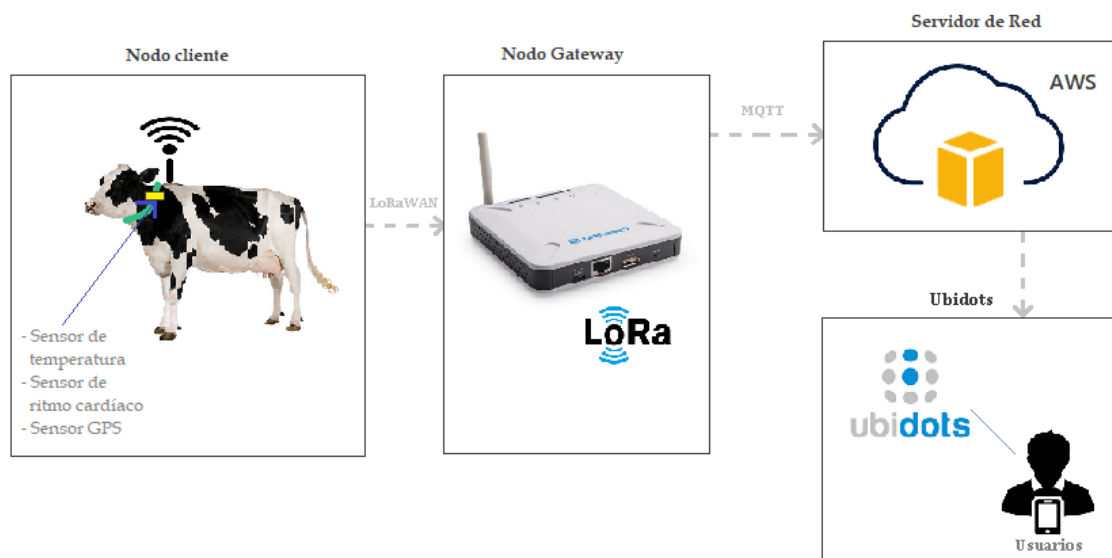
En el presente capítulo se describe el esquema de solución de la arquitectura para el proyecto IoT. Además, se explica el diseño del nodo cliente y del nodo Gateway a través de sus diagramas de flujo. Después, se muestra el diagrama esquemático y el esquema de conexión física de los dispositivos implementados. A continuación, se detalla el desarrollo del hardware y software del proyecto. En el cual se muestra el aspecto final del prototipo de collar (PCB), sus configuraciones (nodo cliente, nodo Gateway, servidor AWS) y el desarrollo de la interfaz de visualización en Ubidots. Finalmente, se detalla el costo del proyecto.

Arquitectura del sistema

En la Figura 26, se presenta el diagrama de bloques de la arquitectura del prototipo de collar para el rastreo y monitoreo de ganado lechero basado en tecnología LoRa para IoT.

Figura 26

Diagrama de bloques de la arquitectura del sistema.



Nota. La grafica presenta el diagrama de bloques del sistema de rastreo y monitoreo para vacas lecheras.

En la solución propuesta las variables a medir son: posición, velocidad de movimiento, desplazamiento, temperatura corporal y ritmo cardíaco. Como se puede apreciar esta arquitectura, está compuesta por cuatro bloques que se describen a continuación:

- **Nodo cliente:** Este bloque se encuentra conformado por tres sensores, el de posición, de temperatura y de ritmo cardíaco, los mismos que miden las variables físicas en las vacas lecheras, posteriormente son leídos por la placa de desarrollo basado en el microcontrolador Atmega328P, por último, la unidad de control LoRa Shield envía estos datos a través de su antena de RF a 915 MHz incorporada por el canal de radio Lora al nodo Gateway.
- **Nodo Gateway:** Este bloque servirá de enrutador de red para recibir los paquetes de datos enviados por el nodo cliente (haciendo uso del protocolo de comunicación LoRaWAN) y a su vez reenviarlos al servidor AWS en internet por medio del protocolo de comunicación MQTT.
- **Servidor de Red:** Este bloque se encarga del alojamiento en la nube. Tiene la función de recibir los datos decodificados por el nodo Gateway del nodo cliente, posteriormente guarda la información adquirida en la base de datos MySQL creada y reenvía la información mediante la interacción con APIs a la interfaz de usuario haciendo uso del protocolo HTTP para su posterior análisis y toma de decisiones. En este bloque también se realiza la programación para el envío de alarmas a Ubidots.
- **Ubidots:** Este bloque se enfoca a facilitar la visualización y procesamiento de los datos de las variables fisiológicas del animal en tiempo real. Proporcionando un entorno más agradable al usuario

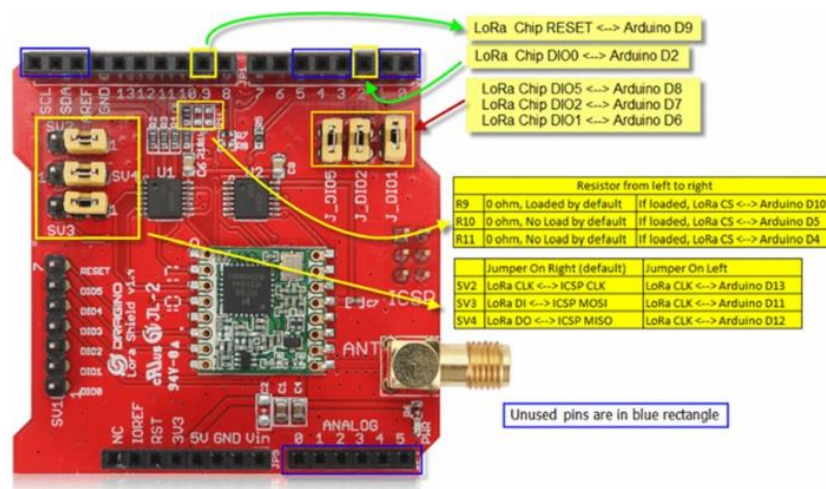
accesible desde un dispositivo móvil u ordenador. De esta manera el usuario se mantiene informado y puede tomar decisiones inmediatas evitando pérdidas económicas.

Diseño del nodo cliente

Para la implementación de este nodo se requiere: Sensores + Lora/GPS Shield + Arduino Uno. Para no tener inconvenientes en la comunicación, es importante conocer el correcto uso de las placas de Dragino Shield. En la Figura 27, se exponen los pines que se utilizan para la comunicación LoRa Shield con Arduino y en la Tabla 10 se detalla más a profundidad.

Figura 27

Módulo Lora Shield con pines para conexión con Arduino.



Nota. Este módulo permite monitorear sin interferencias e inmunidad al ruido.

Tomado de *LoRa Shield*. Obtenido de:

<https://www.aprendiendoarduino.com/tag/lora-shield/>

Tabla 10

Distribución de los pines LoRa Shield con Arduino.

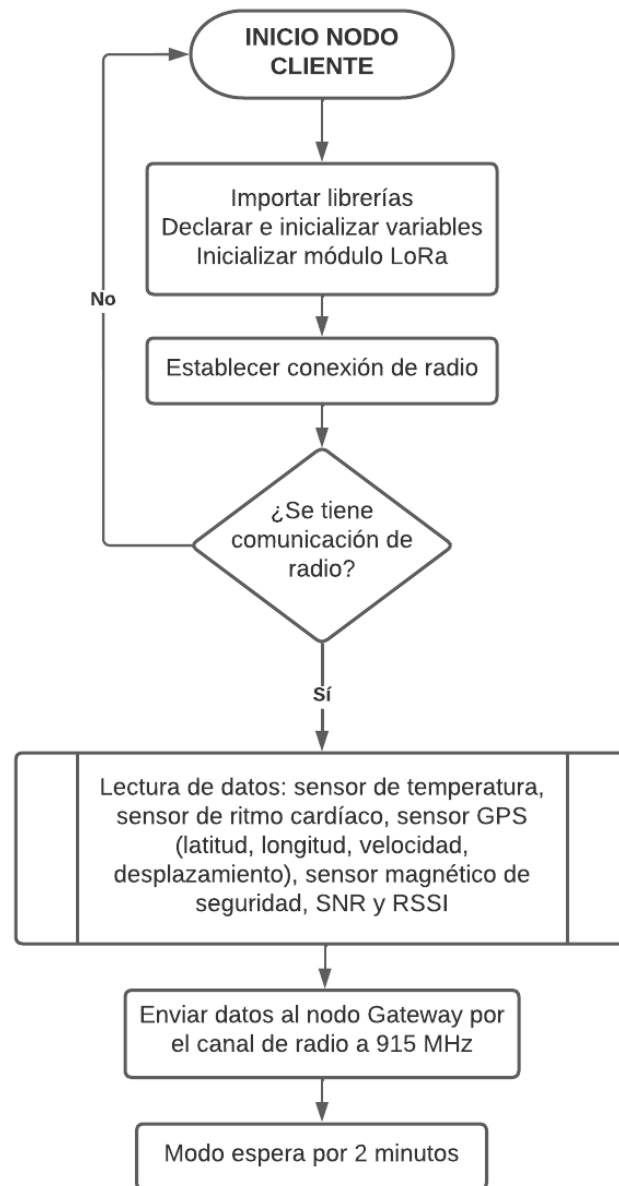
Pines de LoRa Shield+Arduino	Disponibilidad
SLC, SDA, REF GND	Pines utilizables Pin utilizable
D10, D11, D12, D13 D9	Pines digitales utilizables con condición Pin digital no utilizable
D8, D7, D6 D5, D4, D3	Pines digitales utilizables con condición Pines digitales utilizables
D2	Pin digital no utilizable
D1	Pin digital utilizable
D0	Pin utilizable
A0, A1, A2, A3, A4, A5	Pines analógicos utilizables

Nota. Esta tabla muestra la disponibilidad de pines para la conexión. Tomado de *LG01N/OLG01N LoRa Gateway User Manual.*

En la Figura 28, se muestra el diagrama de flujo que describe el software del nodo cliente. La programación de las funciones se realizó en el IDE de Arduino, se inicia importando la librería para la comunicación entre LoRa Shield y la placa de Arduino SPI.h seguido de la librería para la comunicación de radio LoRa.h y las librerías de los sensores para la temperatura Wire.h – Adafruit_MLX90614.h, para la geo localización SoftwareSerial.h – TinyGPS.h y para el ritmo cardíaco PulseSensorPlayground.h. Luego se comprueba que se estableció la conexión de radio, si la comunicación LoRa es exitosa se procede a leer los datos de temperatura, ritmo cardíaco, latitud, longitud, velocidad, desplazamiento, estado del collar, SNR y RSSI. Finalmente se envían los datos concatenados al nodo Gateway cada 2 minutos por el canal de radio LoRa, el cual está configurado en la frecuencia de 915MHz.

Figura 28

Diseño del diagrama de flujo del nodo final.



Nota. Realizado en Lucidchart.

Diseño del nodo Gateway

Para la implementación de este nodo que cumple las funciones de receptor y transmisor solo se utilizará el Gateway de Dragino LoRa LG01-N. Sus interfaces

proporcionan métodos para que los usuarios enlacen la red de sensores a Internet con facilidad. A continuación, se detalla las especificaciones técnicas del LG01-N.

Linux Side:

- Procesador: 400MHz, 24K MIPS
- Flash: 16MB; RAM: 64MB

LoRa Side:

- Controlador de LoRa: AR9331 24K MIPS / Linux
- Chip LoRa: SX2176

Interfaces:

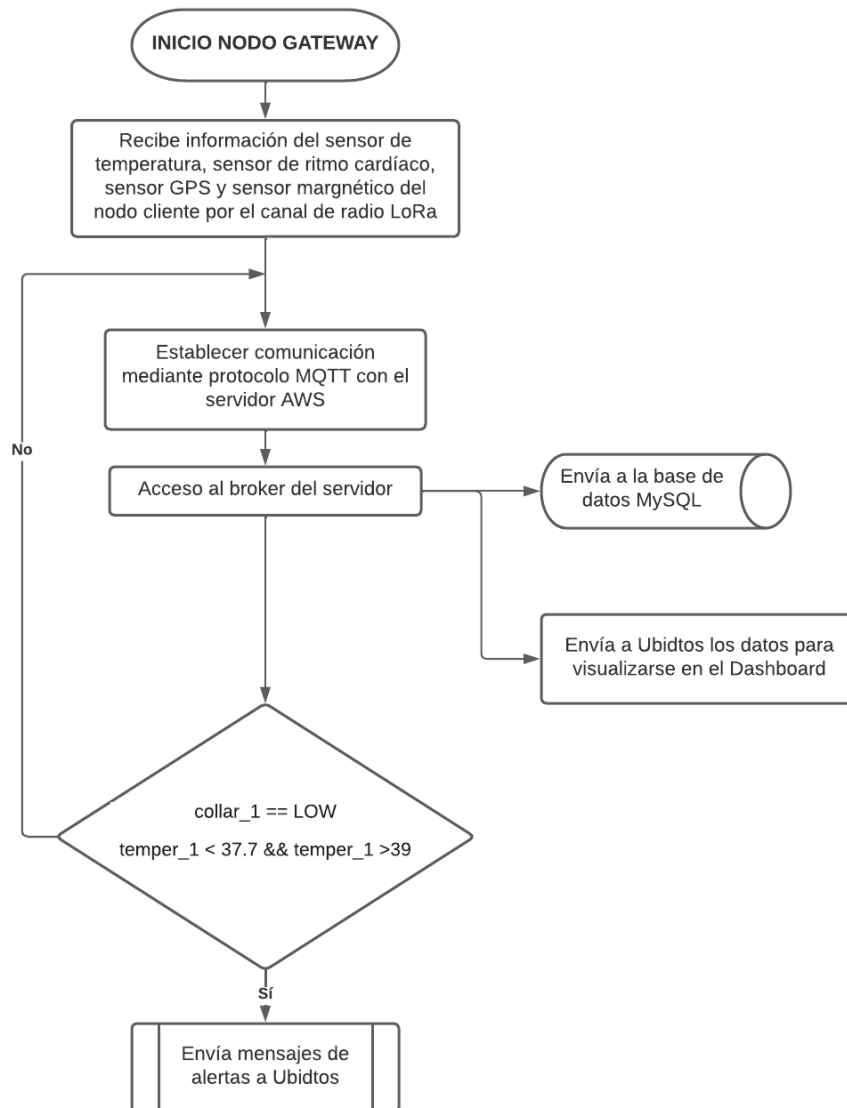
- Puertos RJ45: 10M/100M x 2
- Estándar WiFi: 802.11 b/g/n
- Entrada de energía: 12V DC
- Conector de host USB 2.0 x 1 (MClelectronics, 2021).

La Figura 29, muestra el diagrama de flujo que describe el software del nodo Gateway y la forma de comunicarse este bloque con el servidor en la nube. Las configuraciones de los parámetros de comunicación de red del LG01-N se realizan en la interfaz de Dragino. Primero se recibe las lecturas del sensor de temperatura MLX90614, sensor de ritmo cardíaco, información del sensor GPSNeo-6M y sensor magnético. Se establece la comunicación con el servidor en Amazon vía MQTT, con acceso ya al bróker del servidor se procede a enviar los datos a la base de datos donde se almacenan en formato .php para su posterior visualización en Ubidots. A continuación, se verifica si las variables de temperatura, ritmo cardíaco y latitud/longitud están fuera de los límites permitidos, así también si, la seguridad del collar está en LOW significa que ha sido retirado del animal. Dado el caso que así

sea el bróker del servidor enviara los mensajes de alerta a la interfaz visual en Ubidots.

Figura 29

Diseño del diagrama de flujo del nodo Gateway.



Nota. Realizado en Lucidchart.

Diseño del circuito Electrónico

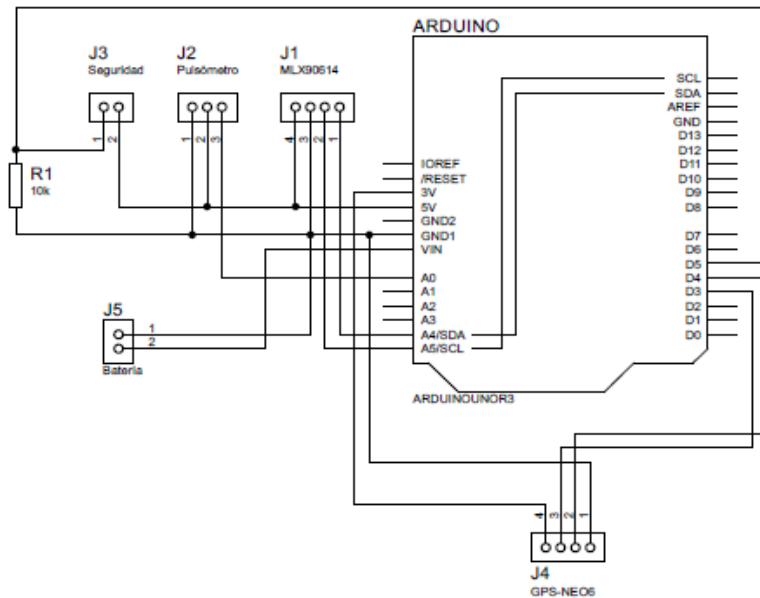
Diagrama de conexiones

Para el diseño e implementación del nodo cliente es necesario realizar su diagrama eléctrico y estructural con la finalidad de presentar la distribución de las

conexiones entre los diferentes elementos. A partir del circuito esquemático de la Figura 30 se generó el esquema de conexión física de la Figura 31.

Figura 30

Diagrama Esquemático.

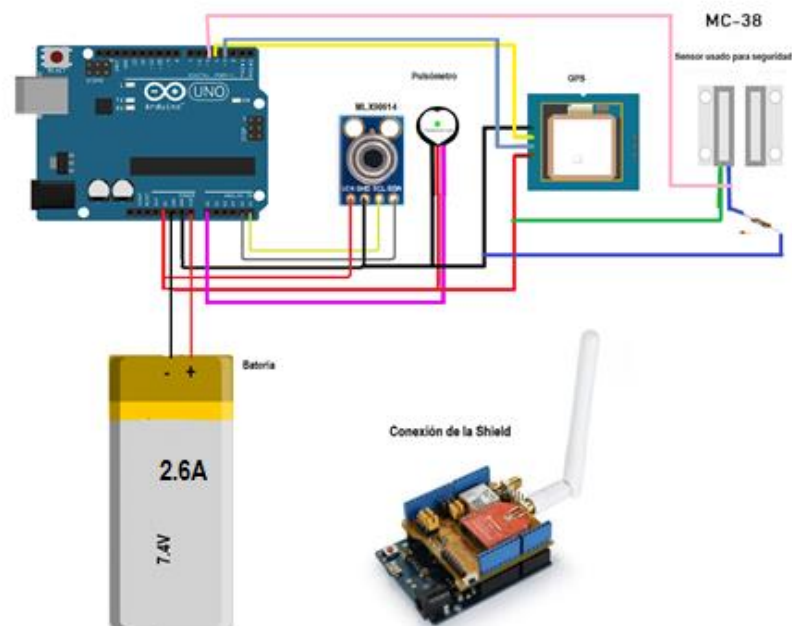


Nota. El gráfico representa el circuito esquemático del nodo cliente que permite la monitorización de vaca lechera, es necesario aclarar que el dispositivo Shield LoRa va conectado en la parte superior del Arduino UNO.

Como alimentación de este nodo que contiene los sensores que se encargan de medir las diferentes variables ya mencionadas con anterioridad se tiene una batería de 7.4V, no se hizo uso de un reductor de voltaje debido que el Arduino puede ser alimentado hasta con 12V, lo demás componentes como son los diferentes sensores se conectan de acuerdo a la figura que se muestra a continuación.

Figura 31

Esquema de conexión física del nodo cliente.



Nota. En el gráfico se puede apreciar la conexión que tiene el nodo sensor, además en la parte “Conexión de la Shield”, se representa la forma en cómo se debe conectar este módulo con Arduino Uno, en este caso la Shield se empotra encima del arduino.

Comparación de sensores seleccionados con equipos patrones

Con el objetivo de que los datos que se han adquirido sean confiables, fue necesario realizar una comparación entre los sensores seleccionados con equipos patrones.

Termómetro GP-300 y el sensor de temperatura MLX90614

El termómetro GP-300 cuenta con un sensor infrarrojo biométrico de alta precisión que lanza un puntero de luz por medio del puntero láser y que a través de cálculos físicos induce la temperatura de la persona, animal u objeto. Es un termómetro digital infrarrojo sin contacto, la distancia a la cual se puede colocar el

termómetro para obtener valores efectivos es de 3 a 5 cm, el rango de medición va de 32°C – 42.9°C, el margen de error es de $\pm 0.2^\circ \text{C}$.

Figura 32

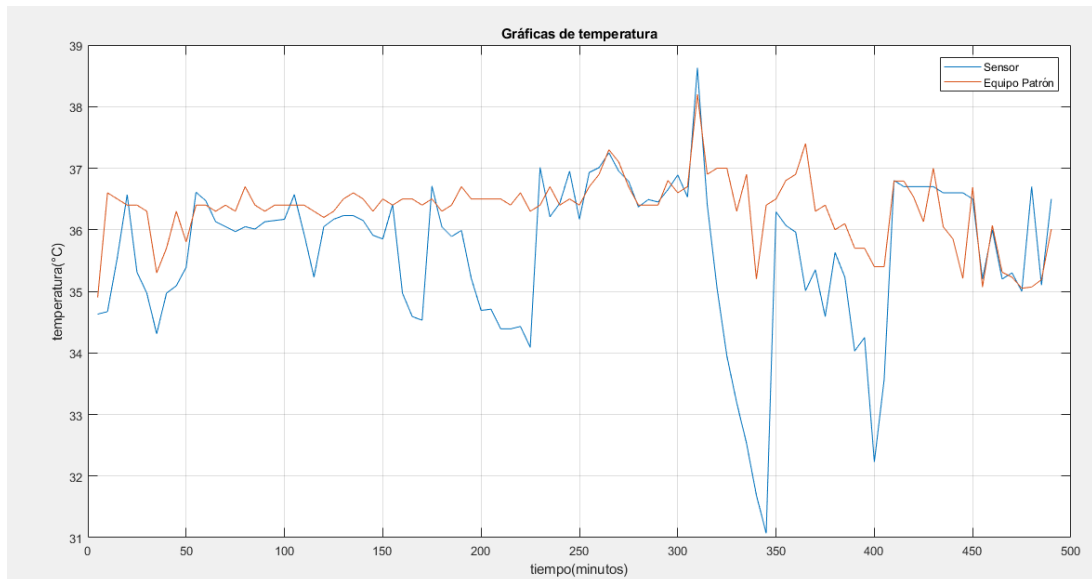
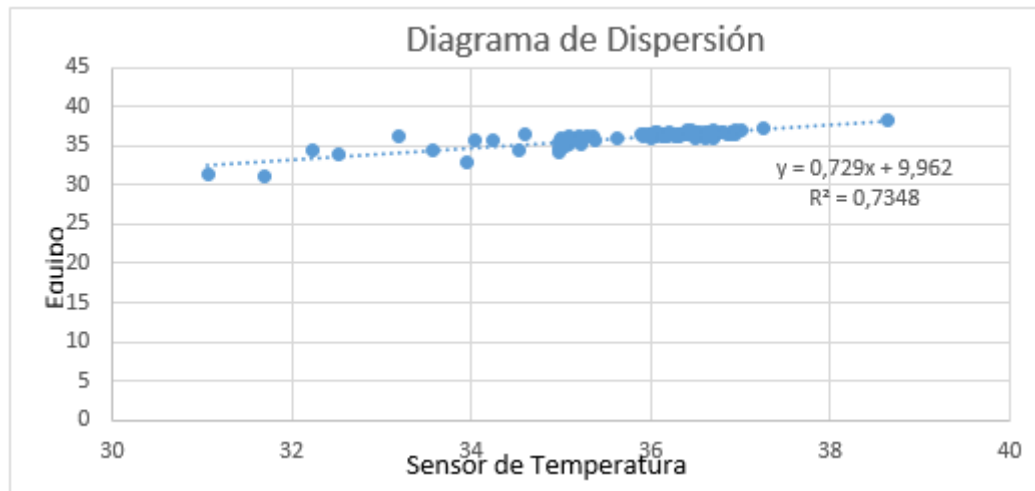
Termómetro GP-300.



Nota. En el gráfico se muestra el equipo patrón para medir la temperatura.

Luego de definir y caracterizar al equipo patrón que se usó para la comparación, se procede a graficar los datos tomados para ver la relación y similitud que tienen los mismos.

En la gráfica 33 se puede apreciar dos curvas, tanto del sensor de temperatura como del termómetro, estas representan los datos tomados para verificar la similitud que tienen, además en esta gráfica se puede ver la tendencia que tienen las muestras de acuerdo al tiempo, estos datos se tomaron en una persona, y como se puede observar la curva del termómetro que fue tomado como base para comparación tiene mayores fluctuaciones ya que la respuesta de este es más rápida. Con respecto a la gráfica 34 se representa la dispersión de los datos, en este caso el coeficiente de relación es igual a 0.7348, por lo tanto, hay una relación directa al ser un valor positivo y como es cercano a 1 la relación es fuerte, entonces los valores del sensor son muy similares al del equipo patrón.

Figura 33*Curvas de temperatura.***Figura 34***Diagrama de dispersión de la temperatura.****Oxímetro Teasiy C101H1 y el sensor de pulso cardíaco***

Este oxímetro permite tener mediciones precisas de la saturación de SpO₂, además e puede usar para detectar la frecuencia del pulso y la fuerza del pulso, sus lecturas son precisas y su pantalla LED de alta definición muestra de manera

práctica la saturación. Además, tiene un rango de medición: 30 BPM – 240 BPM y la precisión es de +/- 2BPM.

Figura 35

Oxímetro Teaisy C101H1.



Nota. En el gráfico se muestra el equipo patrón para medir la frecuencia cardíaca. Obtenido de: <https://s.libertaddigital.com/2020/12/30/oximetro-teaisy-c101h1.jpg>.

Luego de definir y caracterizar al equipo patrón que se usó para la comparación, se procede a graficar los datos tomados para ver la relación y similitud que tienen los mismos.

Figura 36

Curvas del pulso cardíaco.

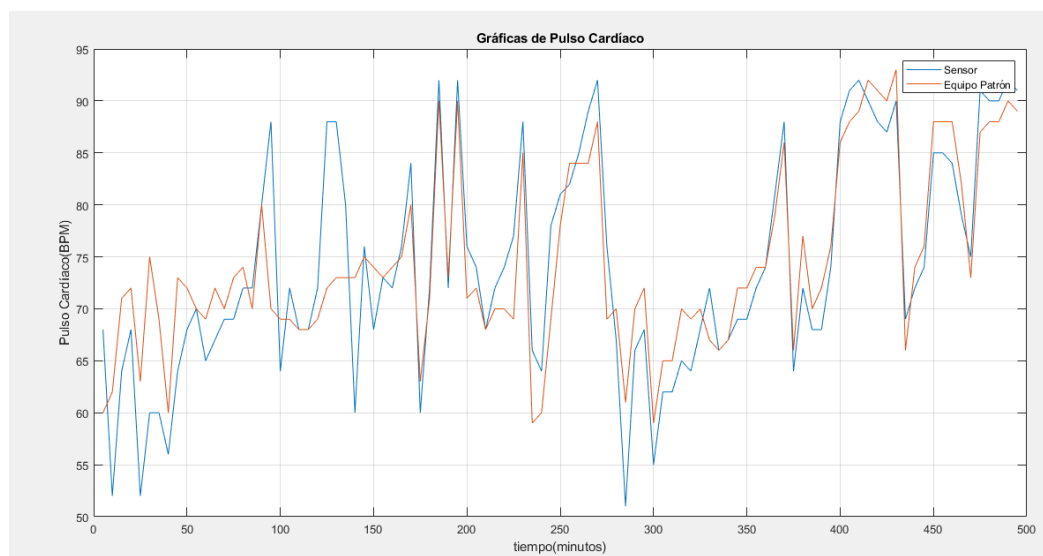
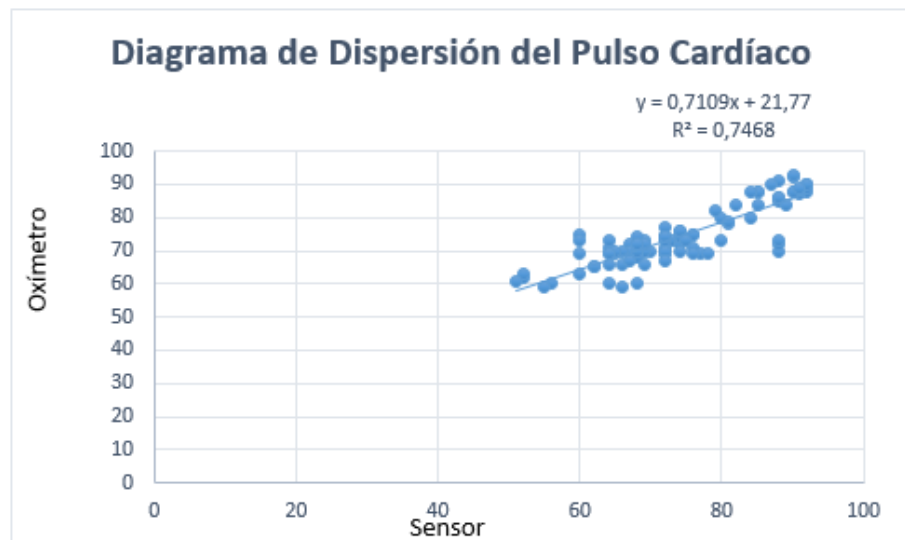


Figura 37

Diagrama de dispersión del pulso cardíaco.



En la gráfica 36 se puede observar las curvas de cada equipo respectivamente, en este caso hay una gran similitud, sin embargo el oxímetro (equipo patrón) tiene una mayor fluctuación ya que su respuesta es más rápida, estos datos se relacionan de forma directa de acuerdo a la gráfica 37, ya que el coeficiente de relación es positivo y muy cercano a 1, teniendo una relación fuerte, es necesario mencionar que de acuerdo a las características del oxímetro su nivel de error es de $\pm 2\text{BPM}$, a pesar de tener este valor los datos que se tomaron son muy similares, por tanto el sensor que se usa es muy efectivo, también es importante saber que la variable a medir en este caso la frecuencia cardíaca, puede cambiar y no necesariamente por afecciones en la salud, sino que es muy sensible y por tanto tiene bastantes variaciones.

Google Maps y GPS NEO-6M 2019

Google Maps permite buscar la posición geográfica de sitios de manera ágil, muestra al usuario cómo llegar y ofrece la posibilidad de utilizar Street View para visualizar la calle en la que está ubicado.

Las coordenadas de Google Maps se encuentran en el sistema WGS84 y se muestra la latitud y la longitud, positiva para Norte y Este, negativa para Sur y Oeste.

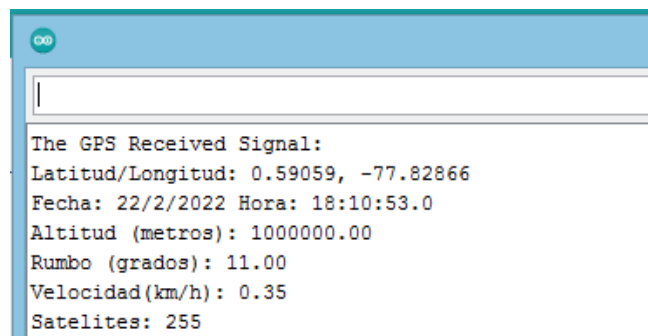
Para obtener las coordenadas del lugar en interés:

- Se hace clic con el botón derecho del ratón en el punto deseado en el mapa para que aparezca un menú con opciones. En el menú, se da clic en la opción ¿Qué hay aquí? y esto muestra las coordenadas.

Una vez seleccionado el programa con el cual se realiza la comparación se procede a tomar los datos de latitud y longitud de un lugar en específico, por tanto con el programa de arduino se reciben los datos que se muestran la Figura 38 y al mismo tiempo con el medio de google Maps dando como resultado las coordenadas que se muestran en la Figura 39. Con el objetivo de verificar la exactitud que tiene el módulo sensor, se toman varias mediciones de geo-localización y los resultados se muestran en la Tabla 11, que permite concluir que a pesar de que los datos se diferencian por las milésimas tanto la latitud y la longitud tienen un error que se encuentra en el rango aceptable, por tanto, los errores absolutos no afectan las mediciones.

Figura 38

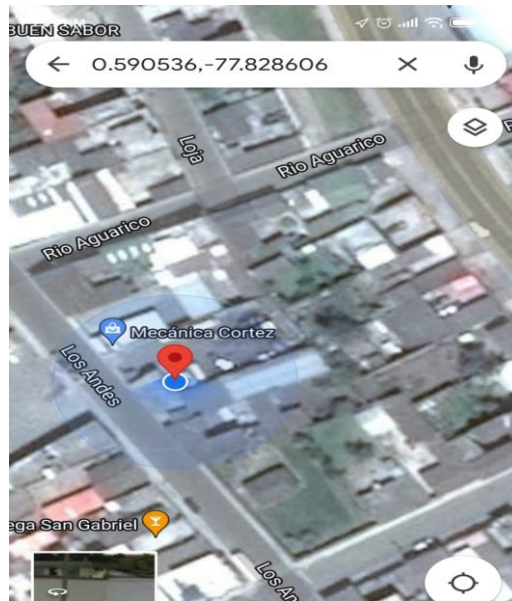
Coordenadas obtenidas por el sensor GPS.



```
The GPS Received Signal:
Latitud/Longitud: 0.59059, -77.82866
Fecha: 22/2/2022 Hora: 18:10:53.0
Altitud (metros): 1000000.00
Rumbo (grados): 11.00
Velocidad(km/h): 0.35
Satelites: 255
```


Figura 39

Coordenadas obtenidas por Google Maps.

**Tabla 11**

Pruebas de geo-localización GPS-NEO vs Google Maps.

Coordenadas Google Maps		Coordenadas GPS-NEO		Error Absoluto	
Latitud	Longitud	Latitud	Longitud	Latitud	Longitud
0.59053	-77.82845	0.59057	-77.82841	-0.00004	-0.00004
0.59073	-77.82870	0.59073	-77.82874	0.00000	0.00004
0.59056	-77.82844	0.59054	-77.82846	0.00002	0.00002
0.59042	-77.82846	0.59041	-77.82836	0.00001	-0.00010
0.59035	-77.82839	0.59035	-77.82829	0.00000	-0.00010
0.59049	-77.82841	0.59040	-77.82830	0.00009	-0.00011
0.59038	-77.82830	0.59041	-77.82830	-0.00030	-0.00000
0.59049	-77.82834	0.59041	-77.82827	0.00008	-0.00007
0.59035	-77.82826	0.59039	-77.82828	-0.00004	0.00002
0.59053	-77.82833	0.59043	-77.82844	0.00010	0.00011

Implementación del Hardware y Software del sistema prototipo IoT

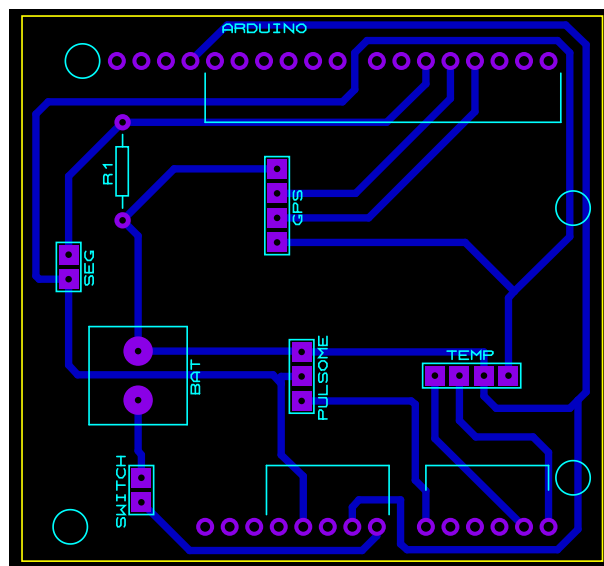
Diseño en PCB del circuito electrónico

El diseño de la placa se realizó teniendo en cuenta que debe ser lo más pequeña posible para que las dimensiones de la caja sean reducidas, así se evita que el animal se estrese debido al peso del prototipo y con esto evitar que los valores de las variables cambien debido al tamaño del prototipo.

En la Figura 40 se tiene el diseño de la placa final, misma que se imprimió y en la cual se soldaron los elementos que se seleccionaron con anterioridad para cumplir con la implementación del prototipo.

Figura 40

Diseño en PCB.



Nota. En el gráfico se puede apreciar el diseño en PCB, misma que se usó para la implementación del prototipo.

Nodo cliente

Para poder usar los sensores se realizó la configuración de cada uno en el software de Arduino además del módulo LoRa Shield, el mismo que se encarga de recibir y procesar la información para luego ser enviada al Gateway, es necesario

tener en cuenta que los datos deben ser enviados a la misma frecuencia en la que trabaja el Gateway, además de importar las librerías necesarias para cada uno de los sensores a configurar, crear funciones para poder transmitir los datos y tener en cuenta la concatenación de todas las variables con el Gateway y así se pueda reconocer al nodo que envía la información.

Programación del nodo cliente

En base a la Figura 28 se realiza la programación del nodo cliente.

Sensor de Temperatura

Para el sensor de temperatura que se ha seleccionado, es necesario incluir la librería Wire y la librería Adafruit_MLX90614 estas dos permiten la programación del termómetro infrarrojo. La librería Wire permite realizar la comunicación con dispositivos por bus I2C (Inter-Integrated Circuit o 2 alambres). Usa dos líneas: SDA (datos) y SCL (reloj) que son los pines que dispone el sensor MLx90614.

En la figura siguiente se puede apreciar la programación que se realizó para que el sensor mida de forma adecuada la temperatura, se tiene la variable tempObjeto de tipo float, en esta se almacena los datos que mide el sensor, además se tiene la función medicionesTemp() de tipo float que se encarga de realizar el proceso de medir la temperatura del animal para luego transformarla a grados Celsius, y finalmente se hace uso de la línea de código Serial.print misma que permite imprimir en el monitor serie de Arduino los datos que se han tomado con el sensor.

Figura 41

Configuración del sensor MLX90614.

```
//*****Sensor de temperatura *****
#include <Wire.h>
#include <Adafruit_MLX90614.h>
Adafruit_MLX90614 mlx = Adafruit_MLX90614(); // Instanciar objeto
float tempObjeto; // Variables a utilizar

float medicionesTemp() {
  /***/TEMPERATURA****/
  //Obtener temperaturas grados Celsius
  tempObjeto = mlx.readObjectTempC();
  // Mostrar información
  Serial.print("Temp. objeto => ");
  Serial.print(tempObjeto);
  Serial.println("°C"); // Envía los valores al puerto serial
  delay(2000);
  return tempObjeto;
}
```

Nota. La imagen expone la configuración que se realizó en Arduino para el sensor de temperatura.

Sensor de Frecuencia Cardíaca

Para la configuración de este sensor, es necesario incluir la librería <PulseSensorPlayground.h> y en la siguiente figura se presenta la programación que se realizó para la toma de datos.

Como se puede apreciar en la imagen, se tiene dos variables PulseWire que es constante y en la cual se pone el valor de cero que indicada la entrada a la que se ha colocado el pin en arduino, en BPM se almacena el valor a medir y finalmente se tiene la función pulsoBPM (), que es la encargada de medir los pulsos para luego imprimirlos en el monitor serie de Arduino.

Figura 42

Configuración del sensor de frecuencia cardíaca.

```
#include <PulseSensorPlayground.h>    // Includes the PulseSensorPlayground Library.
// VariablesPulso
const int PulseWire = 0;              // PulseSensor PURPLE WIRE connected to ANALOG PIN 0
int BPM=0;

int pulsoBPM(){

    BPM = analogRead(PulseWire); // Read the PulseSensor's value.
    Serial.println("♥ A HeartBeat Happened ! ");
    Serial.print("BPM: ");
    Serial.println(BPM);
    delay(20);
    return BPM;
}
```

Nota. La imagen expone la configuración que se realizó en Arduino para el sensor de frecuencia cardíaca.

Sensor de Seguridad

Para el sensor de seguridad se hizo uso del código que se muestra en la siguiente imagen, para lo cual se definió dos variables, reedswitch de tipo int en la cual se asignó el valor de 5 que define el pin de entrada en Arduino y además la variable val de tipo entero que almacena el valor a medir, finalmente se tiene la función collar () que es la encargada de obtener los valores del sensor de seguridad y en caso de que la variable val detecte un cambio de valor de alto a bajo envía un mensaje que el collar se ha desconectado, todo esto se puede observar en el monitor serie.

Figura 43

Configuración del sensor de seguridad.

```
//COLLAR
int reedswitch = 5;

int val =0;
//COLLAR
pinMode (reedswitch, INPUT);

int collar(){
  int valorC;
  val= digitalRead(reedswitch);
  if (val== LOW){
    Serial.println("¡ALERTA! COLLAR DESCONECTADO --> Estatus:0 ");
    delay(3000);
    valorC=0;
  }
  else{
    Serial.println("COLLAR CONECTADO --> Estatus:1 ");
    delay(3000);
    valorC=1;
  }
  return valorC;
}
```

Nota. La imagen expone la configuración que se realizó en Arduino para el sensor de seguridad.

Sensor GPS UBLOX NEO

Para poder realizar la configuración de este módulo sensor es necesario incluir esta librería <TinyGPS.h> ya que permite identificar latitud y longitud, así como otras variables de una forma más sencilla, además se necesita de la librería <SoftwareSerial.h>, ya que esta facilita la emulación por Software del puerto serie y se usa Rx y TX puertos serie por Hardware para la comunicación con la computadora, con esto se verifica los datos que recibe el módulo GPS.

En la imagen siguiente se aprecia la programación para el modulo GPS, en este caso se definen los pines tanto para recepción y transmisión respectivamente, además se instancia el objeto que se usa para llamar en la función gpsdata que obtiene los valores de latitud, longitud velocidad y la fecha con su respectivo

formato, aquí se definieron las variables correspondientes para almacenar los diferentes valores que se obtienen de este módulo

Figura 44

Código para la configuración del módulo GPS.

```
#define RXpin 4
#define TXpin 3
SoftwareSerial mySerial(TXpin,RXpin);
TinyGPS gps;

void gpsdata(TinyGPS &gps)
{
  long latitude, longitude;
  float flatitude,flongitude;
  float velocidadK;
  unsigned long date, time, chars, age;
  int year;
  byte month, day, hour, minute, second, hundredths;
  unsigned short sentences, failed;

  //latitude and longitude in float
  gps.f_get_position(&flatitude, &flongitude, &age);
  Serial.print("Lat/Long(float): ");
  Serial.print(flatitude,5);
  Serial.print(", ");
  Serial.print(flongitude,5);
  Serial.print(" Fijar: "); Serial.print(age); Serial.print("ms.");
  Serial.print("\n");
  velocidadK=gps.f_speed_kmph();
  float velocidadMPS=gps.f_speed_mps();
  Serial.print("Velocidad: ");
  Serial.print(" (mps): ");
  Serial.print(velocidadMPS);
  Serial.print("\n");
```

Nota. La imagen expone la configuración que se realizó en Arduino para el módulo GPS.

Módulo LoRa Shield

Como primer punto es necesario incluir la librería SPI que permite realizar el proceso de comunicación LoRa Shield con la placa de Arduino y la librería LoRa que facilita la transmisión y recepción de los datos por medio de LoRa.

Respecto a la imagen que se indica a continuación, se puede apreciar la configuración de los parámetros que son esenciales para la comunicación por LoRa, estos parámetros son el nombre del nodo ID, el factor de dispersión, la frecuencia en la que se va a trabajar, la tasa de código, el ancho de banda y las palabras que permiten la sincronización para una red pública, por lo tanto los valores que se colocaron hacen referencia lo establecido para este tipo de comunicación y con respecto a la banda de frecuencia, se hace uso de 915MHz por ser un espectro libre, este banda de frecuencia se denomina ISM que es reservada a nivel internacional para estudios industriales, científicos y médicos. (Radio, 2020).

Figura 45

Código para configurar el módulo LoRa.

```
char *node_id = "<4567>";
uint8_t datasend[54];
unsigned int count = 1;
unsigned long new_time,old_time=0;
long lastSendTime = 0;          // last send time
int interval = 14000;          // interval between sends

void setup() {
  Serial.begin(9600);
  mySerial.begin(9600);
  delay(200);

  //COLLAR
  pinMode (reedswitch, INPUT);
  //LoRa
  while (!Serial);
  Serial.print("##### ");
  Serial.print("La comunicación se ha iniciado...");
  Serial.println(" #####");
  while (!LoRa.begin(915000000)) {
    Serial.print(".");
    delay(500);
  }
  LoRa.setSignalBandwidth(125000);
  LoRa.setCodingRate4(5);
  LoRa.setSpreadingFactor(7);
  LoRa.setSyncWord(0x34);
  Serial.print("##### ");
  Serial.println("La comunicación LoRa ha tenido éxito!");
}
```

Nota. En esta imagen se evidencia el código para programar la comunicación por medio de Lora.

Figura 46

Concatenación de cada variable con sus respectivos valores.

```

strcat(data, "");
strcat(data, "");
strcat(data, collar_1);
strcat(data, ",");
strcat(data, "");
strcat(data, lat_1);
strcat(data, ",");
strcat(data, "");
strcat(data, long_1);
strcat(data, ",");
strcat(data, " ");
strcat(data, velo_1);
strcat(data, ",");
strcat(data, "");
strcat(data, pos_1);
strcat(data, ",");
strcat(data, "");
strcat(data, "");
strcat(data, temper_1);
strcat(data, ",");
strcat(data, "");
strcat(data, bpm_1);
strcat(data, ",");
strcat(data, "");
strcat(data, snr_1);
strcat(data, ",");
strcat(data, rssi_1);
strcat(data, "");
strncpy((char *)datasend.data):

```

Nota. En la imagen se aprecia la concatenación de cada variable con sus respectivos datos de acuerdo a cada sensor, para lo cual se hizo uso de la función `strcat` y finalmente se hace uso de la función `strcpy` que permite formar el payload que va a ser enviando, esta cadena de variables concatenadas se almacena en la variable `datasend`.

Figura 47

Código para el envío y recepción de datos.

```

/**ENVIO DE DATOS**/
void SendData()
{
    Serial.println("Los datos se envían...");
    medicionGPS();
    LoRa.beginPacket();
    gpsdata(gps);
    LoRa.print((char *)datasend);
    LoRa.endPacket();
    Serial.println("Payload enviado correctamente!!");
}

/**RECEPCION DE DATOS**/
void onReceive(int packetSize) {
    if (packetSize == 0) return;          // Si no hay mensaje, return
    String mensajeRX = "";
    while (LoRa.available()) {
        mensajeRX += (char)LoRa.read();
    }
    Serial.println("Instrucción recibida: " + mensajeRX);
    Serial.println();
    if(mensajeRX == "on2"){
        Serial.println("Midiendo RSSI y SNR...");
    }
}

```

Nota. En esta imagen se aprecia el código que se realizó para el envío y recepción de datos, se tienen dos funciones `SendData ()`, que es la encargada de contener los métodos para el envío de la carga útil, es necesario aclarar que la función `LoRa.beginPacket` permite la inicialización del envío del paquete de datos por medio de LoRa, mientras que la función `onRceive` es la encargada de recibir los datos, en este caso envía el mensaje midiendo RSSI y SNR cuando el gateway con el nodo final se han comunicado correctamente y el envío de datos ha tenido éxito, estos valores hacen referencia al nivel de ruido que ha tenido la comunicación y el nivel de potencia de la señal, respectivamente.

Finalmente, para recibir los valores de RSSI y SNR se hace uso de `LoRa.packetSnr ()` y `LoRa.packetRssi ()`.

Diseño 3D de la carcasa del nodo cliente

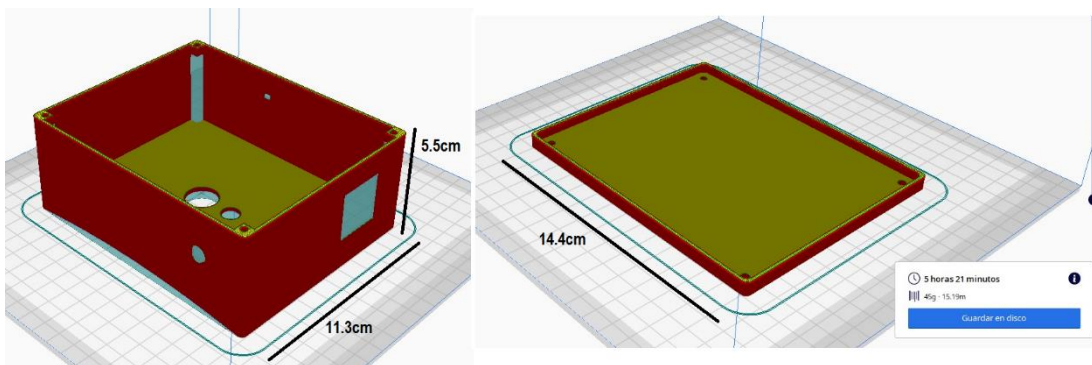
Para poder realizar la caja que se indica en la figura 48, se procedió a realizar el diseño en el software SolidWorks teniendo las siguientes consideraciones:

- La placa en PCB tiene una dimensión de 6x6 cm tanto de largo como de ancho.
- El largo de la batería es de 12 cm y el ancho es de 1.6cm.
- La caja debe contar con cuatros huecos para que se pueda colocar el sensor de temperatura, el de pulso, y las antenas del gateway y de la Shield respectivamente.
- El material debe ser resistente.
- El color de la caja debe ser uno que permita la reflexión de la radiación y con esto de se vean afectados los parámetros a medir.

Por lo tanto, el diseño final es el que se muestra a continuación con sus respectivas medidas.

Figura 48

Diseño de la caja para el nodo cliente.



Nota. En el gráfico se puede apreciar las dimensiones de la caja en la cual se colocan los diferentes sensores.

Nodo Gateway

El gateway LG01-N que se está empleando permite hacer uso de un solo canal LoRa.

Configuración para acceso a internet

Existen dos formas de conectar el gateway a la red conectarse a través de WiFi o conectarse vía Ethernet, se opta por la segunda opción que es la más fácil. Haciendo uso de un cable ethernet, se conecta el puerto LAN del gateway directamente a la PC. Esta obtendrá la IP del puerto LAN del gateway que por defecto para LG01-N es **10.130.1.1**.

Figura 49

Conexión vía Ethernet con conexión directa desde la PC.



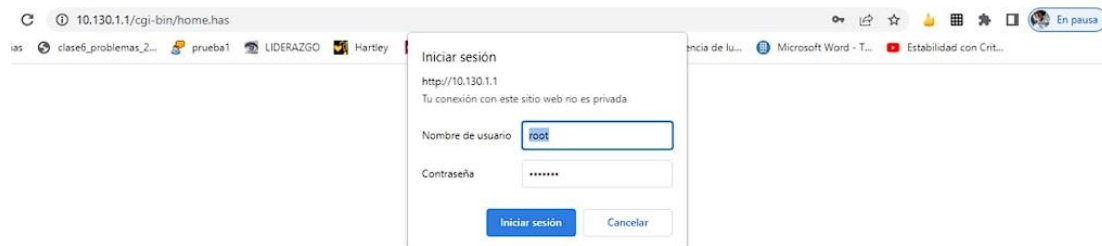
Posteriormente se abre un navegador en la PC y se entra a la dirección IP por defecto 10.130.1.1. Se presentará la interfaz de inicio de sesión (log-in) del LG01-N, las credenciales por defecto son:

User Name: root

Password: dragino

Figura 50

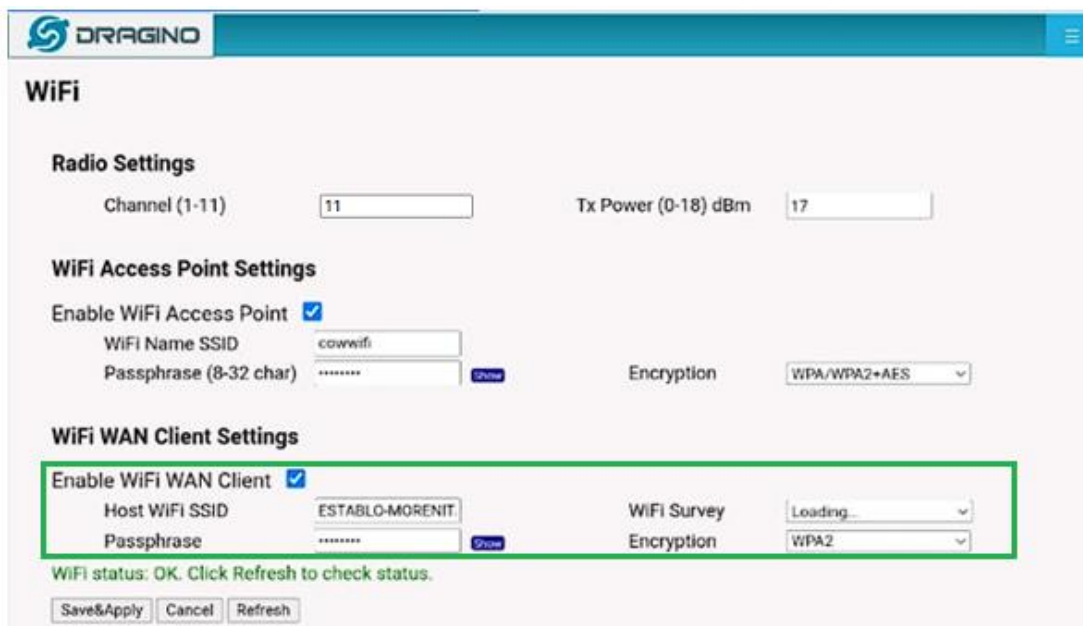
Interfaz de inicio de sesión del Gateway LoRa.



Una vez dentro de la interfaz de usuario del gateway, se procede a vincularlo con el router de la hacienda La Morenita como Cliente WiFi. Para ello, se dirige a la pestaña System → WiFi → WiFi WAN Client Settings.

Figura 51

Configuración del Gateway LoRa a la red WiFi de la hacienda La Morenita.



En la configuración del cliente WiFi WAN, se ingresa el nombre del punto de acceso WiFi y la contraseña, luego se da clic en Guardar y aplicar los cambios. Por último, se desconecta el cable ethernet y se reinicia el gateway, comprobando así que el gateway LoRa tiene acceso a internet.

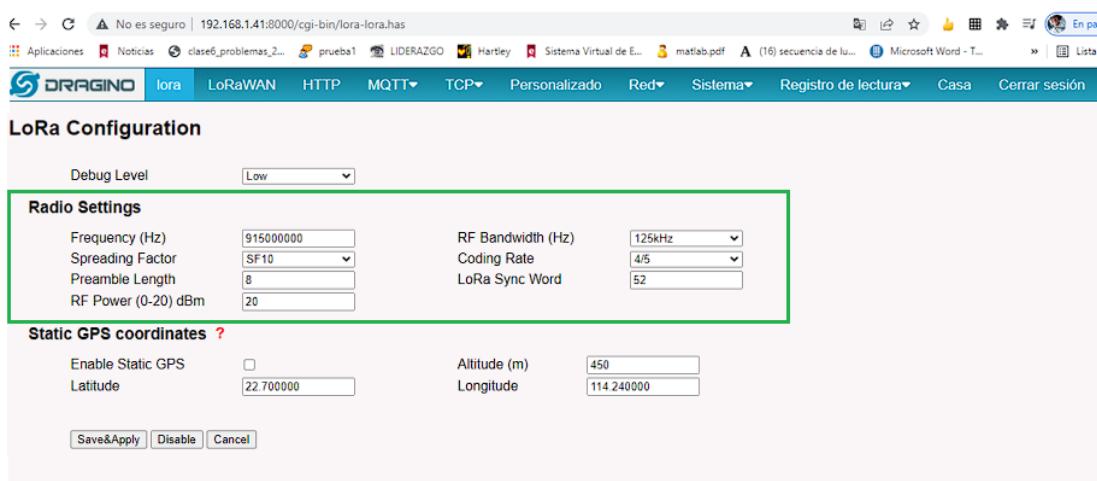
Configuración del modo de operación del Gateway LG01-N

Primero se configura el plan de frecuencia en LG01-N para que el equipo funcione en bandas destinadas a espectro radioeléctrico para uso determinado en bandas libres en Ecuador según la ARCOTEL, corresponde la banda 915-928 MHz específicamente la frecuencia establecida es AU915 MHz para ser usada en la tecnología LoRa.

En una red de área amplia de baja potencia, los parámetros de RF determinantes a ser configurados son: un ancho de banda (BW) de 125 kHz para tener un mayor alcance y mejorar la sensibilidad, un factor de dispersión (SF) de 10 ya que cuanto mayor es el SF más capacidad tiene el receptor de alejarse de la señal de ruido. La tasa de codificación (CR) de 4/5 asegurando la fiabilidad de los datos, la longitud del preámbulo de 8 y la potencia de la señal de 20 dBm. Con esta configuración la comunicación tiene un mejor balance en el enlace y robustez a la pérdida de datos.

Figura 52

Configuración de los parámetros de radio LoRa.



The screenshot shows a web browser window displaying the 'LoRa Configuration' page. The page has a navigation bar with the DRAGINO logo and several menu items: lora, LoRaWAN, HTTP, MQTT, TCP, Personalizado, Red, Sistema, Registro de lectura, Casa, and Cerrar sesión. The main content area is titled 'LoRa Configuration' and includes a 'Debug Level' dropdown set to 'Low'. A green box highlights the 'Radio Settings' section, which contains the following fields:

Parameter	Value
Frequency (Hz)	915000000
Spreading Factor	SF10
Preamble Length	8
RF Power (0-20) dBm	20
RF Bandwidth (Hz)	125kHz
Coding Rate	4/5
LoRa Sync Word	52

Below the radio settings is the 'Static GPS coordinates' section, which includes:

Parameter	Value
Enable Static GPS	<input type="checkbox"/>
Latitude	22 700000
Altitude (m)	450
Longitude	114 240000

At the bottom of the configuration area are three buttons: 'Save&Apply', 'Disable', and 'Cancel'.

Configuración MQTT

Se escogió MQTT como protocolo de comunicación para transmitir la información recolectada por el gateway al servidor en la nube por su sencillez y ligereza. Por tanto, se configura los siguientes parámetros en el gateway: la dirección DNS (farmyng.ga) y el puerto por defecto del bróker 1883, el Client ID, además se habilitan Publish y Subscribe para transmitir los mensajes a los clientes subscriptos al tópicos monitoreoE y para recibir los mensajes desde la plataforma con el tópicos monitoreoR.

Figura 53

Configuración de los parámetros de MQTT en el Gateway LoRa.

The screenshot displays the 'MQTT Client Configuration' interface. The 'General' profile is selected. The 'Broker Address [-h]' is set to 'farmyng.ga', 'Broker Port [-p]' is '1883', and 'Client ID [-i]' is 'GatewayNodo'. The 'Publish' section has 'Enable Publish' checked, 'Quality of Service [-q]' set to 'QoS 0', and 'Topic Format [-t]' set to 'monitoreoE'. The 'Subscribe' section has 'Enable Subscribe' checked, 'Quality of Service [-q]' set to 'QoS 0', and 'Topic Format [-t]' set to 'monitoreoR'. The 'Save&Apply' and 'Cancel' buttons are at the bottom.

A continuación, se configura el puerto TCP 4567 para el canal MQTT por el cual el nodo Gateway enviará los datos de SNR y RSSI.

Figura 54

Configuración de los parámetros del canal MQTT en el Gateway LoRa.

MQTT Channel Management

Add Channel

Local ID:

Remote ID:

API Key:

Delete Channel

Local ID:

Channels Mapping:

Local ID	Remote ID	API Keys
4567	4567	

Estructura del payload o carga útil de LoRa

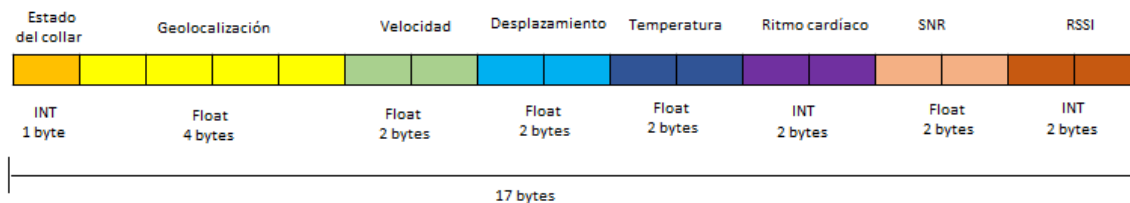
Existen dos formas de conectar el gateway a la red conectarse a través de WiFi o conectarse vía Ethernet, se opta por la segunda opción que es la más fácil. Haciendo uso de un cable ethernet, se conecta el puerto LAN del gateway directamente a la PC. Esta obtendrá la IP del puerto LAN del gateway que por defecto para LG01-N es **10.130.1.1**.

Para definir el formato del payload se procede a analizar los valores de las variables del sistema. Se debe tener en cuenta que lora permite hasta 255 bytes para el envío de información. La trama inicia destinando 1 byte para el estado del collar (0 o 1) porque con 1 byte se puede expresar un número de 0 a 2 no existe inconveniente. Se destina 4 bytes para la geo localización cuyos valores son latitud (0.5648) y longitud (-77.791) dado que con 4 bytes se puede expresar un número de 0 a 4.292.967.296 se justifica. Para finalizar se destinan 2 bytes para cada una de las siguientes variables: la velocidad (5 máx), el desplazamiento (1000m apróx), la temperatura (39.5°C máx), el ritmo cardíaco (80bpm máx), el SNR (-20 dBm apróx) y

el RSSI (-500 apróx), dado que con 3 bytes se puede expresar un numero de 0 a 16.777.215, este valor es adecuado para todas las variables descritas.

Figura 55

Formato del payload LoRa configurado para las variables del prototipo de collar IoT.



Diseño del servidor en la nube AWS

Una vez elegido el software en el que se va a desarrollar el servidor, se procede a crear un usuario para poder obtener los beneficios que brinda la plataforma, por lo tanto, en Google ingresamos el siguiente URL: <https://console.aws.amazon.com/> que nos dirige a la página en donde se puede crear el usuario.

Figura 56

Crear Cuenta en la consola de Amazon.

Iniciar sesión

Usuario raíz
Propietario de la cuenta que realiza tareas que requieren acceso ilimitado. [Más información](#)

Usuario de IAM
Usuario de una cuenta que realiza tareas diarias. [Más información](#)

Dirección de email del usuario raíz

Siguiente

Al continuar, acepta el [Contrato de cliente de AWS](#) u otro acuerdo para los servicios de AWS y el [Aviso de privacidad](#). Este sitio utiliza cookies esenciales. Consulte nuestro [Aviso de cookies](#) para obtener más información.

¿Es nuevo en AWS?

Crear una cuenta de AWS

Instancias de spot de Amazon EC2

Aumente hasta 10 veces la agilidad de las cargas de trabajo de producción sin estado y tolerantes a errores con capacidad de cómputo asequible.

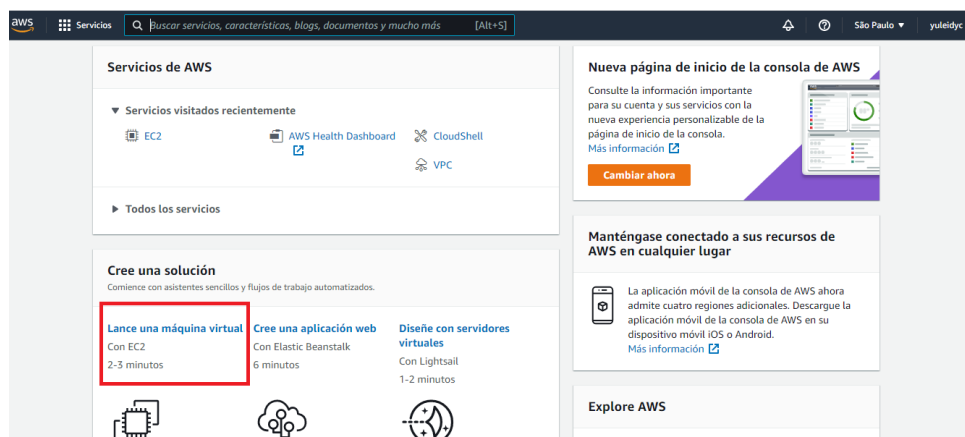
[Más información »](#)

Nota. En la imagen se puede apreciar la ventana que aparece cuando se ingresa al link antes mencionado, para crear la cuenta se presiona en el recuadro rojo, una vez que se aprueba el registro y se crea el usuario ya se puede acceder a los diferentes beneficios de AWS, entre estos, la creación de la instancia necesaria para este caso de estudio.

Teniendo listo el usuario ya se puede crear la máquina virtual, es importante mencionar que Amazon tiene servidores alrededor de todo el mundo, para lo cual es necesario escoger uno que este cercano a Ecuador, por tal motivo se escogió el de Sao Paulo, una vez realizado esta selección, se da clic en el recuadro de color rojo que se muestra en la siguiente imagen.

Figura 57

Lanzamiento de la instancia.



Una vez que se ha dado clic en lanzar una instancia aparece la ventana que se muestra en la Figura 58, en el numeral 1 se coloca el nombre del servidor a crear, este nombre es de acuerdo a lo que el usuario desee colocar, en el numeral 2 se selecciona el sistema operativo que se desee, para este caso se seleccionó Ubuntu en la versión 18.04 ya que está disponible para la capa gratuita y en el numeral 3 el tipo de instancia para lo cual se ha escogido t2.micro que es de propósito general y de tipo free(libre).

Figura 58

Creación de la Instancia-Selección de parámetros.

Nombre

SERVIDOR_FARMING 1

Agregar etiquetas adicionales


▼ Imágenes de aplicaciones y sistemas operativos (Amazon Machine Image) [Información](#)

Una AMI es una plantilla que contiene la configuración de software (sistema operativo, servidor de aplicaciones y aplicaciones) necesaria para lanzar la instancia. Busque o examine las AMI si no ve lo que busca a continuación.


🔍 *Busque en nuestro catálogo completo que incluye miles de imágenes de sistemas operativos y aplicaciones*

Recientes | **Inicio rápido**


Amazon Linux




Ubuntu




Windows



Red Hat



SUSE Linux



🔍 **Buscar más AMI**

Incluye las AMI de AWS, Marketplace y la comunidad

Amazon Machine Image (AMI) 2

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type

ami-01e914058c8b92680 (64 bits (x86)) / ami-0bddd37db53fdcc (64 bits (Arm))

Virtualización: hvm Habilitado para ENA: true Tipo de dispositivo raíz: eos

Descripción

Canonical, Ubuntu, 18.04 LTS, amd64 hionic image build on 2022-04-11

Arquitectura ID de AMI

64 bits (x86) ami-01e914068c8b92680

Apto para la capa gratuita ▼

▼ Tipo de instancia [Información](#) 3

Tipo de instancia

t2.micro Apto para la capa gratuita ▼

Familia: t2 1 vCPU 1 GiB Memoria

Bajo demanda Linux precios: 0.0186 USD por hora

Bajo demanda Windows precios: 0.0232 USD por hora

Comparar tipos de instancias

Figura 59

Aumento de Volumen de almacenamiento.

▼ Configurar almacenamiento Información Avanzado

1x GiB Volumen raíz

i Los clientes que cumplan los requisitos de la capa gratuita pueden obtener hasta 30 GB de almacenamiento magnético o de uso general (SSD) de EBS ✕

Nota. Además, se necesita más almacenamiento para lo cual se eligió 25 GiB y de tipo gp2 (uso general).

Una vez que se han configurado estos parámetros, es necesario habilitar algunos puertos ya que AWS por defecto cierra todo debido a temas de seguridad, los puertos se deben configurar de acuerdo a lo que se vaya a utilizar y es necesario que se habiliten grupos de seguridad para evitar que los atacantes intenten dañar a la instancia creada, así, estos grupos actúan como un firewall virtual, y en caso de que el usuario no habilite los grupos de seguridad AWS ejecuta uno por default, por lo tanto, para el caso estudio se habilitaron los siguientes puertos.

Figura 60

Puertos para el Grupo de Seguridad.

Tipo	Protocolo	Puertos	Especificación
HTTP	TCP	80	HTTP
TCP	TCP	8083	Vesta
TCP	TCP	18093	MQXs
TCP	TCP	18083	MQX
TCP	TCP	8883, 8090, 1883, 8093, 8094	EMQ
HTTPS	TCP	443	HTTPS
TCP	TCP	12000 a 12100	FTP PASIVO
SSH	TCP	22	SSH
TCP	TCP	21	FTP
MYSQL	TCP	3306	SQL

Una vez realizado todo este procedimiento, como punto final, se tiene la creación de llaves, las mismas que permiten conectarse por medio de SSH, lo cual es necesario para poder acceder a la instancia y poder configurar los demás parámetros, en la imagen que se muestra a continuación se puede verificar la creación de las llaves, es muy importante tener en cuenta que estas deben ser guardadas ya que Amazon permite la descarga por una sola vez.

Figura 61

Creación de las llaves.

Nombre del par de claves

El nombre puede incluir hasta 255 caracteres ASCII. No puede incluir espacios al principio ni al final.

Tipo de par de claves

RSA
Par de claves públicas y privadas cifradas por RSA

ED25519
Los pares de claves privadas y públicas cifradas ED25519 (no se admite para instancias de Windows)

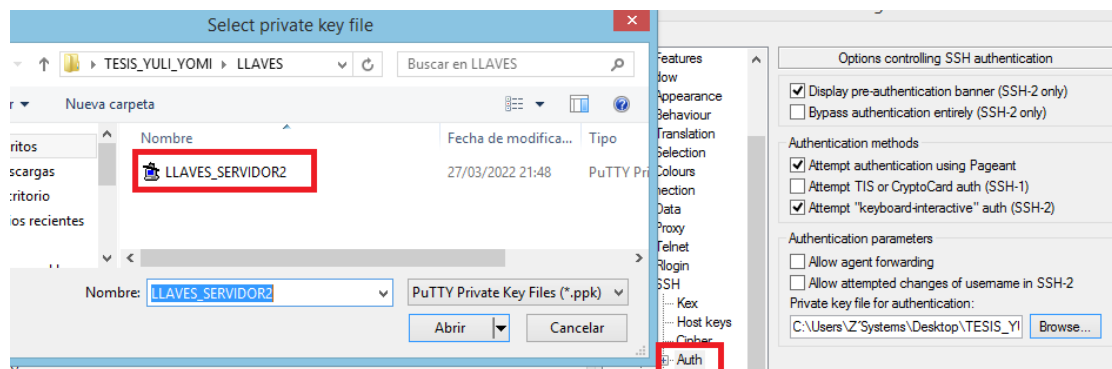
Formato de archivo de clave privada

.pem
Para usar con OpenSSH

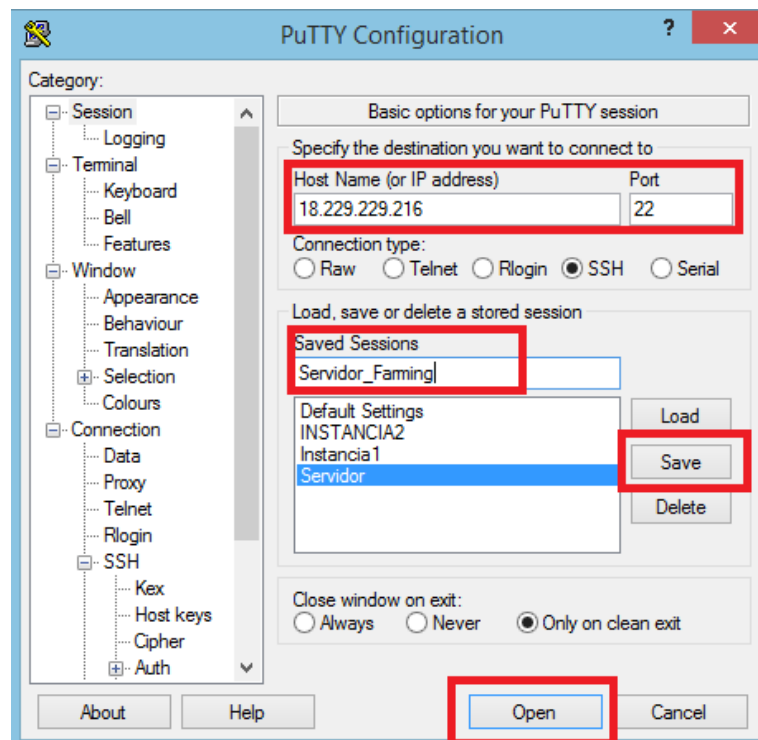
.ppk
Para usar con PuTTY

Cancelar

Una vez creada este par de claves se procede a acceder por medio de Putty a la instancia que se ha creado, este proceso se puede apreciar en la imagen que se tiene a continuación.

Figura 62*Elección de llaves.*

Nota. En la imagen se aprecia la selección de la llave para poder acceder a la instancia por medio del programa Putty a través de SSH.

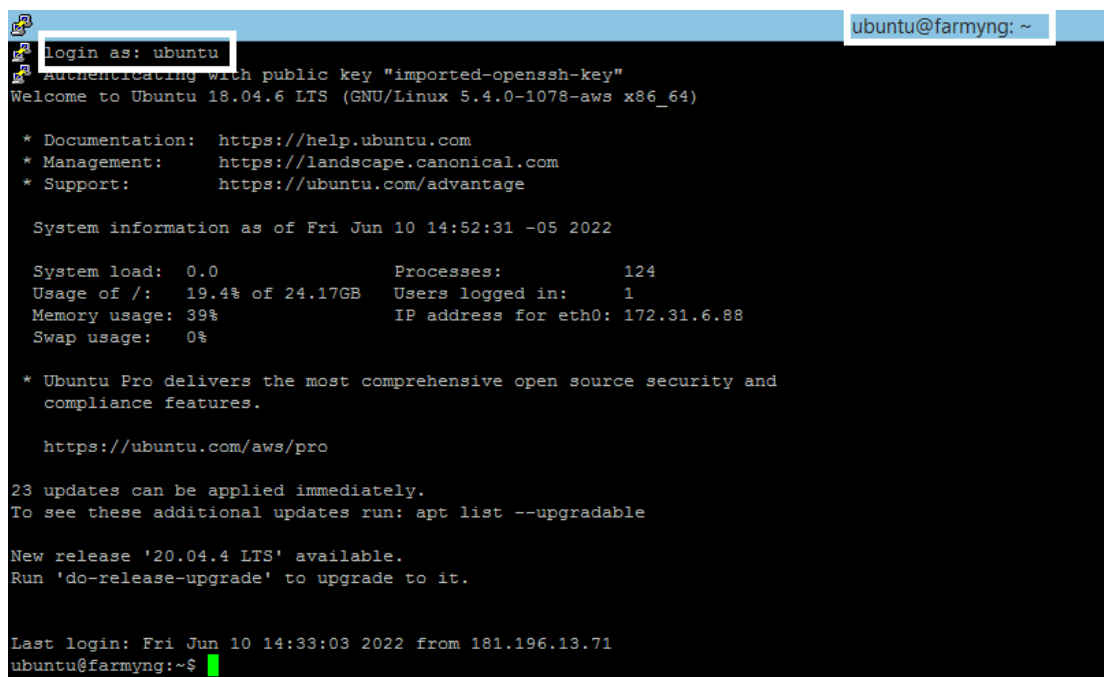
Figura 63*Colocar la dirección IP.*

Nota. En esta imagen se expone los pasos que se deben seguir una vez que se ha seleccionado las llaves, se coloca la dirección IP, luego se coloca en Save, que permite guardar la información para poder acceder de forma directa sin necesidad

de estar repitiendo el proceso de escoger las llaves y finalmente se da en la opción Open y aparece la ventana que se muestra en la siguiente figura.

Figura 64

Acceso a la instancia por medio de SSH.



```

login as: ubuntu
authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1078-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri Jun 10 14:52:31 -05 2022

System load:  0.0          Processes:      124
Usage of /:   19.4% of 24.17GB  Users logged in:  1
Memory usage: 39%          IP address for eth0: 172.31.6.88
Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

23 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '20.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Jun 10 14:33:03 2022 from 181.196.13.71
ubuntu@farmyng:~$

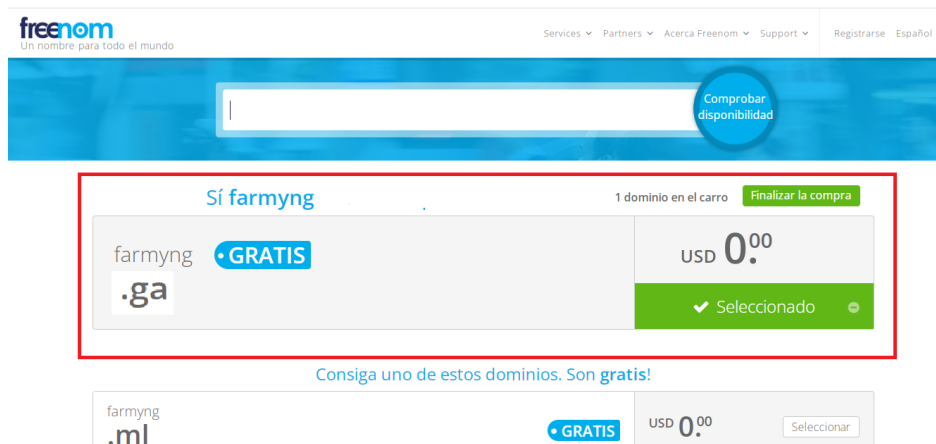
```

Nota. Como se puede apreciar en la imagen, se ha logrado acceder a la instancia, en login as: el usuario es “ubuntu” por defecto.

Una vez que se ha configurado la instancia y se ha verificado que es posible conectarse mediante SSH, es importante que se coloque un DNS o dominio ya que este permite acceder a una página web de forma más rápida, además, la dirección de una web se recuerde de mejor forma, ya que una cadena numérica es mucho menos identificable que un nombre de dominio. Teniendo claro esto, se accedió a la página: <https://www.freenom.com/es/freeandpaiddomains.html>, y como primer paso es colocar el nombre que se desee para el dominio y con ello verificar si está disponible, una vez cumplido este paso se procede a crear el dominio, para este caso el nombre escogido es “farmyng”.ga ya que es el que se pudo obtener de forma gratuita.

Figura 65

Creación de Dominio.



Una vez que se ha creado el dominio, es posible ir modificando y agregando el DNS a la IP que se desea, como se observa en la imagen siguiente.

Figura 66

Asignación de IP y dominio.

Escribe	TTL	Objetivo
A	3600	18.229.229.216
CNOMBRE	3600	farmyng.ga

Nota. Aquí se ha realizado la asignación correspondiente de cada parámetro en A se coloca la IP a la cual se desea asociar, mientras que en CNAME (CNOMBRE) se asocia el nombre de alias con un nombre de dominio auténtico, finalmente en TTL es el ajuste que fija el tiempo en que los datos permanecen válidos antes de que el router los borre, este valor es asignado por defecto, pero el valor límite que se puede agregar es de 86.400 segundos.

Otro factor importante en esta parte de configuración es el nivel de seguridad que tiene la instancia que ha sido creada, y como ya ha sido asociada a un dominio, este es necesario que cuenta con un certificado que le asegure al usuario que al

acceder a este sitio web no se verá comprometida su información, para lo cual se ha hecho uso del certificado SSL, este certificado permite la autenticación de la identificación de un sitio web y autoriza una conexión cifrada (Kaspersky, s.f.), evita que personas con malas intenciones modifiquen la información que se ha enviado por medio de dos sistemas.

Para esto se ha hecho la descarga de los certificados, accediendo a la instancia por medio de SSH y para verificar que ya se cuenta con SSL se puede apreciar la imagen siguiente.

Figura 67

Carpeta que contiene los certificados SSL.

```

root@farmyng:/home/ubuntu# exit
exit
ubuntu@farmyng:~$ ubuntu
ubuntu: command not found
ubuntu@farmyng:~$ sudo su
root@farmyng:/home/ubuntu# cd ..
root@farmyng:/home# cd admin
root@farmyng:/home/admin# cd conf
root@farmyng:/home/admin/conf# cd web
root@farmyng:/home/admin/conf/web# ls
farmyng.ga.apache2.conf      nginx.farmyng.ga.conf_letsencrypt
farmyng.ga.apache2.ssl.conf  ssl.farmyng.ga.ca
farmyng.ga.nginx.conf        ssl.farmyng.ga.crt
farmyng.ga.nginx.ssl.conf    ssl.farmyng.ga.key
nginx.farmyng.ga.conf_letsencrypt  ssl.farmyng.ga.pem
root@farmyng:/home/admin/conf/web#

```

Instalación Node.js

“Node.js es un entorno que permite la ejecución en multiplataforma, de código abierto, para la capa del servidor, basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google, este nodo fue diseñado para crear sitios web en tiempo real con función de inserción” (Capan, 2017)

. Uno de los puntos más fuertes de este node.js, es que es una aplicación natural para exponer la información de una base de datos, El almacenamiento de datos JSON permite que Node.js evite la conversión de datos.

Por tal beneficio se ha hecho uso de este node.js y se ha descargado los archivos necesarios para poder hacer uso del mismo, en la figura que se muestra a continuación se evidencia los archivos de descarga.

Figura 68

Archivos de descarga para instalar Node.js.

```

root@farmyng: /# sudo apt update
Hit:1 http://sa-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
root@farmyng: /# apt install nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
root@farmyng: /# node --version
v8.10.0
root@farmyng: /# apt install npm
Reading package lists... Done
Building dependency tree
Reading state information... Done
root@farmyng: /# npm -v
3.5.2

```

Nota. En la imagen se expone la Instalación y verificación de la versión del node.js.

Una vez que se ha instalado el node.js, es necesario instalar diferentes componentes, entre estos está el protocolo MQTT y MySQL que permitirán almacenar y enviar la información capturada por el nodo cliente.

Configuración del Bróker para MQTT

El bróker es considerado como un Middleware, este es un software que permite la interacción de diferentes aplicaciones, facilitando la conexión y la sincronización, existen diferentes middleware, pero para este caso se hará un enfoque al middleware que está orientado a la mensajería, siendo este, una infraestructura que está establecida en software o hardware y que soporta el envío y

recibo de mensajes entre sistemas, por lo tanto el bróker es un middleware orientado a la mensajería encargado de traducir los mensajes de los transmisores a los receptores. Una de las grandes ventajas de hacer uso del bróker es que permite la Asincronicidad, es decir que, “si la aplicación del receptor falla por alguna motivo, se acumulan en la cola de mensajes para su posterior procesamiento cuando el receptor se reinicie” (Fernández, 2017).

Al tener claro esto, se escoge el protocolo que permitirá el envío de la información, como bien se sabe el enfoque del proyecto es relacionado al IoT y quizás lo mejor sería hacer uso de HTTP, sin embargo, este protocolo es relativamente pesado, por lo cual se hace necesario mencionar a MQTT, siendo este un protocolo que permite la comunicación M2M, es decir máquina a máquina, en donde los dispositivos se comunican por medio de un patrón publicar/suscribir. “Este protocolo es muy usado para casos donde es necesario un consumo energético y ancho de banda muy bajo y es ideal para soluciones de IoT (Galván, s.f.).

Para entender la aplicación de este software se tiene las siguientes consideraciones:

Componentes y acciones clave de MQTT.

- Cliente: dispositivo capaz de difundir mensajes, realizar la suscripción para poder aceptar mensajes.
- Broker: recibe mensajes publicados por el cliente y los envía a los clientes que tienen una suscripción
- Publicar: cuando un cliente envía un mensaje al bróker.
- Tópico: los mensajes deben estar etiquetados con algún tópico o tema.

(Fernández, 2017).

Para la implementación específica del bróker MQTT es necesario escoger el servidor más adecuado, para lo cual se ha seleccionado EMQ, ya que su código es

abierto, además es ideal para el desarrollo de aplicaciones que se van a manejar en tiempo real, además de que es altamente escalable y permite la creación de un clúster con múltiples nodos, lo que facilita el manejo de miles de conexiones.

Pasos a seguir para poder instalar EMQX:

1. Instalar Dependencias.

```
$ sudo apt update && sudo apt install -y \  
apt-transport-https \  
ca-certificates \  
curl \  
gnupg-agent \  
software-properties-common
```

2. Actualizar el índice del paquete apt

```
$ sudo apt update
```

3. Instale la última versión de EMQ X

```
$ sudo apt install emqx  
$ sudo apt install emqx=3.1.0
```

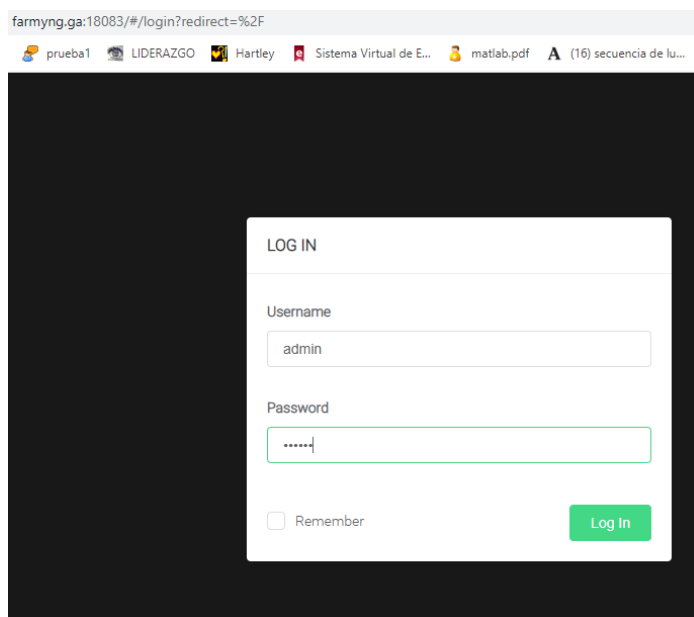
4. Iniciar EMQX

```
$ emqx start  
emqx 3.1.0 is started successfully!
```

Una vez realizada la instalación se accede al archivo que permite configurar ciertos parámetros, como son el puerto *listener.ws.external* que viene configurado por default en 8083 y se debe cambiar al 8093 y el puerto *management.listener.http* que vienen configurado en 8080 y se debe cambiar a 8090. Una vez realizado estos cambios ya se puede acceder desde la página web con el siguiente link: farmyng.ga:18083.

Figura 69

Acceso al bróker por medio de MQTT.



Nota. En la imagen se puede apreciar el acceso al bróker una vez que ha sido instalado en la instancia a través de SSH, el usuario y contraseña por defecto es: admin y public respectivamente.

Instalación de MySQL

Es necesario instalar una base de datos para poder registrar los datos que se toman con cada uno de los sensores, al igual que MQTT, esta base de datos ha sido instalado por comandos una vez que se ha conectado a la instancia por medio de SSH. En la figura siguiente se puede apreciar el proceso de instalación.

Figura 70

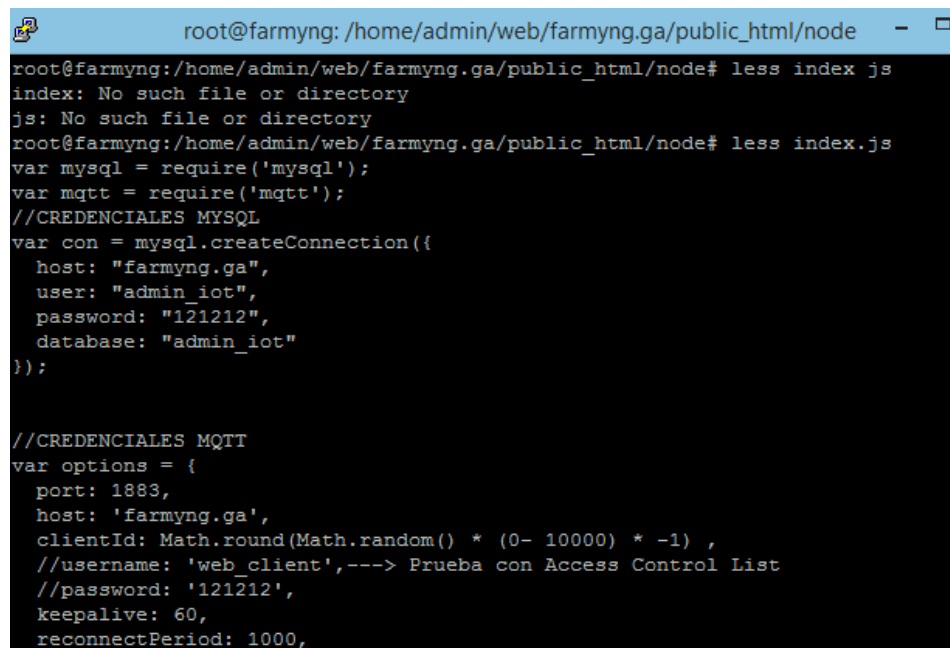
Instalación de mysql.

```
ubuntu@farmyng:~$ sudo su
root@farmyng:/home/ubuntu# cd ..
root@farmyng:/home# cd admin
root@farmyng:/home/admin# cd web
root@farmyng:/home/admin/web# cd farmyng.ga
root@farmyng:/home/admin/web/farmyng.ga# ls
cgi-bin document_errors logs private public_html public_html state
root@farmyng:/home/admin/web/farmyng.ga# cd public-html
bash: cd: public-html: No such file or directory
root@farmyng:/home/admin/web/farmyng.ga# cd public_html
root@farmyng:/home/admin/web/farmyng.ga/public_html# ls
css img index.html js lib node robots.txt
root@farmyng:/home/admin/web/farmyng.ga/public_html# cd node
root@farmyng:/home/admin/web/farmyng.ga/public_html/node# ls
distancia.js node_modules package-lock.json
package.js npm-shrinkwrap package.json
root@farmyng:/home/admin/web/farmyng.ga/public_html/node# npm install mysql
..... / idealTree: timing idealTree Completed in 526ms
```

Una vez que se han configurado los parámetros anteriores, es necesario modificar el archivo index.js, en este se configura la comunicación con el bróker mediante MQTT y la comunicación con mysql, en este archivo se encuentran las credenciales para cada uno, respectivamente, a continuación, se evidencia la configuración.

Figura 71

Configuración de credencial para la comunicación en tiempo real.



```
root@farmyng: /home/admin/web/farmyng.ga/public_html/node
root@farmyng:/home/admin/web/farmyng.ga/public_html/node# less index.js
index: No such file or directory
js: No such file or directory
root@farmyng:/home/admin/web/farmyng.ga/public_html/node# less index.js
var mysql = require('mysql');
var mqtt = require('mqtt');
//CREDENCIALES MYSQL
var con = mysql.createConnection({
  host: "farmyng.ga",
  user: "admin_iot",
  password: "121212",
  database: "admin_iot"
});

//CREDENCIALES MQTT
var options = {
  port: 1883,
  host: 'farmyng.ga',
  clientId: Math.round(Math.random() * (0- 10000) * -1) ,
  //username: 'web_client',---> Prueba con Access Control List
  //password: '121212',
  keepalive: 60,
  reconnectPeriod: 1000,
```

Finalizada la configuración de la credencial para conectarse con el bróker, es necesario que se realice la subscripción al tópic y para recibir la información enviada por el Gateway se usa "#", cuyo código completo se aprecia en el Anexo 1.

Es necesario que se realice el proceso de decodificación una vez que la información se ha recibido, el código presentado en Anexo 2, permite realizar el proceso mencionado.

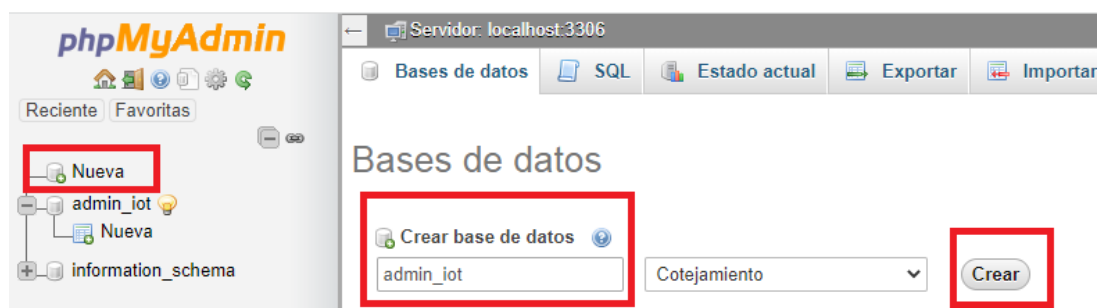
Creación de la Base de datos

La base de datos permite almacenar datos y además conectarles entre sí en una unidad lógica, por lo tanto, es un conjunto de datos estructurados que pertenecen a un mismo contexto y que tiene el objetivo de gestionar electrónicamente cantidades de información de gran magnitud.

Para crear una nueva base de datos dentro de phpMyAdmin, se da click en nueva y en la parte “Crear base de datos” se coloca el nombre que se le desee dar a esta base de datos, la imagen siguiente expone el proceso descrito.

Figura 72

Creación de la base de datos.



Nota. En la parte de cotejamiento se `utf8_spanish_ci`, que corresponde al español tradicional, y permite que se use la ñ y la ch y ll. Finalmente se pulsa en ‘Crear’ y el nombre de la base que se ha creado aparece en la columna de la izquierda.

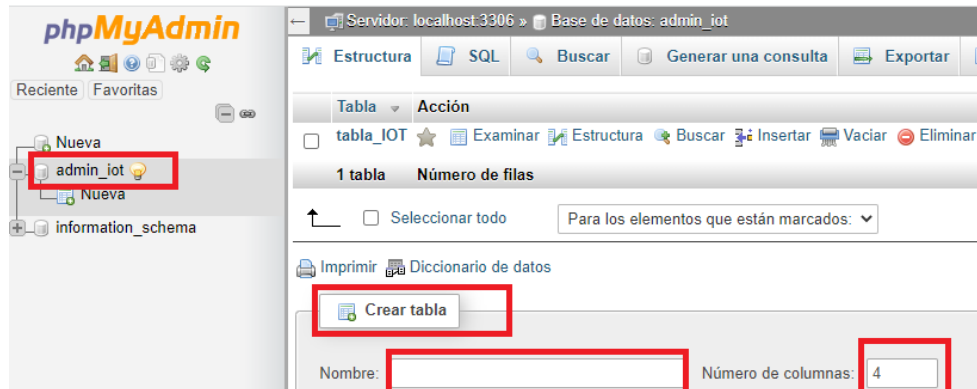
Una vez que ha sido creada la base de datos se procede a crear la tabla que permitirá almacenar los datos que provienen de los sensores, para lo cual, se debe hacer clic en el nombre de la base de datos creada y aquí se coloca el nombre de la tabla y el número de columnas que va a tener.

En las bases de datos se debe añadir un campo extra que debe almacenar un valor único y que permite identificar cada registro, generalmente se le denomina `id`, entonces, la tabla creada cuenta con 11 columnas, mismas que almacenan la

información de id, collar, latitud, longitud, velocidad, distancia, temperatura, bpm, snr, rssi y la fecha_hora.

Figura 73

Creación de la tabla.



Nota. En la imagen se expone la creación de la tabla, una vez realizado este proceso se procede a escribir los nombres y el tipo de dato de cada uno de los campos que tendrá la tabla “tabla_IOT”.

Finalmente, en la imagen siguiente se aprecia la tabla con sus respectivas columnas y tipo de dato.

Figura 74

Tabla de datos.

	id	collar	latitud	longitud	velocidad	posicion	temperatura	bpm	snr	rssi	fecha_hora
<input type="checkbox"/>	16	0	0.5646	-77.7974	0.07	10.7093	18.97	377	0	-157	2022-05-12 12:47:16
<input type="checkbox"/>	29	0	0.5646	-77.7974	0.04	10.6141	19.17	379	0	-157	2022-05-12 12:47:16
<input type="checkbox"/>	27	0	0.5646	-77.7974	0.04	780.459	19.19	381	0	-157	2022-05-12 12:47:16
<input type="checkbox"/>	26	0	0.5646	-77.7974	0.02	10.9837	19.29	380	0	-157	2022-05-12 12:47:16

Desarrollo de la interfaz de visualización en Ubidots

Una vez que se han configurado todos los parámetros para la comunicación y el envío de datos, se necesita que los datos que se han almacenado en la base de datos se puedan visualizar de una forma más dinámica, para lo cual, se ha seleccionado la plataforma de Ubidots, esta tiene una relación directa con AWS IoT Core, este es un beneficio de AWS, para lo cual facilitan el envío de datos de sensores mediante complementos, así AWS IoT Core es un intermediario para enrutar mensajes entre sus dispositivos.

Para poder realizar esta integración, es necesario que, en AWS se creen los siguientes recursos:

- Crear una política, por lo tanto, se dirige a la consola de administración de IAM y luego se da clic en la opción “Políticas” y se selecciona la opción de crear, se continúa con la pestaña de JSON para copiar el código del Anexo 3.
Finalmente, en la pestaña “Revisar”, se coloca el nombre de la política y finalmente clic en “crear”.

Figura 75

Crear política.

The screenshot displays the AWS IAM console interface for creating a new policy. The main heading is 'Políticas (952)' with a sub-heading 'Una política es un objeto en AWS que define permisos.' A red box highlights the 'Acciones' dropdown menu, which includes the 'Crear política' button. Below this, the 'Editor visual' and 'JSON' tabs are shown, with the 'JSON' tab selected. The JSON editor contains the following policy document:

```

{
  "Statement": [
    {
      "Sid": "Visualizador",
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:CreateRole",
        "iam:PutRolePolicy",
        "iam:ListRolePolicies"
      ],
      "Resource": "arn:aws:iam::4:0430:169c:role/Ubido*"
    }
  ]
}

```

The 'Revisar la política' section shows the 'Nombre' field set to 'ubidots_policy' and a 'Resumen' table with the following data:

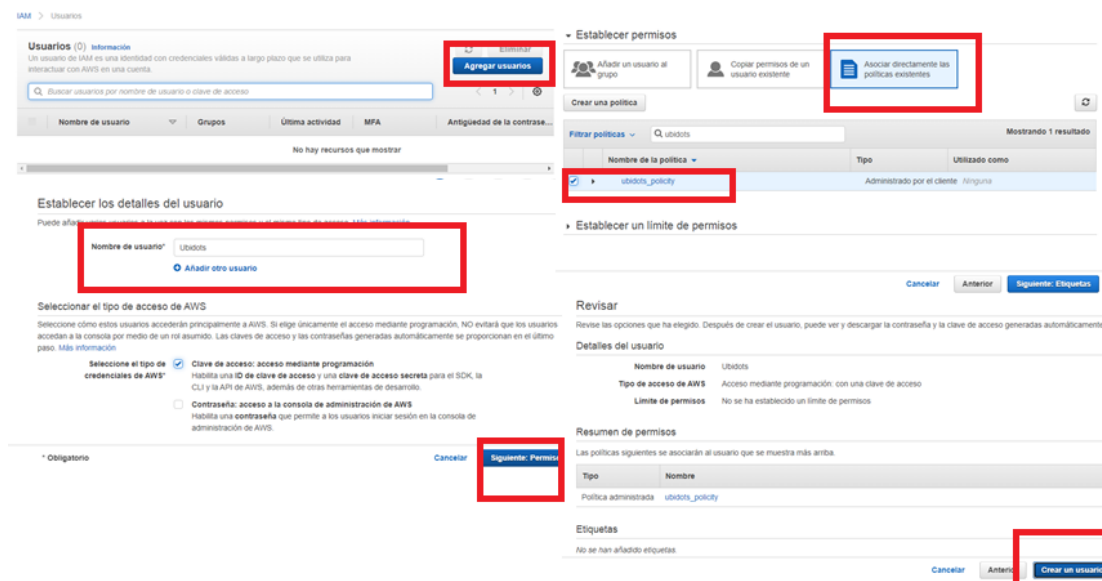
Servicio	Nivel de acceso	Recurso	Condición de solicitud
IAM	Limitado: Enumeración, Lectura, Escritura, Administración de permisos	RoleName string like Ubido*	Ninguna

Nota. La imagen expone los pasos a seguir para la creación de la política.

- Luego se hace clic en "Agregar usuario", se ingresa un nombre para el usuario, se selecciona la opción "Acceso programático", se hace clic en "Adjuntar políticas existentes", y se selecciona la que se creó con anterioridad, hacer clic en "Crear usuario", finalmente se debe descargar las claves con el botón Descargar .csv que se usan para el complemento con Ubido.

Figura 76

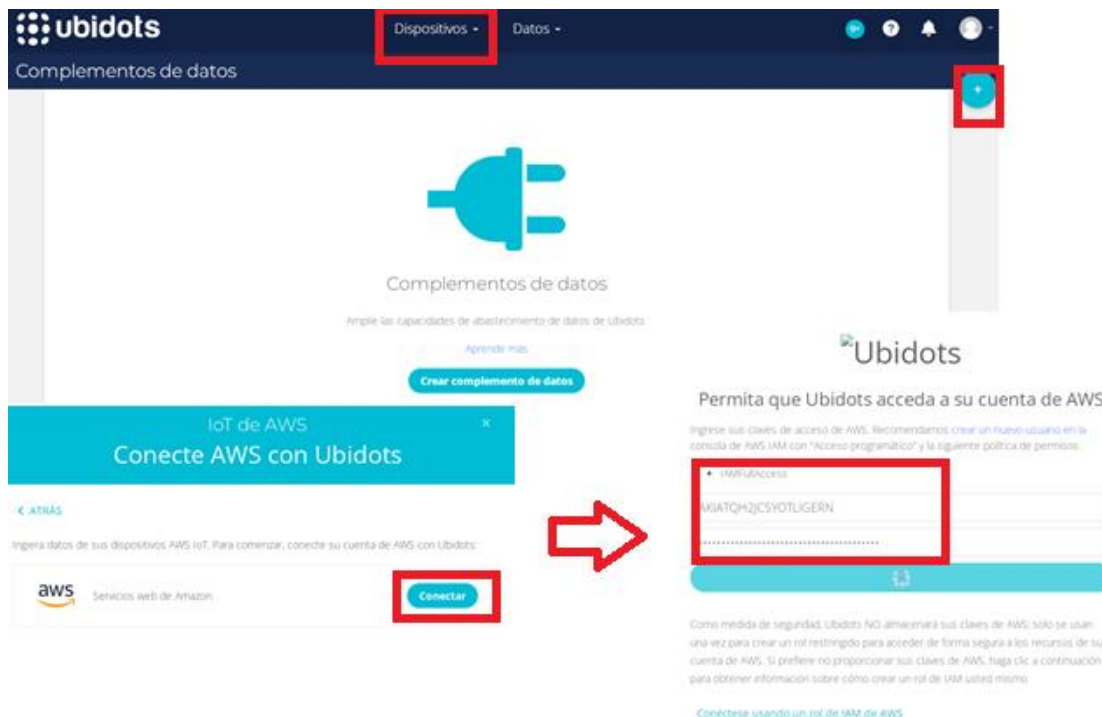
Creación de claves para conexión con Ubidots.



- Crear un complemento de AWS IoT Core, este permite decodificar las cargas útiles JSON de la regla HTTPS de AWS, para lo cual es necesario tener una cuenta en Ubidots, una vez que se ha ingresado a Ubidots, se va a la parte Dispositivos ", hacer" Complementos ", luego haga clic en el ícono " + " para crear un nuevo complemento, se busca el complemento de AWS IoT Core, haga clic en él y aparecerá una ventana de autenticación. Ingrese su ID de clave de acceso de AWS y su clave de acceso secreta creada anteriormente AWS IoT core.

Figura 77

Creación del complemento en Ubidots.



Una vez realizadas estas configuraciones, se redirige nuevamente a complementos, se puede verificar que el complemento ha sido creado y se procede a seleccionar para que se conecte con AWS, luego de esto se obtienen los datos por parte de los sensores, que posteriormente son publicados y aquí AWS IoT Core se encarga de crear una regla de forma automática que permite enrutar el tráfico hacia Ubidots, cuando esto ha sucedido, se crean recursos directamente desde Ubidots, en el cual es necesario colocar la región en donde se encuentra y aparece el formato de envío de datos, para este caso se tiene una carga útil JSON, que permite extraer la etiqueta variable, la etiqueta dispositivo Ubidots y la etiqueta valor de la variable.

Finalmente, cuando se han publicado los datos, Ubidots crea inmediatamente el Devices en donde se encuentra cada una de las variables medidas con sus respectivos valores, como se muestra a continuación.

Figura 78

VARIABLES ASIGNADAS EN UBILOTS.

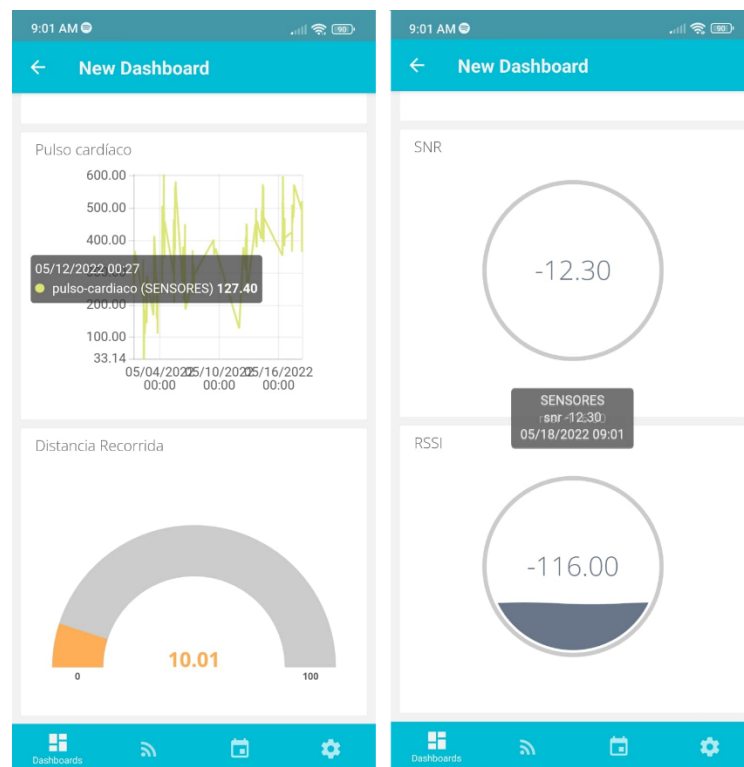
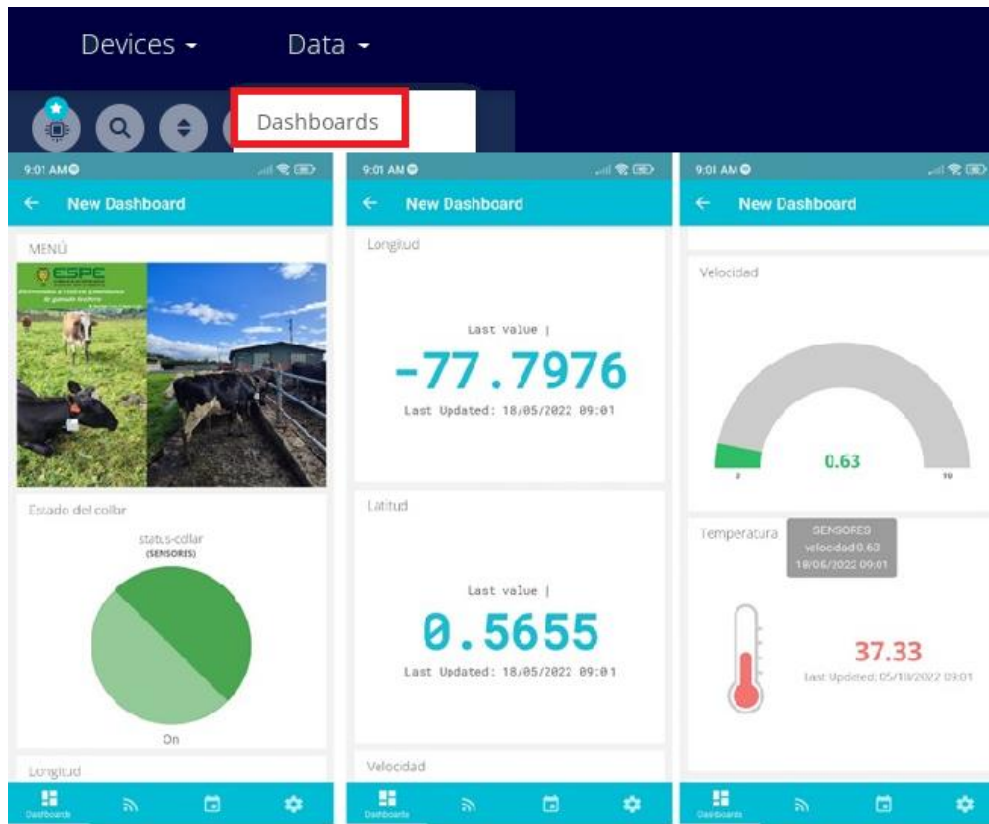
The screenshot shows the 'Devices' page in Ubidots. A device named 'SENSORES' is selected, and its details are shown on the left. The main area displays a grid of variable cards for the device. The variables and their values are:

Variable	Value	Last Activity
latitud	0.59	2 days ago
longitud	-77.83	2 days ago
posicion	10.29	2 days ago
pulso ranfaro	572.00	2 days ago
msi	-157.00	2 days ago
snr	0.00	2 days ago
temperatura	18.95	2 days ago
velocidad	0.31	2 days ago

Para poder tener una mejor visualización de los datos, se dirige a la pestaña Data en el cual se selecciona Dashboard, y en esta ventana es posible agregar íconos que se vinculan con cada una de las variables, esto permite ver los datos en forma gráfica, se selecciona el rango de tiempo que se desea mirar y finalmente la interfaz de visualización de datos se muestra en la imagen que está a continuación.

Figura 79

Interfaz de visualización en Ubidots.



Nota. En la imagen se expone la interfaz que se tiene para la visualización de cada uno de los datos, es necesario mencionar que Ubidots se puede descargar como aplicación para teléfono celular, por lo tanto, se puede tener los datos tanto en el computador como el teléfono.

Prototipo Final

En la Figura 80 se muestra el prototipo final, mismo que cuenta con una carcasa que fue impresa en 3D, el material impide que los elementos que se encuentran en el interior sufran daños por ciertos factores como lluvia, el polvo y otros factores que se encuentran en el ambiente y de un collar de tela para que no provoque ninguna llaga por uso frecuente del mismo. Además, como se ha mencionado que el dispositivo no debe ser invasivo para el animal también es necesario conocer el peso de todo el prototipo final, este valor se muestra en la figura 81, con lo cual se evidencia que no provocará estrés en el animal ya que pesa 401g.

Figura 80

Imagen del prototipo final que se colocó a la vaca.



Nota. En el gráfico se puede apreciar el prototipo final, este consta de una caja impresa en 3D, además de un collar de tela y de los demás elementos que se han mencionado con anterioridad.

Figura 81

Peso del dispositivo final.



Nota. La figura expone el peso del dispositivo final que se coloca al animal.

Costo de Implementación del prototipo de collar

Los equipos que presentaron convenientes en su adquisición fueron los dispositivos LoRa (Gateway LG01-N + Módulo de comunicación Shield GPS). Pero fueron adquiridos en Amazon y el resto de dispositivos por mercado libre. En la Tabla 12, se detalla el costo de implementación del prototipo.

Tabla 12

Elementos y costos del prototipo de collar.

Detalle	Cantidad	Precio Unitario	Precio Total
MLX90614 sensor infrarrojo	1	14,50	14,50
Arduino Uno Genérico	1	16,00	16,00

Detalle	Cantidad	Precio Unitario	Precio Total
Sensor ritmo cardíaco	1	4,50	4,50
Sensor contacto Magnético	1	2,00	2,00
GPS UBLOX NEO 2019	1	18,50	18,50
Batería LIPO 7.4V-1A	1	15,25	15,25
Cargador	1	17,00	17,00
Batería LIPO 7.4V-2.5^a	1	34,00	32,00
Panel Solar 5.5V-160mA	1	9,50	9,50
Cable USB para arduino	1	2,50	2,50
Dragino LORA Shield	1	30,00	30,00
Dragino LORA Gateway	1	130,00	130,00
Collar	1	4,50	4,50
Transporte para realizar pruebas	-	65,00	65,00
Cable de timbre	2m	2,00	2,00
Diseño e impresión de la caja	1	85,50	85,50
Placa final y soldadura	1	22,50	22,50
Otros		86,00	86,50
		Total	557,75

Nota. Esta tabla muestra el precio de cada uno de los elementos y dispositivos que se usaron para la construcción del prototipo, en “otros” el valor hace referencia a componentes adicionales que se compraron, debido a que se tuvo inconvenientes en las pruebas iniciales y se quemaron elementos, además se debe aclarar que el panel solar se lo usó para las primeras tomas de datos, sin embargo se tuvo problemas con la batería, ya que al tener variaciones de voltaje sufrió daños y por tal motivo se decidió eliminar este elemento del prototipo.

Cálculo del área de propagación

Para poder calcular el alcance teórico que va a tener este enlace, se hizo uso del modelo de propagación Hata- Extendido ya que cubre un rango de frecuencias más amplio y es perfecto para la propagación en áreas suburbanas o rurales, por lo tanto, se parte de la ecuación siguiente:

$$L(dB) = 46.3 + 33.9 * \log_{10}(f) - 13.82 * \log_{10}(h_b) - a(h_m) + (44.9 - 6.55 * \log_{10}(h_b)) * \log_{10}(d) + C_m \quad (1)$$

En donde:

- f : es la frecuencia de operación (MHz).
- h_b : es la altura de la antena Gateway respecto del suelo (metros).
- h_m : es la altura de la antena de la Shield respecto del suelo (metros).
- $a(h_m)$: es el factor de corrección de altura de la antena de la estación móvil (nodo final) y que para áreas rurales o suburbanas es el siguiente valor: $(1.1 * \log_{10}(f) - 0.7)h_m - (1.56 * \log_{10}(f) - 0.8)$
- C_m : es la compensación constante que para este caso es de 0dB

Los Datos que se tienen para resolver la ecuación son los siguientes:

- $f = 915MHz$
- $h_b = 2.55m$
- $h_m = 1.37m$

Se procede a reemplazar estos valores en la ecuación.

$$\begin{aligned} L(dB) &= 46.3 + 33.9 * \log_{10}(915) - 13.82 * \log_{10}(2.55) \\ &- (1.1 * \log_{10}(915) - 0.7)(1.37) - (1.56 * \log_{10}(915) - 0.8) \\ &+ (44.9 - 6.55 * \log_{10}(2.55)) * \log_{10}(d) \end{aligned}$$

$$L(dB) = 46.3 + 100.39 - 5.62 - 0.316 + 42.237 * \log_{10}(d)$$

$$L(dB) = 140.754 + 42.237 * \log_{10}(d) \quad (2)$$

La distancia es el dato que necesitamos conocer para tener el alcance de la comunicación, por lo tanto se procede a calcular el valor de pérdidas $L(dB)$ con la siguiente ecuación:

$$L(dB) = P_{Tx} - P_{Rx} + G_{AT} + G_{AR} - L_{conectores} \quad (3)$$

En dónde P_{Tx} es la potencia de transmisión y P_{Rx} es la potencia de recepción mientras que G_{AT} y G_{AR} son las ganancias de las antenas de transmisión y de recepción respectivamente y finalmente $L_{conectores}$ es la perdida que se provoca por causa de los conectores.

Para poder calcular el valor de la potencia de recepción se tiene la ecuación (4) mostrada a continuación:

$$P_{Rx} = S(dBm) + MD \quad (4)$$

$$S(dBm) = -174 + 10 \log(BW) + SNR + NF \quad (5)$$

De acuerdo a los parámetros que se han configurado en la parte de Dragino, el ancho de banda que se utiliza para la frecuencia de 915 MHz es de 125 kHz, la relación señal ruido para un factor de propagación de 7 es de -7.5 dB y la figura de ruido (NF) para *LoRa* es de 6 dB, estos valores se reemplazan en la ecuación 5 y se procede a encontrar el valor de la sensibilidad.

$$S(dBm) = -174 + 10 \log(125) - 7.5 + 6$$

$$S(dBm) = -154.531 \text{ dBm}$$

Además, se sabe que la potencia de transmisión para este caso es de 20 dBm, mientras que las antenas tanto para transmisión y recepción tienen una ganancia de 2 dBi, se considera una pérdida por cables en todo el prototipo de 1.5 dB con un MD de 30 dB. Es necesario aclarar que en el terreno que se coloca el prototipo final es un espacio abierto, sin embargo, si existen pérdidas por obstáculos, en este caso la existencia de árboles y las casas que se encuentran a su alrededor, así se añade 15 dB que es el valor correspondiente a pérdidas por los diferentes obstáculos. Con estos datos se procede a calcular los interrogantes faltantes.

$$P_{Rx} = -154.531 + 30$$

$$P_{Rx} = -124.53 \text{ dB}$$

$$L(\text{dB}) = 20 + 124.53 + 4 - 1.5 - 15$$

$$L(\text{dB}) = 132.03 \text{ dB}$$

$$132.03 = 140.754 + 42.237 * \log_{10}(d)$$

$$\frac{-8.724}{42.237} = * \log_{10}(d)$$

$$d = 0.622 \text{ km}$$

$$A = \pi d^2$$

$$A = \pi(0.622)^2$$

$$A = 1.215 \text{ km}^2$$

Por tanto, de forma teórica la distancia máxima a la cual se debe ubicar el nodo final debe ser a 622 metros de distancia con respecto al Gateway y permite cubrir una superficie de 1.215 km cuadrados.

Capítulo VI

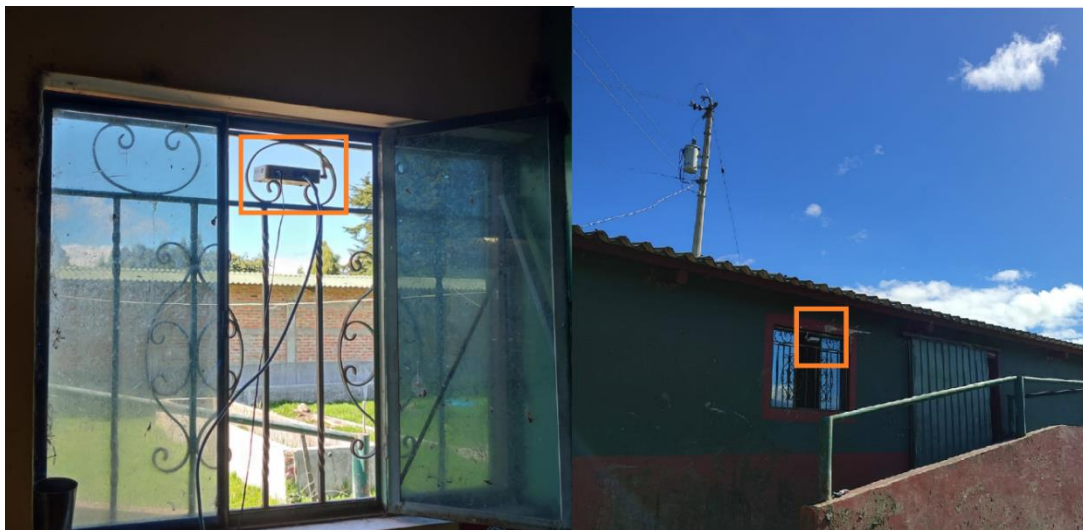
Pruebas y Resultados

Pruebas de alcance

En la Figura 82 que se expone a continuación se visualiza el lugar y la altura respecto al suelo con la que está ubicado el Gateway. En la Tabla 13 se indica sus coordenadas y las de los puntos de límite del área de prueba.

Figura 82

Ubicación del Gateway.



Nota. En la imagen se aprecia la ubicación que tiene el nodo Gateway o estación base.

Tabla 13

Coordenadas del Gateway y puntos de cercado del área de prueba.

Coordenadas	Gateway	Punto A	Punto B	Punto C	Punto D
Latitud	0.5645235	0.5652	0.5673	0.5647	0.5626
Longitud	-77.7975418	-77.7996	-77.7975	-77.7944	-77.7972

Las distancias entre el Gateway y los puntos de cercado se observan en la Figura 82. De esta manera se tiene que la distancia entre el punto C que es el más alejado y el Gateway es de 345 m.

Figura 83

Gráfica de la ubicación del Gateway y puntos de cercado.



Para poder realizar las pruebas de alcance, primero se realizó el cálculo respectivo de acuerdo a los datos que se tienen y de acuerdo a la ubicación en el cual se colocó el Gateway y el nodo cliente, una vez determinada la distancia teórica

SNR	RSSI
7	-112
7	-112
7	-112
7	-112
7	-112
7	-112
7	-112
7	-112
7	-112
7	-112
7	-112
7	-112
7	-112
7	-112
-12.5	-106
-12.5	-157
-12.5	-157
-12.5	-157
-12.5	-157
-12.5	-157

En esta tabla hay diferentes valores de RSSI, “este valor indica la intensidad de la señal recibida, es decir, la potencia de la señal recibida en mW y se mide en dBm y es un valor negativo. Este valor se puede usar como una medida para verificar si el receptor puede "escuchar bien" una señal de un remitente” (MOBILEFISH.com, s.f.). Mientras más cercano a cero sea este valor la señal es más fuerte. Los valores más frecuentes para este parámetro son -30dBm que indica que la señal es fuerte y -120dBm que indica que la señal es débil, teniendo en cuenta estos dos límites y haciendo referencia a la tabla se puede decir que la distancia máxima a la cual se debe colocar el nodo cliente es a 300m con respecto al Gateway, ya que en esa distancia el valor de RSSI es de -157dBm, a esta distancia la señal se empieza a debilitar y con respecto a SNR este parámetro indica

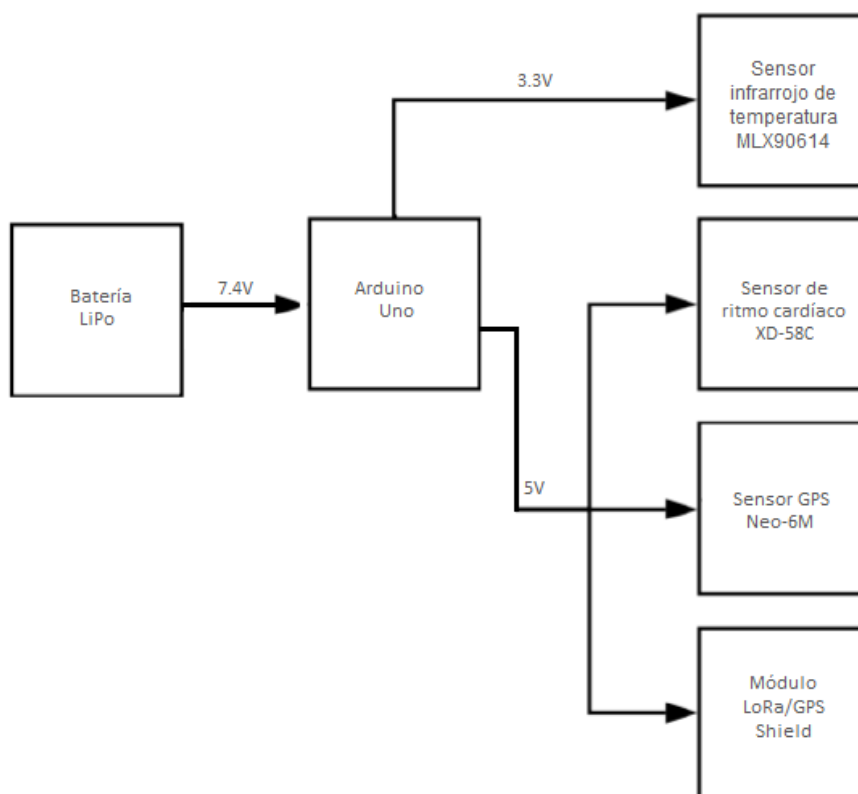
la relación entre la señal de potencia recibida y el nivel de potencia del piso de ruido que hace referencia al área en donde se encuentran las fuentes de señales de interferencia no deseadas mismas que dañan a la señal transmitida, para LoRa se tiene estos rangos : -20dB y +10dB y un valor más cercano a +10dB significa que la señal que se ha recibido está menos corrupta. Como se puede visualizar en la tabla los valores de SNR se acercan a 10dB, por lo tanto, quiere decir que la señal que se ha enviado no ha sido tan afectada por las señales interferentes que se encuentran en el medio.

Consumo energético

Antes de realizar el cálculo del consumo de corriente del prototipo, se muestra un bosquejo del requerimiento energético del sistema.

Figura 84

Diagrama de bloques de la alimentación.



El voltaje máximo requerido por los elementos que integran el nodo es 5 VDC, ya que, el principal objetivo para este proyecto es un dispositivo de bajo consumo y considerando que al Arduino se lo puede alimentar en la entrada con 7-12V, se utilizó como fuente de alimentación una batería de litio de 2 celdas que posee un voltaje de 7.4 V y una tasa de descarga de 2600 mA. En la Tabla 15, se muestra el consumo de corriente de cada componente y el consumo total del prototipo para su funcionamiento.

Tabla 15

Consumo teórico de los componentes del nodo.

Componente	Consumo de corriente DC
Arduino UNO	50 mA
Módulo LoRa Shield	25 mA
Sensor de temperatura MLX 90614	2 mA
Sensor de ritmo cardíaco XD-58C	4 mA
Módulo GPS NEO6-M	37 mA
Total	118 mA

De la tabla anterior se obtuvo que el consumo total es de 118 mA a 5V, con esta información y la mencionada anteriormente se procede hacer el cálculo para saber el tiempo que tardara en descargarse la batería recargable, en base a la siguiente fórmula:

$$t = \frac{P_{almacenada-bateria}}{P_{de-consumo}}$$

Conociendo que la potencia está dada por:

$$P = V * I$$

Entonces,

$$t = \frac{7.4V * 2600mA}{5V * 118mA}$$

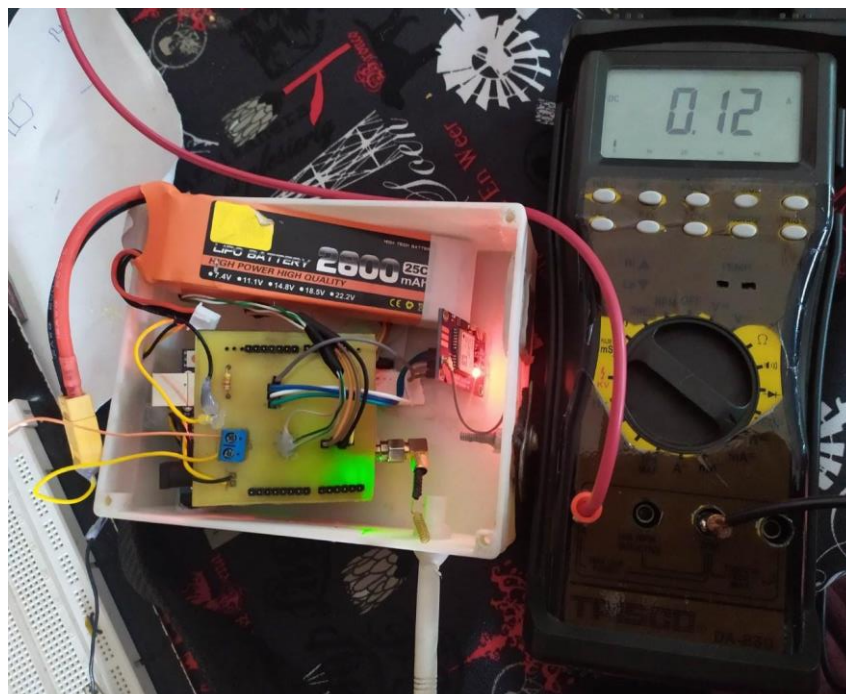
$$t = \frac{19.24}{0.59} = 32.61 h$$

Por lo tanto, la batería puede estar 32.61 horas en funcionamiento sin ser recargada, pero se recomienda no descargar la batería más allá del 50% para mantener su vida útil, dejando 16.305 horas de útil. Suficiente considerando que se buscaba un funcionamiento autónomo de 8 horas aproximadamente para realizar las pruebas.

Ahora se procedió a corroborar el valor de consumo energético con el arrojado por un amperímetro, como se muestra en la figura a continuación es de 120 mA similar al del teórico de 118 mA con un error del 1.69%.

Figura 85

Consumo de corriente del nodo arrojado por un amperímetro.

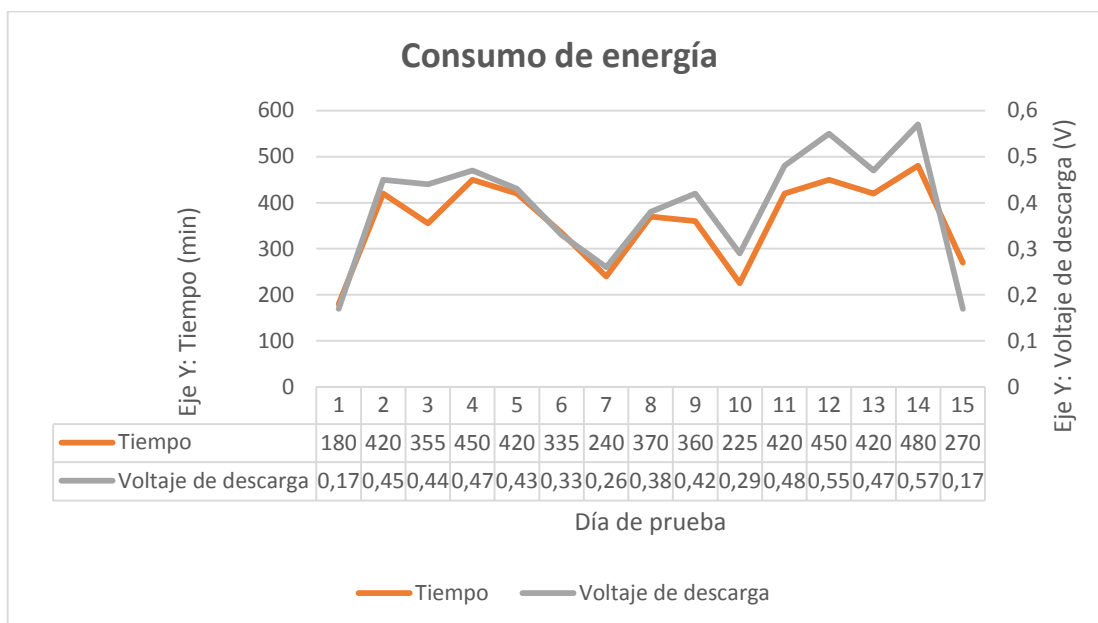


También se muestreo la batería de forma constante tomando el valor al finalizar el período de monitorización a diario, los resultados se pueden observan en

la siguiente Figura 86 con lo que se aprecia que es proporcional a mayor tiempo de uso del dispositivo mayor es el voltaje de descarga de la batería LiPo.

Figura 86

Gráfica del consumo energético del nodo cliente.



Logrando el objetivo de tener un dispositivo de bajo consumo energético, cabe mencionar que las baterías LiPo no se deben descargar por debajo de 3.5V por celda en este caso utilizarla hasta antes que llegue a 7V, ya que esto le podría causar un daño irreparable. En las pruebas de campo la batería el valor más bajo al cual se descargó fue 7.69V en aproximadamente 8 horas.

Pruebas de funcionamiento

En esta etapa, para la validación del funcionamiento del sistema de rastreo y monitoreo, se dividió en tres partes: integración de sensores al nodo, sistema de comunicación LoRa y seguridad del prototipo de collar. Dado que el éxito del sistema en la mayoría está centrado en la transmisión y recepción LoRa, se realizaron dos pruebas específicas: conexión Gateway LoRa – Servidor AWS y conexión Servidor AWS – Ubidots con el objetivo de corroborar la conexión, identificar errores y eficiencia.

Validación de la integración de sensores al nodo

Para la validación de la integración de sensores al nodo se verifica que los datos de cada uno de los sensores estén llegando correctamente, se procederá a conectar la tarjeta Arduino del nodo cliente al computador por cable USB. Luego se ejecuta el programa de la comunicación LoRa en el IDE de Arduino y a través de monitor serie se observa los datos que recogen los tres sensores (temperatura, ritmo cardíaco y GPS), en la siguiente imagen se puede comprobar esto.

Figura 87

Verificación de la toma de datos por cada uno de los sensores.

```

COM3
Enviar
El monitoreo de la vaca Nua ha empezado
---- COMUNICACIÓN LISTA----
---- TRANSMISIÓN A TRAVÉS DE LoRa LISTA
---- INFORMACIÓN ENVIADA CORRECTAMENTE:
1
*****
Adquiriendo Datos GPS:
-----
Lat/Long(float): 0.59044, -77.82849 Fijar: 637ms.
Alt(cm): 285510 Course(10^-2 deg): 9836 Speed(10^-2 knots): 95
Alt(float): 2855.10
Velocidad: (mps): 0.49 (kmph): 1.76
Temp. ambiente => 23.45°C
Temp. objeto => 20.15°C
♥ A HeartBeat Happened !
BPM: 689
COLLAR CONECTADO --> Estatus:1
0.00
-157
 Autoscroll  Mostrar marca temporal
Nueva línea 9600 baudio Limpiar salida

```

Validación del sistema de comunicación LoRa

Habiendo ya validado la integración de sensores al nodo, se procederá a verificar el enlace del sistema de comunicación LoRa, para ello se colocó el Gateway en un lugar alto y con línea de vista, considerando donde pasaría pastando la vaca en los próximos días. El payload que llega al gateway tiene formato hexadecimal el cual es decodificado basado en ASCII posteriormente este conjunto de datos será seccionado para tener una lectura legible de los parámetros de los sensores,

además es necesario decodificarlos en este formato para poder condicionarlos con otras API e interactuar con otras plataformas IoT como lo es Ubidots.

Figura 88

Recepción de datos en el Gateway.

```

Previous Log:
Mon May 9 11:46:33 2022 user.notice root: [IoT.MQTT]:pub_qos[-q]: 0
Mon May 9 11:46:33 2022 user.notice root: [IoT.MQTT]:cafile[--cafile]:
Mon May 9 11:46:34 2022 user.notice root: [IoT.MQTT]:cert[--cert]: /etc/iot/cert/
Mon May 9 11:46:34 2022 user.notice root: [IoT.MQTT]:key[--key]: /etc/iot/cert/
Mon May 9 11:46:34 2022 user.notice root: [IoT.MQTT]:clientID[-i]: dragino_gateway
Mon May 9 11:46:34 2022 user.notice root: [IoT.MQTT]:remote_id: 4567
Mon May 9 11:46:34 2022 user.notice root: [IoT.MQTT]:pub_topic[-t]: sensores
Mon May 9 11:46:34 2022 user.notice root: [IoT.MQTT]:decoder: not set
Mon May 9 11:46:34 2022 user.notice root: [IoT.MQTT]:mqtt_data[-m]: 1.0,0.5655,-77.7968, 0.06,0.02,37.01,355.0,-11.8,-117.0
Mon May 9 11:46:34 2022 user.notice root: [IoT.MQTT]:
Mon May 9 11:46:43 2022 user.notice iot_keep_alive: Internet Access OK: via wlan0-2
Mon May 9 11:46:43 2022 user.notice iot_keep_alive: use WAN or WiFi for internet access now
Mon May 9 11:46:47 2022 daemon.info [e01_nkr_fm0[24734]:
Mon May 9 11:46:47 2022 daemon.info [e01_pkt_fm0[24734]: RNTX-
Receive(HEX):3c343536373e312e302c302e353635352c2d37372e373936382c20302e30302c302e30322c33372e31332c3335362e302c2d31312e302c2d3131372
e30
Mon May 9 11:46:47 2022 daemon.info mqtt_process.sh[1614]: /var/iot/channels/ CREATE 4567
Mon May 9 11:46:47 2022 user.notice root: [IoT.MQTT]: Find Match Entry for 4567
Mon May 9 11:46:48 2022 user.notice root: [IoT.MQTT]: DECODER 4567
Mon May 9 11:46:50 2022 user.notice root: [IoT.MQTT]:
Mon May 9 11:46:50 2022 user.notice root: [IoT.MQTT]:-----

```

Nota. En la imagen se aprecia la recepción de datos de cada sensor en formato HEX, que luego se lo procesa para obtener los datos en formato decimal.

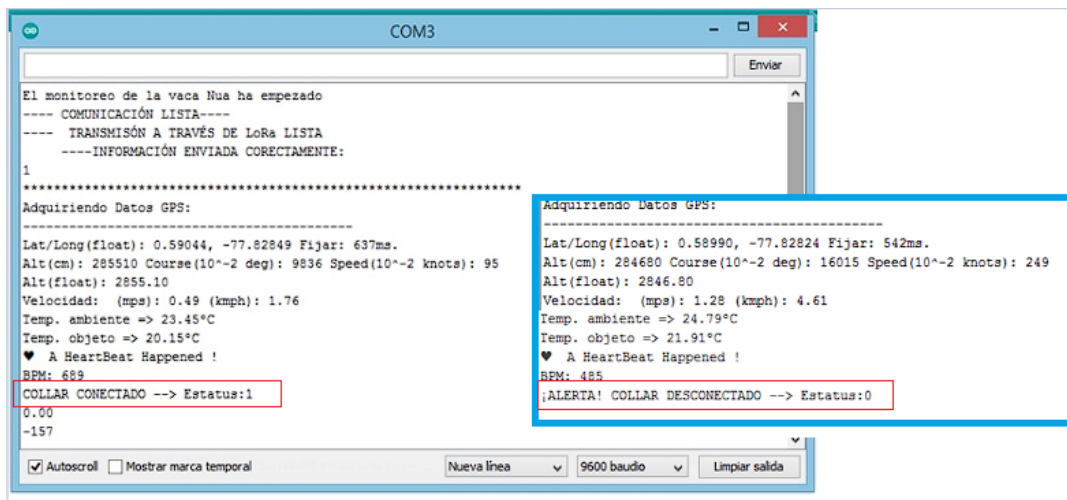
Una vez verificado que los datos se envíen desde el nodo cliente al Gateway, se confirma que la comunicación LoRa ha sido exitosa, por lo tanto, los datos ya son almacenados en la base de datos.

Validación de seguridad del prototipo de collar

Con la finalidad de brindar mayor seguridad al nodo se lo equipó con un sensor magnético el cual se colocó de extremo a extremo, para controlar si el collar ha sido cortado o sacado. Al realizar esta prueba se pretende validar los dos cambios de estado (1=collar conectado, 0=collar desconectado) y el envío de alarma de seguridad al entorno de Ubidots.

Figura 89

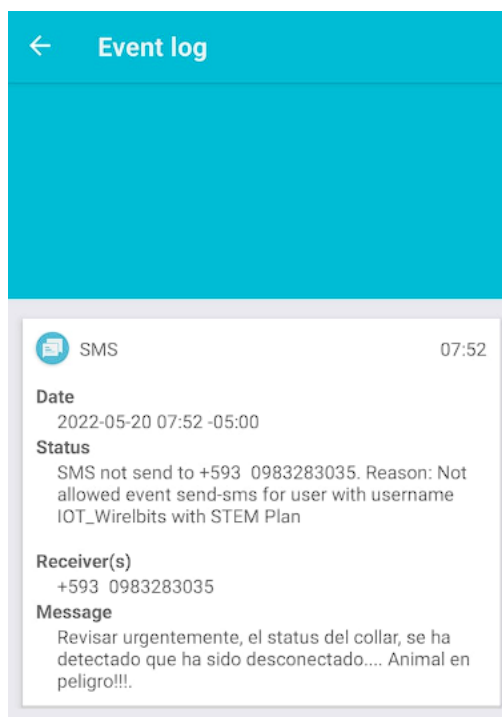
Visualización de cambios de estado de seguridad por el monitor serie de Arduino.



Nota. La imagen permite la supervisión por la plataforma Arduino si el prototipo de collar ha sufrido una violación de seguridad.

Figura 90

Visualización del mensaje de alerta en la plataforma Ubidots.



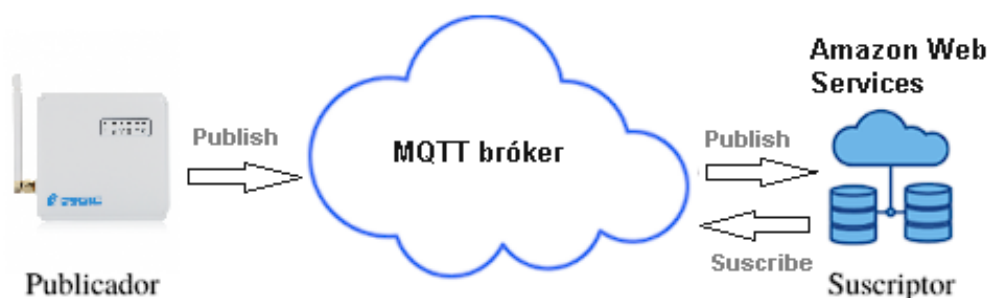
Nota. Realizado en Ubidots.

Pruebas de conexión Gateway LoRa al Servidor AWS

En apartados anteriores se ha mencionado que para enviar la información del gateway al servidor se hace uso del protocolo MQTT. Este protocolo está basado en la comunicación M2M (machine-to-machine), se ejecuta sobre TCP/IP utilizando un patrón publicador/suscriptor (pub-sub) y un servidor central llamado bróker. Los clientes (pub-sub) no se comunican directamente entre sí, sino que intercambian datos por medio del bróker. Un cliente puede publicar un mensaje en un determinado topic o puede suscribirse a este topic, y recibir la información del topic suscripto.

Figura 91

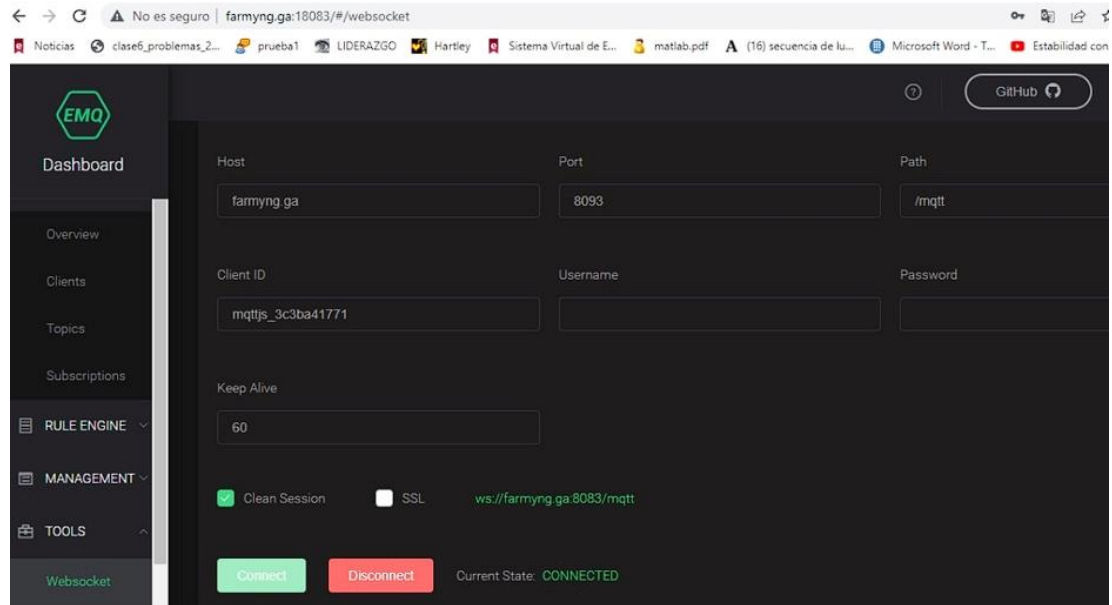
Estructura de la conexión entre el gateway y AWS.



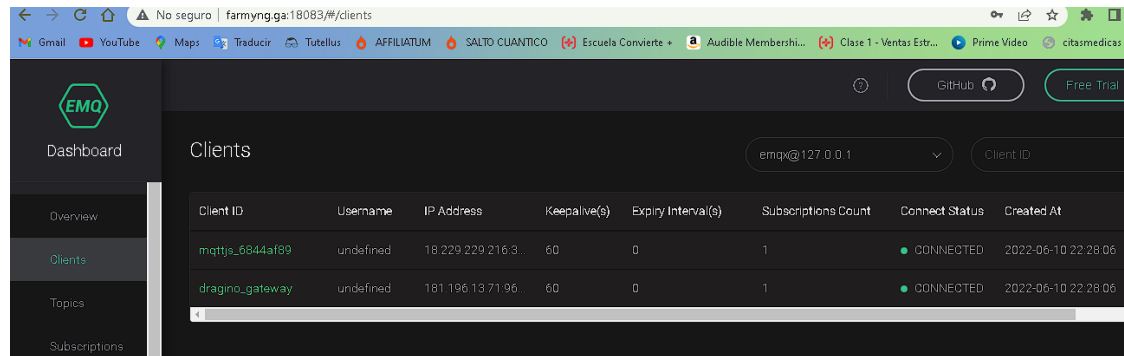
Se accede al bróker por medio de la dirección <http://farmyng.ga:18083/>, se ratifica que el gateway sea un cliente, seguido se dirige a la opción de WebSocket para verificar que los mensajes lleguen al bróker bajo el tópico definido.

Figura 92

Parámetros de configuración del Websocket en el bróker.

**Figura 93**

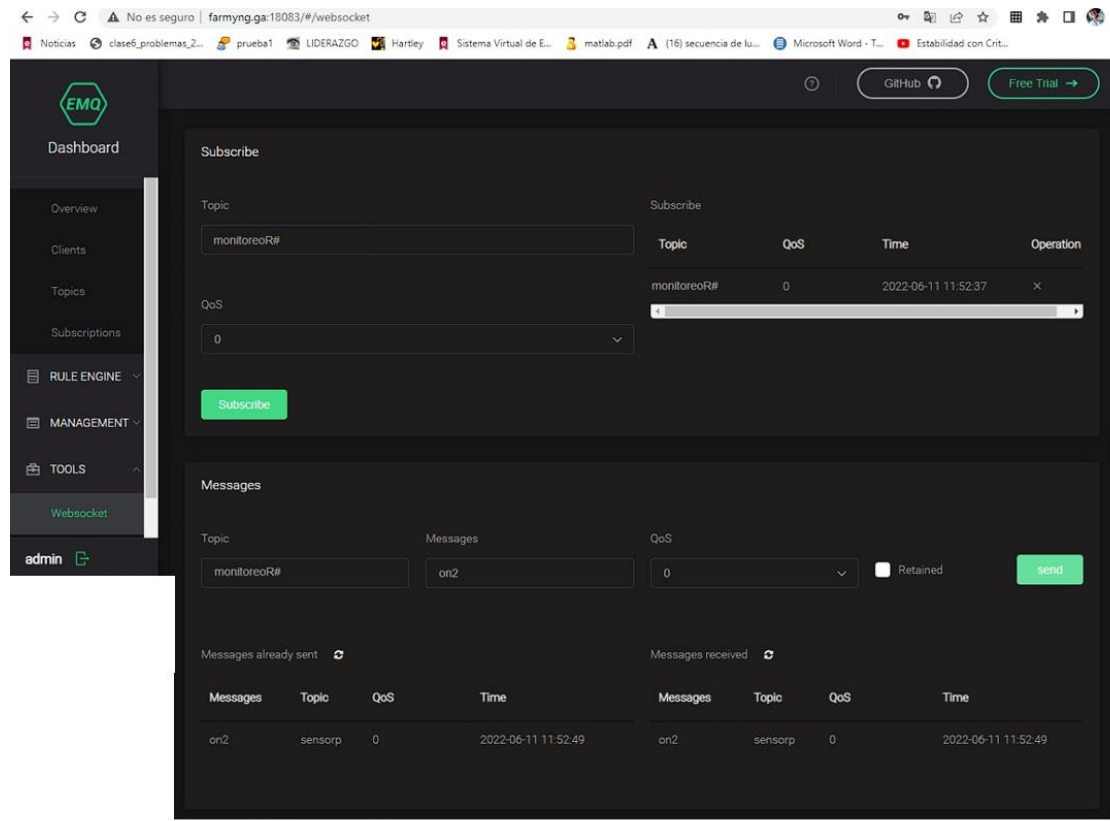
Ratificación del cliente publicador en el bróker.



Nota. Realizado por los autores.

Figura 94

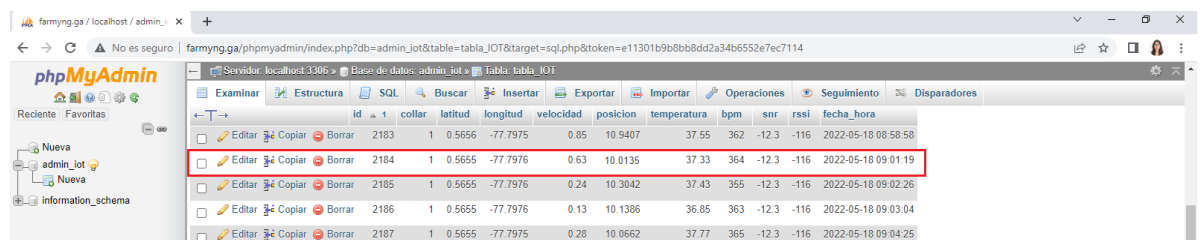
Llegada de los datos del gateway al bróker.



A continuación, se dirige a la base de datos alojada en AWS de Amazon por medio de la dirección <http://farmyng.ga/phpmyadmin/>, las credenciales de acceso son las siguientes: usuario – admin_iot y contraseña – 121212. Se selecciona admin_iot → taba_IOT para visualizar la base de datos.

Figura 95

Llegada de datos a la base de datos en Amazon.



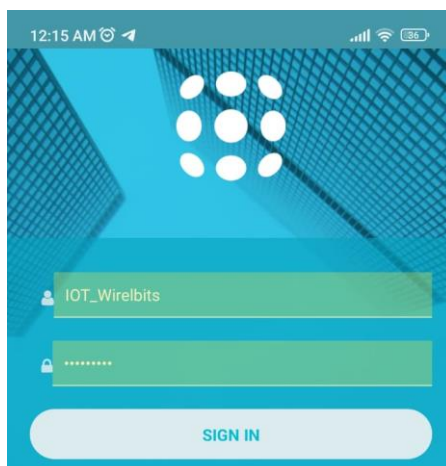
Pruebas de conexión Servidor – Ubidots

Una de las ventajas de la plataforma IoT de Ubidots son sus herramientas visuales, entre estas Ubidots explorer, que es una aplicación para smartphone. De esta forma el usuario podrá acceder a la información directamente desde la versión móvil. Se descarga la app desde Play Store y se accede con las siguientes credenciales.

- Usuario: IOT_Wirelbits
- Contraseña: 121212asd

Figura 96

Ingreso a la aplicación de Ubidots.



Para validar la conexión entre el servidor y la interfaz de usuario se procede a comparar los datos que se encuentran almacenados en la base de datos alojada en AWS con los datos que están siendo mostrados en el Dashboard de Ubidots en tiempo real.

Resultados y análisis de la información obtenida

Los resultados obtenidos de las variables en análisis de posición, velocidad, temperatura, ritmo cardíaco y distancia recorrida corresponden a la monitorización de la vaca durante un periodo de 15 días con envío de datos cada 2 minutos, entre las 6:40 y las 19:40 h. En la figura 97, se muestra como se colocó el dispositivo en el

animal, el cual está diseñado a prueba de lluvias, debido a que la toma de datos se realizó en el mes de mayo el cual tiene un promedio de 27.8 días de lluvia según el estudio de prevención meteorológica y clima San Gabriel, Ecuador de Weather Atlas. Siendo así las condiciones climáticas para los días de prueba lluvia y ligeramente soleado. Cabe mencionar que el dispositivo cuenta con un interruptor de encendido/apagado.

Figura 97

Disposición del dispositivo collar en la vaca Nua.



Para el caso de la temperatura corporal del bovino, se puede observar en las Figuras 98 y 99 su variación a lo largo de los días de prueba. Esta oscila entre 36.71°C y 37.11°C, registrando picos máximos de 39.09, 39.45, 39.47, 39.63 y 39.83°C y mínimos de 35.01, 35.13 y 35.27°C. Aunque se han presentado valores fuera del rango normal de temperatura de un bovino adulto sano, la temperatura registrada por el sensor infrarrojo MLX90615 alcanzó una media de 37,02°C cuyo valor no está lejos del valor real típico de 38.5°C. Se obtiene un error relativo del 3.84%.

$$\%error_{temp} = \frac{t_{esperada} - t_{obtenida}}{t_{esperada}} * 100\%$$

$$\%error_{temp} = \frac{38.5 - 37.02}{38.5} * 100\%$$

$$\%error_{temp} = 3.84\%$$

Por otra parte, cuando se observa una disminución en los valores es debido a que el sensor capta tanto la temperatura corporal del bovino como la del ambiente, por lo que, si el dispositivo llega a separarse del cuello de la vaca se ve una variación, ya que las pruebas son en un ambiente no controlado, la propia convivencia con el resto del ganado afecta la correcta posición del dispositivo porque entre estas se suelen lamer o refregar. Cabe mencionar que durante las horas evaluadas se observó que cuando la vaca realiza el proceso de rumia que generalmente lo hace acostada y pocas veces de pie, la temperatura presenta una ligera disminución y durante la tarde se evidencia los picos de las temperaturas más altas.

Figura 98

Temperatura superficial de la vaca Nua en la Semana 1.

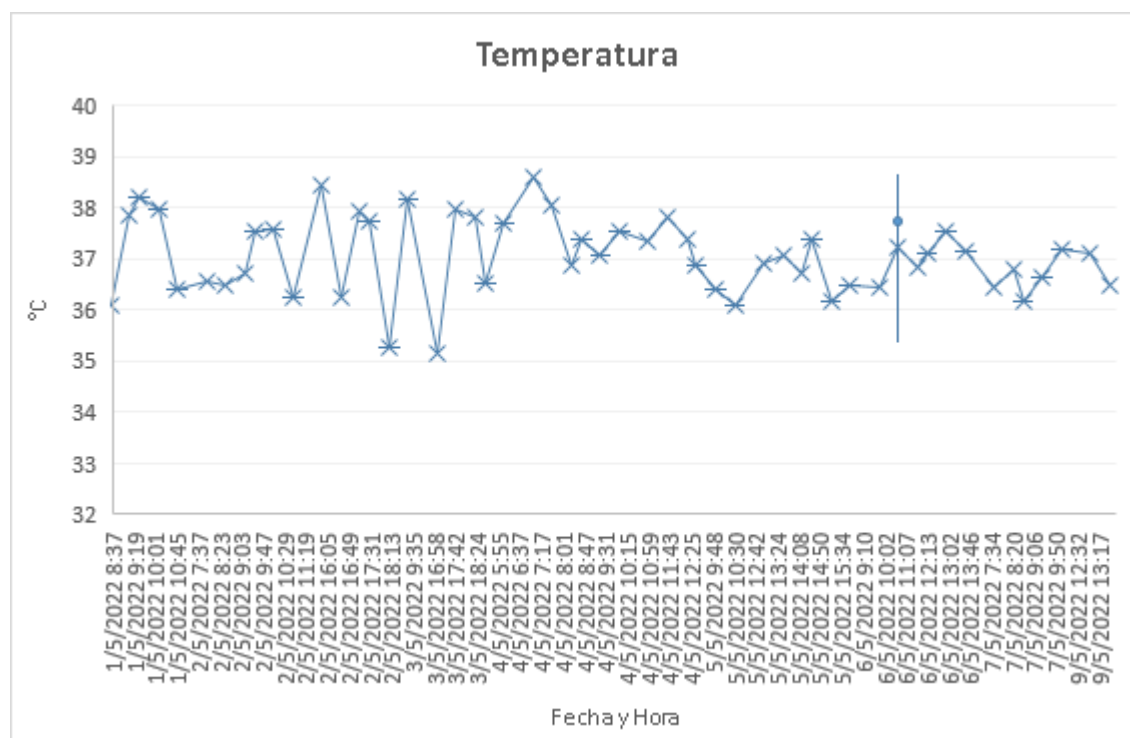
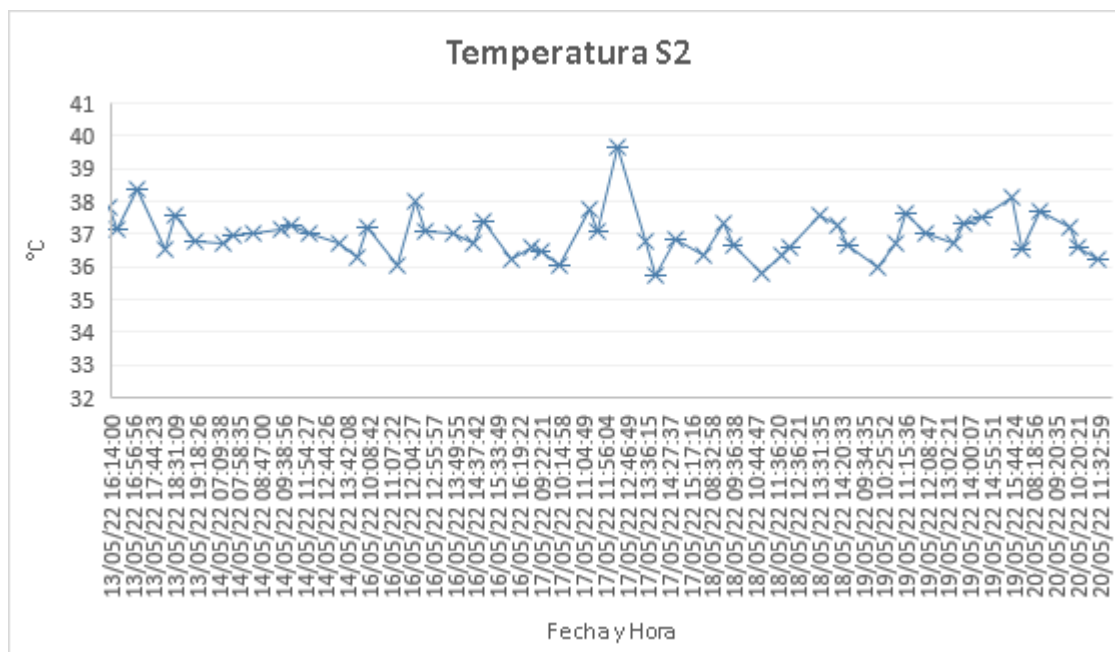


Figura 99

Temperatura superficial de la vaca Nua en la Semana 2.



Para el caso del ritmo cardíaco o pulso del bovino es importante mencionar que los datos obtenidos por el sensor XD-85C arroja los valores sin procesar, es decir sin acondicionamiento del sensor, por lo tanto, para obtener los pulsos por minutos (bpm) es necesario multiplicar por un factor de acondicionamiento de 0.125, ya que el tiempo de muestreo es de 2 min y los bpm se miden cada 15 seg. En las Figuras 100 y 101 se puede observar el ritmo cardíaco se mantiene entre 46 bpm y 57 bpm, estos valores son los esperados teniendo en cuenta el rango normal está entre 40bpm y 80 bpm, además presenta una media de 57 bpm cuyo valor para este caso tampoco está alejado del valor real típico de 60 bpm. Se obtiene un error relativo del 5%.

$$\%error_{bpm} = \frac{BPM_{esperado} - BPM_{obtenido}}{BPM_{esperado}} * 100\%$$

$$\%error_{temp} = \frac{60 - 57}{60} * 100\%$$

$$\%error_{temp} = 5\%$$

El ritmo cardíaco presentó un pico máximo de 80 bpm mientras que su pico mínimo fue de 44 bpm, esto se debe al mismo motivo de la temperatura. Ya que la vaca mueve el cuello para actividades como comer y beber agua en el bebedero gira su cabeza lo que ocasiona que los sensores pierdan contacto directo con la piel del animal. Pero debido a su porcentaje de error en media se puede decir que los valores obtenidos han sido acertados.

Figura 100

Ritmo cardíaco de la vaca Nua en la Semana 1.

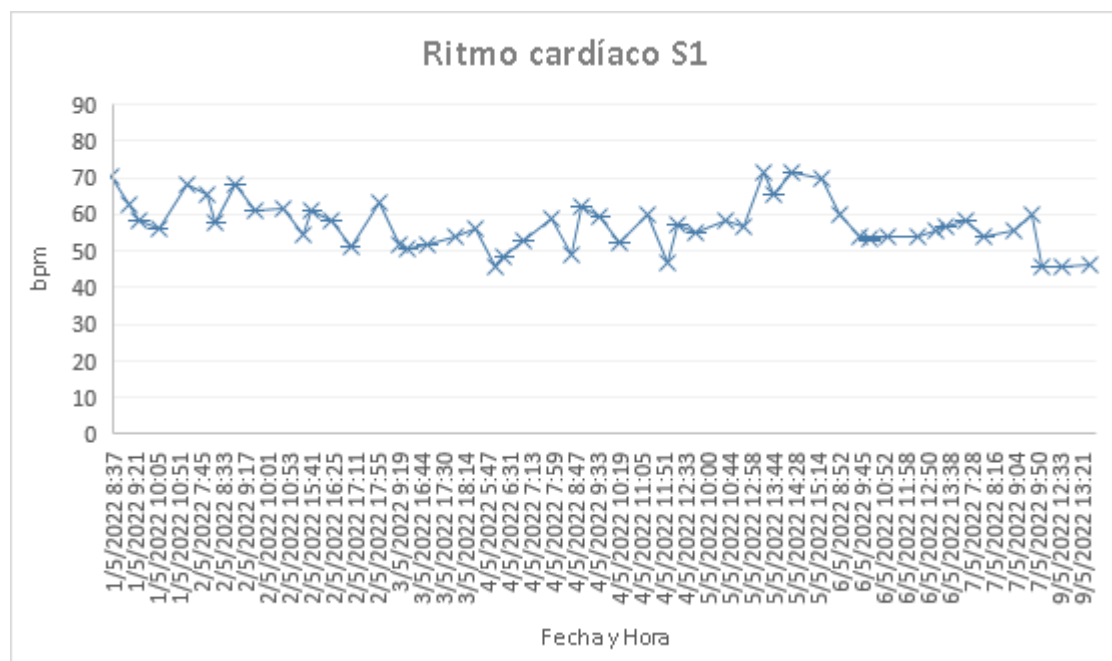
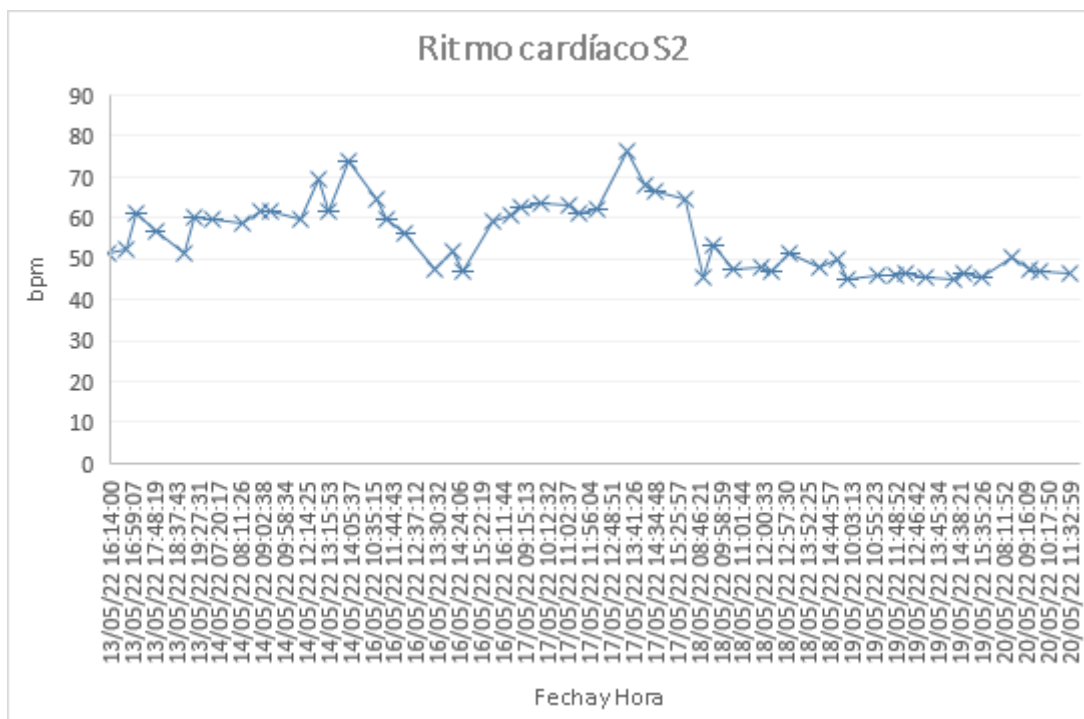


Figura 101

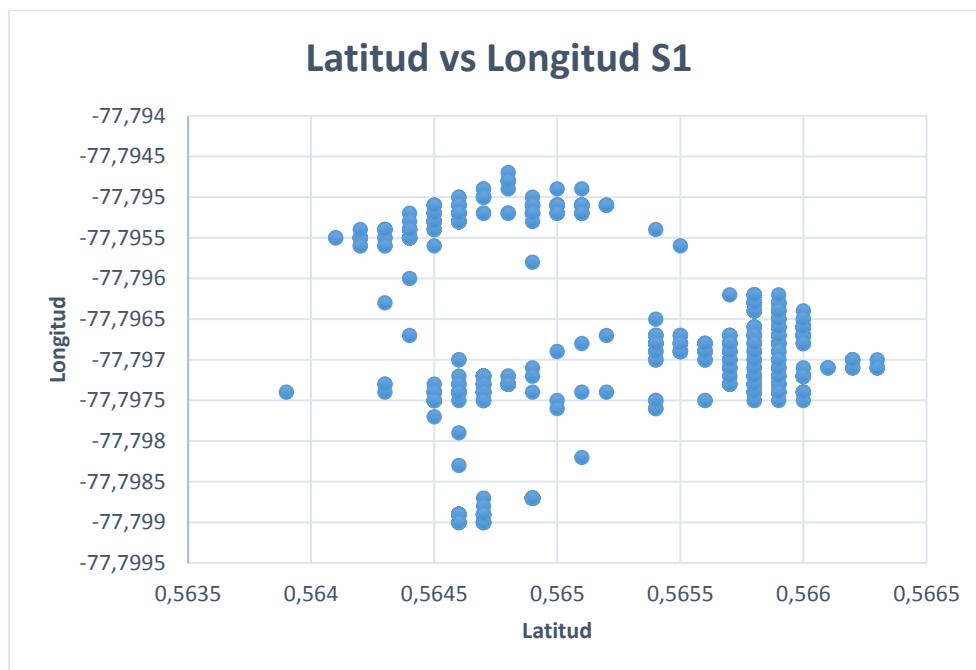
Ritmo cardíaco de la vaca Nua en la Semana 2.



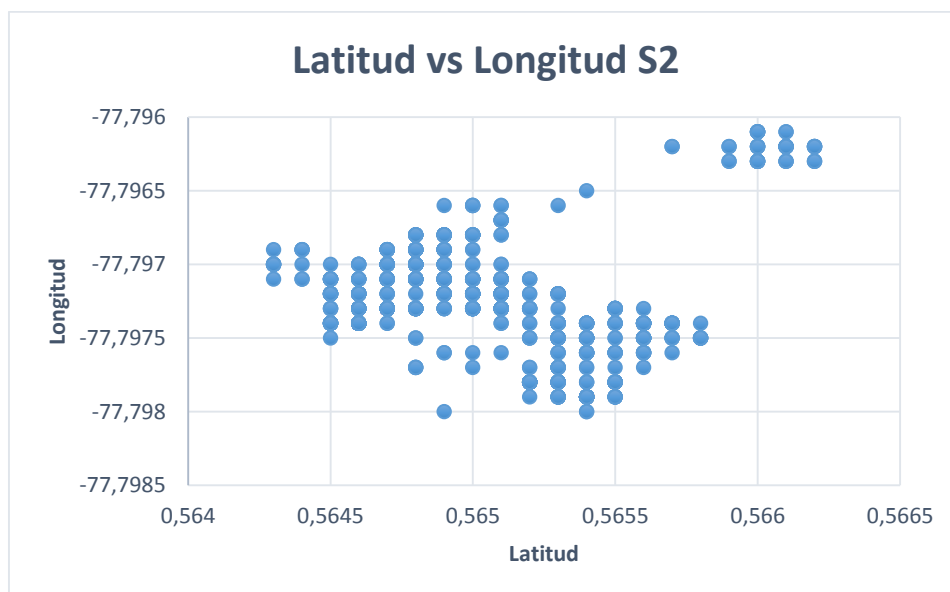
Se puede observar en las Figuras 102 y 103, los datos adquiridos mediante el dispositivo con el sensor GPS NEO-6M, el cual posee una precisión para la posición de 2.5 m pero hay que tener en cuenta que la precisión dependerá en gran medida de la cantidad de satélites que se encuentren cerca para adquirir la señal. Al realizar el análisis de todas las coordenadas de latitud/longitud obtenidas se ha calculado su media dando un valor promedio de 0.5651807/-77.79701. También se puede apreciar que los datos varían a partir de las milésimas, por ello se realizará el cálculo de su desviación estándar dando como resultado 0.000525/0.0007711, se puede decir que la vaca no recorre grandes distancias.

Figura 102

Gráfica de geo-localización de la vaca Nua en la Semana 1.

**Figura 103**

Gráfica de geo-localización de la vaca Nua en la Semana 2.



Para una mejor visualización se hace uso de la herramienta de Google Maps para representar todos los datos obtenidos, en base a esto se puede ver que el recorrido del animal siempre se ha encontrado dentro de los límites establecidos, como se muestra en la Figura 104.

Figura 104

Gráfica de los datos obtenidos de geo-localización en Google Maps.



Este sensor también proporcionó los datos de la velocidad del animal, en las Figuras 105 y 106 se presenta su variación la cual oscila entre 0.02 km/h y 0.06 km/h, alcanzó un pico máximo 4.83 km/h este valor alto se debe, a que se la persiguió en un periodo como objetivo de prueba y mientras que su pico mínimo fue de 0 km/h, que es cuando el animal está en reposo. Por otro lado, se puede

observar variaciones que oscilan entre 1.22 km/h y 2.98 km/h, que se da cuando la vaca se desplaza al bebedero de agua o al ordeño. Estos valores son los esperados puesto que las velocidades de vacas lecheras varían entre 2 a 5 km/h.

Figura 105

Velocidad de la vaca Nua en la Semana 1.

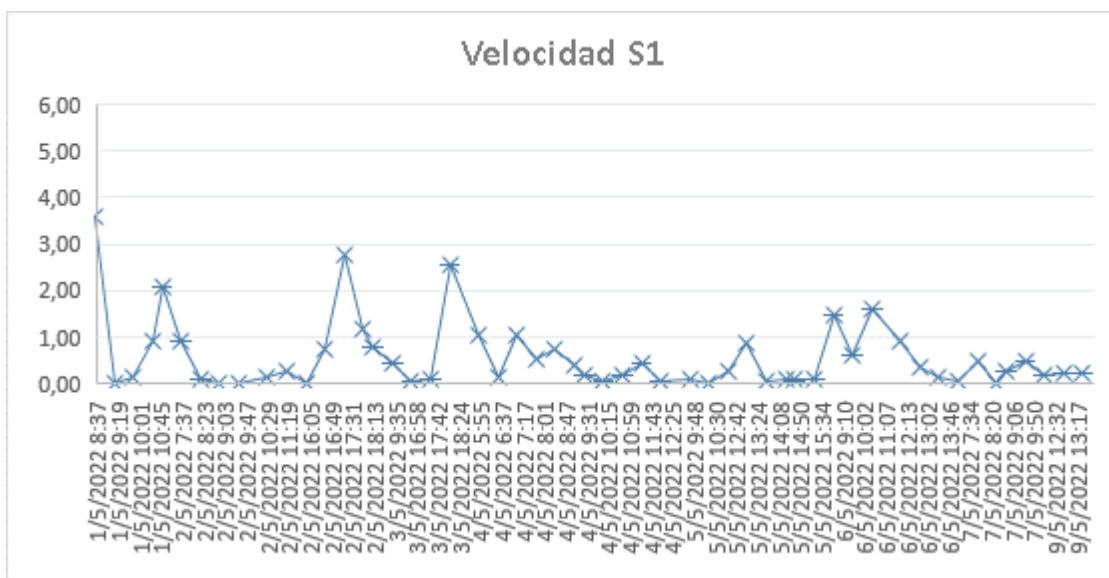
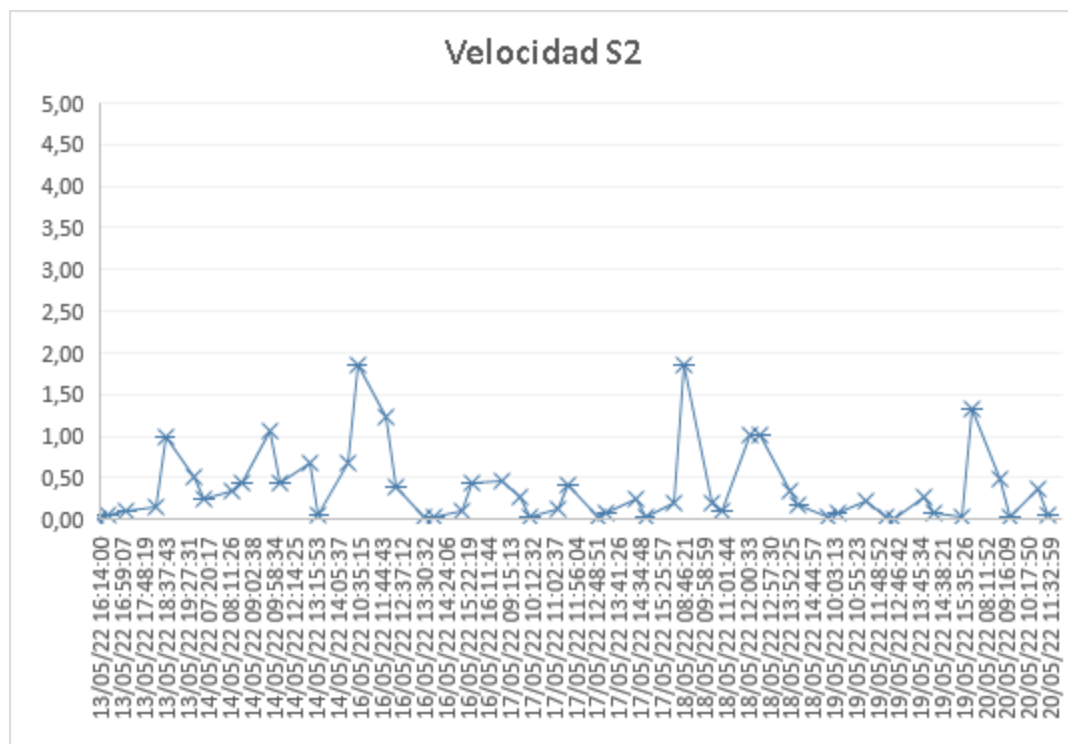


Figura 106

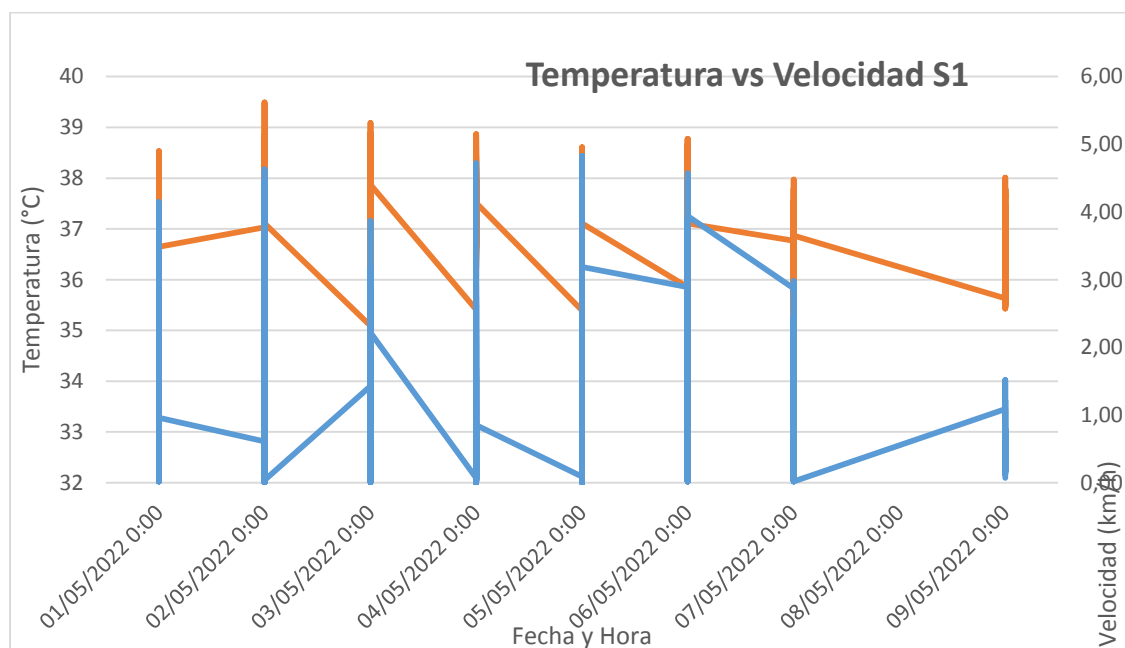
Velocidad de la vaca Nua en la Semana 2.



Con los datos de la temperatura y la velocidad se procede a realizar un análisis de correlación. Es importante mencionar que el aumento de la temperatura del animal es porque ha existido un incremento en su velocidad en ese instante de tiempo, puesto que el movimiento puede incrementar la temperatura entre 0.5 y 3.5°C. Pero esto está sujeto a varios factores, por ejemplo: la temperatura ambiente, cuando la vaca estuvo expuesta a un clima ligeramente soleado esta disminuyó su movimiento para liberar el exceso de calor acumulado, por otra parte, cuando la vaca estuvo expuesta a un clima con temperaturas frías, aumentó su movimiento para poder así mantener su temperatura corporal. Esto se puede apreciar en la Figura 107.

Figura 107

Gráfica superponiendo temperatura vs velocidad.



Para el caso del desplazamiento, en la variable posición, se ha almacenado las distancias que ha recorrido la vaca. Este valor se obtiene a partir de la comparación de dos coordenadas, la primera es un punto de los límites del cercado y la segunda del satélite de Google Maps, estas dos distancias se comparan con la que envía el dispositivo en tiempo real, para lo cual se suma y se obtiene la distancia

recorrida en ese instante de tiempo. Al finalizar el período de prueba y apagar el dispositivo después de aproximadamente 20 minutos el acumulador envía a la base de datos la distancia total final del día.

Los datos del desplazamiento del animal diario se visualizan en la Figura 107, es importante mencionar que las distancias recorridas por vacas lecheras varían entre 2 a 7km. De lo que se puede concluir que la vaca tiende a mantener una media de 1161.9869 m o 1.162 km, este valor se debe a que la vaca lleva una rutina bien establecida por horarios, a las 3:30am se desplaza al ordeño, después a las 5:30am se desplaza a la zona de pastoreo. En la tarde a las 2:30pm vuelve a la sala de ordeño y a las 6pm se desplaza al área donde descansará la noche y ya que el ordeño se encuentra muy cerca del área de pastoreo su desplazamiento no es tan elevado. Pero como se puede observar existen picos en su distancia recorrida siendo el máximo de 2.036 km esto se debe a que a medida que el animal pastorea el área se va cambiando a donde haya comida, de igual forma el área donde vaya a descansar.

Figura 108

Desplazamiento o distancia recorrida de la vaca Nua durante los 15 días de prueba.



Visualización de las alarmas en Ubidots

Una vez que se ha accedido a la aplicación móvil de Ubidots, se dirige a la opción Event log, aquí se podrá visualizar los mensajes de alarma según sea el caso: cuando el collar haya sido desconectado o violentado de alguna forma y cuando los límites de la temperatura han sido superados, esto se puede observar en las siguientes figuras.

Figura 109

Mensaje de alarma cuando el collar es desconectado.

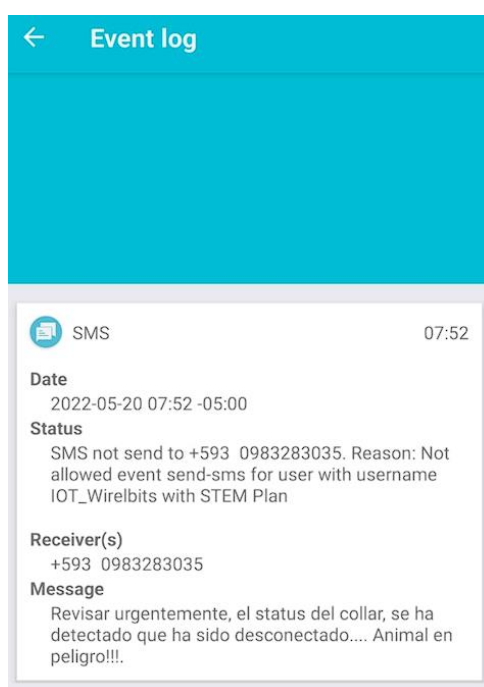
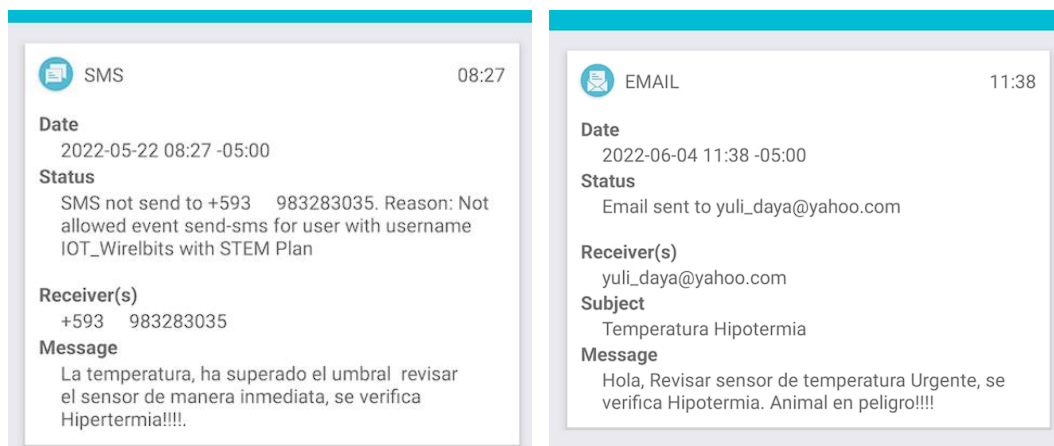


Figura 110

Mensaje de alarma cuando se sobrepasan los límites de temperatura.



Nota. Cabe mencionar que cuando la temperatura sobrepasa los 39°C se considera hipertermia y cuando disminuye los 37.7°C se considera hipotermia.

Representación gráfica en Ubidots

Es importante mencionar que durante el período de pruebas, se presentó un ataque por un malware malicioso, específicamente un troyano, como consecuencia de esto, se tuvo pérdidas irreparables de la primera instancia donde se almacenaba la base de datos, por lo tanto, para obtener los datos que ya fueron guardados, se hizo uso de un ataque de fuerza bruta, con esto se logró recuperar los datos en un archivo .csv para ser visualizado en Excel, sin embargo, es necesario aclarar que la instancia ya no se pudo recuperar, por tal motivo en la plataforma de Ubidots se puede visualizar los datos únicamente de la nueva instancia, cuya base de datos está activa desde el 16 de mayo del 2022. Por ello el análisis y discusión de resultados se lo realizó en Excel con la información de las dos bases de datos ya que al tener mayor cantidad de registros de las variables fisiológicas el análisis es mucho mejor.

Capítulo VII

Conclusiones, Recomendaciones y Trabajos futuros

Conclusiones

Realizar el estudio del arte sobre los parámetros que indican el estado de salud de una vaca en condiciones normales, permitió definir las variables a medir, mismas que se usaron para realizar el monitoreo del animal, y además establecer los rangos límites mínimos y máximos que puede tener la vaca en la temperatura y frecuencia cardíaca, también fue de gran ayuda en la orientación del trabajo de titulación.

Realizar la comparación de cada elemento que forma parte del prototipo, fue de gran ayuda, gracias a esto se pudo definir los sensores, el tipo de batería, el tipo de comunicación que se usó para poder diseñar e implementar este proyecto, es útil esta parte ya que se conoce los beneficios de cada componente y con ello se puede aprovechar al máximo las características de los mismos.

La tecnología LoRa fue la indicada para el desarrollo de este proyecto, además de que permite comunicar dispositivos hasta 30km, uno de los beneficios más sobresalientes es que permite el envío de información de forma ilimitada y con un nivel de seguridad muy bueno y un consumo mínimo de energía.

En base al cálculo de propagación que se realizó para definir el alcance que tiene el enlace, se concluye que de forma teórica el dispositivo Gateway y el prototipo, pueden colocarse a una distancia no mayor a 600m, sin embargo después de realizar las pruebas de cobertura, a una distancia de 345m se tienen un RSSI de -157dBm que representa a una señal débil, por lo tanto el rango máximo de este enlace es de 300m para que la transmisión de información no se vea afectada y

con respecto al SNR de acuerdo a los datos que se tomaron, se determina que la señal no ha sufrido muchos cambios debido a las señales interferentes.

El objetivo de este proyecto es desarrollar un dispositivo electrónico capaz de monitorear y transmitir variables fisiológicas de un bovino, ante lo que se puede afirmar que se ha logrado gracias al uso de herramientas Open Source; que facilitaron la conexión del hardware y software, logrando una vinculación exitosa entre el nodo final, nodo Gateway, servidor web e interfaz gráfica. En cuanto a la interfaz de usuario se considera que cumple con los requisitos funcionales de diseño, esto es, contar con una herramienta de aplicación móvil facilitando la visualización de manera rápida y dinámica de los cambios fisiológicos al usuario y al mismo tiempo ser alertados en caso de presentarse anomalías en la temperatura y la posición del bovino.

En base a la discusión de los resultados, se concluye que el tiempo de recolección de datos ha sido el adecuado, para realizar la valoración en cuanto a funcionalidad y efectividad del prototipo de rastreo y monitoreo de ganado bovino; se determinó que el dispositivo consigue una autonomía de 8 horas. Además, se ha conseguido evidenciar las variaciones de las cinco variables fisiológicas en estudio, de lo que se pudo determinar que la ubicación del dispositivo en el animal, juega un papel importante a tener en cuenta, ya que al ser colocado en el cuello y por algún movimiento del animal este puede separarse centímetros lo que cambiará los valores de los datos de temperatura y ritmo cardíaco, sin embargo, esto no sucede a menudo y la diferencia en la toma de datos no es tan grande, lo que no representa un inconveniente.

Recomendaciones

Es recomendable realizar el análisis y pruebas del enlace de propagación, con ellos se puede determinar a qué distancia máxima y mínima se puede colocar el dispositivo prototipo y evitar pérdida de información.

Es recomendable realizar un cálculo del consumo energético que va a tener todo el dispositivo prototipo, con lo cual se puede escoger la batería adecuada y determinar cuántas horas va a funcionar el equipo de forma continua.

Procurar que la ubicación del dispositivo Gateway LoRa tenga una línea de vista adecuada para asegurar la recepción de los datos y evitar así pérdida de paquetes y retardos.

Trabajos futuros

Se propone el desarrollo de un dispositivo prototipo que permita monitorear y rastrear vacas lecheras, pero con un sistema energético que le brinde al dispositivo la capacidad de ser autónomo, por ejemplo, hacer uso de paneles solares, teniendo en cuenta que ninguno de los elementos que forman parte del prototipo se vea afectado por las variaciones bruscas de voltaje.

La creación de modelos matemáticos para predecir enfermedades en estos animales por medio de las variables que se ha medido es uno de los retos a mejorar para investigaciones futuras.

Por otro lado, la cobertura de este proyecto fue lo suficiente para el área que ese tiempo están usando para pastear a las vacas, pero al terminar el alimento en dichas zonas serán movilizadas a áreas en las que ya no existe cobertura. Por ello se propone adquirir un Gateway LoRa de mayor alcance y de preferencia que sea de exterior y realizar pruebas de campo para determinar el nuevo rango de alcance de la red y analizar la fidelidad de la información obtenida de las variables fisiológicas.

Referencias

- Abellán, M. Á. (09 de 07 de 2019). *Introducción a la programación con AppInventor*. Obtenido de Introducción a la programación con AppInventor: <https://www.programoergosum.es/tutoriales/introduccion-a-la-programacion-con-appinventor/>
- Activities, B. T. (2021 de 03 de 22). *Bo True Activities*. Obtenido de <https://botrueactivities.com/comparativa-entre-sigfox-y-lorawan/>
- Arduino, A. (07 de 03 de 2018). *LoRa*. Obtenido de LoRa: <https://www.aprendiendoarduino.com/tag/lora/>
- Armengol Silvia, B. M. (16 de 09 de 2006). *Sitio Argentino de producción animal*. Obtenido de https://www.produccion-animal.com.ar/etologia_y_bienestar/bienestar_en_general/16-feria_ciencias.pdf
- AWS. (s.f.). *AWS*. Obtenido de <https://aws.amazon.com/es/ec2/instance-types/>
- Capan, T. (2017). *Developers*. Obtenido de <https://www.toptal.com/nodejs/por-que-demonios-usaria-node-js-un-tutorial-caso-por-caso>
- cf_pierola. (06 de 11 de 2016). *Scribd*. Obtenido de <https://es.scribd.com/document/330182506/Diferencia-Del-Ganado-de-Carne-y-Leche>
- CFN. (2017). *FICHA SECTORIAL: Cría y Reproducción de Ganado Bovino*. Subgerencia de Análisis e Información.
- Concepción, R. (19 de 12 de 2019). *Tarjetas de Desarrollo*. Obtenido de Tarjetas de Desarrollo: <https://www.rjconcepcion.com/podcast/tarjetas-de-desarrollo-episodio-8/>
- ConceptoDefinición*. (31 de 01 de 2021). Obtenido de <https://conceptodefinicion.de/ganado-vacuno-o-bovino/>
- CONtextoganadero. (02 de nov de 2017). *Es necesario controlar la temperatura de los animales*. Obtenido de <https://www.contextoganadero.com/ganaderia-sostenible/es-necesario-controlar-la-temperatura-de-los-animales>
- CONtextoganadero. (30 de 04 de 2019). *CONtextoganadero*. Obtenido de <https://www.contextoganadero.com/ganaderia-sostenible/vigile-las-constantes-fisiologicas-en-bovinos>
- CONtextoganadero. (30 de abril de 2019). *Vigile las constantes fisiológicas en bovinos*. Obtenido de <https://www.contextoganadero.com/ganaderia-sostenible/vigile-las-constantes-fisiologicas-en-bovinos>
- CONtextoganadero. (10 de 04 de 2020). *CONtextoganadero*. Obtenido de <https://www.contextoganadero.com/ganaderia-sostenible/conozca-los-valores-de-algunas-constantes-fisiologicas-de-los-animales>
- EcuRed. (11 de 2021). *GSM*. Obtenido de GSM: <https://www.ecured.cu/GSM>
- efor. (2020). *TECNOLOGÍAS DE COMUNICACIÓN PARA IOT*.

- El Productor*. (05 de 07 de 2017). Obtenido de <https://elproductor.com/2017/07/la-trazabilidad-en-bovinos-tarea-pendiente-en-ecuador/>
- Eustat. (2021). *Conexión móvil a internet*. Obtenido de https://www.eustat.eus/documentos/opt_1/tema_150/elem_16575/definicion.html
- Fernández, D. (19 de 12 de 2017). *bigEEK*. Obtenido de <https://blog.big-geek.com/broker-mensajeria/>
- Flores, F. (22 de 03 de 2021). *Cloud Computing: Tipos de nubes, servicios y proveedores*. Obtenido de *Cloud Computing: Tipos de nubes, servicios y proveedores*: <https://openwebinars.net/blog/tipos-de-cloud-computing/>
- Fumero, R. (2016). *CARACTERIZACIÓN DEL DESPLAZAMIENTO EN PASTOREO DE VACAS ANGUS, HEREFORD Y SUS CRUZAS F1 BAJO DIFERENTES DISPONIBILIDADES DE FORRAJE EN CAMPO NATURAL EN TRES MOMENTOS DE DETERMINACIÓN*. Uruguay.
- Galván, P. (s.f.). *SG*. Obtenido de <https://sg.com.mx/revista/51/hola-mundo-iot#:~:text=Funcionamiento%20de%20MQTT&text=Broker%3A%20el%20broker%20es%20el,con%20alg%C3%BAn%20t%C3%B3pico%20o%20tema.>
- Ganadería.com*. (s.f.). Obtenido de <https://www.ganaderia.com/raza/Angus>
- Ganadería.com*. (s.f.). Obtenido de <https://www.ganaderia.com/raza/charolais>
- ganadero, C. (26 de 10 de 2018). *CONtexto Ganadero*. Obtenido de <https://www.contextoganadero.com/internacional/el-internet-de-las-cosas-aplicado-al-mundo-ganadero>
- Gardey, J. P. (2021). *ESPACIO GEOGRÁFICO*. Obtenido de <https://definicion.de/espacio-geografico/>
- Georgia, U. o. (27 de 05 de 2017). *Estrategias de Detección de Celo para Ganado Lechero*. Obtenido de <https://extension.uga.edu/publications/detail.html?number=B1212-SP&title=Estrategias%20de%20Detecci%C3%B3n%20de%20Celo%20para%20Ganado%20Lechero>
- Gómez, A. (2017). Obtenido de [https://ciencia.lasalle.edu.co/cgi/viewcontent.cgi?article=1197&context=zootecnica#:~:text=.%2C%202015\).-,%20Cuando%20la%20temperatura%20aumenta%2C%20se%20produce%20una%20disminuci%C3%B3n%20de%20la,sensible%20es%20el%20animal%20al](https://ciencia.lasalle.edu.co/cgi/viewcontent.cgi?article=1197&context=zootecnica#:~:text=.%2C%202015).-,%20Cuando%20la%20temperatura%20aumenta%2C%20se%20produce%20una%20disminuci%C3%B3n%20de%20la,sensible%20es%20el%20animal%20al)
- GPS GANADOS. (2 de 10 de 2018). Obtenido de <https://www.vptelemetric.cl/gps-para-ganados/>
- Guadalupe Parra, H.-R. (2013). *modulo de clinica de bovinos*. México: Centro Universitario del Sur.
- Hernández, D. d. (2011). *EVOLUCIÓN DE LA TECNOLOGÍA CELULAR GSM HACIA LA GENERACIÓN 3.75*. TLATEMOANI, 10.
- INFAIMON. (14 de 01 de 2020). *Tipos de sensores de desplazamiento y sus aplicaciones industriales*. Obtenido de *Tipos de sensores de desplazamiento*

y sus aplicaciones industriales: <https://blog.infaimon.com/que-son-sensores-de-desplazamiento/>

Kaspersky. (s.f.). *Kaspersky*. Obtenido de <https://latam.kaspersky.com/resource-center/definitions/what-is-a-ssl-certificate>

KeepCoding, R. (26 de 01 de 2022). *Programar aplicaciones Android en Visual Studio es una buena idea*. Obtenido de <https://keepcoding.io/blog/programar-aplicaciones-android-en-visual-studio/>

MAG, CNT y Agrocalidad analizan sistemas tecnológicos para evitar abigeato. (21 de 12 de 2020). Obtenido de <https://www.agrocalidad.gob.ec/mag-cnt-y-agrocalidad-analizan-sistemas-tecnologicos-para-evitar-abigeato/>

Mayorga, F., Freire, C., Sánchez, A., & Vayas, T. (2020). *Análisis Sector ganadero 2014 - 2019*.

MCIelectronics. (14 de 12 de 2021). *Gateway LoRa Dragino LG01-N*. Obtenido de <https://www.mcielectronics.cl/shop/product/gateway-lora-dragino-lg01-n-dragino-29539>

MOBILEFISH.com. (s.f.). Obtenido de <https://lora.readthedocs.io/en/latest/>

Pazmiño, C. A. (2021). *Protocolo LORA para el análisis de medición con GPS y arduino en la industria ganadera del ecuador: una revisión sistemática*. Guayaquil.

Pickering, P. (29 de 06 de 2017). *Desarrollar con LoRa para aplicaciones IoT de baja tasa y largo alcance*. Obtenido de *Desarrollar con LoRa para aplicaciones IoT de baja tasa y largo alcance*: <https://www.digikey.com.mx/es/articles/develop-lora-for-low-rate-long-range-iot-applications>

PRIETO, K. S., & ROJAS SCARPETTA, L. C. (2019). *SISTEMA ELECTRÓNICO PARA EL MONITOREO DE BOVINOS HEMBRA EN SU CICLO ESTRAL*. Bogotá.

Protection, W. A. (25 de 08 de 2016). *Lo que no sabías de las vacas*. Obtenido de *Lo que no sabías de las vacas*: <https://www.worldanimalprotection.cr/blogs/lo-que-no-sabias-sobre-las-vacas#:~:text=Las%20vacas%20les%20encanta%20la%20rutina%3A%20siempre%20lo%20hacen%20las,y%2030%20minutos%20bebiendo%20agua>.

Radio, T. (12 de 06 de 2020). *tele radio*. Obtenido de <https://www.tele-radio.com/la/bandas-de-radio-frecuencia-utilizadas-en-todo-el-mundo/>

Raffino, M. E. (22 de Junio de 2020). *¿Qué es el clima?* Obtenido de <https://concepto.de/clima-2/>

RAUTEC. (11 de 2021). *Sensor de pulso cardiaco*. Obtenido de *Sensor de pulso cardiaco*: <https://sites.google.com/site/rauteccinvestav/experimentacin/sensor-de-pulso-cardiaco>

Roca, C. (2021). *Los 7 Proveedores de Cloud Computing TOP del mercado*. Obtenido de *Los 7 Proveedores de Cloud Computing TOP del mercado*:

<https://www.thepowermba.com/es/transformacion-digital/proveedores-cloud-computing/>

Rose, K., Eldridge, S., & Chapin, L. (Octubre de 2015). *LA INTERNET DE LAS COSAS—UNA BREVE RESEÑA*. Internet Society.

salesforce. (11 de 2021). *Cloud Computing: Aplicaciones en un solo lugar*. Obtenido de Cloud Computing: Aplicaciones en un solo lugar: <https://www.salesforce.com/mx/cloud-computing/>

Salvador, A. (s.f.). *agritotal.com*. Obtenido de <https://www.agritotal.com/nota/efectos-del-estres-calorico-en-vacas-lecheras/>

SENSORS, R. (11 de 2021). *El sensor de temperatura*. Obtenido de El sensor de temperatura: <https://www.rechner-sensors.com/es/documentacion/knowledge/el-sensor-de-temperatura>

Smith, G. M. (09 de 03 de 2020). *¿Qué es un Sensor y Qué Hace?* Obtenido de ¿Qué es un Sensor y Qué Hace?: <https://dewesoft.com/es/daq/que-es-un-sensor>

Sosa, C., Tello, E., & Lara, D. (s.f.). *Enfoque para Generar Aplicaciones Orientadas a Servicios para IoT mediante el Desarrollo Dirigido por Modelos*. Tamaulipas, México.

Soyomar, J. A. (2016). *MONITOREO EN LA GANADERIA BOVINA A TRAVÉS DE REDES GSM/GPRS*. Medellín.

TIMING, S.-B. P. (11 de 2021). *Sistema de Posicionamiento Global*. Obtenido de Sistema de Posicionamiento Global: [https://www.gps.gov/spanish.php#:~:text=El%20Sistema%20de%20Posicionamiento%20Global%20\(GPS\)%20es%20un%20sistema%20de,civiles%20en%20todo%20el%20mundo.](https://www.gps.gov/spanish.php#:~:text=El%20Sistema%20de%20Posicionamiento%20Global%20(GPS)%20es%20un%20sistema%20de,civiles%20en%20todo%20el%20mundo.)

Valerio, D. (s.f.). Obtenido de http://www.uco.es/zootecniaygestion/img/pictorex/08_09_53_tema1_ganado_bovino.pdf

Wikipedia. (26 de julio de 2021). *Posición*. Obtenido de <https://es.wikipedia.org/wiki/Posici%C3%B3n>

Wikipedia. (24 de junio de 2021). *Velocidad de los animales*. Obtenido de https://es.wikipedia.org/wiki/Velocidad_de_los_animales

World, R. W. (11 de 2021). *What is zigbee?* Obtenido de What is zigbee?: <https://www.rfwireless-world.com/Terminology/what-is-zigbee.html>

zone, a. (11 de 2021). *Qué es Android: todo sobre el sistema operativo de Google*. Obtenido de Qué es Android: todo sobre el sistema operativo de Google: <https://www.adslzone.net/reportajes/software/que-es-android/>