



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y COMUNICACIONES

Tema:

**SISTEMA DE MONITOREO PARA LA AVÍCOLA “SAN MIGUEL” CON
UNA RED DE SENSORES INALÁMBRICOS BASADOS EN LA
TECNOLOGÍA LPWAN**

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo a la
obtención del título de Ingeniero en Electrónica y Comunicaciones

ÁREA: Física y Electrónica

LÍNEA DE INVESTIGACIÓN: Sistemas de Control

AUTOR: Ángel Leonardo Orosco Tonato

TUTOR: Ing. Ana Pamela Castro Martin Mg.

Ambato – Ecuador

Septiembre – 2022

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Titulación con el tema: SISTEMA DE MONITOREO PARA LA AVÍCOLA “SAN MIGUEL” CON UNA RED DE SENSORES INALÁMBRICOS BASADOS EN LA TECNOLOGÍA LPWAN, desarrollado bajo la modalidad Proyecto de Investigación elaborado por el señor Orosco Tonato Ángel Leonardo, estudiante de la Carrera de Ingeniería Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, septiembre 2022

Ing. Ana Pamela Castro Martin Mg.

TUTOR

AUTORÍA

El presente Proyecto de Investigación titulado: SISTEMA DE MONITOREO PARA LA AVÍCOLA "SAN MIGUEL" CON UNA RED DE SENSORES INALÁMBRICOS BASADOS EN LA TECNOLOGÍA LPWAN es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, septiembre 2022



Ángel Leonardo Orosco Tonato

CC: 0503970899

AUTOR

APROBACIÓN TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Orosco Tonato Ángel Leonardo, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA DE MONITOREO PARA LA AVÍCOLA “SAN MIGUEL” CON UNA RED DE SENSORES INALÁMBRICOS BASADOS EN LA TECNOLOGÍA LPWAN, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con el señor Presidente del Tribunal.

Ambato, septiembre 2022.

Ing. Carlos Sánchez Rosero.

PRESIDENTE SUBROGANTE DEL TRIBUNAL

Ing. Santiago Manzano

PROFESOR CALIFICADOR

Ing. Juan Pablo Pallo

PROFESOR CALIFICADOR

DERECHOS DE AUTOR

Autorizó a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizó su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, septiembre 2022.



Ángel Leonardo Orosco Tonato

CC: 0503970899

AUTOR

DEDICATORIA

Lleno de alegría, y con mucho amor dedico este trabajo, a mis padres Vidal Orosco y Gloria Tonato, quien con su apoyo incondicional me supieron guiar hacia tan anhelado objetivo.

A mis hermanos, quienes con su comprensión y aliento me incentivaron a superarme día a día

A Dios quien me ha bendecido con esta maravillosa familia y ha obrado para que mi formación profesional hoy sea motivo de felicidad.

Ángel Leonardo Orosco Tonato

AGRADECIMIENTO

Al finalizar tan arduo camino me encuentro muy contento de alcanzar esta meta tan importante como lo es mi titulación, pero dicho camino no hubiese sido lo que fue sin todas las personas que conocí a lo largo de él, por tal motivo agradezco:

A Dios y al Universo quienes conspiraron para que cada uno de los días encajaran como una pieza de rompecabeza forjando así el camino hacia mi meta.

A mi querida alma mater Universidad Técnica de Ambato por ofrecerme la oportunidad de tener una educación de calidad.

A mi querida Facultad de Ingeniería en Sistemas Electrónica e Industrial quien me brindo sus aulas, biblioteca, y laboratorios para adquirir nuevos conocimientos.

A mis docentes quienes con sus experiencias inculcaron valores y los conocimientos necesarios de esta noble carrera.

A mi Tutora Ing. Ana Pamela Castro Martin quien con su arduo esfuerzo y dedicación fomento a que se desarrolle de gran manera dicho trabajo de titulación.

A mi madre Gloria Tonato pilar fundamental en toda mi vida estudiantil quien nunca desarto y siempre confió en mí.

A mi Padre Vidal Orosco quien con su sabiduría y consejos me supo guiar hacia mi meta.

A mis Hermanos quien con su apoyo y ocurrencias hicieron que mi vida universitaria sea manejable.

A mis amigos quienes me brindaron su amistad, una amistad llena de lealtad y solidaridad.

A mi novia quien llego en el momento indicado para brindarme el empujoncito extra que necesitaba para culminar mi carrera.

A Avícola San Miguel quien presto sus instalaciones y permitió realizar mi trabajo de titulación en él.

¡¡¡Muchas Gracias!!!

Ángel Leonardo Orosco Tonato

ÍNDICE GENERAL

APROBACIÓN DEL TUTOR	ii
AUTORÍA	¡Error! Marcador no definido.
APROBACIÓN TRIBUNAL DE GRADO	¡Error! Marcador no definido.
DERECHOS DE AUTOR	¡Error! Marcador no definido.
DEDICATORIA	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL	viii
ÍNDICE DE FIGURAS	xiii
ÍNDICE DE TABLAS	xvi
RESUMEN EJECUTIVO	xviii
ABSTRACT	xix
CAPÍTULO I	1
MARCO TEÓRICO	1
1.1 Tema de investigación	1
1.2 Antecedentes Investigativo	1
1.2.1 Contextualización del Problema	3
1.2.2 Fundamentación Teórica.....	5
Corporación nacional de avicultores del ecuador (CONAVE).....	5
Avicultura	5
Graja avícola	6
Avícola San Miguel	7
Pollo de engorde	7
Crianza en un área limitada	7
Temperatura y humedad dentro de una granja agrícola.....	8
Ventilación.....	9

Iluminación	10
Smart Farms.....	10
Metodología para el Diseño de Smart-Farms	11
Sensores inalámbricos	12
Internet de las cosas.	13
LPWAN (Low Power Wide Area Network).	14
LoRaWAN.....	16
Capas de LoraWAN.....	17
Arquitectura de Red LoraWAN.....	18
Clases de LoraWAN	19
Capa Física de LoRa	20
Modulación LoRa	22
CSS (Chirp Spread Spectrum)	22
Pulsos <i>Chirp</i>	23
1.3 Objetivos.....	26
1.3.1 Objetivo General	26
1.3.2 Objetivos específicos	26
Capítulo II.....	27
Metodología	27
2.1 Materiales.....	27
2.2 Métodos	27
Modalidad de la Investigación.....	27
Investigación Aplicada	27
Investigación Documental	27
Investigación de Campo	27
Investigación Experimental	28
Recolección de Información	28

2.3	Procesamiento y Análisis de Datos.....	28
2.4	Desarrollo del proyecto	28
Capitulo III		30
Resultados Y Discusión.....		30
Análisis y Discusión de los Resultados.....		30
	Desarrollo de la propuesta	30
	Variables detectadas	33
	Selección del Microcontrolador	35
	Selección de Sensor de Temperatura y Humedad	36
	Selección de Sensor de lúmenes	37
	Selección de Sensor de CO2.....	37
	Gateway	38
	Selección de Actuadores tipo Relé	38
	Servidor IoT.....	39
	Diseño de Nodos Finales	40
	Representación del módulo LoRa.....	40
	Circuito Impreso del Nodo 1	41
	Circuito Impreso del Nodo 2	43
Diseño mecánico		45
	Material de construcción de la carcasa	45
	Diseño mecánico esclavo (Nodo 1)	47
	Diseño mecánico esclavo (Nodo 2)	48
Construcción e implementación		49
	Circuito Impreso del Nodo 3	51
	Diseño mecánico maestro Gateway (Nodo 3)	54
	Acondicionamiento y adquisición de datos	56
	Etapa de Adquisición de Datos en los Esclavos	56

Funcionamiento del Esclavo 1 (Nodo 1)	58
Transmisión de datos Esclavo 1 (Nodo 1)	59
Funcionamiento del Esclavo 2 (Nodo 2)	61
Transmisión de datos Esclavo 2 (Nodo 2)	62
Etapa de Adquisición de Datos del Maestro (Nodo 3)	63
Funcionamiento del Maestro (Nodo 3).....	64
Funcionamiento del Proceso A	68
Funcionamiento del Proceso B	69
Funcionamiento del Proceso C	72
Servidor IoT.....	73
Implementación de Thinger.io.....	74
Aplicación.....	76
Implementación de REMOTEXY	76
Implementación Telegram	79
Notificaciones Telegram.....	81
Instalación de Nodos Finales	84
3.1 Verificación de Hipótesis	85
Análisis de resultados	85
Mediciones Nodo 1 y Nodo 2.....	86
Recepción de datos Nodo 3 (Maestro).....	87
Visualización de datos en RemoteXY	88
Visualización de datos en Thinger.io.....	89
Pruebas diarias del sistema	90
Error de medición	92
Presupuesto del proyecto	95
Análisis Costo Beneficio	96
Capítulo IV	99

Conclusiones y Recomendaciones.....	99
Conclusiones.....	99
Recomendaciones.....	100
Bibliografía	101

ÍNDICE DE FIGURAS

Figura 1. Granja avícola.....	6
Figura 2: Galpón San Miguel.....	7
Figura 3. Crianza por zonas con una criadera. [15]	9
Figura 4. Smart Farms.[18][19]	11
Figura 5: Metodología de diseño de Smart-Farms[16]	12
Figura 6. Crianza por zonas con una criadera. [15]	13
Figura 7. Internet de las cosas.[20]	14
Figura 8. Tecnología LPWAN con y sin licencia.[19].....	15
Figura 9. LPWAN surge como la técnica de IoT más rápida.[19].....	16
Figura 10: Protocolo de comunicación LoRaWAN. [23]	16
Figura 11:Tama LoraWAN [21].	17
Figura 12:Trama Mac [21]......	18
Figura 13:Arquitectura de re de Lorawan [24].	18
Figura 14. Modelo OSI para la tecnología LoRa. [23]	20
Figura 15. Diferentes modulaciones para una misma información. [23].....	21
Figura 16: Relación frecuencia vs tiempo de la modulación CSS. [25]	22
Figura 17: Pulsos <i>upchirp</i> y <i>downchirp</i> . [23].....	23
Figura 18: Diagrama de las etapas de crianza.....	31
Figura 19: Arquitectura típica de un sistema Lora.....	34
Figura 20: Arquitectura del sistema de monitoreo.....	34
Figura 21. Configuración de pines del módulo Heltec LoRa 32.	40
Figura 22. a) Representación visual lora b) Configuración de pines c) Diseño esquemático.....	41
Figura 23.a) b) Medidas reales del módulo LoRa c) Representación física del módulo LoRa.....	41
Figura 24. Esquema electrónico del Nodo 1.	42
Figura 25. a) Diseño del circuito impreso del Nodo 1 b) Diseño 3D de la PCB del Nodo 1.....	43
Figura 26. Esquema electrónico del Nodo 2.	44
Figura 27. a) Diseño del circuito impreso del Nodo 2 b) Diseño 3D de la PCB del Nodo 2 Fuente: Investigador.....	44

Figura 28. Diseño mecánico del Nodo 1.....	48
Figura 29. Diseño mecánico del Nodo 2.....	48
Figura 30. Ruteado de pistas en maquina CNC.	49
Figura 31. Montaje de elementos electrónicos en las placas nodo 1(a) y nodo 2(b).	49
Figura 32. Montaje del módulo LoRa en las placas.....	50
Figura 33. a) Ensamblaje del Nodo 1 b) Ensamblaje del Nodo 2.....	51
Figura 34. Esquema electrónico del Nodo 3.	53
Figura 35. a) Diseño del circuito impreso del Nodo 3 b) Diseño 3D de la PCB del Nodo 2.....	54
Figura 36. Diseño mecánico del Nodo 3.....	55
Figura 37. a) Montaje de elementos electrónicos en las placas Gateway b) Montaje del módulo LoRa en las placas.	55
Figura 38. Ensamblaje del Nodo 3.....	56
Figura 39. Elementos electrónicos de los esclavos.	56
Figura 40. Secuencia de comunicación LoRa y DHT 22.....	57
Figura 41. Diagrama de flujo de nodo 1.	59
Figura 42. Conversión de milivoltios a ppm de CO2. [33].....	61
Figura 43. Funcionamiento del filtro Kalman con Arduino.....	61
Figura 44. Diagrama de flujo del nodo 2.	62
Figura 45. Elementos electrónicos del maestro.....	63
Figura 46. Red disponible a 146.3 metros de distancia.	64
Figura 47. Diagrama de flujo del nodo 3.	67
Figura 48. Diagrama de flujo del Proceso A.....	69
Figura 49. Diagrama de flujo Proceso B.....	72
Figura 50. Diagrama de flujo Proceso C.....	73
Figura 51. Pantalla principal de Thinger.io.....	74
Figura 52. Nuevo dispositivo agregado a Thinger.io.....	74
Figura 53. HMI de variables para el Nodo 1.....	75
Figura 54. HMI de variables para el Nodo 2.....	75
Figura 55. Pantalla principal de RemoteXY.	76
Figura 56. Instalación de la aplicación RemoteXY	77
Figura 57. Agregar nuevo dispositivo inteligente.....	77
Figura 58. Establecer comunicación entre LoRa32 y el dispositivo móvil.	78

Figura 59. HMI desarrollado en RemoteXY.....	78
Figura 60. Creación de un BOT en Telegram.....	79
Figura 61. Comandos para obtener el IDBot.....	80
Figura 62. Configuración de respuestas rápidas del BOT.....	81
Figura 63. Medición 1 de temperatura del Nodo1 y Nodo2.....	81
Figura 64. Notificación de temperatura “Muy Fría”, recibida.....	82
Figura 65. Medición 2 de temperatura del Nodo1 y Nodo2.....	82
Figura 66. Notificación de temperatura “Muy Alta”, recibida.....	83
Figura 67. Medición CO2 existente en el galpón.....	83
Figura 68. Notificación de CO2 “Muy Alto”, recibida.....	84
Figura 69. Conexiones de corriente alterna hacia el galpón.....	84
Figura 70. Disposición de los Nodos 1 y Nodo 2.....	85
Figura 71. Proceso de tratamiento de datos.....	86
Figura 72. Datos obtenidos del Nodo 1 y Nodo 2.....	86
Figura 73. Datos recibidos en el Nodo 3 (Maestro).....	87
Figura 74. Datos recibidos RemoteXY.....	88
Figura 75. Datos recibidos en Thinger.io.....	89

ÍNDICE DE TABLAS

Tabla 1 Ventajas y Desventajas	3
Tabla 2: Temperatura de crianza [13] de pollos Ross 308 para engorde. [15]	8
Tabla 3: Efectos de varios contaminantes en el aire. [13].....	9
Tabla 4: Requisitos de un sistema IoT.	14
Tabla 5: Características de las capas LoRaWAN.	17
Tabla 6: Características de las clases de Lora Mac.....	19
Tabla 9. Microcontroladores con tecnología LORA.....	23
Tabla 11. Comparación de módulos DHT disponibles en el mercado.....	24
Tabla 13. Comparación de módulos de lúmenes.	24
Tabla 15. Comparación de módulos de CO2.	24
Tabla 17. Comparación relé de activación.	25
Tabla 19. Comparación entre servidores IoT.	25
Tabla 7. Variables detectadas con su ventaja y desventaja.	33
Tabla 8. Descripción de arquitectura.	34
Tabla 10. Datos Técnicos del Módulo LORA ESP32 WiFi.	35
Tabla 12. Datos técnicos del sensor DHT22.....	36
Tabla 14. Datos técnicos del sensor LM393.	37
Tabla 16. Datos técnicos del módulo MG-811.	38
Tabla 18. Datos técnicos del módulo Relay 2CH.	39
Tabla 20. Requerimientos del Nodo 1.....	42
Tabla 21. Requerimientos del Nodo 2.....	43
Tabla 22. Propiedades mecánicas y físicas del PLA.....	45
Tabla 23. Dimensiones físicas de los dispositivos electrónicos.....	46
Tabla 24. Requerimientos del Nodo 3.....	51
Tabla 25. Datos medidos en el Nodo 1 y Nodo 2.	87
Tabla 26. Datos recibidos en el Nodo 3.	88
Tabla 27. Datos recibidos en RemoteXY.....	88
Tabla 28. Datos recibidos en Thinger.io.	89
Tabla 29. Datos obtenidos en el día uno de medición.....	90
Tabla 30. Datos obtenidos en el día dos de medición.....	90
Tabla 31. Datos obtenidos en el día tres de medición.....	91
Tabla 32. Error de medición en temperatura.....	92

Tabla 33. Error de medición en humedad.	93
Tabla 34. Error de medición en lúmenes.	93
Tabla 35. Error de medición en CO2.	94
Tabla 36. Error relativo Máximo y Mínimo.....	94
Tabla 37. Presupuesto de inversión del proyecto.....	95
Tabla 38. Consumo energético del sistema de monitoreo.....	97
Tabla 39. Consumo energético por horas del sistema.....	97
Tabla 40. Cargos tarifarios únicos Cotopaxi.....	97
Tabla 41. Cargos tarifarios únicos Cotopaxi.....	98
Tabla 42. Planilla de luz del sistema.	98

RESUMEN EJECUTIVO

El presente trabajo de investigación describe el desarrollo de un sistema de monitoreo para la avícola “San Miguel” con una red de sensores inalámbricos basados en la tecnología LPWAN ubicada en la provincia de Cotopaxi cantón San Miguel de Salcedo, de esta manera los operadores del galpón de pollos estarán informados en todo momento sobre las condiciones ambientales como: temperatura, humedad, lúmenes y concentración de CO2 existente.

El prototipo Lora con IoT de monitoreo consta de 3 elementos importantes: dos nodos configurados como esclavos, un nodo configurado como maestro y los respectivos servidores IoT para dispositivo móviles como para computadoras o laptops, en este caso se utilizó un microcontrolador Lora ESP32. El Nodo1 cuenta con 4 sensores para medir la temperatura, humedad, lúmenes, y CO2. El Nodo 2 cuenta con 3 sensores para la medición de la temperatura, humedad y lúmenes. El Nodo 3 recibe datos del Nodo 1 y 2. El servidor IoT llamado Thinger.io recibe datos del Nodo 3, estos datos se muestran en una interfaz de usuario. Se desarrolló una appmóvil con la herramienta de aplicación del servidor en la nube RemoteXY para mostrar los datos del sensor.

El Nodo 3 tiene la capacidad de enviar alarmas de alta temperatura, baja temperatura y alta concentración de CO2 en el ambiente mediante la utilización del sistema de mensajería instantánea que posee Telegram, esta funcionalidad le brinda una ayuda a los operadores del galpón debido a que no poseen instrumentos de medición de dichas condiciones ambientales. Generalmente este proceso de medición lo realizan únicamente con el sentido de tacto y en base a su experiencia en la crianza de pollos, por lo tanto, contar con un sistema inteligente de monitoreo brinda una ventaja dentro de la crianza de pollos.

La implementación del sistema de monitoreo para la avícola proporciona un mejor manejo de los recursos como el gas de las criadoras, también permitió identificar la concentración de CO2 en el ambiente para ventilar adecuadamente el galpón. Además, el conocimiento de las condiciones ambientales en tiempo real y un adecuado cuidado del ave mejora la calidad de vida de los pollos.

Palabras clave: IoT, Lora, sistema de monitoreo, comunicación inalámbrica, avícola

ABSTRACT

The present research work describes the development of a monitoring system for the "San Miguel" poultry farm with a network of wireless sensors based on LPWAN technology located in the province of Cotopaxi, canton of San Miguel de Salcedo, in this way the operators of the shed of chickens will be informed at all times about the environmental conditions such as: temperature, humidity, lumens and existing CO₂ concentration.

The prototype monitoring system with Lora and IoT technology has 3 important elements: two slaves nodes (Node 1 and 2), one master node (Node 3), and a IoT servers for mobile devices such as computers or laptops. The Lora ESP32 microcontroller was used for the monitoring system. Node1 has 4 sensors to measure temperature, humidity, lumens, and CO₂. Node 2 has 3 sensors for measuring temperature, humidity and lumens. Data collected by Node 1 and 2 is sent to Node 3. IoT server called Thingier.io receives data from Node 3, this data is shown in a user interface. An appmovil was developed with the application tool of RemoteXY cloud server to show the sensor data.

Node 3 can send alarms of high temperature, low temperature and high concentration of CO₂ in the environment through the use of the instant messaging system that Telegram has. The automatic alarm system brings operators an easy way to know if environment conditions are optimal in the poultry farm. Generally, this measurement process is carried out only with the sense of touch and based on their experience in raising chickens, therefore, having an intelligent monitoring system provides an advantage in raising chickens.

The implementation of the monitoring system for the poultry farm provides better management of resources such as gas from the brooders, it also allowed identifying the concentration of CO₂ in the environment to adequately ventilate the shed. In addition, the knowledge of the environmental conditions in real time and an appropriate bird care improves the quality of life of chickens.

Keywords: IoT, Lora, Server, Poultry.

CAPÍTULO I

MARCO TEÓRICO

1.1 Tema de investigación

SISTEMA DE MONITOREO PARA LA AVÍCOLA “SAN MIGUEL” CON UNA RED DE SENSORES INALÁMBRICOS BASADOS EN LA TECNOLOGÍA LPWAN

1.2 Antecedentes Investigativo

De acuerdo con la investigación realizada se describen proyectos relacionados con metodologías, tecnologías y sistemas de monitoreo aplicados para establecer soluciones a través de sistemas IoT enfocadas en el sector avícola.

En el año 2018, Andrés Osorio y Carlos Gonzáles, de la Universidad Javeriana, Colombia presenta el proyecto de titulación denominado “Desarrollo e implementación de una aplicación Web para el monitoreo en tiempo real de temperatura, presión y humedad al interior de los galpones de aves de engorde utilizando redes de sensores inalámbricos”, misma que presenta una red de sensores inalámbricos que miden variables como la temperatura, presión y humedad en el galpón, además esta información es almacenada en una base de datos SQL para la realización de consultas y visualizadas a través de la aplicación Web. La tecnología que aplica en esta investigación es un protocolo de comunicación ZigBee en una topología estrella [1].

En el año 2019, Ricardo Torres, de la Universidad de Israel, Ecuador, presenta el proyecto de investigación “Red de módulos XBee para el control y monitoreo de temperatura e intensidad luminosa para criaderos avícolas mediante modo API con visualización HMI”, donde presenta una maqueta simulando un galpón, en el cual se controla las variables de una manera automática. Además, el control y monitoreo que se realiza a través de una interfaz Humano Maquina (HMI) para controlar y mantener las variables en valores óptimos necesarios para el cuidado de los pollos. La tecnología que aplica en esta investigación es un protocolo de comunicación XBee con topología mixta [2].

En el año 2020, María del Rosario Vargas y Criseldo Luna, del Tecnológico Nacional de México, presenta el proyecto de titulación denominado “Prototipo para la Automatización del Control del Ambiente en el Área Avícola del Instituto Tecnológico de Huejutla”, en el cual se desarrolla un prototipo de control con la tecnología WiFi de un sistema avícola donde los sensores miden temperatura y humedad (DTH11). La información es enviada a un microcontrolador que procesa los datos, también incorpora una aplicación Android realizada con el software Android Studio que permite al usuario visualizar las variables medidas mediante un dispositivo móvil. Los datos son presentados al usuario en tiempo real esto permite un monitoreo constante de las variables ambientales dentro del área avícola [3].

En el año 2020, Muhammad Faiz Haji Hambali , Ravi Kumar Patchmuthu y Au Thien Wan, presentada en la 8th International Conference on Information and Communication Technology (ICoICT) en Yogyakarta, Indonesia, desarrolla la investigación titulada “IoT Based Smart Poultry Farm in Brunei” que consiste en un sistema de monitoreo y un sistema de alerta para granjas. El sistema de alerta se activa cuando las variables medidas se encuentran fuera de un umbral. El proyecto fue desarrollado utilizando la tecnología IoT y una red de sensores inalámbricos (WSN), la alerta es enviada a través de varios medios como SMS, WhatsApp, correo electrónico y mediante una interfaz web donde se muestran los valores en tiempo real de las variables de la granja avícola [4].

En el año 2020, Jenny Priyanka Mondol, Khan Raqib Mahmud, Muhammad Golam Kibria, Abul Kalam Al Azad, presentada en la 2nd International Conference on Advanced Information and Communication Technology (ICAICT) en Dhaka, Bangladesh, presenta la investigación denominada “IoT based Smart Weather Monitoring System for Poultry Farm” realiza un monitoreo en tiempo real de variables con la tecnología IOT, se utiliza un sensor de temperatura y humedad DHT11. El sistema recopila los datos y los transmite a un servidor en la nube donde se almacenan y son comparados con un umbral continuamente. Al momento en que la variable supera el umbral envía una alerta al teléfono inteligente, y activa una señal acústica en la granja avícola [5].

La electrónica en la actualidad se ha convertido en un medio para solucionar diversos problemas que nos presentan el vida cotidiana, trabajo, e incluso en pequeños emprendimientos, pero la misma avanza muy rápido, tal es así que en la Tabla 1 se presenta las ventajas y desventajas de los antecedentes investigativas reunidas para el presente proyecto de investigación denominado: “Sistema De Monitoreo Para La Avícola “San Miguel” Con Una Red De Sensores Inalámbricos Basados En La Tecnología Lpwan.”

Tabla 1 Ventajas y Desventajas

Ventajas	Desventajas
<ul style="list-style-type: none"> • La sensorización en tiempo real es algo común en la actualidad. • Las aplicaciones móviles facilitan la monitorización preventiva. • La tecnología IoT amplía el rango de monitorización. • Existen múltiples plataformas IoT gratuitas. 	<ul style="list-style-type: none"> • Las Tecnologías de RF como Zigbee, Xbee, entre otras son limitadas. • El monitoreo se lo realizaba de forma local. • IoT es obsoleta sin acceso a internet. • La Gateway es el centro de toda red IoT.

Elaborado por: El investigador

1.2.1 Contextualización del Problema

En el Ecuador se han desarrollado grandes industrias tanto agrícolas, frutícolas, avícolas, piscícolas, entre otras, han permitido dinamizar la economía del país y al mismo tiempo alimentar a la población ecuatoriana incluso en situaciones de emergencia. El Ecuador es un país autosustentable en la producción de proteína animal en especial para la avícola, debido a que se puede cultivar diferentes granos, y elaborar alimentos balanceados para la crianza de pollos. Según la Corporación Nacional de Avicultores del Ecuador (CONAVE) y el Instituto Nacional de Estadística y Censo (INEC) para el 2006 se reportó un valor aproximado de granjas avícolas de 197 en la provincia de Tungurahua, distribuidas en los cantones de Ambato, Pelileo, Baños, Cevallos y Patate [6]. A nivel nacional se estima una población avícola de 224 millones de pollos de engorde, 9.8 millones de gallinas ponedoras y una producción de 48 000 000 de huevos por semana [6]. Sin embargo, el acelerado crecimiento de

esta industria ha dado lugar a implementación de criaderos o galpones que en su mayoría son instalados de manera empírica y poco tecnificada.

Entre los mayores retos que se enfrenta un productor de pollos en el país es la calidad de la genética del animal y las condiciones físicas en donde son criados [7]. El desconocimiento en el momento de preparar el galpón para la llegada de los pollos puede contribuir al aumento de decesos de aves. Los galpones deben cumplir parámetros en sus condiciones ambientales para recibir a los pollos, por ejemplo, limpieza, temperatura ideal, ventilación adecuada, entre otros. Una vez los pollos llegan a su nuevo hogar, lo que garantiza su buen crecimiento es el constante control de los operarios a las condiciones del galpón. La temperatura es una de las variables físicas más importante que oscila entre 28° C en la etapa inicial y 24° C en la etapa final de crecimiento, el encargado del galpón observa los cambios del comportamiento en las aves para tomar cualquier acción correctiva [7]. Así mismo, otras magnitudes físicas se controlan según el criterio del operario, su conocimiento empírico o por observación del comportamiento de las aves.

En cuanto a la innovación tecnológica dentro de la industria avícola, existe presupuesto para invertir en tecnología a futuro, pero no existe el suficiente conocimiento para encaminarlo en el contexto de la industria 4,0 es decir la industria avícola ecuatoriana carece de departamentos especializados en innovación y transformación digital en el marco de la Industria 4,0. La crianza de pollos en Ecuador es un área de oportunidad para promover la cultura de transformación digital implementando tendencias tecnológicas actuales que encaminen a la incorporación de Granjas Inteligentes [8].

En el cantón Salcedo provincia de Cotopaxi se encuentra la granja avícola “San Miguel” cuyo galpón de 24 m² tiene la capacidad de criar a un máximo de 1200 pollos. Los animales ingresan a la granja de un día de nacidos y se venden después de tres semanas. En el proceso de crianza de pollos, los operarios de la granja son los encargados de asegurar el bienestar y buen crecimiento de los pollos. Los parámetros como temperatura, luminosidad, humedad son controlados de forma manual y empíricamente. Para mantener una temperatura apropiada se utiliza sistemas de calefacción con campanas con GLP. Entre los mayores inconvenientes de este sistema es su imposibilidad de contar con una realimentación de temperatura. El dióxido de

carbono producido por las campanas de GLP, la luminosidad y humedad del lugar son variables desconocida para el operario, y no existe monitoreo de las variables físicas que intervienen en la crianza de pollos.

El presente trabajo de investigación se basa en la implementación de un sistema de monitoreo inteligente de las magnitudes físicas más importantes para la crianza de pollos desde un día de edad hasta las tres semanas, estas magnitudes son temperatura, humedad, lúmenes y CO₂. La tecnología utilizada para el desarrollo de este proyecto está basada en LPWAN (Low Power Wide Area Networks) para realizar una red de sensores inalámbricos. Dicho sistema permitirá reducir la tasa de decesos en la crianza de pollos, también se obtendrá datos reales de las condiciones ambientales en el criadero lo cual permitirá al encargado tomar decisiones acertadas para mejorar la salud de los pollos.

1.2.2 Fundamentación Teórica

Corporación nacional de avicultores del ecuador (CONAVE)

Fundada en el año de 1994, la Corporación Nacional de Avicultores del Ecuador (CONAVE), nace de la necesidad de agrupar a los productores avícolas ecuatorianos, con el fin de contribuir a la inocuidad, soberanía alimentaria del país y generación de empleo directo e indirecto a un amplio sector del país, Conave busca el bienestar del sector avícola representándolos ante los diferentes actores de la sociedad como son: Entidades gubernamentales, Organismos internacionales, Medios de comunicación, Universidades y centros de estudio; y Consumidores, actualmente agrupa aproximadamente al 80% de productores del país, liderando acciones de orden sanitario, ambiental y de promoción de consumo de proteína animal [9].

Avicultura

La avicultura ha tenido un gran crecimiento en el país, al ser uno de los productores de cárnicos más importantes, este abarca desde la producción de huevos hasta la carne de pollo y otras aves como patos, pavos y otros. La crianza difiere entre una especie a otra, pero para aves livianas se estima una duración aproximada de 18 a 20 semanas de edad, en cambio para especies criollas y razas pesadas el tiempo de crianza varía entre 24 a 30 semanas de edad [10].

En cada especie varia la tasa de mortalidad, pero en la mayoría de los casos esto se debe a enfermedades a causa de un mal control del habitat de los mismos. Existen otros factores como la calidad de los alimentos que influyen en el ritmo de crecimiento, la coloración de la piel, aspectos que influyen en la calidad de la carne [10].

Granja avícola

Las granjas avícolas con instalaciones que están destinadas a la cría de aves, así como pavos, pollos, gallinas y otros con el fin ser comercializados. Estas instalaciones son comúnmente conocidas como galpones, los cuales deben tener una estructura que permite de alguna manera controlar variables ya sea de manera manual o automática como las corrientes de aire, ventilación en temporadas calurosas, y controlar otras variables que afectan en el crecimiento de las aves tal es el caso de la temperatura, humedad, iluminación y otros aspectos que pueden variar en función de la localización del galpón [11].

Estas instalaciones también han sido focalizadas para simplemente realizar la producción de huevos como se muestra en la Figura 1, criaderos para los primeros 18 días aproximadamente en la crianza de aves, y finalmente en criaderos para el engorde. Por lo cual, esta parte de la industria alimenticia ha crecido continuamente y se ha ido especializando debido a que las condiciones necesarias para cada uno de los aspectos antes mencionados difieren en función de etapa de crecimiento [11].



Figura 1. Granja avícola.

Fuente: Investigador

Avícola San Miguel

Avícola San Miguel se encuentra ubicado en la Provincia de Cotopaxi Cantón Salcedo parroquia Mulliquindil Santa Ana es una avícola artesanal que actualmente cuenta con 4000 pollos distribuidos en diferentes galpones con capacidad de 1200 pollos cada uno de ellos, los cuales son criados desde el primer día de nacimiento hasta los 18 días, donde se procede a realizar su venta. En la avícola el control de variables directas que afectan a los pollos tales como alimentación, peso, estado saludable del pollo, y el control de las variables como la temperatura, humedad, lúmenes, CO₂, se los realiza de forma empírica, al igual que la alimentación, generando así diversos problemas en el método de crianza siendo para el productor un sacrificio por el tiempo que se toma en realizar el control personalizado tanto de la cría como del galpón en sí.



Figura 2: Galpón San Miguel

Fuente: Investigador

Pollo de engorde

El proceso de crianza de pollos de engorde está influenciado por varios elementos, dentro de los cuales tenemos: manejo de la parvada, la buena calidad del alimento suministrada, salud respiratoria, salud digestiva y las condiciones climáticas. Esta información fue obtenida de un manual de engorde de la genética Ross 308, la más utilizada en todo el país y que se encuentra en el siguiente estudio [12].

Crianza en un área limitada

En este sistema de crianza se requiere de una fuente permanente de calor, en muchos casos se utiliza una campana de gas o criadoras. En este caso se debe considerar que las aves deben estar confinadas en un solo punto, proporcionando iluminación de

forma puntual. En galpones que disponen de cortinas es muy común ubicar las criadoras en el centro del galpón para reducir el área de calentamiento y ahorrar energía [13].

Temperatura y humedad dentro de una granja agrícola.

Las variables que tienen una mayor importancia son la temperatura y la humedad, que en ocasiones van de la mano para mantener un ambiente idónea para la crianza de aves. Estos aspectos poseen una gran influencia en el proceso de crianza y desarrollo de las aves, factores que repercuten en la calidad de la carne, de la misma manera tener un mayor control para evitar enfermedades en las aves, ya que a bajas temperaturas las aves pueden adquirir enfermedades como neumonía y al estar en ambientes húmedas existe una mayor proliferación de bacterias causantes de enfermedades. Para el crecimiento de pollos la temperatura debe mantener diferentes rangos de temperaturas que varían en función de su etapa de crecimiento y ubicación en el galpón como se muestra en la Tabla 2, de igual manera la humedad relativa como un valor aproximado puede estar entre un rango de 65% a 75%. Estos datos varían en función de la etapa de crecimiento y la especie de las aves [14].

Tabla 2: Temperatura de crianza [13] de pollos Ross 308 para engorde. [15]

Temperatura por zonas (°C)			
Edad(días)	Temp en Galpón (°C)	Borde de la Criadora (A)	2 m desde el borde (B)
1	30	32	29
3	28	30	27
6	27	28	25
9	26	27	25
12	25	26	25
15	24	25	
18	23	24	
21	22	23	
24	21	21	
27	20	20	

En la Figura 3 se puede observar el diagrama para determinar el área de gradientes de temperatura y la forma más adecuada para instalar una fuente de calor dentro de un galpón de crianza de pollos.

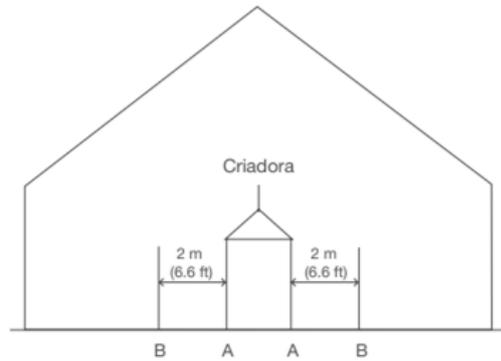


Figura 3. Crianza por zonas con una criadera. [15]

Ventilación

La ventilación hace referencia a la calidad del aire en el ambiente de crianza de pollos, se debe diferenciar claramente entre el concepto de renovación de aire y enfriar al ave. Para una adecuada ventilación sin resfriar a los pollitos se debe instalar entradas de aire en la parte superior del galpón y controlar el flujo de aire.

Muchos galpones utilizan las campanas de calentamiento a base de GLP (Gas Licuado de Petróleo), lo cual genera la presencia de bióxido de carbono y puede envenenar a las aves, además que los pollos también generan monóxido de carbono, bióxido de carbono y amoníaco, compuestos químicos nocivos para su propia salud [13]. En la Tabla 3 se resumen los principales factores contaminantes en el aire y los efectos que tienen los mismo en aves criadas en galpones.

Tabla 3: Efectos de varios contaminantes en el aire. [13]

Factor	Efecto
Polvo	Daño en el recubrimiento interno del tracto respiratorio. Incrementa la susceptibilidad a enfermedades.
Monóxido de Carbono	100ppm reducen la presencia de oxígeno en el ambiente.

Bióxido de Carbono	Más 0.35% causa Ascitis. Es fatal a niveles altos.
Amoniaco	10ppm dañan la superficie pulmonar. 20ppm aumenta susceptibilidad a enfermedades respiratorias 50 ppm reducen las tasas de crecimiento.
Humedad	Sus efectos varían dependiendo de la temperatura. 29 °C y 70% HR afecta el crecimiento

Iluminación

La iluminación es un factor muy importante para que el pollo pueda ganar peso, muchas granjas avícolas practican la utilización de luz continua para este objetivo. El funcionamiento de estos sistemas consiste en un periodo prolongado de iluminación seguido de una breve oscuridad de aproximadamente media hora. Durante el proceso de crianza se debe ir reduciendo gradualmente la intensidad de la luz hasta llegar a los 21 días con una intensidad de 10 lux.

Por último, se debe mantener una iluminación uniforme durante todo el galpón, siendo los focos y lámparas fluorescentes una gran opción para este fin generado un ahorro significativo en las planillas de luz [12].

Smart Farms.

Un sistema SMART-Farm (Sustainable, Manageable, Accessible Rural Technologies) consiste en tecnologías de redes de sensores y actuadores inalámbricos de bajo costo que se aplican con el objetivo de transmitir información útil para gestionar diferentes recursos de manera eficiente en una comunidad, gracias al uso del control de acciones[16].

La automatización de granjas a través de tecnología IoT tiene el propósito de crear granjas sostenibles, más eficientes y eficaces. En las Smart Farms se monitorean varios aspectos tal es el caso de variables atmosféricas como temperatura, humedad, etc. Conocer estas variables es esencial para promover un control en la salud de los animales, así como de los alimentos proporcionados a los mismos al encontrarse en el

mismo ambiente. Como se puede observar en la Figura 4 las Smart farms están compuestas por los sensores, actuadores, y el medio de transmisión etc [17].



Figura 4. Smart Farms.[18][19]

Metodología para el Diseño de Smart-Farms

El diseño de una Smart-Farm debe tener en cuenta las características y condiciones socioculturales de la comunidad rural donde se va instalar, es decir, los sensores, actuadores y controladores deben ser seleccionados cuidadosamente para que puedan acoplarse a las necesidades y conocimientos de los operarios. Estas consideraciones durante el diseño de una Smart-Farm permiten revalorizar los componentes existentes en el medio y proporcionar un desarrollo local sustentable desde el punto de vista ambiental, conservación de los recursos naturales y aspectos socio-ambientales. En la Figura 5 se puede observar los pasos de la metodología para diseño de una Smart-Farm.

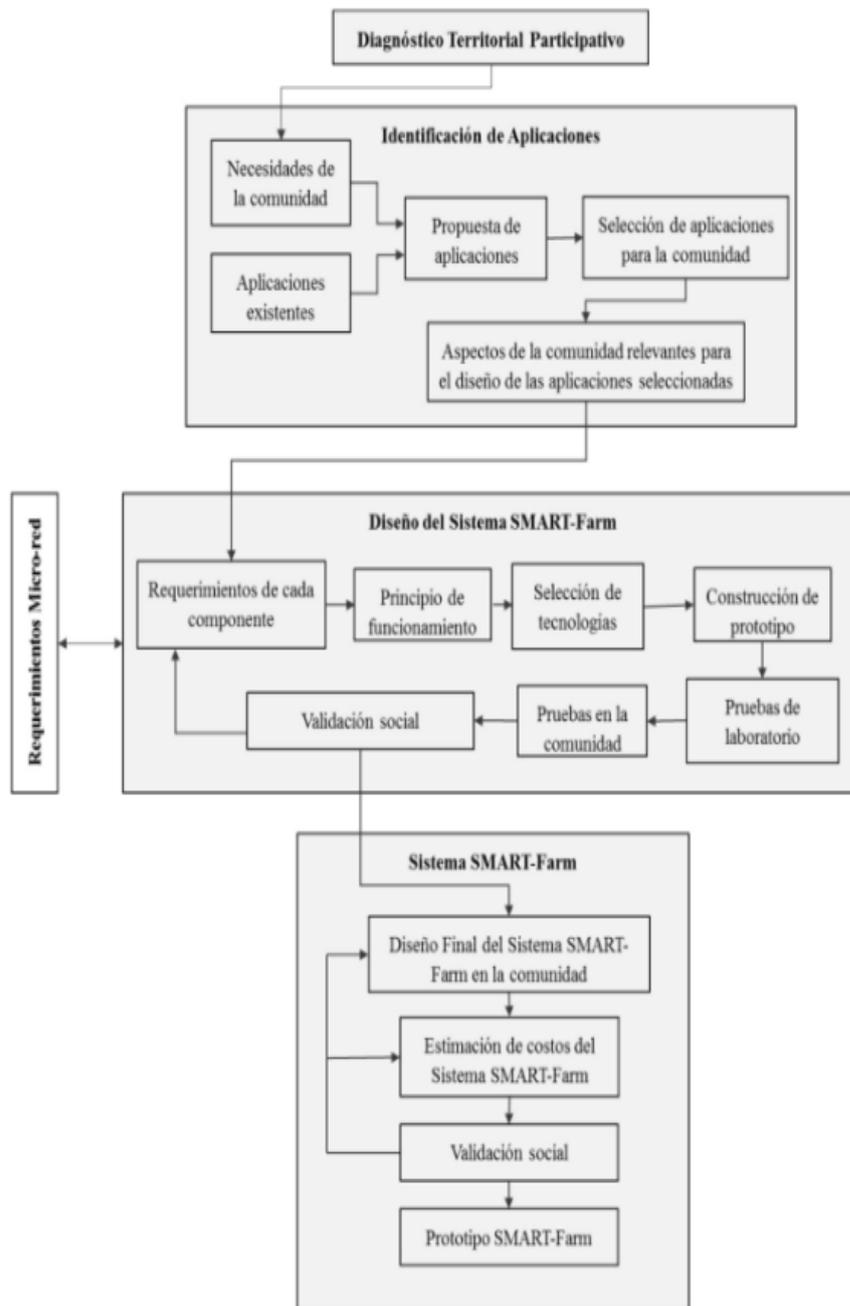


Figura 5: Metodología de diseño de Smart-Farms[16].

Sensores inalámbricos

Los sistemas WSN (Wireless Sensors Networks) son conocidos como una red de sensores inalámbricos con una gran cantidad de dispositivos pequeños, independientes y se identifican como nodos de sensores, se instalan cerca de un fenómeno para monitorear, almacenar y transmitir los datos obtenidos de forma inalámbrica. Los sistemas WSN son muy aplicables dentro de las industrias, domótica, detección de

magnitudes físicas en un ambiente, entornos militares, entre otros. En la Figura 6 se puede observar una red de sensores y un servidor donde se reciben los datos enviados [12].

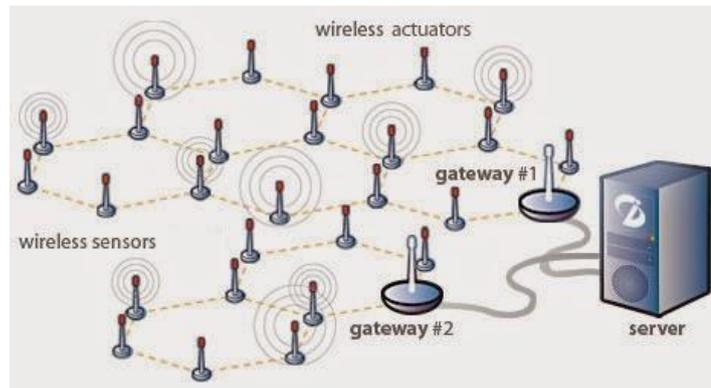


Figura 6. Crianza por zonas con una criadera. [15]

Internet de las cosas.

El internet de las cosas se enfoca en la evolución entre las relaciones de los objetos convencionales y las personas como se muestra en la Figura 7, de la misma manera es la digitalización del mundo físico. Cabe recalcar que la mayoría de los datos de los dispositivos que se encuentran en contacto con las personas, son transmitidos a la red, los cuales se sincronizan para mejorar la interacción con las personas y de la misma manera entre ellos. Mediante la utilización de sistemas embebidos es factible adquirir información a través de sensores que posteriormente serán conectados a una red, con una característica fundamental que es el monitoreo y control de manera remota que provocan acciones o eventos[20]. En la Figura 7 se puede observar las aplicaciones del internet de las cosas en la sociedad.

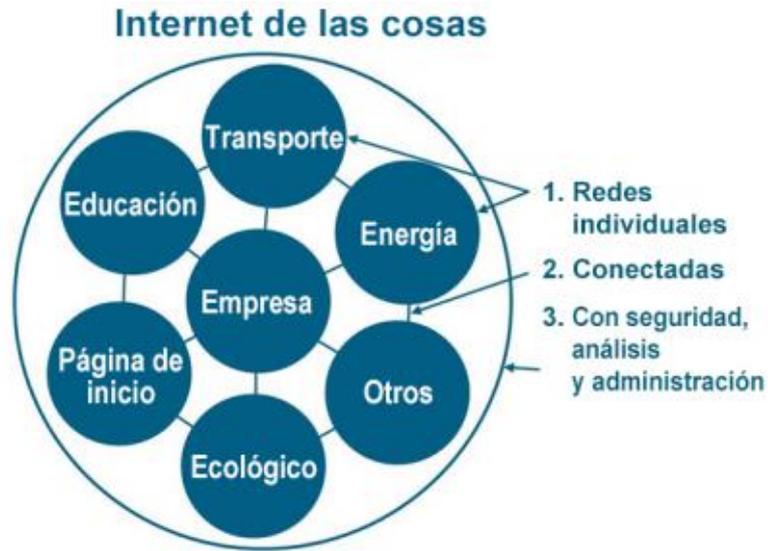


Figura 7. Internet de las cosas.[20]

Los dispositivos y las redes de comunicación para la implementación IoT cumplen con ciertos requisitos como se indica en la siguiente Tabla 4.

Tabla 4: Requisitos de un sistema IoT.

Dispositivos	Redes de comunicación
<ul style="list-style-type: none"> • Bajo consumo de energía • Reducido tamaño • Movilidad en las comunicaciones • Bajo coste de producción de los dispositivos finales 	<ul style="list-style-type: none"> • Servicio de localización • Comunicación segura • Redes de conexión de amplio alcance (varios Km) • Bajo coste de producción de los elementos de red

Elaborado por: El investigador

Fuente: [21]

LPWAN (Low Power Wide Area Network).

Esta red ha sido parte de la innovación tecnológica relacionada con el Internet de las cosas (IoT). La comunicación es uno de los pilares que han permitido la integración de sensores, programas y otras tecnologías para la transferencia de datos a una red. Actualmente, se han incorporado nuevos protocolos de tecnologías LPWAN, así como LoRa, NB-IoT y Sigfox. Cada uno de estos protocolos son considerados líderes, esto gracias a su gran accesibilidad, el bajo consumo de energética y en gran medida a la rentabilidad obtenido como un medio de comunicación. Algunos de los beneficios

que otorga este protocolo se basa en el bajo consumo energético y el gran alcance, así como en términos de latencia y calidad de servicio (QoS). Por lo cual, el uso de estos dispositivos es esencial en aplicaciones donde el envío de información no sea excesiva y sea necesario que la duración de la batería sea prolongada a lo largo de varios años [22].

Para la utilización de una red LPWAN existen varias tecnologías, las cuales pueden tener o no una licencia para su aplicación como se muestra en la Figura 8. Para un manejo de datos en tiempo real se aplican topologías basadas en LPWAN. En la siguiente imagen se muestra tecnologías LPWAN que pueden ser utilizadas principalmente para el diseño de una red de sensores para el envío de datos a bajo costo y a grandes distancias [19].



Figura 8. Tecnología LPWAN con y sin licencia.[19]

Varios protocolos o técnicas usadas son comparadas con tecnologías LPWAN. Algunos de los aspectos que son base para una comparación entre tecnologías de comunicación son el tiempo de vida de las baterías y el alcance de estos. La tecnología LPWAN tiene un mejor desempeño al comparar con otras tecnologías como Local Area Network (LAN), Personal Area Network (PAN) y un teléfono celular[19]. Esta comparación se aprecia en la Figura 9.

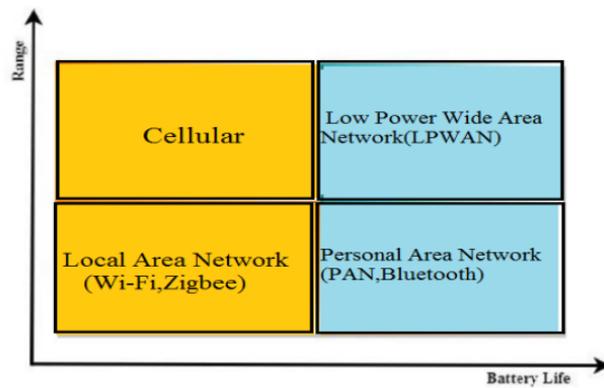


Figura 9. LPWAN surge como la técnica de IoT más rápida.[19]

Cabe recalcar que la Tecnología LPWAN es idónea para la transmisión de información y ampliamente utilizada para aplicaciones de monitoreo un ejemplo de ello en la agricultura inteligente. Otras tecnologías de comunicación como Bluetooth, Zig-Bee e identificación por radio frecuencia (RFID) tienen un menor rendimiento considerando el ancho de banda y la distancia [19].

LoRaWAN

LoRaWan es un protocolo de comunicación diseñado y desarrollado para optimizar el uso del protocolo de capa física de LoRa. En la Figura 10 se puede observar el funcionamiento de este protocolo y como brinda soporte a las capas superiores como de enlace, red y transporte. Este protocolo permite mejorar el funcionamiento y optimizar la vida útil de la batería de alimentación de los nodos, la capacidad de red, la calidad de servicio y la seguridad en la transmisión de los datos [7].

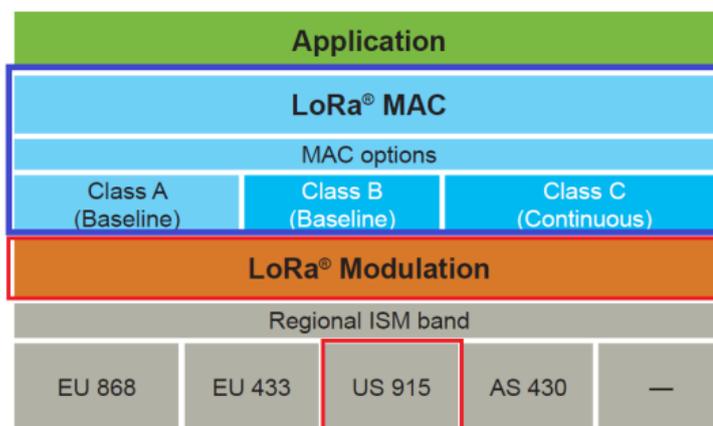


Figura 10: Protocolo de comunicación LoRaWAN. [23]

Capas de LoraWAN

LoraWAN se encuentra dividido en dos capas tanto física como mac las cuales se detallan en la Tabla 5 sus características.

Tabla 5: Características de las capas LoRaWAN.

Capa Física	Capa Mac
<ul style="list-style-type: none">• Establece conexión• Opera En la banda ISM• Opera debajo de 1Ghz• Modulación Lora para enlaces de largo alcance	<ul style="list-style-type: none">• Gestiona los canales y parámetros de conexión.• Dependiendo de la direccionalidad y los tiempos de espera existen tres clases de dispositivos clase A, B y C.

Elaborado por: El investigador

Fuente:[24]

Trama física LoraWAN: en la Figura 11 se puede observar la trama física de Lora

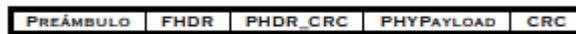


Figura 11:Tama LoraWAN [21].

Preámbulo o Preámbulo de sincronización: Define el esquema de modulación del paquete, lo cual permite:

- Definir los parámetros de modulación de forma sencilla.
- Definir los parámetros de forma individualizada para cada paquete.
- Indicar el esquema de transmisión para cada paquete en función de la distancia de la Gateway (el objeto puede estar en movimiento) y por lo tanto ajustar el rendimiento y consumo eléctrico según las circunstancias.

El preámbulo finaliza con un byte de sincronización lo que permite:

- Diferenciar entre las redes loras que están emitiendo en la misma banda de frecuencias.
- El dispositivo solo escuche las emisiones que tiene su mismo byte de sincronización.
- La concurrencia de distintas redes Lora en un mismo espacio.

Payload

- El tamaño máximo de 255 byte
- El tamaño está definido por un slo byte en la cabecera de la trama
- El tamaño de bytes es suficiente para la transmisión de geolocalización

PHYPayload

- Contiene la trama de la capa de acceso al medio LoraWAN

Trama Mac LoraWAN: en la Figura 12 se puede observar la trama Mac de Lora

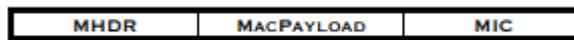


Figura 12:Trama Mac [19].

Cabecera de la trama Mac (MHDR): Se Identifica principalmente el tipo de mensaje que contine Mac Payload seguido del Campo de integridad (MIC)

Arquitectura de Red LoraWAN

LoraWAN posee una arquitectura de red como se puede observar en la Figura 13 la cual se detalla a continuación cada uno de sus componentes.

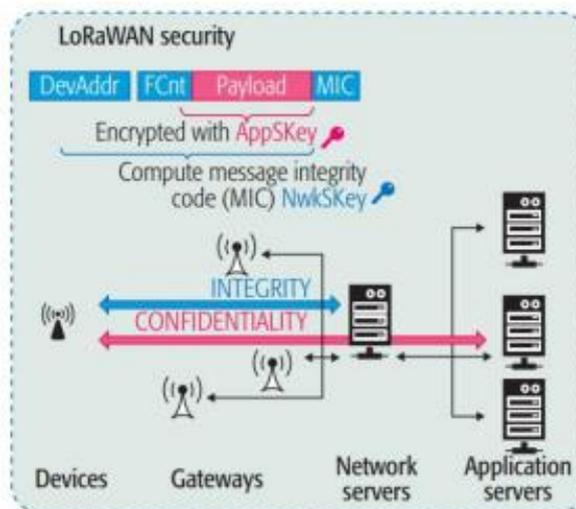


Figura 13:Arquitectura de re de Lorawan [24].

- **Dispositivos:** Dispositivos integrados de comunicación de baja potencia Encargados de recoger información y enviar al gateway [24].
- **Gateway:** pasarela de las transmisiones entre los dispositivos y servidores de red.

- **Servidor de Red:** Autentica los nodos, comprueba la integridad de los mensajes y los enruta a los servidores de aplicación.
- **Servidor de Aplicación:** Incorpora mecanismos de confidencialidad de mensajes y gestiona las solicitudes.

Clases de LoraWAN

LoraWAN tiene tres principales clases A, B y C como se observa en la siguiente Tabla 6 las cuales se detallan a continuación.

Tabla 6: Características de las clases de Lora Mac

Clase A	Clase B	Clase C
<ul style="list-style-type: none"> • La planificación de las TX corre a cargo del propio dispositivo final. • La recepción esta solo permitida después de una Tx completada correctamente. • Está formada por dos ventanas separadas de recepción. • Si la primera ventana ya ha recibido los datos la segunda ventana se deshabilita. • Bajo consumo energético. • Es la clase más eficiente. 	<ul style="list-style-type: none"> • Permite la creación de ventanas de recepción sin la necesidad que se produzca una TX previa. • Se aumenta la capacidad del dispositivo final de recibir datos. • Aumento de consumo eléctrico. 	<ul style="list-style-type: none"> • Esta en modo de recepción permanente • Se interrumpe la recepción solo cuando se transmite. • Presenta una mejor conexión entre los dispositivos finales y la Gateway a cambio de un alto consumo energético.

Elaborado por: El investigador en base a [21]

Capa Física de LoRa

El microcontrolador LoRa tiene una modulación de espectro ensanchado desarrollado y patentado por la empresa Semtech, con una velocidad de datos y sensibilidad de 125kHz o 500kHz cuando funciona en un canal que tiene un ancho de banda fijo, estos valores pueden ser para uplink o downlink. Adicionalmente esta nueva tecnología dispone de una optimización adaptativa de la potencia y la velocidad de transmisión de datos que basado en factores de dispersión ortogonales. Esta característica favorece el aprovechamiento de las baterías de alimentación.

En lo que se refiere al modelo OSI (Modelo de Interconexión de Sistemas Abiertos por sus siglas en inglés) la tecnología LoRa se encuentra en la capa física como se puede observar en la Figura 14. El medio de transmisión es el aire, por donde viajan las ondas RF (Radio Frecuencia) desde un dispositivo LoRa hasta un Gateway o viceversa [23].

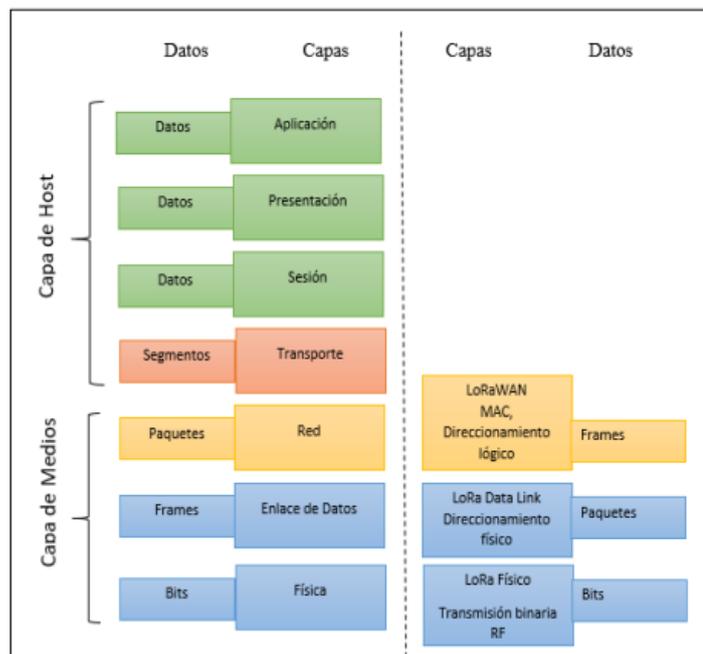


Figura 14. Modelo OSI para la tecnología LoRa. [23]

Por otro lado, en la capa física se utiliza al *bit* como la unidad básica de información, en la capa de enlace de utilizan los *frames* que son básicamente la unión de varios bits en paquetes. Entonces, en la capa física se especifican las normas eléctricas, físicas funciones y procedimientos que van aplicarse, también se especifican los voltajes, frecuencias y ancho de pulso para establecer la comunicación entre nodos. Sin

embargo, la capa de enlace es la encargada de mover las tramas de datos entre los nodos adyacentes que componen una red y la capa física mueve los bits entre los nodos adyacentes de la red [23]. En la Figura 15 se puede observar una lista de modulaciones utilizadas para transmitir la misma información digital, en este caso se pueden observar las siguientes:

- ASK: Modulación por desplazamiento de amplitud.
- BFSK: Modulación por desplazamiento de frecuencia, Binario.
- BPSK: Modulación por desplazamiento de fase, Binario.

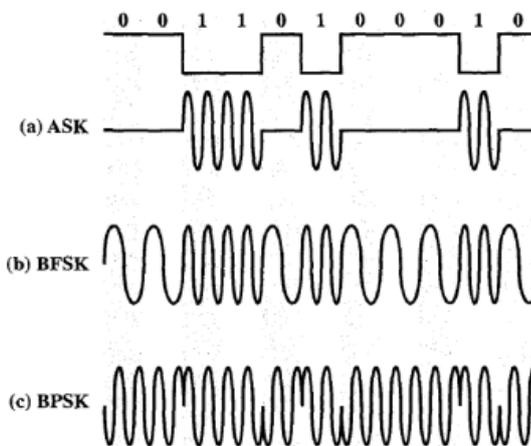


Figura 15. Diferentes modulaciones para una misma información. [23]

Los objetivos principales de esta capa son:

- Definir el medio físico por donde viaja la comunicación
- Definir las características de materiales y eléctricas que se implementarán en la comunicación
- Definir características funcionales de la interfaz

Finalmente, las funciones que debe ejecutar el *hardware* en esta capa son las siguientes:

- Transmitir el flujo de bits a través del medio
- Manejar las señales eléctricas del medio de transmisión
- Garantizar la conexión

Modulación LoRa

La modulación LoRa puede ser entendida o tiene un funcionamiento similar a la *Multiple Frequency Shift Keying* (MFSK). La modulación MFSK es una técnica de transmisión de datos digitales que utiliza diferentes tonos para transmitir información, estos tonos pueden ser paralelos o concurrentes, secuenciales y una combinación de varios tonos. Por otro lado, LoRa también está basado en la modulación CSS (Spread Code), básicamente esta modulación se trata de un pulso *chirp* que sucede a una frecuencia más alta que la frecuencia de la información que se desea transmitir; esto produce un esparcimiento del ancho de banda (BW) de la señal transmitida. Este tipo de modulación ha sido muy utilizada en aplicaciones espaciales debido a su robustez y seguridad en la transmisión de la información. Sin embargo, la tecnología LoRa es una de las primeras aplicaciones de bajo costo para uso comercial y educativo [23].

La tecnología LoRa es más aplicada con la utilización del tráfico *uplink* que su contraparte *downlink*; debido que los dispositivos están pensados para que envíen información hacia un *Gateway* y de esta manera visualizar los datos a través del internet o mediante el acceso a la red implementada. Las bandas que puede utilizar LoRa están delimitadas por la región donde se van implementar estas pueden ser de 433 MHz, 868MHz o 915 MHz [23].

CSS (Chirp Spread Spectrum)

La técnica de modulación CSS utilizada por LoRa en la capa física, consiste en una variación lineal de la frecuencia en el tiempo con el objetivo de codificar la información que se desea transmitir, como se muestra en la Figura 16. Esta variación lineal es igual tanto para el transmisor como para el receptor y equivalentes al desfase en el tiempo de la señal, lo cual facilita la decodificación y la señal es inmune a los efectos Doppler [25].

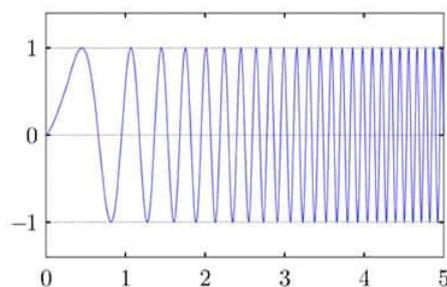


Figura 16: Relación frecuencia vs tiempo de la modulación CSS. [25]

Pulsos Chirp

Se puede definir a la modulación de pulsos como un tipo modulación de espectro ensanchado con acceso múltiple por división de códigos. Como se explicó antes su funcionamiento se basa en la utilización de pulsos chirp, que incrementan o disminuyen su frecuencia durante un periodo de tiempo, como se muestra en la Figura 17.

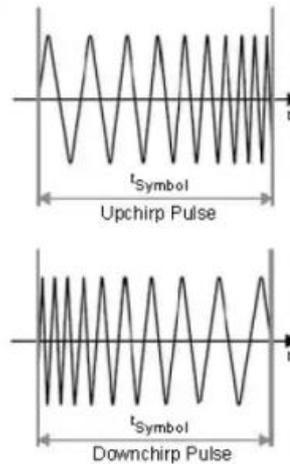
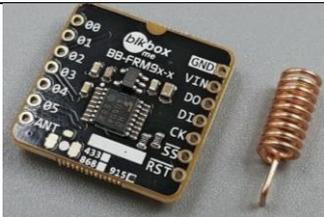
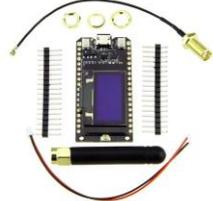


Figura 17: Pulsos *upchirp* y *downchirp*. [23]

Lora

Tabla 7. Microcontroladores con tecnología LORA.

Modelo	Lora Sx1276	ESP32	Lora Esp32 Sx 1278
Fotografía			
Voltaje	3-5 Vdc	3.3 Vdc	3.3Vdc a 7 Vdc
Frecuencia	915 Mhz	80 a 240 Mhz	868 – 915 Mhz
Potencia	100 mW	900 mW	1100 mW
Consumo Corriente	9.9 mA	Wifi 180 mA	200 mA
Tranceiver Lora Incluido	Si	No	Si
LCD	No	No	Si

Costo	20 USD	30 USD	40 USD
-------	--------	--------	--------

Fuente: Investigador

Sensor DTH

Tabla 8. Comparación de módulos DHT disponibles en el mercado.

Modelo	DHT 11	DHT 22
Representación		
Voltaje (Vdc)	3-5	3-6
Humedad relativa	5%	5%
Temperatura	0°C a 50°C	-40°C a 80°C
Numero de pines	4	4
Tiempo de medición (s)	2	1

Fuente: Investigador

Sensor de Lúmenes

Tabla 9. Comparación de módulos de lúmenes.

Módulo Lúmenes		
Modelo	LM393	LDR
Representación		
Voltaje	5 Vdc	5 Vdc
Numero de pines	4 pines	4 pines
Salidas	Analógico – Digital	Analógico
Precisión de voltaje	2.0 mV	3.0 mV

Fuente: Investigador

Sensor de CO2

Tabla 10. Comparación de módulos de CO2.

Módulo CO2		
Modelo	MG-811 V1	MG-811 V2

Representación		
Voltaje	5 Vdc	5 Vdc
Numero de pines	4 pines	5 pines
Salidas Análoga	2.7 – 4.1 V	2.7 – 4.1 V
Salida Digital	Alarma	Alarma
Rango	0 a 10000 ppm	0 a 10000 ppm
Precisión	+/- 100 ppm	+/- 100 ppm

Fuente: Investigador

Relé

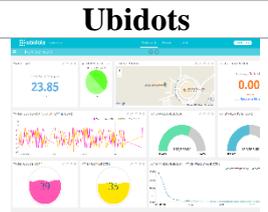
Tabla 11. Comparación relé de activación.

Módulo Relé			
Modelo	Relay 2CH	G3MB-202P	SSR-60 DA
Representación			
Voltaje	5 Vdc	5 Vdc	3 - 32 Vdc
Corriente	20 mA	160 mA	50 mA
Carga Salida	250 Vac	240 Vac	380 Vac
Corriente Salida	10 Amp	2 Amp	60 Amp
Frecuencia	50/60 Hz	50/60 Hz	50/60 Hz
Funcionamiento	ON/OFF	ON/OFF	ON/OFF
Canales	2	2	1

Fuente: Investigador

Servidor IOT

Tabla 12. Comparación entre servidores IoT.

Servidor IoT			
Modelo	Thinger IO	ThingSpeak	Ubidots
Representación			

Dispositivo	2	4	20
Software	Abierto	De pago	De pago
Muestreo	1 segundo	15 segundos	1 segundo
Mensajes	0	3 millones/año	10 SMS / 1
Tiempo	Ilimitado	30 días	30 días
Precio	Gratis	45/año	49/mes

Fuente: Investigador

1.3 Objetivos

1.3.1 Objetivo General

- Implementar un sistema de monitoreo inteligente para la avícola “San Miguel” con una red de sensores inalámbricos basados en la tecnología LPWAN (Low Power WideArea Networks).

1.3.2 Objetivos específicos

- Analizar las principales magnitudes físicas que afectan a la crianza de los pollos de un día hasta las tres semanas.
- Diseñar un sistema inteligente de monitoreo con una red de sensores inalámbricos bajo la tecnología LPWAN.
- Implementar el sistema de monitoreo con visualización de variables en un HMI y un servidor IoT.

CAPITULO II

METODOLOGÍA

2.1 Materiales

En el presente proyecto se realizó una selección de materiales de acuerdo a investigaciones anteriores tales como artículos científicos, proyectos de investigación con características similares. Además, se efectuó una selección de los módulos de comunicación LPWAN.

2.2 Métodos

Modalidad de la Investigación

La finalidad del presente proyecto es implementar un sistema de monitoreo inteligente para la avícola “San Miguel” con una red de sensores inalámbricos basados en la tecnología LPWAN (Low Power WideArea Networks), por ese motivo se aplicó las siguientes modalidades de investigación:

Investigación Aplicada

El presente se ha definido como una investigación aplicada porque se ha utilizado los conocimientos adquiridos durante la formación académica diseñar, construir e implementar un sistema de monitoreo inteligente para una avícola.

Investigación Documental

Se utilizó la investigación documental para la indagación, interpretación y presentación de datos e información sobre los tipos de tecnología de comunicación inalámbrica y el protocolo de comunicación LPWAN que se puede considerar dentro del IoT. También se debe recabar información sobre las condiciones ambientales ideales para el desarrollo de pollos de 1 día hasta tres semanas de edad y de esta manera realizar un adecuado diseño de la red de sensores.

Investigación de Campo

La Investigación de Campo estuvo dirigida netamente a analizar el proceso de crianza de pollos en la granja “San Miguel” considerando etapas: recepción, instalación, acondicionamiento, alimentación, y monitoreo. Además, se ha mantenido una

comunicación constante con los operarios de la granja para conocer todos los detalles sobre el proceso de crianza.

Investigación Experimental

Por otra parte, también se realizó una Investigación Experimental para desarrollar un diseño e implementación de un sistema de monitoreo inteligente mediante una red de sensores inalámbricos en la granja avícola.

Recolección de Información

Para lograr la recolección de información se empleó libros, revistas, fuentes online y proyectos desarrollados, así como guías prácticas y manuales de construcción por lo que se tomará en cuenta bases de datos confiables que permitan el desarrollo del proyecto.

2.3 Procesamiento y Análisis de Datos

El procesamiento y análisis de datos se realizará considerando los siguientes pasos:

- Revisión de información recopilada.
- Análisis de la información y características de las avícolas inteligentes
- Planteamiento de la solución a la falta de monitoreo de variables físicas en la crianza de pollos.
- Control y verificación de los datos obtenidos mediante la depuración del sistema.

2.4 Desarrollo del proyecto

La presente investigación se realizó, en base a las siguientes actividades:

1. Recopilación de información sobre las condiciones ideales para la crianza de pollos y análisis del proceso actual de crianza de la granja “San Miguel”
2. Análisis de las funcionalidades de las avícolas inteligentes, equipos con tecnología LPWAN disponibles en el mercado ecuatoriano y la tecnología LPWAN.
3. Diseño de un sistema de monitoreo y red de sensores inalámbricos de magnitudes físicas para los galpones.

4. Establecimiento de una red maestro esclavo entre los sensores y la central de monitoreo.
5. Programación de un HMI.
6. Investigación y desarrollo de página web para los análisis estadísticos de los datos subidos a la nube.
7. Pruebas de funcionamiento de cada uno de los sensores para medición de temperatura, humedad, lúmenes y CO₂.
8. Pruebas del funcionamiento del HMI y comunicación entre los sensores.
9. Instalación del sistema de monitoreo en la granja avícola.
10. Corrección del funcionamiento y reprogramación.
11. Análisis de los resultados y conclusiones obtenidas durante el desarrollo del proyecto.
12. Elaboración del informe final.

CAPITULO III

RESULTADOS Y DISCUSIÓN

El siguiente capítulo presenta el desarrollo del proyecto de investigación, donde describimos a profundidad cada uno de las partes que conforman el sistema de monitorización preventiva, tales como: Dispositivos electrónicos, software, lenguaje de programación, etc. A la vez se presentan los resultados obtenidos de acuerdo a cada una de las pruebas desarrolladas en el campo.

Análisis y Discusión de los Resultados

Hoy en día la implementación de sistemas de monitoreo inteligentes para mejor control de variables tales como: temperatura, humedad, luz, etc. Se han vuelto una necesidad debido a los grandes beneficios que estas prestan como son: el ahorro de tiempo, monitorización constante en base a la generación de notificaciones en caso de alguna anomalía, la recolección de datos y análisis de los mismos, acceso remoto desde cualquier lugar del mundo, etc.; tal es el caso que la implementación de un sistema de monitoreo de alerta temprana es la mejor opción para un galpón de crianza de pollos debido a la naturaleza delicada que presentan este tipo de aves.

El presente proyecto se desarrolló utilizando software y hardware libre y de paga, dispositivos electrónicos que se encuentran en el mercado local. Cabe destacar que el servidor IOT a utilizar es Thingier IO medianamente gratuito y de fácil manejo, compatible con otros programas tanto libres como de paga. Los módulos implementados son de fácil instalación y amigable con el usuario.

Desarrollo de la propuesta

Proceso de crianza en Avícola San Miguel

En la avícola San Miguel se pueden distinguir diferentes etapas dentro del proceso de crianza de pollos, los mismos que se detallan en la Figura 18.

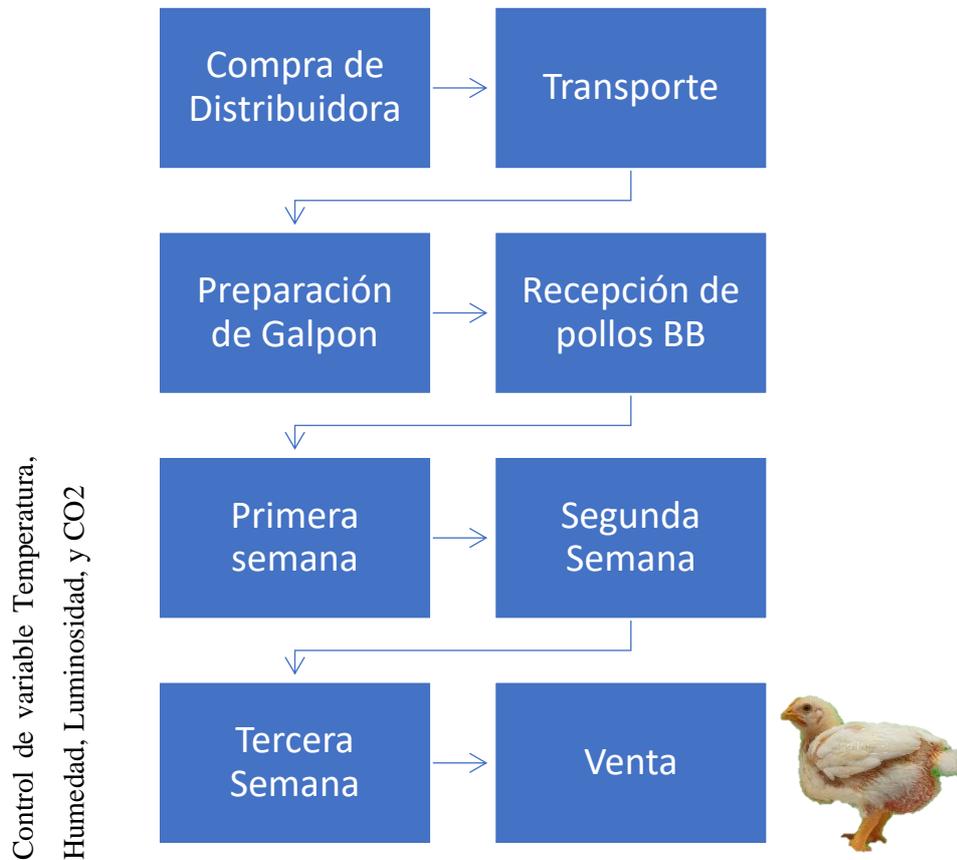


Figura 18: Diagrama de las etapas de crianza.

Fuente: Investigador

Compra: La avícola san Miguel realiza una compra a la Distribuidora Aviment ubicada en la ciudad de Riobamba, el cual los transporta los pollos hasta la ciudad de Salcedo donde los operarios de la planta de crianza se encargan de recibirlos.

Transporte: Una vez realizado el pedido en la Distribuidora Aviment, los pollos son enviados a la ciudad de Salcedo directamente hacia los galpones de la avícola San Miguel respetando las siguientes normas:

- Cantidad exacta de pollos de un día por cada uno de los cartones de transporte para evitar asfixia.
- Sistema de seguridad con cuerdas en los cartones para evitar pérdida de los pollos por caída o golpes.

- Monitoreo cada cierto tiempo de la carga para verificar que el viaje se lleve con normalidad.
- La entrega es personal y recibida por el operador de la avícola.
- El personal se asegura que la carga este normal y no haya perdidas.

Preparación de Galpón: La avícola San Miguel consta de 6 galpones de diferentes capacidades los cuales no pueden ser usado en su totalidad, debido a que estos deben permanecer vacíos un tiempo prudente de una semana para después ser limpiados y esterilizados con diversos químicos. Esta preparación se realiza siempre que una remesa nueva necesite el espacio.

Recepción de Pollos de un día: Son recibidos por el personal de la avícola San Miguel para ser ubicado en uno de los galpones previamente preparados. En el momento que se liberan los pollos los comederos deben estar surtidos de maíz y los bebederos llenos con agua con azúcar para que estos se ambienten.

Primera Semana: Es crucial debido a que en esta semana los pollos bebes son más propensos a enfermedades y por ende a un alto índice de decesos. Los factores que afectan a los pollos son: falta de vacunas, exceso de calor, exceso de humedad, exceso de luz, exceso de gases pestilentes, entre otros. Su alimentación se realiza mediante balanceada inicial 1 y en el agua se vierte endroxil para prevenir toda clase de enfermedades, esto se repite por tres días. Posteriormente, hasta que cumplan 7 días se agrega en los bebederos una vitamina que refuerce la vitalidad de los pollos manteniendo la comida.

Segunda Semana: En esta semana se aplica las vacunas de Bumboro y Newcastle a cada uno de los pollos bebé, la comida se mantiene constante hasta el día 14 con su respectiva monitorización de temperatura humedad y gas.

Tercera Semana: La comida se cambia de inicial 1 a inicial 2 hasta el día 18, continuando con la monitorización de temperatura humedad y gas.

Venta: La comida se cambia a balanceado de engorde hasta el día de su venta, esta puede ser bajo pedido o en el local comercial de la avícola en los días de feria ya sea este jueves o domingos.

VARIABLES DETECTADAS

Las variables más importantes para un correcto funcionamiento del galpón de pollos durante el proceso de crianza son: Temperatura, humedad, lúmenes y la cantidad de oxígeno presente en el galpón. En la Tabla 13 se realiza un análisis de estas variables detectando sus ventajas y desventajas.

Tabla 13. Variables detectadas con su ventaja y desventaja.

Variable	Ventaja	Desventaja
Temperatura	<ul style="list-style-type: none">• Evitar enfermedades como la gripe• Mejora el sistema inmunológico	<ul style="list-style-type: none">• Consumo de cilindro de gas.• Generación de CO2• Produce sed en las aves
Humedad	<ul style="list-style-type: none">• Mejora el crecimiento de las aves.• Mantiene el sistema digestivo de las aves funcionando correctamente.	<ul style="list-style-type: none">• Asfixia en las aves.• Un ambiente seco produce la muerte en las aves.
Lúmenes	<ul style="list-style-type: none">• Agiliza el proceso de crianza de las aves.	<ul style="list-style-type: none">• Aumento de alimento en las aves.• Consumo energético elevado.
Oxígeno	<ul style="list-style-type: none">• Mejora la respiración en las aves.• Carne de buena calidad.	<ul style="list-style-type: none">• Asfixia en las aves.• Intoxicación de la carne.

Fuente: Investigador

Una vez analizado la condición en la que se encuentra la avícola se pretende realizar un sistema de monitoreo inteligente el cual busca mejorar las condiciones ambientales que son factores clave para el buen crecimiento avícola, para esto se debe realizar la arquitectura y diseño de dicho sistema, como a continuación se detalla.

3.1.1.2 Arquitectura del sistema de medición

El sistema de medición mediante la implementación de sensores inteligentes está basado en un modelo de protocolo de comunicaciones Maestro-Esclavo. Este modelo consiste en la configuración de un dispositivo como maestro, el mismo que será el encargado de controlar a uno o más dispositivos secundarios o esclavos [26]. En la Figura 19 se puede identificar arquitectura típica de sistema de monitoreo mediante la tecnología LoRa.



Figura 19: Arquitectura típica de un sistema Lora.
Elaborado por: El Investigador

En la Figura 20 se observa la arquitectura del sistema de monitoreo de la granja avícola “San Miguel”

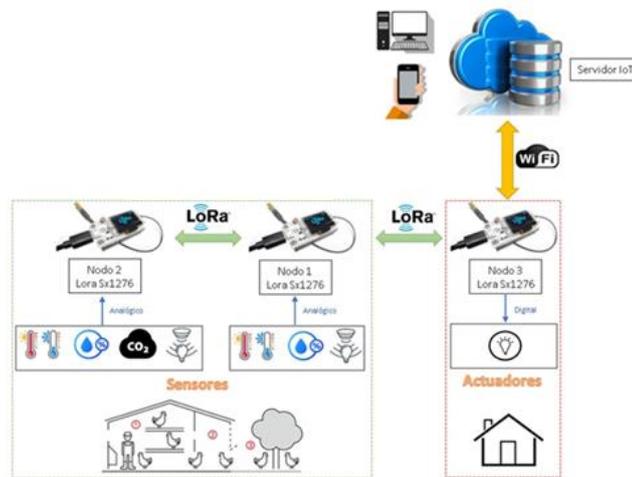


Figura 20: Arquitectura del sistema de monitoreo

En la Tabla 14 se da a conocer la arquitectura con la cual se ha desarrollado el sistema de monitoreo para los galpones de la granja San Miguel.

Tabla 14. Descripción de arquitectura.

Número de nodo	Función
Nodo 1	Son los encargados de la adquisición de las variables físicas (Temperatura humedad Lúmenes)
Nodo 2	Son los encargados de la adquisición de las variables físicas (Temperatura humedad Lúmenes, Co2)
Gateway	Es la encargada de recibir los datos tratados en los nodos finales para después envíalos al servidor.
Servidor IoT	Es el lugar donde se almacena la información adquirida.
Equipos Finales	Son los dispositivos en los cuales van a obtener la información de forma práctica y amigable con el usuario.

Diseño de Nodo final

El nodo final es el encargado de la adquisición de la información mediante un determinado número de sensores asociados a un microcontrolador, el cual mediante algoritmos transforma una señal física en una señal que entienda los dispositivos pertinentes, en el siguiente apartado se realizó una comparativa entre dispositivos para poder seleccionar los mejores elementos para dicho nodo.

Selección del Microcontrolador

En esta sección se realiza una comparación de los dispositivos de comunicación inalámbrica, de bajo costo y bajo consumo de energía. Dentro de estas tecnologías se ha podido detectar los módulos Lora SX1276, ESP32 y Lora ESP32. Esta tecnología permitirá desarrollar la red maestro esclavo entre los sensores de monitoreo y el dispositivo de visualización.

El medidor inalámbrico de temperatura, humedad, Co2 y lúmenes contara con el módulo Lora ESP32 por ser compacto y de bajo consumo. La información recolectada por este dispositivo se enviará al dispositivo LORA ESP32 configurado como maestro y, este a su vez enviará la información de cada uno de los sensores hacia un dashboard desarrollado en un servidor IoT. Entonces el módulo seleccionado para la comunicación es la Lora ESP32 (Maestro y Esclavo), en la Tabla 15 se puede observar los datos técnicos de dicho módulo.

Tabla 15. Datos Técnicos del Módulo LORA ESP32 WiFi.

Datos Técnicos	
Voltaje de alimentación:	Usb 5v, batería 3,7V
Frecuencia LoRa:	915 MHz
Rango de temperatura de trabajo:	40 ° C a + 90 ° C
Wifi:	2,4Ghz
Bluetooth:	4,2 con BLE
Conector JST GH 1,25mm	dos pines para batería
Interfaces de comunicación:	UART x3 SPI x2 I2C x2 I2S x1 Entrada ADC de 12 bits x18

	Salida DAC de 8 bits x2 GPIO x28 GPI x7 Flash SPI 8MB SRAM 520Kb Pantalla Oled de 0,96" - 128x64 pixeles
Velocidad de datos:	150 Mbps @ 11n HT40, 72 Mbps @ 11n HT20, 54 Mbps @ 11g, 11 Mbps @ 11b
Potencia de transmisión:	19,5 dBm @ 11b, 16,5 dBm @ 11g, 15,5 dBm @ 11n

Fuente: Investigador

Selección de Sensor de Temperatura y Humedad

Los sensores utilizados para medición de temperatura y humedad son de tipo capacitivos y cuentan con un termistor que mide el aire circundante y puede obtener los datos de forma digital para ser enviados hacia un controlador. En la Tabla 8 se muestran las dos versiones de este dispositivo existentes en el mercado local.

Se opta por la opción del sensor DHT22 debido a que tiene un rango de medición mayor de -40°C a 80°C y es suficiente para medir la temperatura ambiente del galpón donde se encuentran los pollos recién llegados a la avícola. La humedad relativa es igual en los dos casos por lo cual no es una característica decisiva. El DHT 22 puede realizar mediciones de las variables cada segundo, volviéndolo más rápido que el sensor DHT11. En la Tabla 16 se puede observar los datos técnicos de este tipo de sensor.

Tabla 16. Datos técnicos del sensor DHT22.

Datos Técnicos	
Alimentación	3 a 5VCD y E / S
Consumo de	2.5mA máximo durante la conversión
Humedad de	0-100% con 2-5% de precisión
Temperatura de	40 a 80 °C con $\pm 0,5^{\circ}\text{C}$ Precisión
Muestreo máximo de	0,5 Hz (una vez cada 2 segundos) 4 pines, 0.1" de separación

Fuente: Investigador

Selección de Sensor de lúmenes

Este tipo de sensor utiliza una foto resistencia para determinar la cantidad de lúmenes existentes en una zona, mediante el potenciómetro lineal se realiza una comparación de voltaje para activar o desactivar la salida digital del sensor. Además, convierte la cantidad de lúmenes en un voltaje entre 0 y 5 Vdc, con una resolución aproximada de 2mV dependiendo del tipo de sensor. En la Tabla 9 se muestra una comparación entre los dos sensores existentes en el mercado ecuatoriano.

En este caso se opta por utilizar el sensor LM393 debido a que tiene una resolución menor al sensor LDR esto servirá para que el dispositivo de medición sea más preciso. Por último, este sensor tiene dos tipos de salidas, analógicas y digitales, permitiendo configurarlo de acuerdo a las necesidades del proyecto. En la Tabla 17 se puede observar los datos técnicos de este tipo de sensor.

Tabla 17. Datos técnicos del sensor LM393.

Datos Técnicos	
Voltaje máximo:	5 Vdc
Potencia:	90 mW
Temperatura:	-25°C ~ 75°C
Resistencia a 10 Lux:	mayor 50 K ohmios
Resistencia a 0 Lux	mayor 0,9 M ohmios

Fuente: Investigador

Selección de Sensor de CO2

Este sensor es utilizado para determinar la calidad de aire en una zona específica, tienen una buena sensibilidad y selectividad al Co2. Tiene una dependencia de baja humedad y temperatura, es muy resistente para ambiente difíciles. También es capaz de detectar la concentración de CO2 con respecto a la temperatura. Actualmente existen diversas formas de encapsular este sensor en un módulo para comunicarlo con Arduino. En la Tabla 10 se muestran los módulos más comerciales y que tienen una disponibilidad en el mercado local.

En este caso se pudo determinar que los dos sensores tienen las mismas características, sin embargo, entre las versiones de los se optó por la utilización del módulo MG-811 V2 porque es más fácil de conseguir en el Ecuador. En la Tabla 18 se pueden observar los datos técnicos de este sensor.

Tabla 18. Datos técnicos del módulo MG-811.

Datos Técnicos	
Se basa en el principio de electrolito sólido	
Muy buena sensibilidad al CO ₂	
Rango de detección:	350 – 10.000
Consumo	140 ± 20 mA.
Temperatura de funcionamiento:	20 – 50 °C.
BAJA dependencia de temperatura y humedad.	
Alta estabilidad y repetibilidad.	
Baja sensibilidad a CO y Alcoholes.	
Rapidez de respuesta.	

Fuente: Investigador

Gateway

La Gateway es la encargada de recibir los datos de los nodos finales para después enviar al servidor IoT, también es el encargado de enviar las señales preventivas en el caso de ser necesario para lo cual se utilizó relés y un microcontrolador como lo especificamos en el apartado de selección de microcontrolador, donde se seleccionó el microcontrolador Lora SP32.

Selección de Actuadores tipo Relé

En lo que se refiere al sistema de alarmas se requiere de salidas digitales del tipo relé para activar señales luminosas LED de 120 voltios. Este tipo de elementos sirven para evitar el consumo energético en el caso que se presente una señal de alarma durante un tiempo prolongado. En el mercado local se dispone de diferentes tipos de relés dentro de los cuales se puede identificar a los electromagnéticos y los relés de estado sólido. En la Tabla 11 se puede observar una comparación entre los relés disponibles en el mercado local, donde se puede evidenciar 3 tipos de módulos: relé

electromagnético de dos canales, relé SSR de dos canales y finalmente un relé de estado sólido con activación SSR pero 1 canal.

En este caso se decidió utilizar el módulo Relay 2CH debido a que dispone de pines de activación individuales para cada una de las cargas, además requiere de una corriente muy baja para su activación (20mA) lo que facilita su compatibilidad con cualquier microcontrolador. Los datos técnicos del módulo Relay 2CH se muestran en la Tabla 19.

Tabla 19. Datos técnicos del módulo Relay 2CH.

Datos Técnicos	
Voltaje de operación	5 Vdc
Señal de Control	3.3 o 5 Vdc
Canales	2
Capacidad Max	10A/250Vac o 10A/30Vdc
Corriente Max	10A (NO) o 5A (NC)
Tiempo de acción	10ms / 5ms
Entradas optoacopladas	
Indicadores tipo LED para cada canal.	

Fuente: Investigador

Servidor IoT

Hoy en día se puede acceder a un sin número de opciones en lo que se refiere a servidores IoT como se puede observar en la Tabla 12 para visualización y control de dispositivos inteligentes a través del internet. Varias plataformas que ofrecen este tipo de servicios tienen un costo mensual por utilización. Sin embargo, existen otro tipo de servidores que ofrecen los mismos servicios, pero de forma gratuita, a continuación, se realiza una comparación técnica entre los servidores de IoT más utilizados considerando costos, conectividad, funcionalidad y apariencia.

Para el desarrollo de este proyecto de tesis se decidió utilizar el servidor IoT llamado Thinger IO debido a que es una plataforma de código abierto y no tiene un costo de mantenimiento. Aunque la plataforma está limitada por el número de dispositivos que se pueden conectar simultáneamente, en el presente diseño se requiere de un punto de conexión debido a que se estableció una red maestro esclavo con todos los sensores inteligentes.

Diseño de Nodos Finales

Representación del módulo LoRa

El proceso de creación de elementos electrónicos inicia con la identificación y disposición de todos los pines según el fabricante, en este caso el módulo LoRa 32 desarrollado por la empresa Heltec tienen la configuración que se muestra en la Figura 21, dispone de 36 pines.

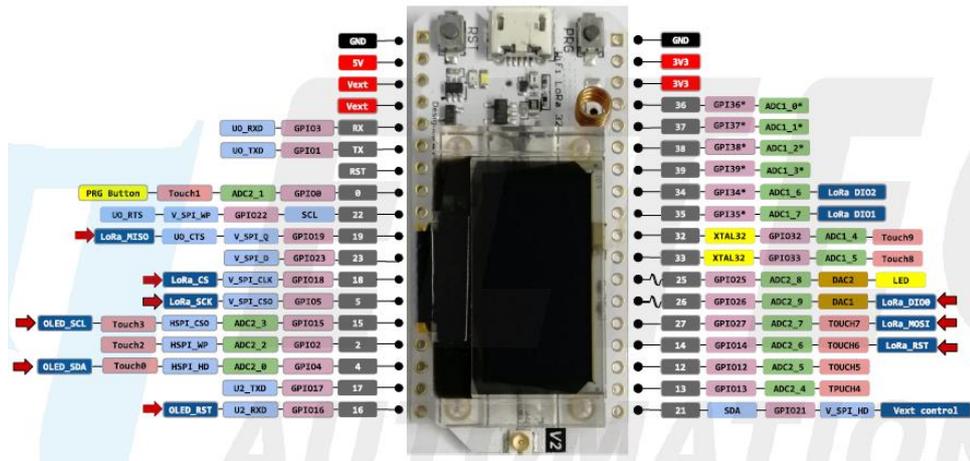
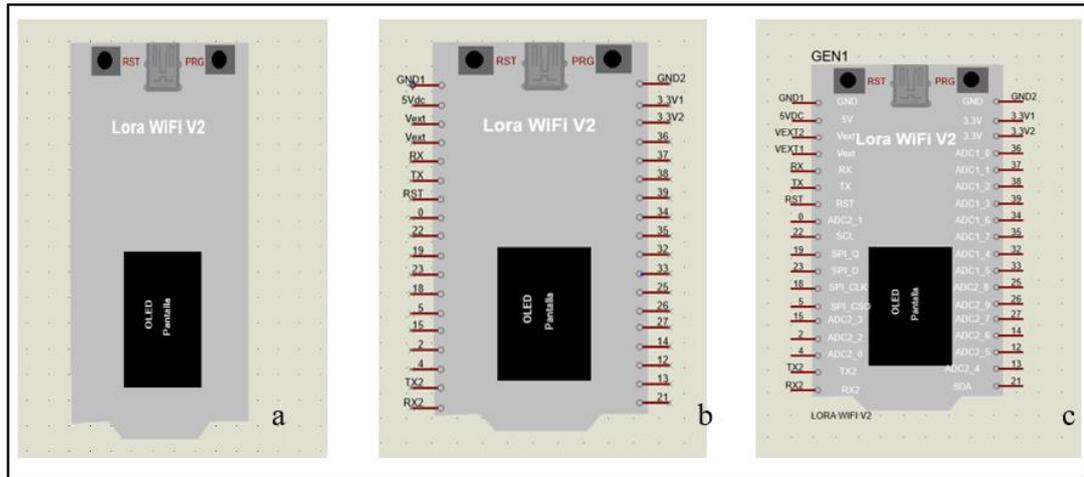


Figura 21. Configuración de pines del módulo Heltec LoRa 32.

Fuente: Investigador

El diseño de los nodos finales se realizó en el software Proteus, debido a que hoy por hoy es un software bastante completo, gracias al extenso número de librerías que componen al mismo.

Una vez identificado la disposición de los pines del módulo LoRa se procede a utilizar rectángulos de diferentes colores con el objetivo de generar una representación visual muy parecida a la realidad, En la Figura 22 sección a, b, c, se puede observar el proceso para el diseño esquemático del módulo LoRa.



a b c

Figura 22. a) Representación visual lora b) Configuración de pines c) Diseño esquemático
 El siguiente paso es crear una representación física del módulo LoRa que será utilizado en el diseño de los circuitos impresos. Se obtiene las medidas físicas del módulo mediante la utilización de un calibrador o Pie de Rey, 26 x 52 cm de ancho y largo, respectivamente. En la Figura 23 sección a, b, c se puede observar las medidas y la representación física del módulo LoRa.

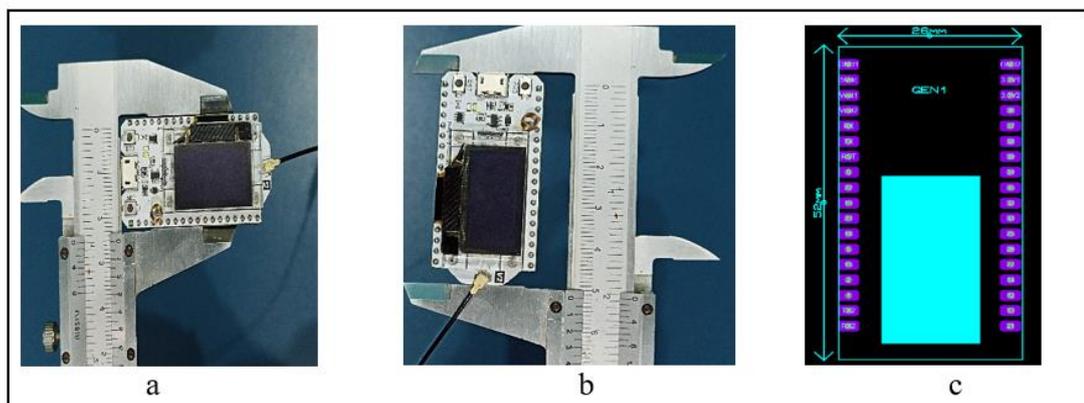


Figura 23.a) b) Medidas reales del módulo LoRa c) Representación física del módulo LoRa.
 Fuente: Investigador

Circuito Impreso del Nodo 1

En el diseño de los circuitos impresos de los sensores inteligentes se debe tener un especial cuidado de los voltajes de funcionamiento, debido a que el módulo LoRa 32

funciona a 3.3 voltios y sus pines no pueden soportar un voltaje superior a este valor. En la Tabla 20 se puede observar los voltajes de operación de cada uno de los sensores y del módulo LoRa.

Tabla 20. Requerimientos del Nodo 1.

Requerimientos	
Voltaje de operación	Valor
Entrada LoRa	5Vdc
DHT22	3.3Vdc
LDR	3.3Vdc
Reset	3.3Vdc

Fuente: Investigador

A continuación, se procede a ubicar conectores del tipo con-sil de 2, 3 y 4 pines señalados con etiquetas correspondientes a los sensores, fuente de alimentación, pulsadores y salidas extra. Adicionalmente se utilizan pines tipo output/input para realizar las conexiones entre todos los elementos electrónicos del circuito, en la Figura 24 se puede observar el esquema electrónico del Nodo 1.

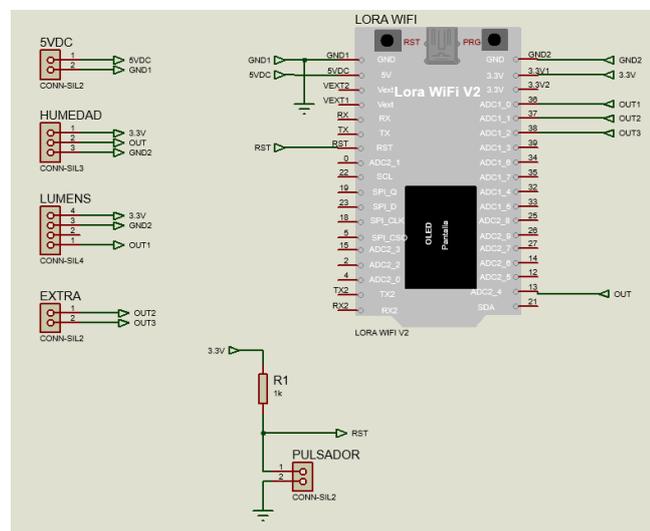


Figura 24. Esquema electrónico del Nodo 1.

Fuente: Investigador

Utilizando el software Proteus En esta sección del diseño se debe considerar la cercanía desde el microcontrolador hacia los conectores, para reducir al máximo el número de pistas requeridas por el circuito.

Finalmente, se obtiene la visualización en 3D del circuito impreso diseñado, donde se puede verificar la correcta disposición de todos los elementos electrónicos.

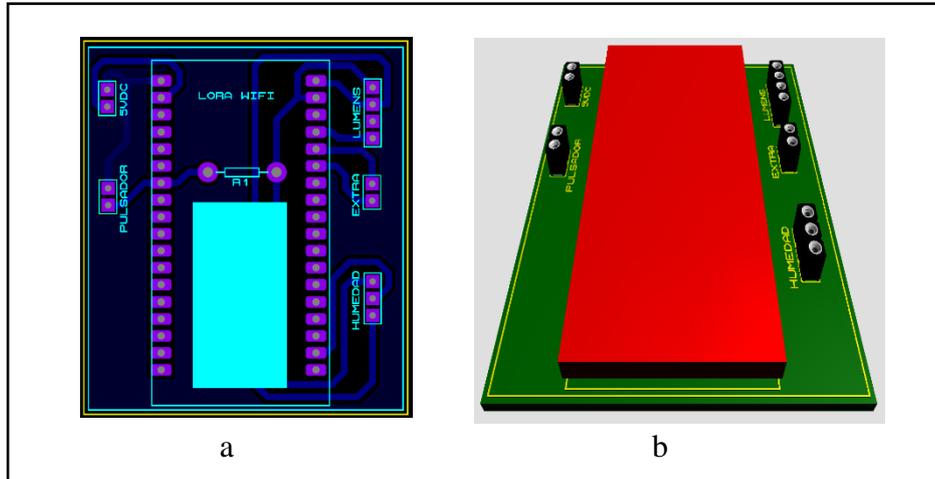


Figura 25. a) Diseño del circuito impreso del Nodo 1 b) Diseño 3D de la PCB del Nodo 1.

Fuente: Investigador

Circuito Impreso del Nodo 2

En el caso del nodo 2 se debe considerar la utilización de voltajes entre 3.3Vdc y 5Vdc para cada elemento electrónico, por lo tanto, se estable los requerimientos del circuito en la Tabla 21.

Tabla 21. Requerimientos del Nodo 2.

Requerimientos	
Voltaje de operación	Valor
Entrada LoRa	5Vdc
DHT22	3.3Vdc
LDR	3.3Vdc
Reset	3.3Vdc
MG811	5Vdc

Fuente: Investigador

Se aplica el mismo proceso utilizado para el diseño esquemático del nodo 1 en el software Proteus utilizando conectores del tipo con-sil de 2, 3 y 4 pines para cada uno

de los elementos electrónicos necesarios en este circuito. En la Figura 26 se puede observar el diseño esquemático correspondiente al nodo 2.

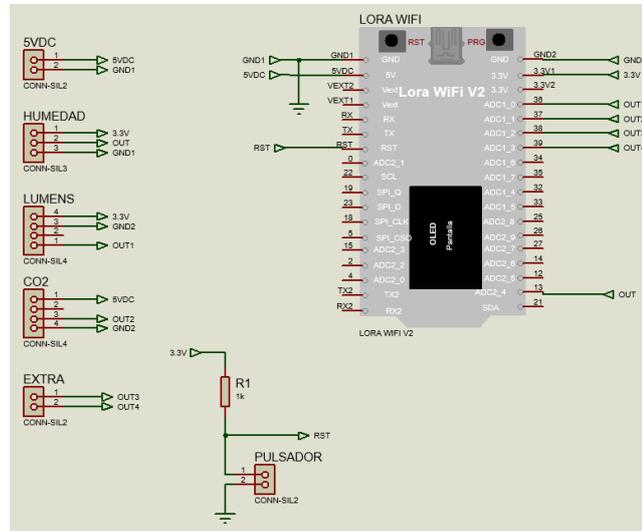


Figura 26. Esquema electrónico del Nodo 2.
Fuente: Investigador

El diseño del circuito impreso se realiza colocando los elementos electrónicos en las cercanías de los pines del módulo LoRa con el objetivo de reducir el número de pistas requerido para su funcionamiento. En la Figura 27 sección a) se puede observar el diseño del circuito impreso correspondiente al nodo 2.

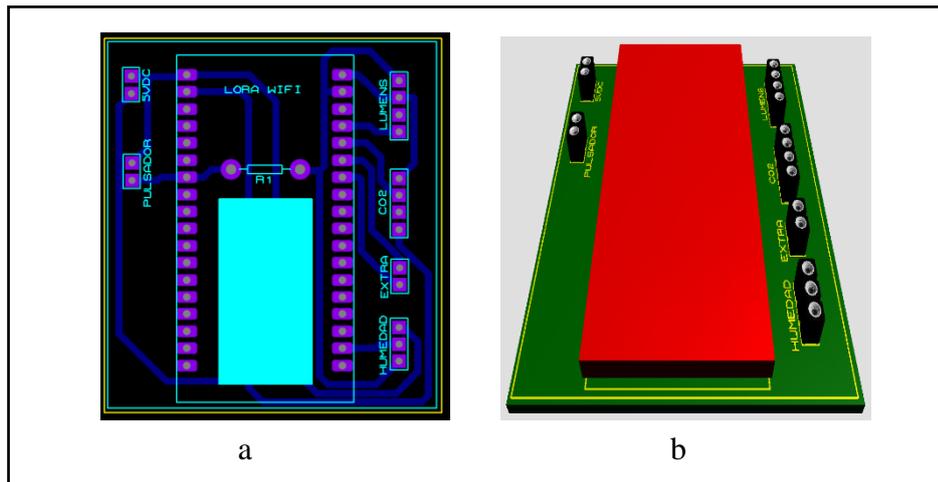


Figura 27. a) Diseño del circuito impreso del Nodo 2 b) Diseño 3D de la PCB del Nodo 2
Fuente: Investigador

Finalmente, se realiza una visualización 3D del circuito diseñado para verificar la disposición física de los elementos electrónicos. En la Figura 27 sección b) se puede observar la PCB del nodo 2.

Diseño mecánico

En esta sección se establece el proceso realizado para construir una carcasa adecuada para cada uno de los sensores inteligentes utilizando formas, materiales y técnicas de prototipado rápido mediante impresión 3D. Este proceso de fabricación cumple con tareas específicas en base a las necesidades requeridas por los dispositivos de medición y las condiciones ambientales del galpón de pollos en la avícola “San Miguel”.

Material de construcción de la carcasa

Dentro de los procesos de fabricación de prototipos académicos las técnicas de impresión 3D son muy utilizadas y uno de los materiales con mayor presencia en el mercado local es el PLA (ácido poliláctico). Este polímero de alta resistencia y biodegradable permite fabricar carcasas resistentes y con un acabado profesional, como se muestra en la Tabla 22.

Tabla 22. Propiedades mecánicas y físicas del PLA.

Características PLA	
Módulo de Elasticidad	3-4 GPa
Alargamiento a la ruptura	2-5%
Resistencia mecánica a la flexión	100MPa
Resistencia mecánica a la tensión	60-70 MPa
Temperatura de fusión	170-200 °C

Fuente: Investigador

Por lo tanto, el PLA ofrece las características necesarias para soportar las condiciones ambientales existentes dentro del galpón de pollos donde se implementará los equipos de medición. Previamente al dimensionamiento de la carcasa se debe realizar una medición del ancho y largo de cada uno de los componentes electrónicos para calcular el área y volumen requerido por cada uno de ellos con las fórmulas (1) y (2). Cabe recalcar que se realizaría una aproximación de la forma de los dispositivos electrónicos a la figura geométrica de un cubo.

$$A = b \times l \quad (1)$$

Donde:

A: área del dispositivo electrónico

b: ancho del dispositivo electrónico

l: largo del dispositivo electrónico

$$V = A * h \quad (2)$$

Donde:

V: volumen del positivo electrónico

A: área del dispositivo electrónico

h: altura del dispositivo electrónico

En la Tabla 23 se puede observar las dimensiones físicas de cada uno de los elementos electrónicos utilizados en el desarrollo del presente proyecto de tesis.

Tabla 23. Dimensiones físicas de los dispositivos electrónicos.

Elemento	Ancho	Largo	Alto	Área	Volumen
DHT22	4 cm	1.3 cm	1 cm	5.2 cm ²	5.2 cm ³
LDR	1.45 cm	2.47 cm	1.8 cm	3.58 cm ²	6.4 cm ³
LoRa	2.53 cm	5.17 cm	1.46 cm	13 cm ²	18.98 cm ³
Conector 5Vdc	1.46 cm	3.79 cm	1.15 cm	5.53 cm ²	6.35 cm ³
MG811	3.12 cm	4.2 cm	2.67 cm	13.1 cm ²	34.97 cm ³
Relé	3.8 cm	4.9 cm	1.7 cm	18.62 cm ²	31.65 cm ³
PCB Nodo 1	4.7 cm	5.45 cm	2.51 cm	25.61 cm ²	64.28 cm ³
PCB Nodo 2	4.95 cm	5.65 cm	2.51 cm	27.96 cm ²	70.17 cm ³
PCB Nodo 3	4.6 cm	5.7 cm	2.51 cm	26.22 cm ²	65.81 cm ³

Fuente: Investigador

Diseño mecánico esclavo (Nodo 1)

El diseño de la carcasa del dispositivo inteligente configurado como esclavo (Nodo 1) fue realizado en un software de diseño mecánico 3D con licencia educativa. En este diseño se establece las dimensiones reales de todos los elementos electrónicos necesarios para una correcta medición y transmisión de los datos hacia el dispositivo maestro. El área mínima requerida para este dispositivo se especifica a continuación:

$$\text{Área}_{\text{Nodo1}} = A_{\text{DHT22}} + A_{\text{LDR}} + A_{\text{LoRa}} + A_{\text{Conector}} + A_{\text{PCB}}$$

$$\text{Área}_{\text{Nodo1}} = (5.2 + 3.58 + 13 + 25.61)\text{cm}^2$$

$$\text{Área}_{\text{Nodo1}} = 47.39 \text{ cm}^2$$

Por otro lado, el volumen mínimo requerido por este dispositivo electrónico se calcula de la siguiente manera:

$$\text{Volumen}_{\text{Nodo1}} = V_{\text{DHT22}} + V_{\text{LDR}} + V_{\text{LoRa}} + V_{\text{Conector}} + V_{\text{PCB}}$$

$$\text{Volumen}_{\text{Nodo1}} = (5.2 + 6.4 + 18.98 + 6.35 + 64.28) \text{ cm}^3$$

$$\text{Volumen}_{\text{Nodo1}} = 101.21 \text{ cm}^3$$

Una vez que se ha determinado el área y volumen mínimo requerido por todos los elementos electrónicos se procede a realizar el diseño de la carcasa, cabe recalcar que este software permite realizar el ensamble tridimensional de todos los componentes para verificar una correcta integración y sujeción de los componentes. En la Figura 28 se puede observar el diseño y el ensamblaje de los elementos electrónicos con la carcasa.

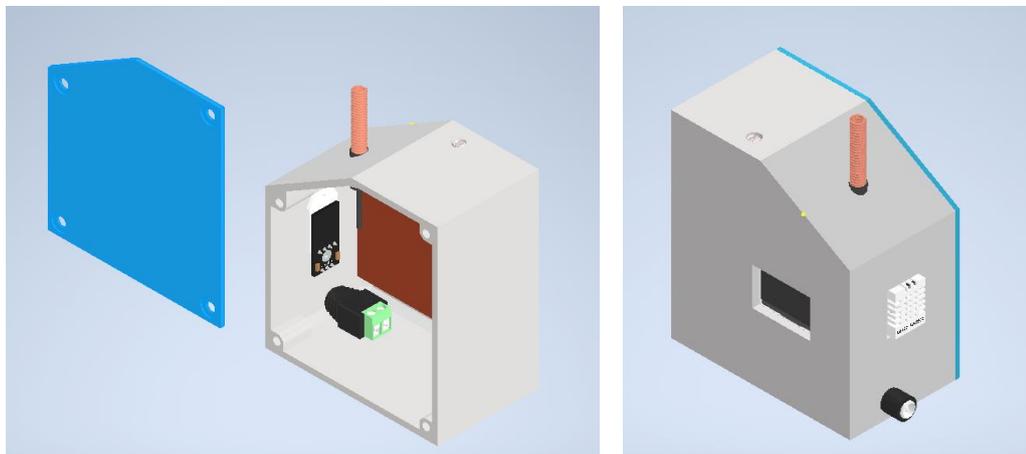


Figura 28. Diseño mecánico del Nodo 1.
Fuente: Investigador

Diseño mecánico esclavo (Nodo 2)

Las dimensiones físicas requeridas para el segundo dispositivo inteligente configurado como esclavo se calculan utilizando los valores descritos anteriormente. El área mínima requerida para este dispositivo se describe a continuación:

$$\text{Área}_{\text{Nodo2}} = A_{\text{DHT22}} + A_{\text{LDR}} + A_{\text{LoRa}} + A_{\text{Conector}} + A_{\text{MG811}} + A_{\text{PCB}}$$

$$\text{Área}_{\text{Nodo2}} = (5.2 + 3.58 + 13 + 13.1 + 27.96)\text{cm}^2$$

$$\text{Área}_{\text{Nodo2}} = 62.84 \text{ cm}^2$$

El volumen mínimo requerido por este dispositivo electrónico se calcula de la siguiente manera:

$$\text{Volumen}_{\text{Nodo2}} = V_{\text{DHT22}} + V_{\text{LDR}} + V_{\text{LoRa}} + V_{\text{Conector}} + V_{\text{MG811}} + V_{\text{PCB}}$$

$$\text{Volumen}_{\text{Nodo2}} = (5.2 + 6.4 + 18.98 + 6.35 + 34.97 + 70.17) \text{ cm}^3$$

$$\text{Volumen}_{\text{Nodo2}} = 142.07 \text{ cm}^3$$

Entonces, se procede a realizar el diseño de la carcasa ensamblando todos los elementos electrónicos, en este caso se debe considerar que el sensor MG811 necesita medir el CO2 existente en el aire, motivo por el cual debe estar ubicado en el exterior de la carcasa. En la Figura 29 se puede observar el diseño mecánico de la carcasa del Nodo 2.

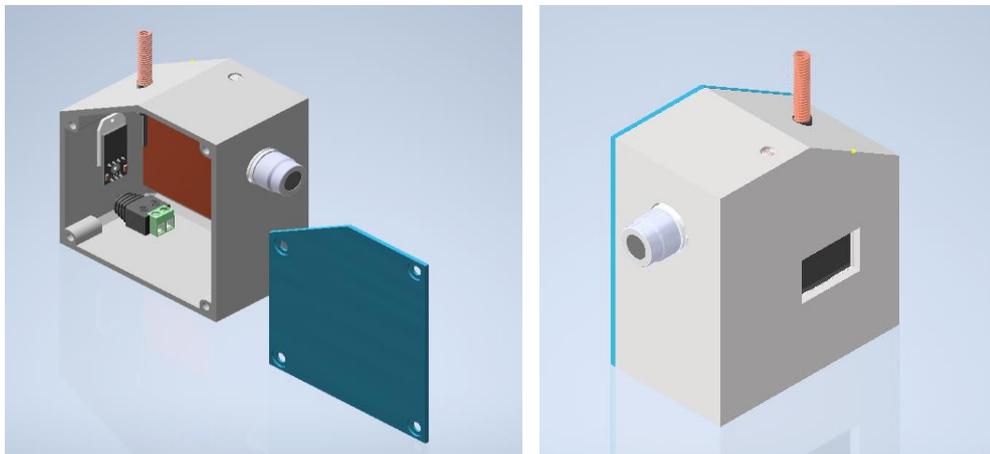


Figura 29. Diseño mecánico del Nodo 2.
Fuente: Investigador

Construcción e implementación

En esta sección se describe el proceso de construcción e implementación tanto de los componentes electrónicos como de la carcasa mecánica, para lo cual se inicia con la construcción del circuito PCB utilizando el servicio de ruteado con CNC. Esta técnica de construcción de circuitos electrónicos es muy utilizada hoy en día y los costos de este servicio son cada vez más económicos. Cabe recalcar que este proceso es el mismo para los 3 nodos, motivo por el cual solamente se describirá el caso del módulo esclavo Nodo 1, como se muestra en la Figura 30

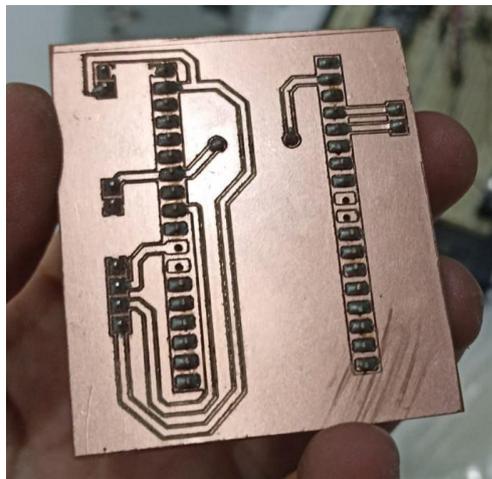


Figura 30. Ruteado de pistas en maquina CNC.
Fuente: Investigado

Una vez realizado el ruteado de las pistas de los circuitos electrónicos se procede al montaje de los conectores a 90°, resistencia y espadines hembra sobre cada una de las placas. Posteriormente se aplica un proceso de soldadura con estaño y un cautín de 60W, el resultado de este proceso de muestra en la Figura 31

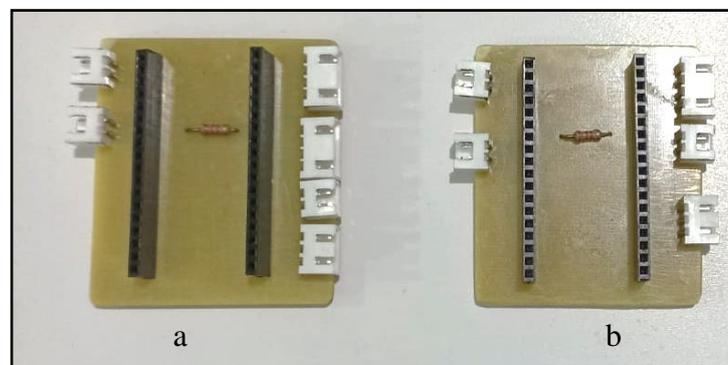


Figura 31. Montaje de elementos electrónicos en las placas nodo 1(a) y nodo 2(b).
Fuente: Investigador

Se debe realizar una verificación de la correcta disposición de los elementos electrónicos realizando el montaje del controlador LoRa en los pines dispuestos para su conexión. En la

Figura 32 se puede observar el montaje del módulo LoRa en cada una de las placas diseñadas.

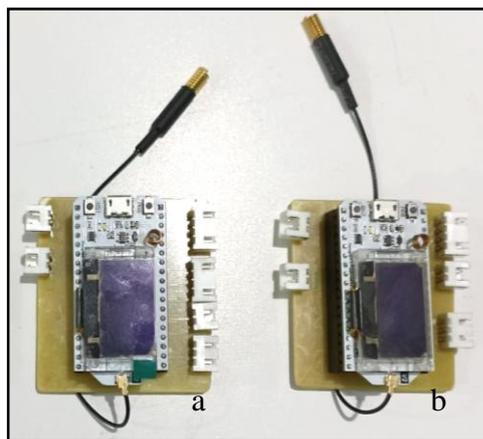
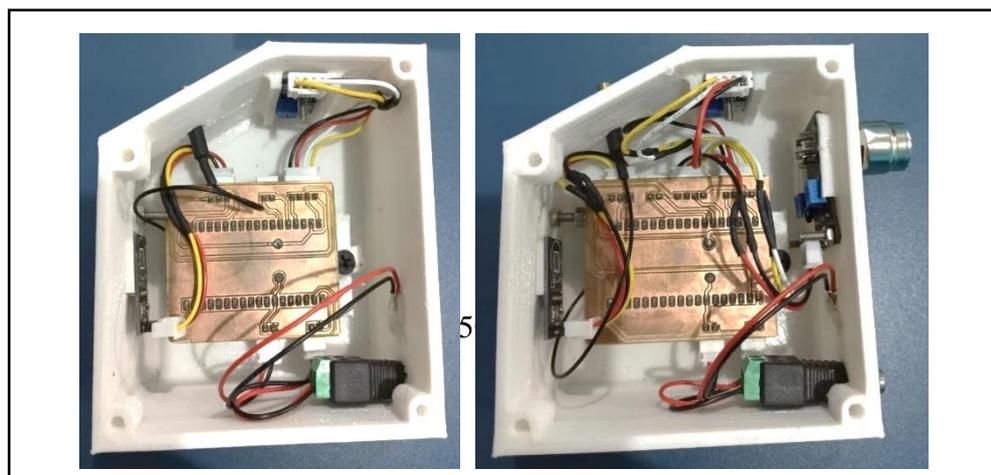


Figura 32. Montaje del módulo LoRa en las placas.
Fuente: Investigador

La implementación mecánica de todos los componentes electrónicos en la carcasa se realiza después de verificar la correcta impresión utilizando el material PLA. En la Figura 33 sección a) y b) se puede observar el ensamblaje del Nodo 1 y Nodo 2.



a

b

Figura 33. a) Ensamblaje del Nodo 1 b) Ensamblaje del Nodo 2.
Fuente: Investigador

Diseño de Gateway

Circuito Impreso del Nodo 3

En el caso del dispositivo inteligente configurado como maestro se requiere el acondicionamiento de un pulsador como reset y dos salidas digitales de 3.3 voltios. En la Tabla 24 se puede observar los requerimientos de este circuito.

Tabla 24. Requerimientos del Nodo 3.

Requerimientos	
Voltaje	Valor
Entrada LoRa	5Vdc
Módulo Relé	5Vdc
Activación Salidas	3.3Vdc
Reset	3.3Vdc

Fuente: Investigador

El módulo relé utilizado en el diseño del circuito del dispositivo maestro tiene un rango de activación en la señal de control de 0 – 5Vdc, siendo 0 voltios una condición de OFF y 5Vdc representa una condición de ON. Sin embargo, el módulo LoRa solamente puede generar 3.3Vdc en sus pines digitales, por lo tanto, se debe calcular el voltaje mínimo de activación del relé utilizando la siguiente fórmula (3).

$$I_C = \frac{V_{CC}}{R_R} \quad (3)$$

Donde:

V_C : Voltaje del relé

R_R : Resistencia de base.

I_C : Corriente del colector

Utilizando la hoja técnica del módulo relé se obtuvieron los valores de $5V_{dc}$ y 70Ω para las variables de VC y RR, respectivamente, de esta manera se puede calcular la corriente del colector.

$$I_C = \frac{V_{CC}}{R_R} = \frac{5}{70}$$

$$I_C = 71.4 \text{ mA}$$

Por otro lado, la corriente de base de activación del transistor J3Y conectado a la bobina del relé electromagnético se calcula utilizando la formula (4).

$$I_b = \frac{I_C}{hFE} \quad (4)$$

Donde:

I_b : Corriente de base del transistor

I_C : Corriente del colector del transistor

hFE : Ganancia de corriente del transistor

Para el módulo relé utilizado en este proyecto se tiene un $hFE = 300$, aplicando la formula se tiene:

$$I_b = \frac{I_C}{hFE} = \frac{71.4 \text{ mA}}{300}$$

$$I_b = 238 \mu A$$

El voltaje de activación mínimo requerido por el módulo relé se calcula con la formula (5).

$$V_{RB} = V_C - V_{be} - V_{LED} \quad (5)$$

Donde:

V_{RB} : Voltaje de activación relé

V_C : Voltaje del relé

V_{be} : Voltaje de activación del transistor

V_{LED} : Voltaje de caída en el led de visualización

Los valores obtenidos de la hoja técnica del módulo relé son: 5Vdc, 0.7Vdc y 1.95Vdc para las variables V_C , V_{be} y V_{LED} , respectivamente. Entonces aplicando la formula (5) se obtienen lo siguiente:

$$V_{RB} = V_C - V_{be} - V_{LED}$$

$$V_{RB} = 5Vdc - 0.7Vdc - 1.95Vdc$$

$$V_{RB} = 2.35 Vdc$$

Entonces, este módulo requiere un valor mínimo de activación de 2.35V, las salidas de 3.3 voltios del módulo LoRa son capaces de activar sin problemas el relé electromagnético. En la Figura 34 se puede observar el esquema electrónico del Nodo

3

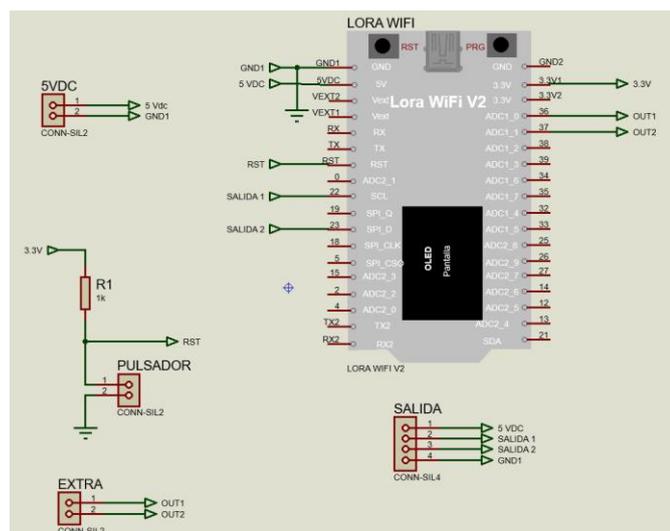
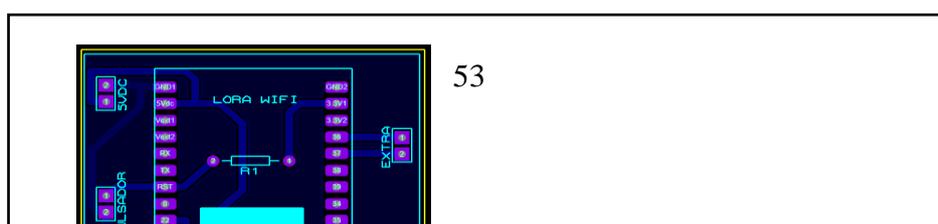
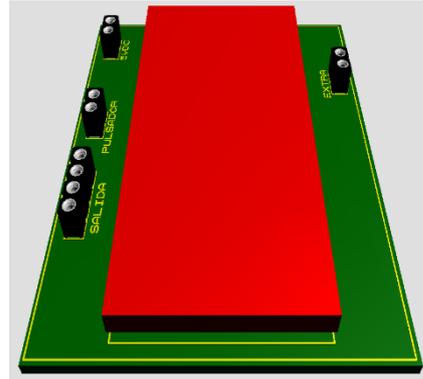


Figura 34. Esquema electrónico del Nodo 3.

Fuente: Investigador

El diseño del circuito impreso para el módulo configurado como maestro se muestra en la Figura 35 Sección a), donde se puede apreciar la disposición de los conectores electrónicos que servirán para realizar las respectivas conexiones externas con el módulo relé y el pulsador de reset, Finalmente, se realiza una verificación de la disposición de los componentes electrónicos mediante la herramienta de visualización 3D que dispone el software Proteus, como se muestra en la Figura 35 Sección b).





a

b

Figura 35. a) Diseño del circuito impreso del Nodo 3 b) Diseño 3D de la PCB del Nodo 2.
Fuente: Investigador

Diseño mecánico maestro Gateway (Nodo 3)

De la misma manera que en los casos anteriores, el módulo configurado como maestro tiene diferentes dispositivos electrónicos, por lo tanto, se realiza un cálculo del área mínima requerida utilizando los valores descritos previamente.

$$Area_{Nodo3} = A_{Relé} + A_{LoRa} + A_{Conector} + A_{PCB}$$

$$Area_{Nodo3} = (18.62 + 13 + 5.53 + 26.22)cm^2$$

$$Area_{Nodo3} = 63.37 cm^2$$

El volumen mínimo requerido por este dispositivo electrónico se calcula de la siguiente manera:

$$Volumen_{Nodo3} = V_{Relé} + V_{LoRa} + V_{Conector} + V_{PCB}$$

$$Volumen_{Nodo3} = (31.65 + 18.98 + 6.35 + 65.81) cm^3$$

$$Volumen_{Nodo3} = 122.79 cm^3$$

Una vez realizado los cálculos del área y volumen de los elementos electrónicos se procede a desarrollar el diseño mecánico de la carcasa para el módulo configurado como maestro, como se muestra en la Figura 36.

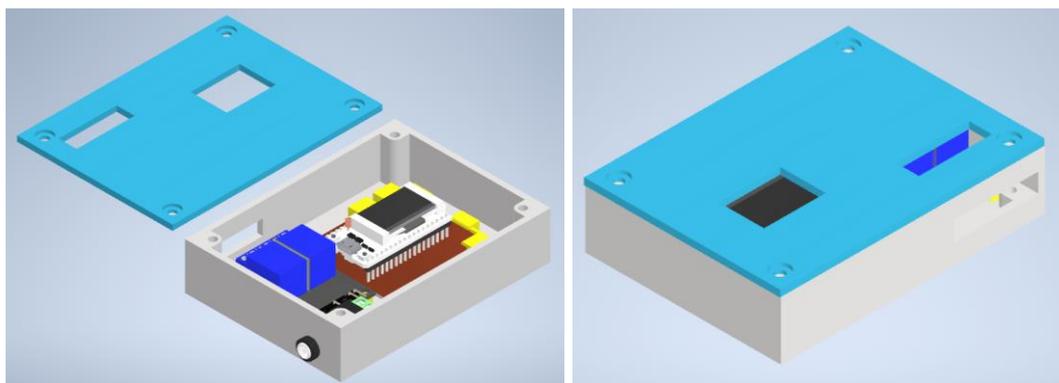


Figura 36. Diseño mecánico del Nodo 3.
Fuente: Investigador

En la Figura 37 sección a) y b) pretende dar a conocer el proceso que se desarrolló para el montaje del nodo 3 Gateway

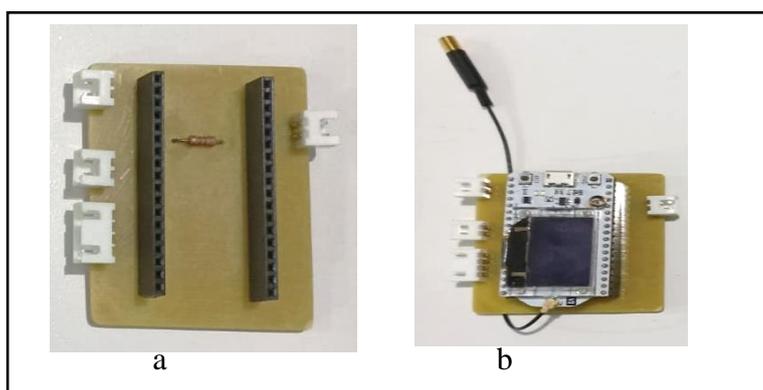


Figura 37. a) Montaje de elementos electrónicos en las placas Gateway b) Montaje del módulo LoRa en las placas.
Fuente: Investigador

Por último, en la Figura 38 se muestra el ensamblaje de la carcasa del módulo configurado como Gateway (maestro), en este dispositivo inteligente se ha ubicado dos ranuras para integrar fácilmente las cargas de 120 Vac como señales luminosas tipo led.

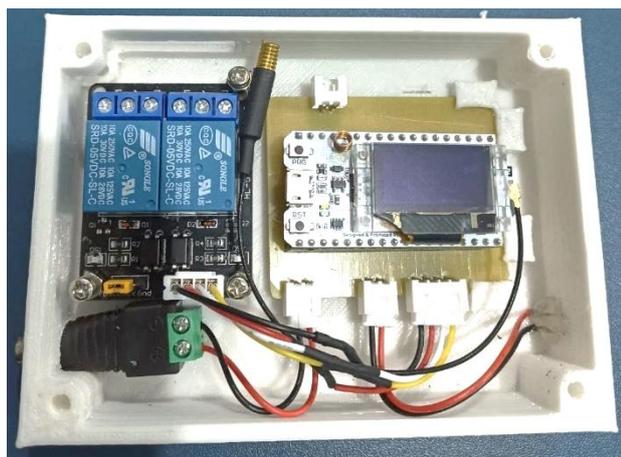


Figura 38. Ensamblaje del Nodo 3.
Fuente: Investigador

Acondicionamiento y adquisición de datos

Etapa de Adquisición de Datos en los Esclavos

Los sensores inteligentes son los encargados de medir las variables físicas de temperatura, humedad, lúmenes y la cantidad de dióxido de carbono presente en el ambiente (CO₂), para lo cual utilizamos la disposición de los elementos electrónicos como se muestra en la Figura 39. En el presente proyecto existe dos diferentes elementos inteligentes para medir las condiciones del galpón de mejor manera, a los cuales se les denominara como esclavo1 y esclavo2, respectivamente.

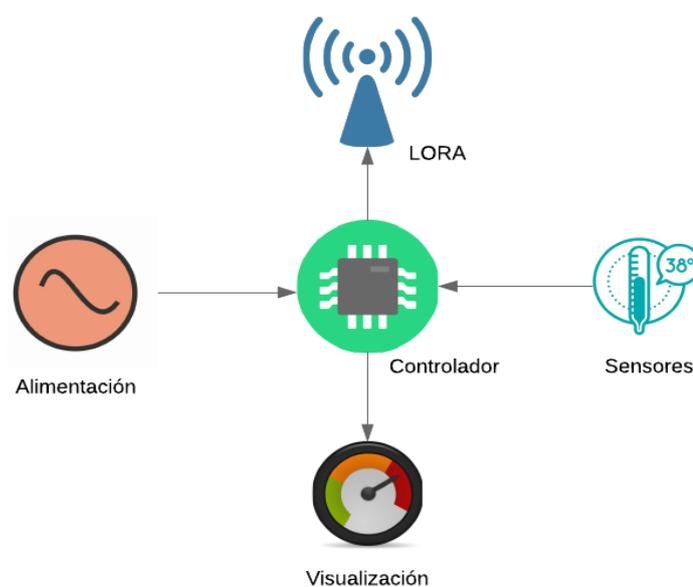


Figura 39. Elementos electrónicos de los esclavos.
Fuente: Investigador

Configuración Esclavo 1 (Nodo 1)

El nodo 1 configurado como esclavo es el encargado de adquirir las variables de temperatura y humedad desde el sensor digital DHT22. Este sensor detecta la humedad mediante funcionamiento capacitivo y la temperatura es adquirida desde una medición del aire con un termistor. La humedad relativa es adquirida con la formula (6).

$$\text{Humedad Relativa (RH)} = \frac{\text{Gramos vapor agua presente (g/m}^3\text{)}}{\text{gramos agua saturación (g/m}^3\text{)}} * 100 \quad (6)$$

La comunicación implementada entre el controlador LoRa y el módulo DHT22 se basa en el protocolo popular conocido como 1-wire de Dallas, la información de las dos variables a través de un solo cable. El sensor envía 5 bytes de información (40 bits seguidos) cada vez que el controlador le solicite una actualización de datos poniendo el pin de conexión en Vcc [27]. El esquema de comunicación entre los dos dispositivos electrónicos se puede observar en la Figura 40.

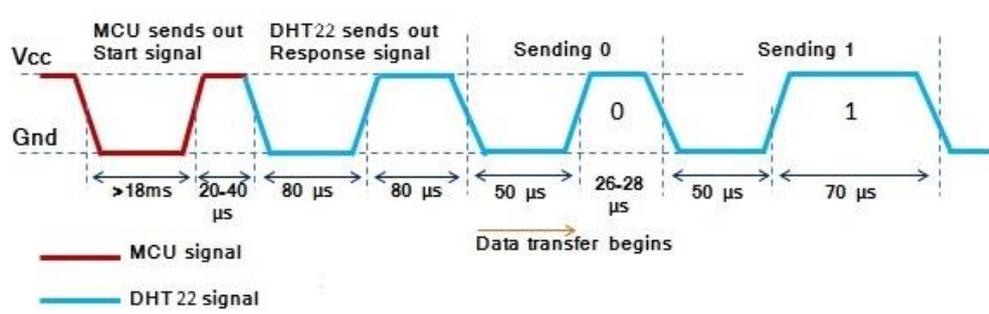


Figura 40. Secuencia de comunicación LoRa y DHT 22.

Otra variable de medición es la cantidad de lúmenes presentes dentro del galpón de pollos, para lo cual se utilizó el módulo LDR. Este elemento foto eléctrico varía su resistencia en función de la cantidad de luz disponible en el medio ambiente. Este valor es enviado hacia el controlar como una señal analógica que varía en un rango de 0 – 5 Vdc. Por otro lado, la formula (7) es la utilizada para linealizar el sensor y obtener la conversión de voltaje a lúmenes.

$$R = A * L^\alpha \quad (7)$$

Donde:

R: Resistencia en ohms

L: Iluminación en lux

A y α : Constantes LDR

Se debe considerar que existen una gran variedad de componentes LDR en el mercado local, por este motivo se utilizó una respuesta típica de la dependencia entre iluminación y resistencia de una LDR en una escala logarítmica.

Funcionamiento del Esclavo 1 (Nodo 1)

Las bandas libres que se pueden utilizar con fines educativos o para el desarrollo de prototipos son las siguientes:

- Banda 2,4 GHz usando WiFi, Bluetooth o Zigbee.
- Banda 5 GHz usado WiFi
- Banda de 868 MHz con opción de usar Zigbee, LoRa o Sigfox
- Banda de 434 MHz.
- Frecuencias RFID (13,56 MHz, 125 kHz, etc) [28].

En este caso se utilizó la banda de 434 MHz porque es una banda libre y las librerías “Heltec.h” utilizadas para la programación están desarrolladas con esta configuración. Cuando el dispositivo Lora32 es encendido se asigna una dirección y un destino dentro de la red, esta información es muy importante cuando se requiere enviar o recibir información de otro dispositivo inteligente, en este caso se utilizaron los valores de 0xBD y 0xFD para las variables de “Local Adress” y “Destino”.

Mediante la utilización de la librería “DHT.h” el microcontrolador Lora32 puede comunicarse con el sensor de humedad DHT22. Los valores de medición de este sensor son de -40°C a 80°C con una precisión de $\pm 5^\circ\text{C}$ y un rango de humedad de 0 a 100% RH con una precisión de 2%. Los valores son obtenidos ejecutando la librería con los comandos readHumidity() y readTemperature() para leer la humedad y la temperatura, respectivamente. La librería utilizada entrega los valores en grados centígrados y en porcentaje, por lo tanto, no se quiere de ninguna conversión adicional. La lectura del sensor de luminosidad se realiza mediante una lectura del pin analógico A0, donde se encuentra conectado el sensor. Este valor tiene un rango de operación de 0 a 5Vdc, valores que son almacenados en la variable denominada como “ldr”. En la Figura 41 se puede observar el diagrama de flujo de la programación realizada para la

adquisición de datos de las variables y la transmisión de la información hacia el maestro con su respectiva programación en el Anexo A, B y C.

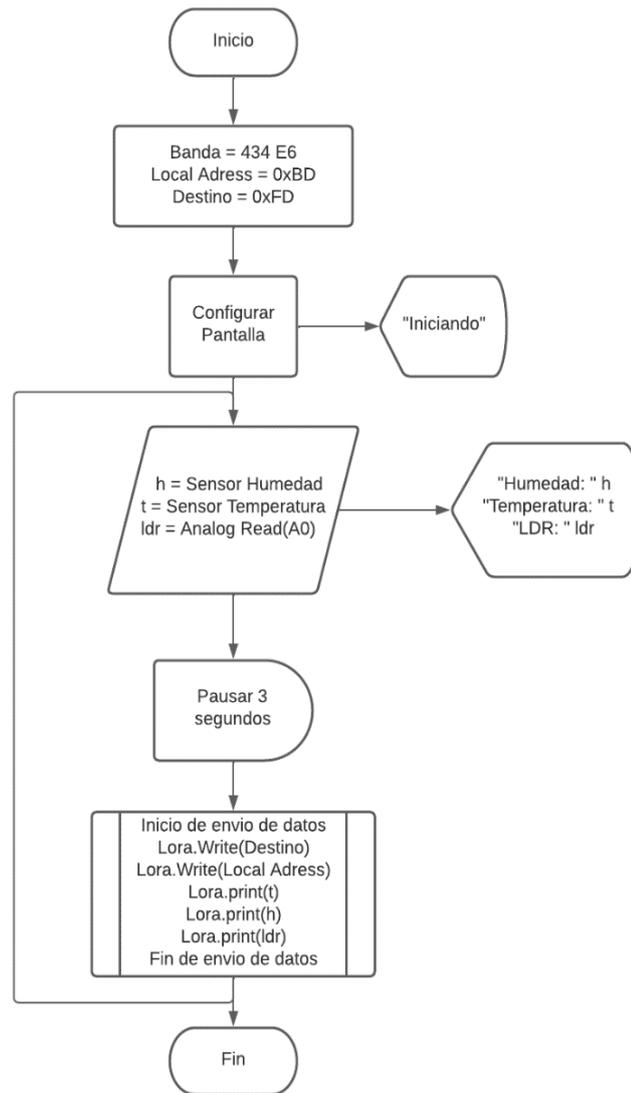


Figura 41. Diagrama de flujo de nodo 1.
Fuente: Investigador

Transmisión de datos Esclavo 1 (Nodo 1)

La transmisión de los datos se realiza utilizando la banda de transmisión a 434 MHz. Una vez que el dispositivo adquirido los valores de los sensores de temperatura, humedad y el valor de luminosidad se procede armar la nueva trama donde se especifican los siguientes valores:

$[Destino, LocalAddress, t, h, ldr]$

Donde:

Destino: Dirección hacia dónde se va enviar la información.

LocalAdress: Dirección actual del dispositivo.

t: Valor de la temperatura medida

h: Valor de la humedad medida

ldr: Valor de la luminosidad medida

Configuración Esclavo 2 (Nodo 2)

El nodo 2 tiene un funcionamiento similar el nodo 1 debido a que es el encargado de medir la temperatura, humedad, lúmenes y la cantidad de CO₂ disponible en el ambiente. Este nodo estará ubicado físicamente en una zona diferente al nodo 1 para que se pueda obtener mediciones de distintas partes del galpón. El módulo MG -811 es el encargado de realizar la medición de la calidad del aire disponible dentro del galpón de pollos, para lo cual se debe utilizar la formula (8) de conversión de voltaje en ppm (partículas por millón de CO₂).

$$V_s = V_0 + \Delta V_s / (\log_{10} 400 - \log_{10} 1000) * (\log_{10} C_{CO_2} - \log_{10} 400) \quad (8)$$

Donde:

V_s: voltaje de salida del sensor

V₀: voltaje de *V_s* cuando el sensor mide 400 ppm

V_c: voltaje de *V_s* cuando el sensor mide 1 000 ppm

ΔV_s : es la diferencia de voltaje entre *V₀* – *V_c*

C_{CO₂}: Es la cantidad de concentración del CO₂ en ppm

Para realizar un cálculo adecuado de la cantidad de CO₂ existentes en el ambiente se debe utilizar la tabla de conversión de milivoltios a ppm que se muestran en la Figura 42. Se debe considerar que los valores mostrados en esta grafica fueron obtenidos a una temperatura de 28°C, una humedad relativa de 65% y una cantidad de oxígeno del 21%.

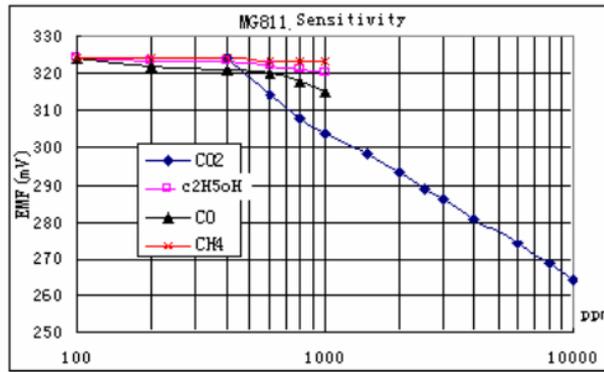


Figura 42. Conversión de milivoltios a ppm de CO₂. [28]

Funcionamiento del Esclavo 2 (Nodo 2)

Este nodo 2 utiliza las librerías mencionadas anteriormente en el nodo 1, estas son: “Heltec.h” para establecer la comunicación entre los dispositivos LoRa y “DHT.h” para obtener las mediciones de temperatura y humedad. Sin embargo, este dispositivo inteligente cuenta con un sensor de CO₂ para lo cual se utilizó la librería “CO2Sensor.h”.

La banda de transmisión configurada en el esclavo 2 es igual a 434 MHz, en lo que se refiere a las variables de “Local Adress” y “Destino” los valores asignados son 0xBB y 0xFD, respectivamente. Los valores de humedad y temperatura son obtenidos con los comandos readHumidity() y readTemperature() y almacenados en las variables “h” y “t”, respectivamente. Para obtener los valores de concentración de CO₂ se requiere la implementación de un filtro de Kalman mediante la librería “SimpleKalmanFilter.h”, este algoritmo sirve para identificar el estado oculto de un sistema dinámico; además sirve para eliminar el ruido blanco presente en la señal original. En la Figura 43 se puede observar el comportamiento de esta librería.



Figura 43. Funcionamiento del filtro Kalman con Arduino.

Fuente: Investigador

En la Figura 41 se puede observar el diagrama de flujo de la programación realizada para la adquisición de datos de las variables y la trasmisión de la información hacia el maestro.

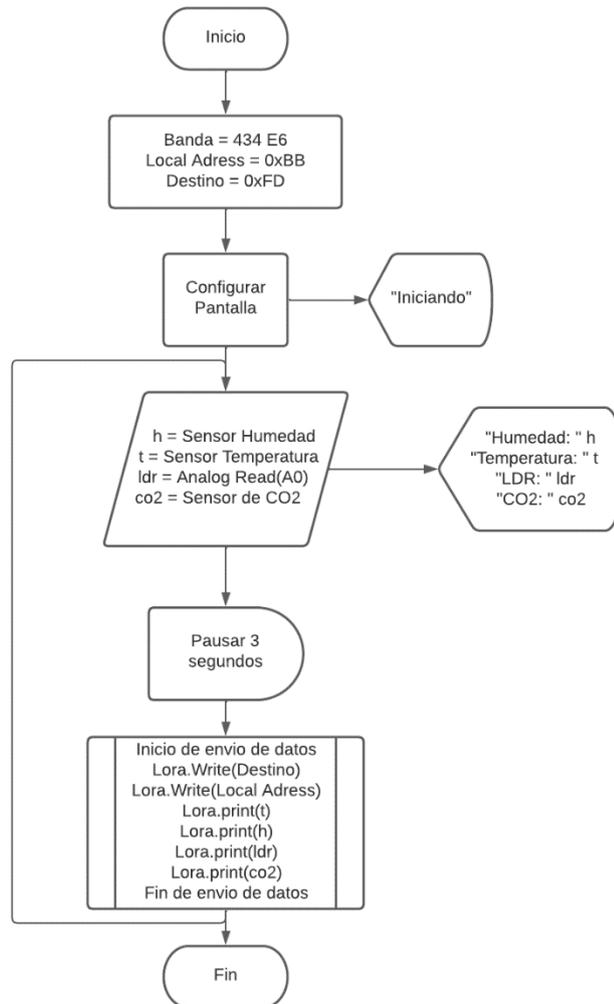


Figura 44. Diagrama de flujo del nodo 2.
Fuente: Investigador

Transmisión de datos Esclavo 2 (Nodo 2)

La transmisión de los datos se realiza utilizando la banda de transmisión a 434 MHz. Una vez que el dispositivo adquirido los valores de los sensores de temperatura, humedad, valor de luminosidad y la ppm de CO2 se procede armar la nueva trama donde se especifican los siguientes valores:

$[Destino, LocalAddress, t, h, ldr, co2]$

Donde:

Destino: Dirección hacia dónde se va enviar la información.

LocalAdress: Dirección actual del dispositivo.

t: Valor de la temperatura medida

h: Valor de la humedad medida

ldr: Valor de la luminosidad medida

co2: Cantidad de ppm de CO2 existentes en el ambiente

Etapa de Adquisición de Datos del Maestro (Nodo 3)

Por otro lado, una correcta selección del microcontrolador con comunicación LoRa y WiFi sirve para implementar una red Maestro – Esclavo. Esta red se establece considerando que la Avícola San Miguel no posee una red WiFi local para que los sensores inteligentes puedan enviar los valores de los sensores hacia el servidor IoT. En la Figura 45 se puede observar los elementos electrónicos necesarios para la implementación del dispositivo inteligente encargado de recibir la información de los sensores a través de un protocolo de comunicación tipo LORA, a este dispositivo se le conocerá como el maestro.

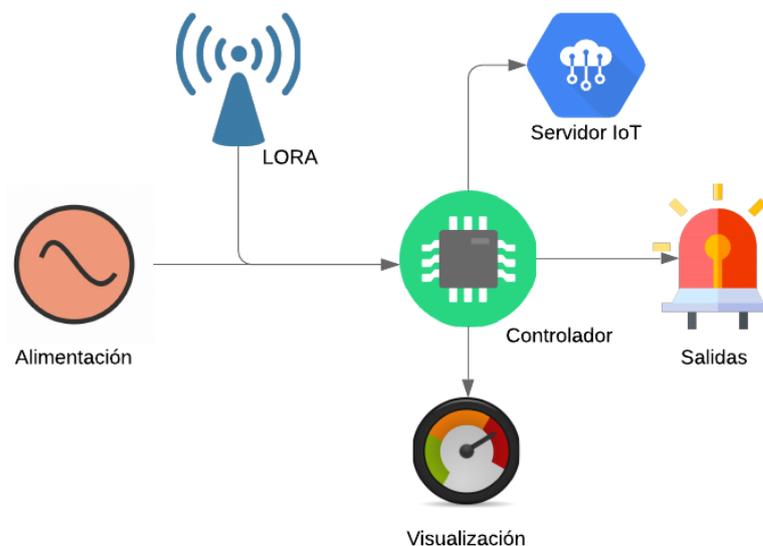


Figura 45. Elementos electrónicos del maestro.

Fuente: Investigador

Configuración Maestro (Nodo 3)

El módulo LoRa 32 configurado como maestro es el encargado de conectarse a una red WiFi disponible en una zona de cobertura del protocolo de comunicación LoRa. En este caso se pudo identificar la disponibilidad de una red a 146.3 metros de distancia en línea recta entre la avícola y la red disponible como se puede observar en la Figura 46.



Figura 46. Red disponible a 146.3 metros de distancia.
Fuente: Investigador

Por otro lado, el módulo maestro será el encargado de activar/desactivar señales luminosas tipo focos LED a 120 Vac, estas señales de aviso servirán para que los operarios de la avícola puedan identificar rápidamente la existencia de unas condiciones ambientales extremas en el galpón de pollos. Además, el maestro podrá enviar los datos hacia un servidor IoT a través del internet y de esta manera los operadores podrán acceder a esta información en cualquier momento y determinar el correcto funcionamiento del galpón.

Funcionamiento del Maestro (Nodo 3)

El funcionamiento del Nodo 3 configurado como maestro requiere de varias librerías para manejar los diferentes protocolos de comunicación y recibir los datos desde los

esclavos, así como también subir la información a los servidores IoT. Las librerías utilizadas son las siguientes:

- “**RemoteXY.h**” esta librería manipula la información para ser visualizada en un HMI desarrollado en un dispositivo móvil mediante una aplicación móvil (apk).
- “**ThingerESP32.h**” utilizada para visualizar los datos en el servidor Thinger.io mediante una computadora.
- “**CTBot.h**” esta librería sirve para crear un bot de respuestas automáticas en el programa Telegram para dispositivos móviles.
- “**Heltec.h**” utilizada para recibir la información desde los esclavos LoRa32.
- “**WiFi.h**” esta librería sirve para establecer una comunicación inalámbrica WiFi con el dispositivo LoRa32.

Una vez que se han establecido correctamente las librerías se proceden a realiza una configuración de los diferentes protocolos de comunicación y de los servidores IoT por separado. La configuración de servidor REMOTEXY para visualización en un dispositivo móvil de las variables medidas del galpón consiste en los siguientes parámetros:

- **WiFi_SSID:** esta variable almacena el nombre de la red WiFi disponible, la misma que servirá para enviar la información a través del internet hacia los demás dispositivos.
- **WiFi_PASSWORD:** en esta variable se agrega la clave de la red WiFi para que el microcontrolador pueda acceder a una conexión de internet.
- **CLOUD_SERVER:** esta variable contiene el servidor que se utiliza para desarrollar el HMI para un dispositivo móvil, esta dirección es constante y corresponde a lo siguiente "cloud.remotexy.com"
- **CLOUD_PORT:** el puerto de comunicación también es un valor constante igual a 6376.
- **CLOUD_TOKEN:** esta variable es muy necesaria porque corresponde a la dirección única que tiene el dispositivo móvil donde se va implementar el HMI para visualizar todas las variables.

El HMI desarrollado para el dispositivo móvil se encuentra almacenado en la página web de “remotexy.com” desde ahí puede acceder el microcontrolador a las siguientes variables:

- **M1T:** Variable de temperatura obtenida por el módulo 1.
- **M1H:** Variable de humedad obtenida por el módulo 1.
- **M1I:** Variable de luminosidad obtenida por el módulo 1.
- **M2T:** Variable de temperatura obtenida por el módulo 2.
- **M2H:** Variable de humedad obtenida por el módulo 2.
- **M2I:** Variable de luminosidad obtenida por el módulo 2.
- **M2CO2:** Variable de ppm de CO2 obtenida por el módulo 2.

Por otro lado, la comunicación con el servidor Thingier.io se realiza para visualizar los valores de temperatura, humedad, luminosidad y ppm de CO2 en una computadora, adicionalmente este servidor permite modificar las alarmas implementadas en el nodo 3. Las variables necesarias para este tipo de comunicación son:

- **USERNAME:** esta variable almacena en nombre del usuario creado en la página web de thingier.io, donde se realizó el HMI.
- **DEVICE_ID:** en esta variable se ingresa el nombre del dispositivo creado en la página web de thingier.io, este puede
- **DEVICE_CREDENTIAL:** este valor alfanumérico es el encargado de vincular el microcontrolador con el HMI desarrollado en la página web de thingier.io.
- Para finalizar con la configuración de la comunicación con la página web de thingier.io se debe inicializar el objeto mediante el comando: **thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL).**

La creación de un bot inteligente en la aplicación móvil de Telegram se realiza utilizando el siguiente comando:

- **BOT_TOKEN:** el número de token corresponde a un número único que cada dispositivo móvil dispone, es utilizado para vincular el bot con el usuario.

Por último, se realiza la configuración de los parámetros para la comunicación LoRa entre los Nodos 1, 2 y 3; para lo cual se utilizó lo siguiente:

- **Banda:** esta es la frecuencia de la red LoRa implementada en este caso el valor es de 434 MHz.
- **Local Adress:** esta es la dirección del dispositivo configurado como maestro, en este caso es 0xFD.

En la Figura 47 se puede observar el diagrama de flujo correspondiente a la primera etapa del nodo 3 (maestro). En esta sección se realiza la configuración de todos los protocolos de comunicación con los comandos antes mencionados.

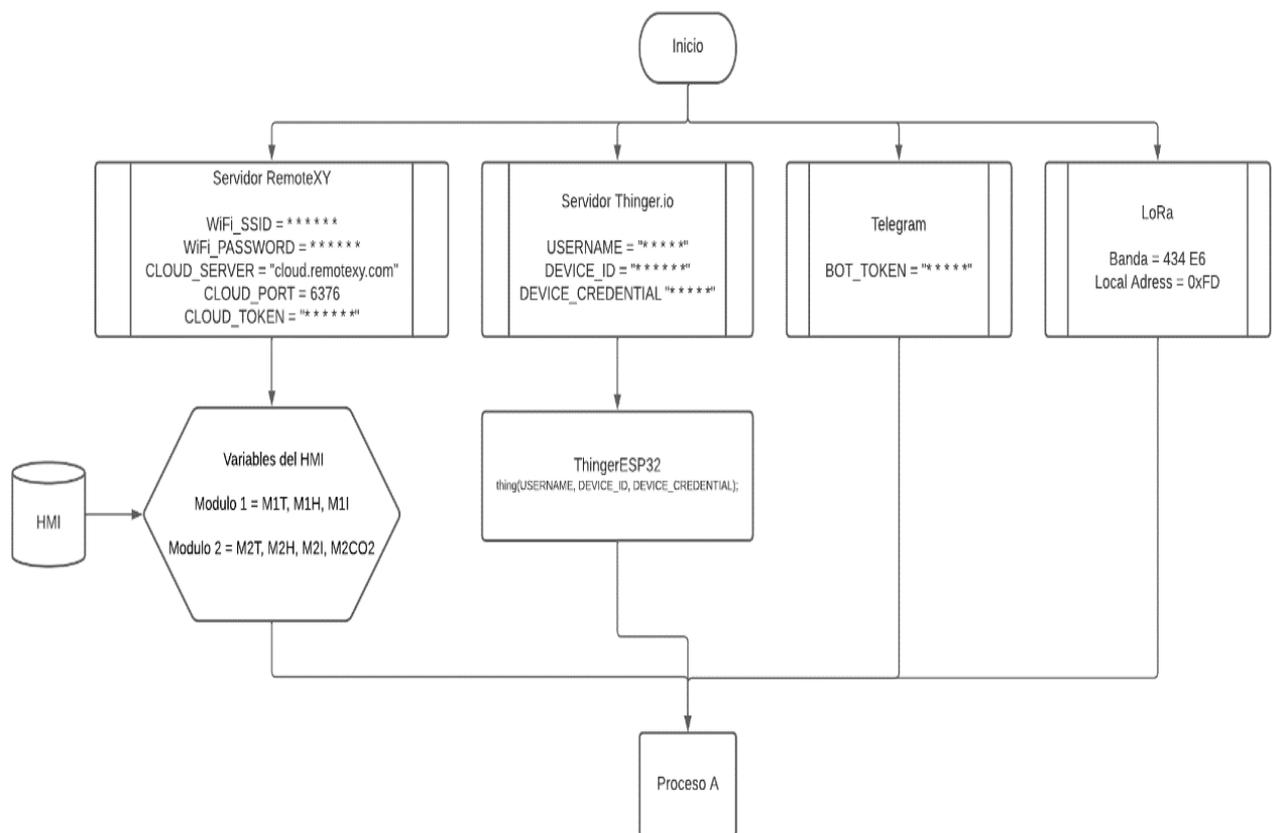


Figura 47. Diagrama de flujo del nodo 3.

Fuente: Investigador

Funcionamiento del Proceso A

El proceso A consiste en la recepción de las tramas de datos enviadas por los Nodos 1 y Nodo 2 hacia el Maestro o Nodo 3. El proceso inicia con la creación de variables para almacenar los valores de temperatura, humedad, luminosidad y ppm de CO2 enviadas por los dos dispositivos inteligentes. Las variables creadas para este fin son las siguientes:

- Modulo 1: tN1, hN1 y ldrN1
- Modulo 2: tN2, hN2, ldrN2 y CO2N2.

Una vez establecidas las variables en el maestro se procede a recibir las tramas de datos enviadas por los dos esclavos, esta información es recibida por el maestro de la siguiente manera:

- Módulo 1: [*Destino, LocalAdress, t, h, ldr*]
- Módulo 2: [*Destino, LocalAdress, t, h, ldr, co2*]

Esta información es separada y almacenada localmente en diferentes viables utilizando los comandos de separación siguientes:

- tN1 = s.separa(mensaje_rcbNd1, ',', 0);
- hN1 = s.separa(mensaje_rcbNd1, ',', 1);
- ldrN1 = s.separa(mensaje_rcbNd1, ',', 2);
- tN2 = s.separa(mensaje_rcbNd2, ',', 0);
- hN2 = s.separa(mensaje_rcbNd2, ',', 1);
- ldrN2 = s.separa(mensaje_rcbNd2, ',', 2);
- CO2N2 = s.separa(mensaje_rcbNd2, ',', 3);

Finalmente, la información recibida se muestra en la pantalla del dispositivo LoRa 32 para que el usuario pueda visualizar las variables de los dos diferentes módulos. Esta información se muestra por 10 segundos cada una. En la Figura 48 se puede observar el diagrama de flujo del Proceso A.

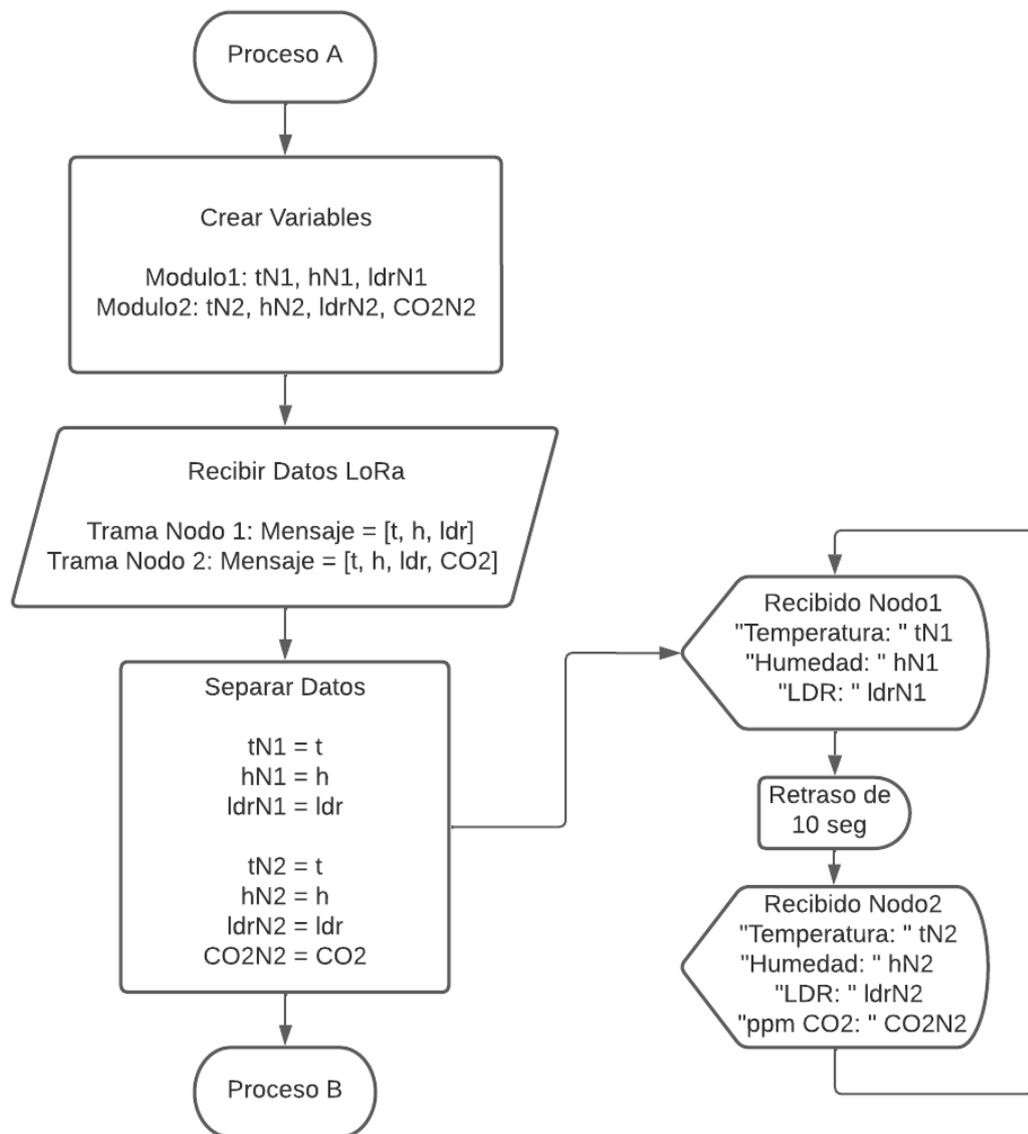


Figura 48. Diagrama de flujo del Proceso A.

Fuente: Investigador

Funcionamiento del Proceso B

El proceso B corresponde a las funciones de actualización de datos en los servidores Thinger.io para visualización en computadoras y el servidor REMOTEXY para visualización mediante un dispositivo móvil en una aplicación (apk). El proceso inicia a partir de la recepción de los datos mediante la red Maestro – Esclavo con el protocolo LoRa desde los dos esclavos (Nodo 1 y Nodo 2). Una vez que se ha establecido la comunicación correctamente y el microcontrolador ha detectado la conexión de un dispositivo móvil los datos son actualizados cada 10 segundos aproximadamente, dependiente de la calidad de servicio de internet disponible. Para manipular las

variables creadas en el HMI del servidor REMOTEXY se utilizan los siguientes comandos:

- RemoteXY.M1T = tN1
- RemoteXY.M1H = hN1
- RemoteXY.M1I = ldrN1

De esta manera se actualizan los valores de temperatura, humedad y luminosidad del Nodo 1, posteriormente se utilizan los siguientes comandos para las variables de temperatura, humedad, luminosidad y ppm de CO2 del Nodo 2.

- RemoteXY.M2T = tN2
- RemoteXY.M2H = hN2
- RemoteXY.M2I = ldrN2
- RemoteXY.CO2 = CO2N2

Por otro lado, para enviar los datos del Nodo 1 como temperatura, humedad y luminosidad al servidor de Thinger.io se utilizan los siguientes comandos:

- thing["Temp_N1"] = outputValue(tNd1)
- thing["Hum_N1"] = outputValue(hNd1)
- thing["Ldr_N1"] = outputValue(ldrNd1)

En el caso del Nodo 2 se utilizan los siguientes comandos para actualizar los datos recibidos de las variables de temperatura, humedad, luminosidad y ppm de CO2.

- thing["Temp_N2"] = outputValue(tNd2)
- thing["Hum_N2"] = outputValue(hNd2)
- thing["Ldr_N2"] = outputValue(ldrNd2)
- thing["CO2_N2"] = outputValue(CO2Nd2)

Finalmente, se establecen los valores para las alarmas de temperatura baja, temperatura alta y alta concentración de CO2 en el ambiente. Después de una revisión bibliográfica e investigación de campo se pudo determinar que la temperatura mínima para los pollos bebes es de 18°C con una máxima de 25°C, por otro lado, la ppm máximos que pueden soportar antes de entrar en asfixia es de 700ppm. El microcontrolador realiza las siguientes comparaciones con estos valores

If $tN1 > \max_temp$ OR $tN2 > \max_temp \Rightarrow Alarma$ AND "Mensaje"

Donde:

tN1: Temperatura medida del Nodo 1.

tN2: Temperatura medida del Nodo 2.

max _temp: Temperatura máxima permitida.

Alarma: Consiste en la activación de una alarma visual como un foco.

Mensaje: Mensaje enviado al usuario por Telegram en este caso es:
"Temperatura muy alta"

En el caso de la temperatura baja se utiliza el siguiente comando para configurar esta alarma y avisar de forma oportuna al operario sobre esta condición.

If $tN1 < \min_temp$ OR $tN2 < \min_temp \Rightarrow Alarma$ AND "Mensaje"

Donde:

min _temp: Temperatura mínima permitida.

Alarma: Consiste en la activación de una alarma visual como un foco.

Mensaje: Mensaje enviado al usuario por Telegram en este caso es:
"Temperatura muy baja"

Por último, la acción de alarma para la condición de exceso de CO2 en el ambiente, se realiza de la siguiente manera:

If $CO2N2 > ppmco2 \Rightarrow Alarma$ AND "Mensaje"

Donde:

CO2N2: Cantidad de CO2 medida desde el Nodo 2 en ppm.

ppmco2 : Valor máximo de CO2 permisible en el ambiente en ppm.

Alarma: Consiste en la activación de una alarma visual como un foco.

Mensaje: Mensaje enviado al usuario por Telegram en este caso es: "CO2 muy alto"

En la Figura 49 se puede observar el diagrama de flujo correspondiente al Proceso B con sus respectivos ciclos de programación y las variables utilizadas.

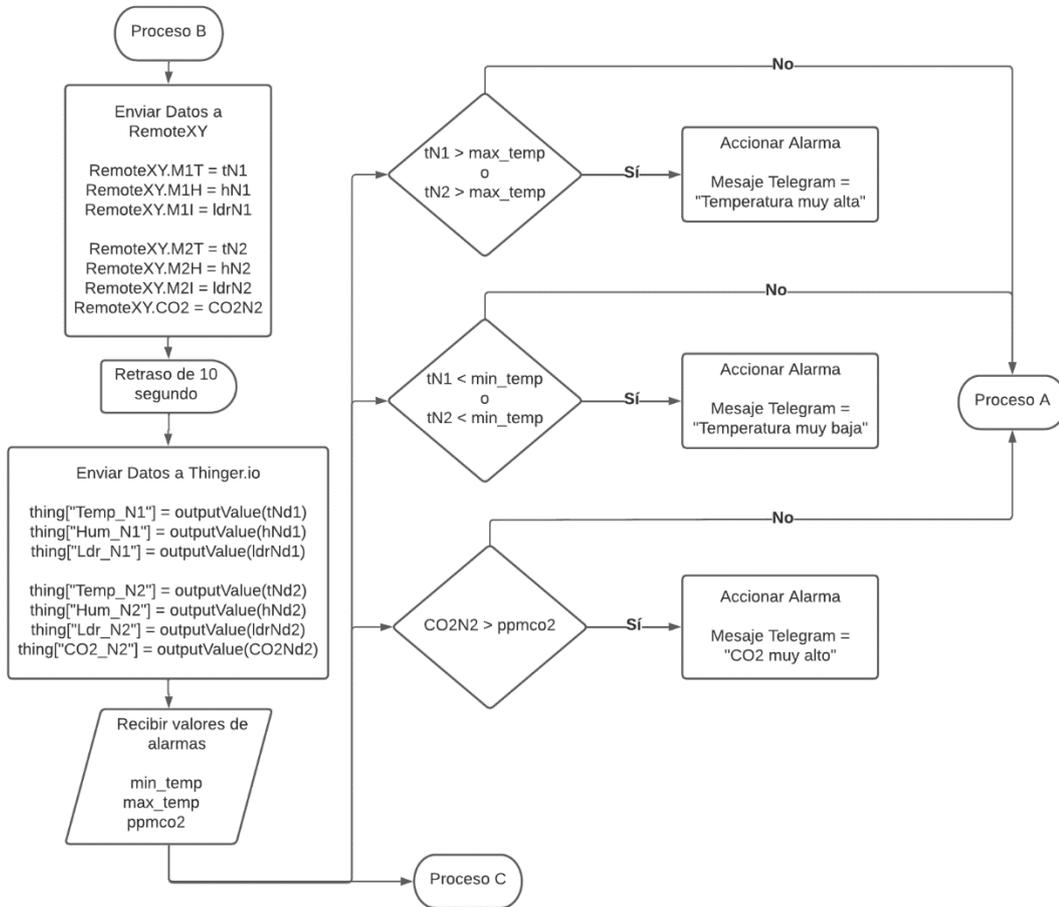


Figura 49. Diagrama de flujo Proceso B.

Fuente: Investigador

Funcionamiento del Proceso C

El Proceso C corresponde a la programación de un BOT de respuestas automáticas, el mismo que servirá para consultar las variables de temperatura, humedad, iluminación y ppm de CO2 de cualquiera de los Nodos 1 o Nodo 2, además se puede consultar los valores de las alarmas como: temperatura máxima, temperatura mínima y ppm de CO2 permisibles. La interacción con el usuario es muy amigable debido al microcontrolador LoRa32 configurado como maestro (Nodo 3) cuenta con un menú de posibles respuestas y sugerencias para el usuario, esto es de gran ayuda para guiarlo hacia la información que desea consultar al sistema de sensores inteligentes. En la Figura 50 se puede observar el diagrama de flujo para el Proceso C.

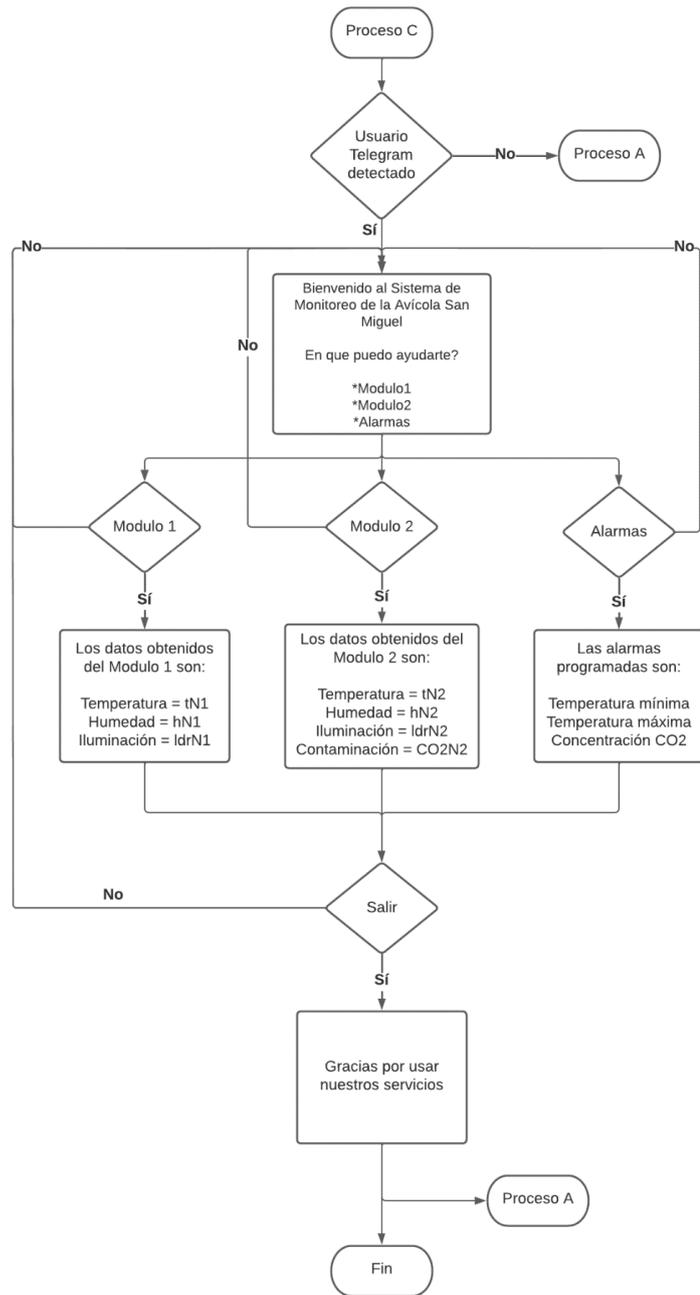


Figura 50. Diagrama de flujo Proceso C.
Fuente: Investigador

Servidor IoT

Los datos son enviados a servidores IoT para que los usuarios puedan acceder a la información de la Avícola San Miguel en tiempo real y desde cualquier dispositivo, ya sea una computadora o un smartphone. En el caso de Thingier.io se ha desarrollado una interface gráfica (HMI) para observar las variables de temperatura, humedad, luminosidad y ppm de CO2 desde una computadora, cabe recalcar que las variables se encuentran diferencias por etiquetas y por nodos. Por otro lado, REMOTEXY es una

página web que permite desarrollar interfaces gráficas para dispositivos móviles que se comunican mediante Wifi con el microcontrolador LoRa32 configurado como maestro.

Implementación de Thinger.io

Para visualizar y manipular las variables del sistema inteligente desde una computadora se decidió utilizar el servidor IoT llamado Thinger IO debido a que es una plataforma de código abierto y no tiene un costo de mantenimiento. Sin embargo, esta plataforma está limitada por el número de dispositivos que se pueden conectar simultáneamente, pero en este caso solamente se requiere de un punto de conexión debido a que se estableció una red maestro esclavo con todos los sensores inteligentes. En la Figura 51 se observa la pantalla principal de la página web Thinger.io.

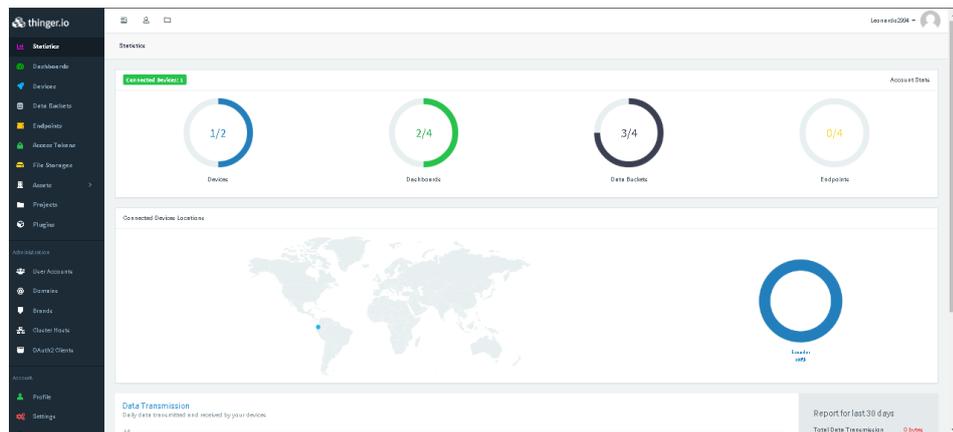


Figura 51. Pantalla principal de Thinger.io.

Fuente: Investigador

A continuación, se procede a agregar un nuevo dispositivo con las respectivas credenciales para que el Nodo 3 pueda comunicarse con el servidor y actualizar las variables de monitoreo en tiempo real como se observa en la Figura 52.

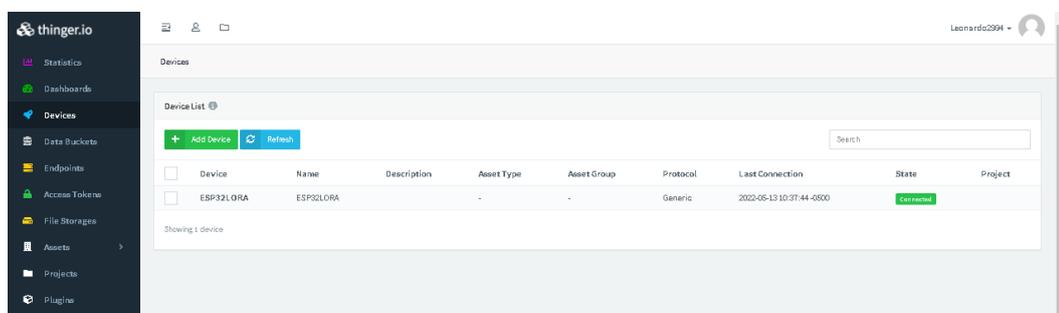


Figura 52. Nuevo dispositivo agregado a Thinger.io.

Fuente: Investigador

En este caso se ha desarrollado un HMI ubicando las variables medidas de cada uno de los Nodos por separado, en la Figura 53 se observa las variables de temperatura, humedad, luminosidad, lux, el historial de las medidas obtenidas en un rango de tiempo variable, este puede ser de 1 min, 10 min, 1 hora, 1 día, etc.

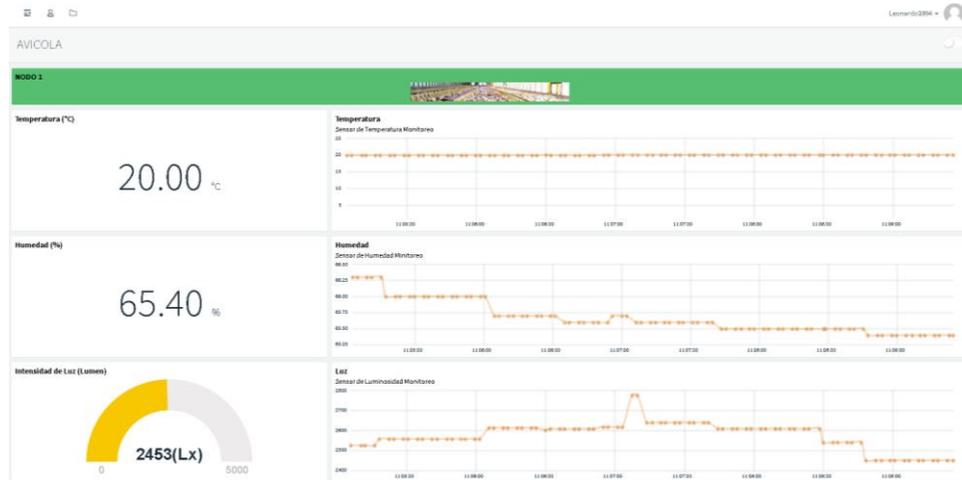


Figura 53. HMI de variables para el Nodo 1.

Fuente: Investigador

En la segunda parte del HMI se puede observar las variables del Nodo 2 como son: temperatura, humedad, iluminación y ppm de CO2 con los valores de 21.4°C, 54.5%, 2033 Lux y 300ppm, respectivamente. De la misma manera el Nodo 2 cuenta con un historial para visualizar el comportamiento de las variables en el tiempo como se puede observar en la Figura 54.



Figura 54. HMI de variables para el Nodo 2.

Fuente: Investigador

Aplicación

Implementación de REMOTEXY

La página web REMOTEXY permite desarrollar interfaces gráficas para diferentes aplicaciones con la gestión de varios elementos de control, visualización y decoración ubicados en cualquier parte de la pantalla del smartphone. En la Figura 55 se puede observar la pantalla principal para desarrollar la interfaz gráfica, donde visualizan elementos de control, sensores del teléfono, números y elementos de visualización.

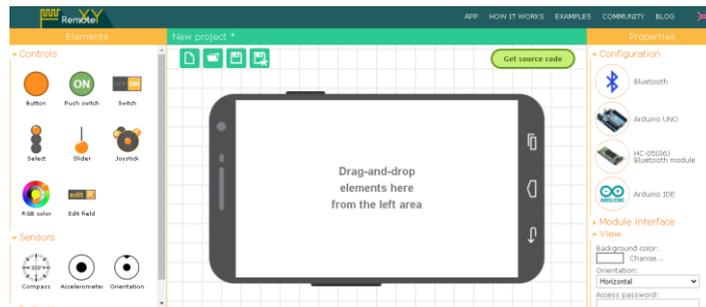


Figura 55. Pantalla principal de RemoteXY.

Fuente: Investigador

En esta sección se crean las variables que servirán para mostrar los valores de temperatura, humedad, iluminación y ppm de CO₂ obtenidos por el Nodo 3 desde los Nodos 1 y Nodo 2 ubicados en el galpón de pollos. Cabe recalcar que una vez que se haya terminado de implementar todos los elementos sobre la pantalla del dispositivo móvil se debe generar el código de programación que será implementado posteriormente en el microcontrolador LoRa32, para lo cual se debe activar la opción de “*Get Source Code*”.

Una vez programado el microcontrolador se procede a configurar la aplicación denominada como “*Remote XY: Arduino Control*” en el dispositivo móvil del usuario que será el encargado de monitorear la Avícola San Miguel. En la Figura 56 se puede observar este proceso de instalación y los iconos de identificación para acceso rápido.

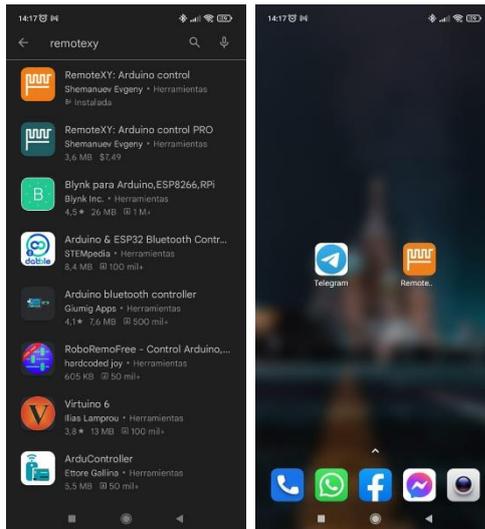


Figura 56. Instalación de la aplicación RemoteXY
Fuente: Investigador

Para establecer una correcta comunicación entre el dispositivo móvil y el microcontrolador Lora32 configurado como maestro (Nodo 3) se debe agregar un nuevo dispositivo utilizando los siguientes parámetros:

- Cloud server IP o URL: `cloud.remotexy.com`
- Port: `6375`
- Device Token: `“* * * * *”`

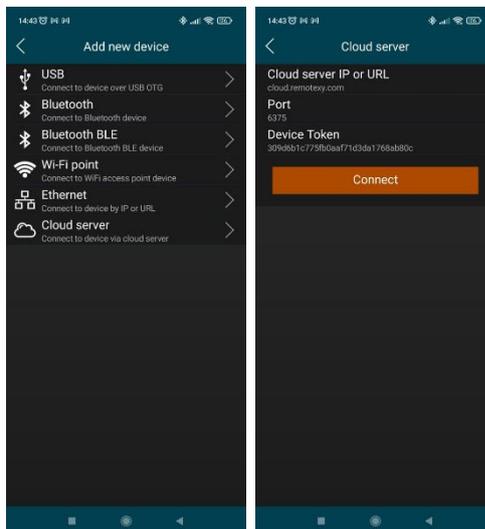


Figura 57. Agregar nuevo dispositivo inteligente.
Fuente: Investigador

Los parámetros utilizados corresponden a la programación implementada en el diagrama de flujo de la Figura 47 en el microcontrolador Nodo 3. De la misma manera en la Figura 57 se deben implementar los mismos parámetros con el objetivo de

establecer una comunicación adecuada entre los componentes de la red. Finalmente, se debe presionar el botón “Connect” para que el dispositivo verifique los parámetros suministrados y pueda agregar correctamente el dispositivo inteligente de monitoreo. Sin embargo, la comunicación se establece cuando existe el servicio de internet en el dispositivo móvil, ya sea por WiFi o servicio de datos por operadora telefónica. En la Figura 58 se puede observar el dispositivo agregado correctamente en la aplicación RemoteXY y también se muestra el acceso de la aplicación a la información suministrada por el Nodo3.

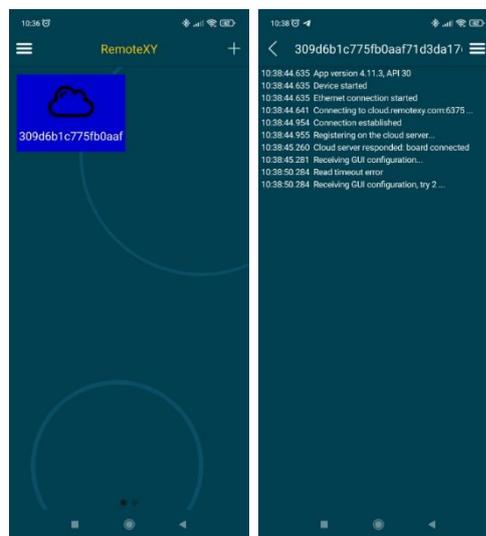


Figura 58. Establecer comunicación entre LoRa32 y el dispositivo móvil.

Fuente: Investigador

Finalmente, con una conexión exitosa se procede a mostrar el HMI desarrollado con la visualización en tiempo real de las variables de temperatura, humedad, luminosidad y ppm de CO2 de los Nodo 1 y Nodo 2, como se puede observar en la Figura 59.

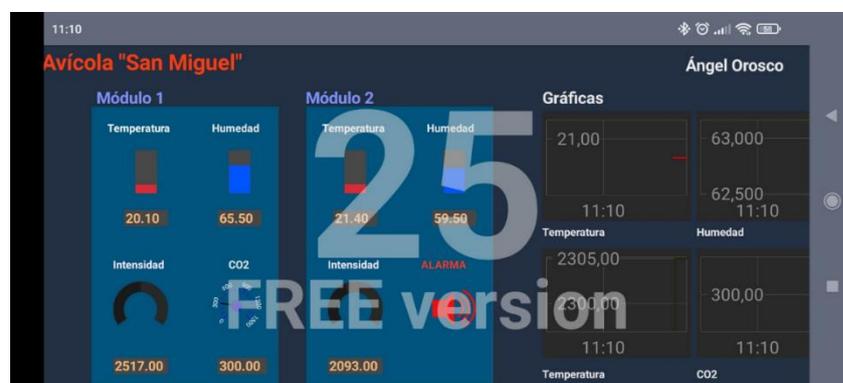


Figura 59. HMI desarrollado en RemoteXY.

Fuente: Investigador

Implementación Telegram

Se ha desarrollado un BOT de respuestas rápidas utilizando el servicio de mensajería de Telegram, además este servicio servirá para enviar alertas a los operadores de la Avícola San Miguel en caso de existir una condición de emergencia por temperatura alta, temperatura baja o por una alta concentración de CO2 en el galpón de pollos. Una de las ventajas que presenta Telegram es que los BOT desarrollados no deben estar asociados a un número de teléfono, facilitando su implementación en cualquier dispositivo móvil.

El proceso de creación de un BOT inicia agregando al usuario de Telegram llamado “BotFather” y enviando la siguiente secuencia de mensajes:

- **“/start”**: Mensaje para iniciar la creación de un nuevo BOT.
- **“/newbot”**: Comando utilizado para crear un nuevo BOT.
- **“Avicola”**: Nombre del BOT creado y donde se recibirá toda la información del sistema de monitoreo inteligente.
- **“AvicolaBot”**: Nombre del usuario para el BOT creado previamente.
- **“TOKEN”**: Dirección HTTP API necesaria para comunicarse con el microcontrolador LoRa32.

En la Figura 60 se puede observar la creación de un BOT en Telegram aplicando los comandos antes mencionados.

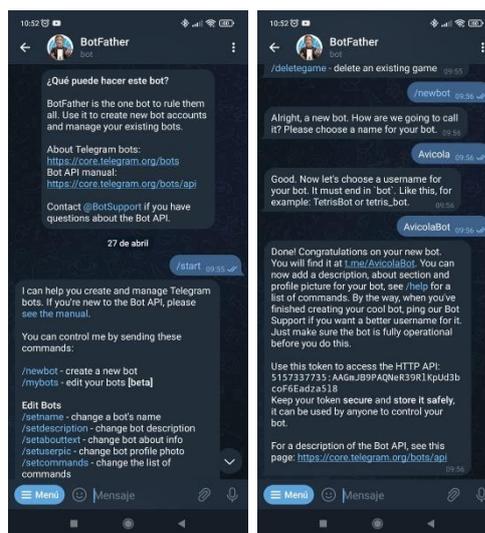


Figura 60. Creación de un BOT en Telegram.

Fuente: Investigador

Otro de los parámetros necesarios para implementar una correcta comunicación entre el dispositivo móvil con Telegram y el Nodo 3 es el denominado IDBOT, para lo cual se debe aplicar los comandos que se muestran en la Figura 61, de la siguiente manera:

- **“IDBot”**: Usuario propio de Telegram.
- **“/getid”**: Comando utilizado para obtener la información del ID del dispositivo móvil.

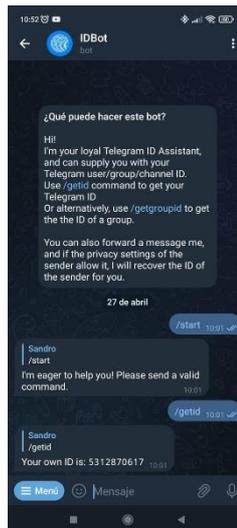


Figura 61. Comandos para obtener el IDBot.

Fuente: Investigador

La configuración de las respuestas automáticas se realiza con el objetivo de guiar al usuario hasta la información que desea consultar del sistema inteligente de monitoreo de variables de la avícola. A continuación, se describen los comandos utilizados:

- **“/Ayuda”**: Mensaje de bienvenida para el usuario.
- **“/Modulo1”**: Esta opción permite al usuario acceder a la información de las variables medidas por el Nodo 1.
- **“/Modulo2”**: Esta opción permite al usuario acceder a la información de las variables medidas por el Nodo 2.
- **“/Alarmas”**: Esta opción permite al usuario acceder a los valores de temperatura mínimo, máximo y concentración máxima de ppm de CO2.
- **“/Salir”**: Reinicia el BOT de respuestas rápidas y envía un mensaje de despedida.

En la Figura 62 se puede observar la interacción del BOT con el usuario para obtener información sobre los Nodo 1 y Nodo 2.

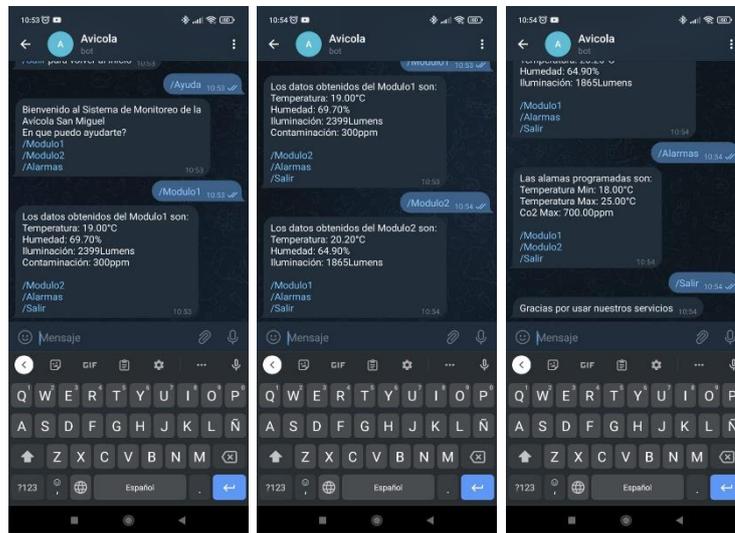


Figura 62. Configuración de respuestas rápidas del BOT.
Fuente: Investigador

Notificaciones Telegram

En esta sección se realiza las pruebas de funcionamiento de las notificaciones en la aplicación de celular llamada “*Telegram*”, para lo cual se utilizaron varios casos, por ejemplo: en el caso 1 se modificó mediante el servidor *Thinger.io* la temperatura mínima del sistema a un valor de 21.5°C, esto quiere decir que para cualquier medición de los nodos 1 y nodo 2 bajo este valor el módulo configurado como maestro debe enviar una notificación al usuario. En la Figura 63 se observa los valores actuales que tienen cada uno de los nodos.

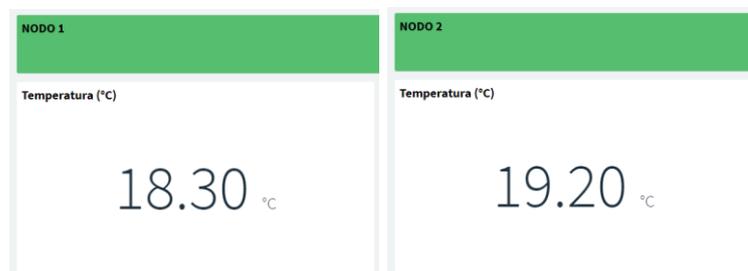


Figura 63. Medición 1 de temperatura del Nodo1 y Nodo2.

Fuente: Investigador

Por lo tanto, en la Figura 64 se observa las notificaciones recibidas tanto del Nodo 1 como del Nodo 2, informando al usuario que la temperatura del galpón de pollos está muy fría. En caso de que la temperatura vuelva subir sobre el valor mínimo las

notificaciones se detienen automáticamente. Cuando la alarma se encuentra activada, el nodo maestro envía mensajes constantemente en un periodo de 10 segundos.



Figura 64. Notificación de temperatura “Muy Fria”, recibida.

Fuente: Investigador

El caso 2 de pruebas se realizó modificando la temperatura máxima a un valor de 18.5°C, con el objetivo de validar el correcto funcionamiento de las notificaciones. En la Figura 65 se puede observar los valores de temperatura obtenidos por el nodo 1 y nodo 2 durante la prueba realizada.

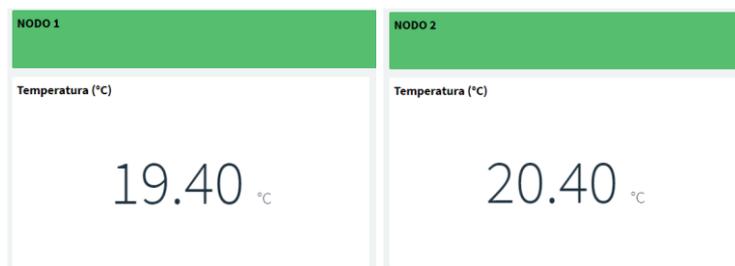


Figura 65. Medición 2 de temperatura del Nodo1 y Nodo2.

Fuente: Investigador

En este caso el usuario ha recibido un mensaje de temperatura “Muy Alta” a su aplicación de *Telegram*, tanto para el nodo 1 como para el nodo 2. En la Figura 66 se puede observar todas las notificaciones recibidas en el dispositivo móvil del usuario.



Figura 66. Notificación de temperatura “Muy Alta”, recibida.

Fuente: Investigador

Finalmente, en el caso 3 se realizó una prueba de la cantidad de CO₂ en ppm existente en el galpón, en este caso se consideró un valor de prueba de 700ppm como máximo y toda medición que supere dicho valor será considerado como una señal de alarma. En la Figura 67 se observa una medición de 1500ppm de CO₂ existentes en el galpón de pollos.

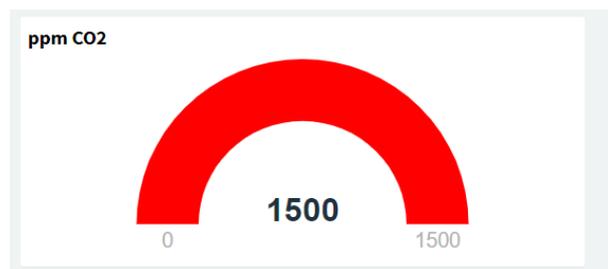


Figura 67. Medición CO₂ existente en el galpón.

Fuente: Investigador

En la Figura 68 se observa las notificaciones recibidas en el dispositivo móvil del usuario para el caso de alta concentración de CO₂ en el galpón de pollos, esto significa que se debe ventilar el galpón para bajar la contaminación del ambiente.



Figura 68. Notificación de CO2 “Muy Alto”, recibida.

Fuente: Investigador

Las notificaciones son recibidas en periodos de 10 segundos siempre y cuando las señales de alamar se mantengan activados, el usuario no deja de recibir las notificaciones hasta que no se encuentre solucionado el problema que ha originado la alarma.

Instalación de Nodos Finales

La instalación de los 3 módulos que componen el sistema de monitoreo con sensores inalámbricos inicia con la implementación de unas conexiones de corriente alterna desde los exteriores del galpón. Para lo cual se utilizó cable STP 2 x 16 AWG con una capacidad máxima de 13A y 600V, como se muestra en la Figura 69.



Figura 69. Conexiones de corriente alterna hacia el galpón.

Fuente: Investigador

Una vez realizado las conexiones pertinentes de voltaje se procede a instalar unas bases de madera de 200x200x3 mm de ancho, largo y alto, respectivamente; a una altura de 45 cm del suelo, como se observa en la Figura 70. Adicionalmente, se ubicó un tomacorriente a unos 25 cm de la base de madera, donde se procedió a conectar los Nodos 1 y Nodo 2.



Figura 70. Disposición de los Nodos 1 y Nodo 2.

Fuente: Investigador

3.1 Verificación de Hipótesis

Análisis de resultados

Inicialmente se debe considerar que las variables de temperatura, humedad, lúmenes y CO₂ no tienen una relación con el tiempo, sino más bien con las condiciones ambientales que se presenten durante el día y la noche. Por lo tanto, se ha decidido establecer los siguientes tiempos para la adquisición de datos y visualización en los valores en los módulos y en los respectivos servidores IoT, como se detalla en los diagramas de flujo de las Figura 41, Figura 44 y Figura 47.

- Nodo 1 y Nodo 2 adquisición de datos cada 1 segundo.
- Envío de datos al Nodo 3 (maestro) cada 3 segundos.
- Envío de datos al servidor Thinger.io 1 vez cada 2 minutos.
- Envío de datos al servidor RemoteXY 1 vez cada 2 minutos..

En la Figura 71 se observa el tratamiento de los datos para su respectivo análisis comparativo.



Figura 71. Proceso de tratamiento de datos.

Fuente: Investigador

Mediciones Nodo 1 y Nodo 2

En la Figura 72 se obtienen los valores medidos de las variables de temperatura, humedad, lúmenes y CO2 del Nodo 1 y Nodo 2 con su respectiva presentación en la pantalla OLED del módulo Lora 32.

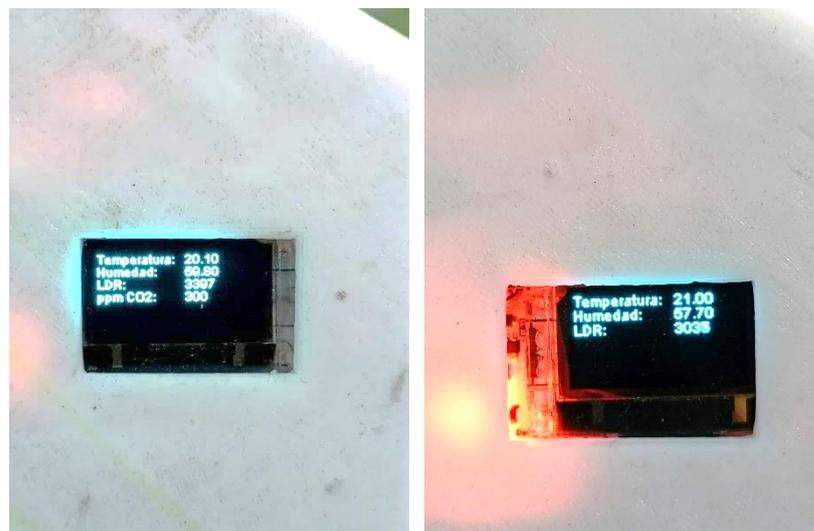


Figura 72. Datos obtenidos del Nodo 1 y Nodo 2.

Fuente: Investigador

Por otro lado, estos valores son representados en la Tabla 25, donde se obtiene el promedio de cada una de las variables aplicando la formula: $Promedio = \sum N \text{ valores} / N$

Tabla 25. Datos medidos en el Nodo 1 y Nodo 2.

	Nodo 1	Nodo 2	Promedio
Temperatura	20.1 °C	21 °C	20.55 °C
Humedad	59.80 %	57.7 %	58.75 %
Lúmenes	3397 lúmenes	3035 lúmenes	3216 lúmenes
CO2	300 ppm		300 ppm

Fuente: Investigador

Recepción de datos Nodo 3 (Maestro)

En la Figura 73 se puede observar los valores recibidos del Nodo 1 y Nodo 2 en el Nodo 3 para ser enviados posteriormente hacia los servidores IoT. En este nodo se realiza la visualización de los datos cada 20 segundos.



Figura 73. Datos recibidos en el Nodo 3 (Maestro).

Fuente: Investigador

En la Tabla 26 se observan los datos que ha recibido el Nodo 3 después que el Nodo 1 y Nodo 2 enviara la información mediante la comunicación Lora.

Tabla 26. Datos recibidos en el Nodo 3.

	Nodo 1	Nodo 2	Promedio
Temperatura	20.1 °C	21 °C	20.55 °C
Humedad	60.10 %	58.10 %	59.1 %
Lúmenes	4023 lúmenes	4095 lúmenes	4059 lúmenes
CO2	300 ppm		300 ppm

Fuente: Investigador

Visualización de datos en RemoteXY

Una vez que el Nodo 3 ha recibido los datos del Nodo 1 y Nodo 2 se procede a enviar hacia el servidor RemoteXY para que el usuario pueda visualizar esta información en la aplicación móvil. Sin embargo, se debe considerar que la información es observada con un retardo dependiendo del servicio de internet disponible tanto para enviar como para recibir los datos, como se observa en la Figura 74.



Figura 74. Datos recibidos RemoteXY.

Fuente: Investigador

De la misma manera que en los casos anteriores se procede a registrar los valores recibidos en la aplicación móvil en la Tabla 27.

Tabla 27. Datos recibidos en RemoteXY.

	Nodo 1	Nodo 2	Promedio
Temperatura	20.1 °C	21.1 °C	20.6 °C
Humedad	59.8 %	51.1 %	55.45 %
Lúmenes	3487 lúmenes	3115 lúmenes	3301 lúmenes
CO2	300 ppm		300 ppm

Fuente: Investigador

Visualización de datos en Thinger.io

Finalmente, en la Figura 75 se muestran los valores recibidos de las variables de temperatura, humedad, lúmenes y CO2 en el servidor Thinger.io. La visualización de estos datos solamente se puede realizar mediante la utilización de una computadora o laptop.

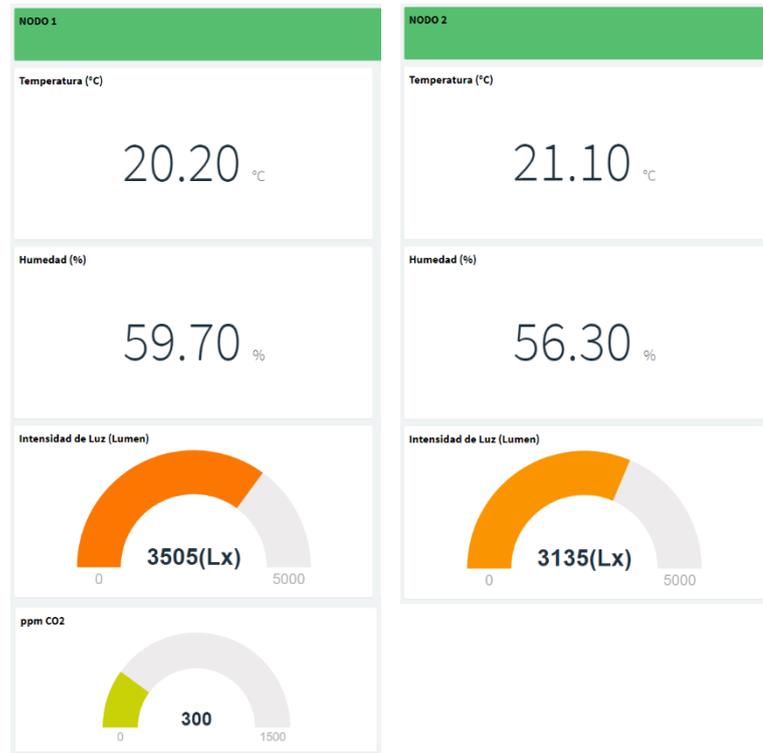


Figura 75. Datos recibidos en Thinger.io.

Fuente: Investigador

En la Tabla 28 se observa los datos recibidos en el servidor Thinger.io y su respectivo promedio.

Tabla 28. Datos recibidos en Thinger.io.

	Nodo 1	Nodo 2	Promedio
Temperatura	20.20 °C	21.1 °C	20.65 °C
Humedad	59.7 %	56.3 %	58 %
Lúmenes	3505 lúmenes	3135 lúmenes	3320 lúmenes
CO2	300 ppm		300 ppm

Fuente: Investigador

Pruebas diarias del sistema

El segundo conjunto de pruebas realizadas consistió en recopilar las variables de temperatura, humedad, lúmenes y CO2 obtenidas por el sistema en tres días diferentes, en intervalos de una hora, iniciando desde las 6am hasta las 9pm. En la Tabla 29 se puede observar los valores obtenidos, cabe recalcar que la medición se realiza dentro del galpón y el operario intenta mantener unas condiciones ambientales controladas. Los horarios de medición son los mismos para los tres días y se muestran en las Tabla 29, Tabla 30 y Tabla 31 .

Tabla 29. Datos obtenidos en el día uno de medición.

Fecha y Hora	Temperatura °C	Humedad %	Lúmenes Lux	CO2 ppm
01/07/2022 06:00	20	85	3590	328
01/07/2022 07:00	20,5	83	3740	496
01/07/2022 08:00	20	80,5	3890	372
01/07/2022 09:00	20,5	78,6	4040	477
01/07/2022 10:00	21	78	4190	460
01/07/2022 11:00	21,5	78	4340	469
01/07/2022 12:00	22	69	4490	493
01/07/2022 13:00	23	65	4364	465
01/07/2022 14:00	24	60	4238	309
01/07/2022 15:00	24,5	55	4112	495
01/07/2022 16:00	24	57	3986	487
01/07/2022 17:00	23	60	3860	498
01/07/2022 18:00	22	75	3734	348
01/07/2022 19:00	22	75	3608	449
01/07/2022 20:00	21,5	78	3482	471
01/07/2022 21:00	21	80	3356	419

Tabla 30. Datos obtenidos en el día dos de medición.

Fecha y Hora	Temperatura °C	Humedad %	Lúmenes Lux	CO2 ppm
02/07/2022 06:00	21,5	82,5	3725	373
02/07/2022 07:00	22	80,5	3875	520
02/07/2022 08:00	21,5	78	4025	452
02/07/2022 09:00	22	76,1	4175	477
02/07/2022 10:00	22,5	75,5	4325	428
02/07/2022 11:00	23	75,5	4475	519
02/07/2022 12:00	23,5	66,5	4625	498
02/07/2022 13:00	24,5	62,5	4499	480
02/07/2022 14:00	25,5	57,5	4373	388

02/07/2022 15:00	26	52,5	4247	435
02/07/2022 16:00	25,5	54,5	4121	354
02/07/2022 17:00	24,5	57,5	3995	377
02/07/2022 18:00	23,5	72,5	3869	474
02/07/2022 19:00	23,5	72,5	3743	419
02/07/2022 20:00	23	75,5	3617	365
02/07/2022 21:00	22,5	77,5	3491	426

Fuente: Investigador

Tabla 31. Datos obtenidos en el día tres de medición.

Fecha y Hora	Temperatura °C	Humedad %	Lúmenes Lux	CO2 ppm
03/07/2022 06:00	22,5	81	3625	538
03/07/2022 07:00	23	79	3775	417
03/07/2022 08:00	22,5	76,5	3925	409
03/07/2022 09:00	23	74,6	4075	350
03/07/2022 10:00	23,5	74	4225	485
03/07/2022 11:00	24	74	4375	509
03/07/2022 12:00	24,5	65	4525	373
03/07/2022 13:00	25,5	61	4399	398
03/07/2022 14:00	26,5	56	4273	465
03/07/2022 15:00	27	51	4147	504
03/07/2022 16:00	26,5	53	4021	506
03/07/2022 17:00	25,5	56	3895	399
03/07/2022 18:00	24,5	71	3769	514
03/07/2022 19:00	24,5	71	3643	558
03/07/2022 20:00	24	74	3517	362
03/07/2022 21:00	23,5	76	3391	521

Fuente: Investigador

De acuerdo a las mediciones realizadas se puede interpretar que las condiciones ambientales del galpón son relativamente constantes todos los días, esto se debe a la utilización de criadoras a gas para mantener una temperatura constante durante todo el día. Sin embargo, se puede observar un decremento en la humedad en las horas donde existe más intensidad del sol desde las 12:00 hasta las 15:00. De la misma manera los lúmenes varían de acuerdo a la cantidad de luz solar existente y que ingresa al galpón por las ventanas y techo transparente, también se debe considerar que existen focos dentro del galpón para aumentar la velocidad de crecimiento de los pollos. Finalmente, la cantidad de CO2 es variable, sin llegar a un nivel de contaminación debido a que existe una buena ventilación dentro del galpón, pero si el operador se olvida de abrir las ventanas puede ocasionar asfixia en los pollos.

Error de medición

Para validar los datos medidos se utilizó tres diferentes aplicaciones disponibles en un celular modelo 31061110AG marca Xiaomi, estas aplicaciones son:

- Room Temperature: utilizada para medir la temperatura interna de un cuarto.
- Hygrometer: sirve para medir la humedad del ambiente.
- Lux Light Meter: utilizado para medir la cantidad de lúmenes existentes en el ambiente.

Aplicando la fórmula de error relativo se pretende cuantificar los valores obtenidos por parte del sistema de medición y las aplicaciones del dispositivo móvil dentro del galpón para pollos, para lo cual se utilizó la siguiente fórmula:

$$Error = \frac{|valor\ real - valor\ medido|}{valor\ real} * 100\%$$

Donde:

Valor real: Es el Valor medido por la aplicación dedicada a ese tipo de variables.

Valor Medido: Es el valor obtenido por el sistema desarrollado.

En la Tabla 32 se observa 15 valores correspondientes al error de medición de la variable de temperatura, donde el 100% representa valores totalmente fuera del rango de medición real y un valor de 0% significa que los datos medidos son exactamente iguales a los valores reales del ambiente analizado.

Tabla 32. Error de medición en temperatura.

Muestra	Temperatura °C		Error Relativo (%)
	Sistema	Aplicación	
1	28,4	27,5	3,2727
2	23,1	23,6	2,1186
3	26,3	25,5	3,1373
4	26,3	26,7	1,4981
5	24,2	24,4	0,8197
6	23,1	23,2	0,4310
7	27,3	26,5	3,0189
8	26,3	26,9	2,2305
9	25,2	24,9	1,2048
10	25,2	25,9	2,7027
11	28,4	28,9	1,7301

12	28,4	28,6	0,6993
13	23,1	23,6	2,1186
14	25,2	24,4	3,2787
15	24,2	24,8	2,4194

Fuente: Investigador

Por otro lado, en la Tabla 33 se muestra el error obtenido de las 15 mediciones para la variable de temperatura, donde el máximo error encontrado es de 8.25% y el mínimo es de 0.77%, cabe recalcar que esta variable tiene varios factores que involucran su variabilidad durante el día.

Tabla 33. Error de medición en humedad.

Muestra	Humedad %		Error Relativo (%)
	Sistema	Aplicación	
1	81	88,3	8,2673
2	79	79,8	1,0025
3	76,5	77,1	0,7782
4	74,6	69,4	7,4928
5	74	80,7	8,3024
6	74	78,4	5,6122
7	65	66,5	2,2556
8	61	63,0	3,1746
9	56	59,4	5,7239
10	51	55,1	7,4410
11	53	55,0	3,6364
12	56	54,3	3,1308
13	71	73,8	3,7940
14	71	69,6	2,0115
15	74	69,6	6,3218

Fuente: Investigador

La variable de lúmenes se ve afectada por los focos instalados internamente en el galpón de pollos y por el ingreso de luz natural del sol a través de las ventanas y el techo de material transparente, la cuantificación del error se observa en la Tabla 34.

Tabla 34. Error de medición en lúmenes.

Muestra	Lúmenes (Lux)		Error Relativo (%)
	Sistema	Aplicación	
1	3625	3879	6,5481
2	3775	3662	3,0857
3	3925	4125	4,8485
4	4075	3871	5,2700
5	4225	4056	4,1667

6	4375	4475	2,2346
7	4525	4344	4,1667
8	4399	4091	7,5287
9	4273	4401	2,9084
10	4147	4247	2,3546
11	4021	4051	0,7406
12	3895	3995	2,5031
13	3769	3995	5,6571
14	3643	3315	9,8944
15	3517	3587	1,9515

Fuente: Investigador

Finalmente, el error de medición obtenido para la variable de CO2 se muestra en la Tabla 35.

Tabla 35. Error de medición en CO2.

Muestra	CO2		Error Relativo (%)
	Sistema	Aplicación	
1	538	506	6,3241
2	417	409	1,9560
3	409	401	1,9950
4	350	329	6,3830
5	485	490	1,0204
6	509	520	2,1154
7	373	358	4,1899
8	398	405	1,7284
9	465	460	1,0870
10	504	505	0,1980
11	506	541	6,4695
12	399	401	0,4988
13	514	516	0,3876
14	558	525	6,2857
15	362	358	1,1173

Fuente: Investigador

Como se puede observar se ha tomado 15 muestras de cada variable monitorizada lo cual muestra los resultados en la siguiente en la

Tabla 36 donde los valores no muestran ser mayores al 10% por tal motivo los equipos implementados no representan un peligro para las aves y van a mejorar la productividad en dicho galpón.

Tabla 36. Error relativo Máximo y Mínimo

<p>Temperatura</p> <p>Error relativo Máximo: 0,4310</p> <p>Error relativo Mínimo: 3,2787</p>
<p>Humedad</p> <p>Error relativo Máximo: 8,3024</p> <p>Error relativo Mínimo: 0,7782</p>
<p>Lux</p> <p>Error relativo Máximo: 9,8944</p> <p>Error relativo Mínimo: 0,7406</p>
<p>CO2</p> <p>Error relativo Máximo: 6,4695</p> <p>Error relativo Mínimo: 0,1980</p>

Presupuesto del proyecto

En esta sección se realiza una descripción detallada de los elementos utilizados en el desarrollo del presente proyecto considerando los materiales electrónicos utilizados como sensores, controladores y alimentación. También se muestran los valores correspondientes a los servicios de impresión 3D y servicio de ruteado CNC para el desarrollo de los circuitos electrónicos. Finalmente se muestran los costos de implementación en el galpón y costos administrativos.

Tabla 37. Presupuesto de inversión del proyecto.

MATERIALES				
Item	Materiales	Cant	V. Unidad	V. Total
1	Modulo Heltec ESP32 LoRa	3	\$60	\$180
2	Modulo Relé	1	\$5.6	\$5.6
3	Sensor de temperatura y humedad DHT22	3	\$15	\$45
4	Sensor MQ2 para detectar CO2	1	\$60	\$60
5	Sensor de lúmenes LDR	3	\$9.6	\$28.8
6	Conector a 5 Vdc	3	\$2.25	\$6.75
7	Cable STP 2 x16 AWG	1	\$20	\$20
8	Cables electrónicos protoboard	1	\$35	\$35
9	Cargador 5Vdc	3	\$6.5	\$19.5
			TOTAL	\$400.65

SERVICIOS				
Item	Materiales	Cant	V. Unidad	V. Total
1	Construcción de circuitos	3	\$45	\$135
2	Impresión 3D	3	\$42.5	\$127.5
3	Instalación eléctrica	2	\$25	\$50
			TOTAL	\$312.5
OFICINA				
Item	Materiales	Cant	V. Unidad	V. Total
1	Artículos de oficina: Bolígrafo, papel, etc.	1	\$80	\$80
			TOTAL	\$80
CAPITAL DE TRABAJO				
Item	Materiales	Cant	V. Unidad	V. Total
1	Transporte y salida a campo	1	\$150	\$150
			TOTAL	\$150

PRESUPUESTO	
Activos Fijos	\$793.15
Capital de trabajo	\$150
Subtotal	\$943.15
Imprevistos 10%	\$94.3
TOTAL	\$1037.46

Análisis Costo Beneficio

El presente proyecto se desarrolló con materiales de fácil adquisición lo cual hace que tenga un impacto económico favorable para su adquisición y su implementación en avícolas que necesiten monitorear contantemente las condiciones ambientales de los pollos, este tipo de sistemas es importante para garantizar un ambiente ideal para el crecimiento y desarrollo saludable de los pollos recién nacidos.

En esta sección se realizará un análisis del funcionamiento del sistema de monitoreo inteligente con sensores inalámbricos instalado en la Avícola “San Miguel”. Este sistema consta de 3 Nodos o módulos conectados en diferentes ubicaciones del galpón, para alimentar cada uno de los módulos se utilizó un cargador de 12Vdc a 3 amperios con una potencia de 36W, como se muestra en la Tabla 38.

Tabla 38. Consumo energético del sistema de monitoreo.

Maquina	Voltaje	Potencia	Frecuencia	Corriente
Nodo 1	110 Vac	36 W	60 Hz	0.327 Amp
Nodo 2	110 Vac	36 W	60 Hz	0.327Amp
Nodo 3	110 Vac	36 W	60 Hz	0.327Amp
Total	110 Vac	108 W	60 Hz	0.98 Amp

Fuente: Investigador

Considerando la utilización de los tres dispositivos eléctricos durante 24 horas se obtiene el consumo energético descrito en la Tabla 39.

Tabla 39. Consumo energético por horas del sistema.

Maquina	Potencia W	Horas de trabajo (h)	KWh
Sistema de monitoreo inteligente	108	1	0.108
	108	8	0.864
	108	240	24.9

Fuente: Investigador

Según pliego tarifario vigente se considera tarifa general de bajo voltaje sin demanda ya que el suministro en el punto de entrega es inferior a 600 voltios (V). Esto quiere decir que el usuario o la empresa que tenga implementado el sistema de monitoreo tendría que pagar lo siguiente:

- Un cargo por comercialización en USD/consumidor-mes, independiente del consumo de energía.
- Cargos incrementales por energía expresados en USD/kWh, en función de la energía consumida [25].

En la Tabla 40 se puede observar los cargos tarifarios únicos por parte de la empresa eléctrica de la provincia de Cotopaxi.

Tabla 40. Cargos tarifarios únicos Cotopaxi.

Rango de consumo	Energía (usd/kwh)	Comercialización
1-300	0,082	1,414
Superior	0,093	1,414

Fuente: Investigador

A continuación, se procede a calcular los cargos tarifarios por KWh consumidos por el sistema de monitoreo en el tiempo de 1 hora, 8 horas y 240 horas (1 mes) de funcionamiento continuo.

Tabla 41. Cargos tarifarios únicos Cotopaxi.

Maquina	\$/KWh	KWh	Dólares
Sistema de monitoreo inteligente	0,093	0.108	\$ 0.01
	0,093	0.864	\$ 0.08
	0,093	24.9	\$ 2.31

Fuente: Investigador

Una vez que se obtienen el costo de funcionamiento del sistema de monitoreo de la Avícola “San Miguel” se procede a calcular el costo de la planilla de consumo eléctrico para los intervalos de tiempo de 1 hora, 8 horas y 30 días correspondientes a un mes de funcionamiento. En este punto se debe considerar pagos fijos que se debe realizar a la empresa eléctrica son el 1.41\$ para comercialización y 6.25\$ de alumbrado público. En la Tabla 42 se puede observar el pago de la planilla de luz mensual que pagaría la Avícola “San Miguel” con la utilización del sistema de monitoreo.

Tabla 42. Planilla de luz del sistema.

PLANILLA	USD 30 días	USD 8 horas	USD en 1 hora
Comercialización	\$ 1,41	\$ 0.047	\$ 5.87*10 ⁻³
Consumo energético	\$ 2.31	\$ 0.08	\$ 0.01
Alumbrado Publico	\$ 6,25	\$ 0.20833	\$ 0,02603
TOTAL, planilla	\$ 9.97	\$ 0.335	\$ 0.0419

Fuente: Investigador

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- Mediante la revisión bibliográfica y la investigación de campo aplicada en este proyecto se pudo determinar que las principales variables físicas que afectan el crecimiento de los pollos de 1 a 3 semanas son: temperatura, humedad, lúmenes y la presencia de CO₂ en el galpón. La temperatura es muy importante porque evita que los pollos se enfermen de gripe aviar y pierdan la vida por su corta edad debido a esto se estableció que la temperatura normal es superior a los 20°C y por debajo a los 25°C. Una humedad excesiva en el galpón es perjudicial para la crianza de pollos debido a que aparecen hongos, nematodos, moho, entre otros organismos perjudiciales para la salud por o tanto para evitar eso la humedad debe estar debajo del 60%. Los lúmenes aceleran el crecimiento de los pollos por lo cual los lúmenes deben estar en el rango de los 3216 lúmenes y el CO₂ puede ser causa de asfixia en los pollos.

Se implementó una red de sensores inteligentes utilizando microcontroladores

- Lora Esp32 debido a las características técnicas que el mismo ofrece como lo es su distancia de 3km, donde el Nodo 1 y Nodo 2 fueron configurados como esclavos y el Nodo 3 es el maestro de la red, la utilización de las librerías para los diferentes sensores, como el “DHT.h” en el microcontrolador Lora32 ayuda a comunicarse con el sensor de humedad DHT22. Los valores de medición de este sensor son de -40°C a 80°C con una precisión de $\pm 5^\circ\text{C}$ y un rango de humedad de 0 a 100% RH con una precisión de 2%. Los valores son obtenidos ejecutando la librería con los comandos `readHumidity()` y `readTemperature()` para leer la humedad y la temperatura, respectivamente
- El sistema inteligente desarrollado cuenta con varias opciones para monitorear, visualizar y alamar a los propietarios del galpón de pollos “San Miguel”. Una de las opciones de visualización móvil se presenta a través de la aplicación RemoteXY, donde el usuario puede acceder a la información de las variables medidas desde un teléfono celular y la plataforma Thinger.io permite visualizar y manipular los valores de las alarmas desde un computador. Por otro lado, se

utilizó la aplicación de Telegram para enviar un mensaje de texto al usuario cuando se presente una condición de emergencia en el galpón, ya sea por alta temperatura, baja temperatura o por un alto nivel de concentración de CO₂.

- Se pudo verificar el correcto funcionamiento del sistema de alarmas utilizando casos de prueba donde se forzó a medir una temperatura superior a 25°C para activar las alarmas de alta temperatura, se repitió el proceso para baja temperatura en un rango menor a 20°C y finalmente las alarmas de alta concentración de CO₂ para una cantidad superior a 700ppm, el módulo relé utilizado para la visualización de los actuadores tiene un rango de activación en la señal de control de 0 – 5Vdc, siendo 0 voltios una condición de OFF y 5Vdc representa una condición de ON.

Recomendaciones

- Para una implementación adecuada de un sistema de monitoreo inteligente, la avícola “San Miguel” debe contar con un servicio de internet permanente, de esta manera la información puede ser observada por los operadores. Sin embargo, se debe considerar que los dispositivos móviles deben contar con un servicio de datos para recibir los mensajes de texto en la aplicación de Telegram.
- La interpretación de los datos puede ser analizada mediante la implementación de una red neuronal que permita identificar el comportamiento de estas variables de acuerdo a la edad de los pollos.
- Para trabajos futuros se debe analizar el control de las criadoras de gas, la apertura y cierre de ventas, la activación y desactivación de los focos de iluminación, es decir, implementar un sistema de monitoreo y control dentro del galpón de pollos.

BIBLIOGRAFÍA

- [1] A. Osorio Betancur and C. A. González, “Desarrollo e implementación de una aplicación web para el monitoreo en tiempo real de temperatura, presión y humedad al interior de los galpones de aves de engorde utilizando redes de sensores inalámbricos,” Pontificia Universidad Javeriana, Cali, 2018.
- [2] R. Torres, “RED DE MÓDULOS XBEE PARA EL CONTROL Y MONITOREO DE TEMPERATURA E INTENSIDAD LUMINOSA PARA CRIADEROS AVÍCOLAS MEDIANTE MODO API CON VISUALIZACIÓN HMI,” Universidad Tecnológica de Israel , Quito , 2019.
- [3] M. del R. Vargas and C. Luna, “Prototipo para la Automatización del Control del Ambiente en el Área Avícola del Instituto Tecnológico de Huejutla,” Instituto tecnológico de Huejutla, México , 2020.
- [4] M. F. H. Hambali, R. K. Patchmuthu, and A. T. Wan, “IoT Based Smart Poultry Farm in Brunei,” in *2020 8th International Conference on Information and Communication Technology, ICoICT 2020*, Jun. 2020, pp. 1–5, doi: 10.1109/ICoICT49345.2020.9166331.
- [5] J. P. Mondol, K. R. Mahmud, M. G. Kibria, and A. K. Al Azad, “IoT based smart weather monitoring system for poultry farm,” in *2020 2nd International Conference on Advanced Information and Communication Technology, ICAICT 2020*, Nov. 2020, pp. 229–234, doi: 10.1109/ICAICT51780.2020.9333535.
- [6] P. Pomboza-Tamaquiza, R. Guerrero-López, D. Guevara-Freire, and V. Rivera, “Granjas avícolas y autosuficiencia de maíz y soya: caso Tungurahua-Ecuador,” *Estud. Soc. Rev. Aliment. Contemp. y Desarro. Reg.*, vol. 28, no. 51, pp. 0–0, Jan. 2018, doi: 10.24836/es.v28i51.511.
- [7] A. H. Silva Bastidas, “Consumo voluntario y rendimiento a la canal en pollos de engorde con residuos pos cosecha de *Theobroma cacao* L.,” 2016, Accessed: Aug. 19, 2021. [Online]. Available: <https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/23701>.

- [8] D. Barzallo, *Análisis de la Innovación Tecnológica Avícola Ecuatoriano en el Contexto De Industria 4.0 | Investigación Tecnológica ISTCT*, vol. 1. 2019.
- [9] CONAVE, “INICIO | CONAVE,” *Corporacion Nacional de avicultores del Ecuador*. <https://www.conave.org/> (accessed Mar. 07, 2022).
- [10] N. Mihailov, D. Todorov, L. Iliev, and P. Mashkov, “Investigation of an efficient poultry lighting solution,” 2014, doi: 10.1109/SIELA.2014.6871875.
- [11] W. Sarachai, P. Ratnapinda, and P. Khumwichai, “Smart notification system for detecting fan failure in evaporative cooling system of a poultry farm,” in *ECTI DAMT-NCON 2019 - 4th International Conference on Digital Arts, Media and Technology and 2nd ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering*, Apr. 2019, pp. 296–299, doi: 10.1109/ECTI-NCON.2019.8692266.
- [12] D. F. Ávila Pesántez and M. F. Vásquez Barrera, “Diseño de un sistema de control gerencial de plantas avícolas utilizando redes de sensores inalámbricos con tecnología open hardware,” Pontificia Universidad Católica del Ecuador Sede Ambato, Ambato, 2015.
- [13] S. A. NAVAS TÚQUERRES and R. M. MALDONADO BRITO, “EVALUACIÓN DE LAS RAZAS DE POLLOS PARRILLEROS ROSS 308 Y COBB 500 EN CONDICIONES DE ALTURA,” Universidad Técnica del Norte, Ibarra, 2009.
- [14] M. M. Estrada Pareja and S. M. Márquez Girón, “Interacción de los factores ambientales con la respuesta del comportamiento productivo en pollos de engorde,” Facultad de Ciencias Agrarias, 2005.
- [15] Aviagen, “Manual de Manejo Pollo de Engorde,” Huntsville, 2018.
- [16] A. Suarez, ““ESTRATEGIAS DE SUSTENTABILIDAD DE MICRO-REDES/SMART-FARM EN LA COMUNIDAD MAPUCHE JOSÉ PAINECURA DE HUEÑALIHUEN,”” Pontificia Universidad Católica de Chile, Santiago, 2018.
- [17] O. JiHye, D. H. Noh, and Y. H. Sohn, “Empirical test of Wi-Fi environment

- stability for smart farm platform,” in *Proceedings of the 2017 4th International Conference on Computer Applications and Information Processing Technology, CAIPT 2017*, Mar. 2018, vol. 2018-Janua, pp. 1–5, doi: 10.1109/CAIPT.2017.8320656.
- [18] Proximvo, “Custom IoT Solutions for Poultry Farms - Smart Farming Australia,” 2018. .
- [19] M. L. Liya and D. Arjun, “A Survey of LPWAN Technology in Agricultural Field,” in *Proceedings of the 4th International Conference on IoT in Social, Mobile, Analytics and Cloud, ISMAC 2020*, Oct. 2020, pp. 313–317, doi: 10.1109/I-SMAC49090.2020.9243410.
- [20] D. Evans, “The Internet of Things: How the Next Evolution of the Internet Is Changing Everything,” 2011.
- [21] I. Ordóñez Monfort, “Estudio de la arquitectura y el nivel de desarrollo de la red LoRaWAN y de los dispositivos LoRa.,” Jun. 2017, Accessed: Mar. 07, 2022. [Online]. Available: <http://hdl.handle.net/10609/64365>.
- [22] M. Iqbal, A. Y. M. Abdullah, and F. Shabnam, “An Application Based Comparative Study of LPWAN Technologies for IoT Environment,” in *2020 IEEE Region 10 Symposium, TENSYP 2020*, Jun. 2020, pp. 1857–1860, doi: 10.1109/TENSYP50017.2020.9230597.
- [23] B. Jiménez and A. Daniel, “Estudio de la cobertura de la modulación LoRa en la banda de 915[Mhz] en un ambiente urbano,” Universidad de Chile, Santiago, 2018.
- [24] F. Delgado Ferro, J. Navarro Ortiz, N. Chinchilla Romero, and J. J. Ramos Muñoz, “Arquitectura LoRaWAN para entornos sin cobertura,” Oct. 2021, Accessed: Mar. 07, 2022. [Online]. Available: <http://hdl.handle.net/10481/71143>.
- [25] P. Pickering, “Desarrollar con LoRa para aplicaciones IoT de baja tasa y largo alcance,” Jun. 29, 2017. <https://www.digiquey.com/es/articulos/develop-lora-for-low-rate-long-range-iot-applications> (accessed Jul. 13, 2022).

- [26] L. P. Martín, J. Carlos, and G. Baños, “enginy@eps Comunicación serie Maestro/Esclavo,” 2007.
- [27] A. Gómez, “DHT22 - TFG ,” 2016. <https://sites.google.com/site/tfgalbertogomez/home/proyectos/Sensores/DHT22> (accessed Jul. 13, 2022).
- [28] “MG811 CO2 Sensor Features,” Accessed: Jul. 13, 2022. [Online]. Available: <http://www.hwsensor.com>.

ANEXOS

Anexo A

Programación correspondiente al Nodo 1 (Esclavo) del sistema de monitoreo inteligente, desarrollado en el software Arduino IDE.

```
//Librería comunicación lora
#include "heltec.h"
#include <SimpleKalmanFilter.h>

SimpleKalmanFilter simpleKalmanFilter(2, 2, 0.02);

//Librería DHT humedad y temperatura
#include<DHT.h>

//Libreriar modulo Co2
#include "CO2Sensor.h"

//Configura del modulo
CO2Sensor co2Sensor(37, 0.99, 100);

//Variables de humedad y temperatura
#define DHTPIN 13
#define DHTTYPE DHT22

DHT dht(DHTPIN,DHTTYPE);
// LDR
const int ldrPin = A0;

#define BAND      433E6 //you can set band here directly,e.g.
868E6,915E6

String outgoing;           // outgoing message

byte localAddress = 0xBB;  // address of this device
byte destination = 0xFD;  // destination to send to

byte msgCount = 0;        // count of outgoing messages
long lastSendTime = 0;    // last send time
int interval = 2000;      // interval between sends

//Variables de los sensores
float h,t;
int ldrValor;
int val = 0;
int co2_val = 0;

void setup()
{
  Heltec.begin(true /*DisplayEnable Enable*/, true
/*Heltec.Heltec.Heltec.LoRa Disable*/, true /*Serial Enable*/, true
/*PABOOST Enable*/, BAND /*long BAND*/);
  Heltec.display->init();
  Heltec.display->flipScreenVertically();
  Heltec.display->setFont(ArialMT_Plain_10);
}
```

```

Heltec.display->clear();
Heltec.display->drawString(0, 0, "Iniciando!");
Heltec.display->display();
delay(1000);
Serial.begin(115200);
pinMode(ldrPin, INPUT);
dht.begin();
co2Sensor.calibrate();
}

void loop()
{
  //Sensor de humedad
  h = dht.readHumidity();
  t = dht.readTemperature();
  co2_val = simpleKalmanFilter.updateEstimate(co2Sensor.read());
  val = co2_val;

  if (co2_val <= 300) {
    val = 300;
  }

  if (co2_val >= 1500) {
    val = 1500;
  }

  // Serial.print("Humedad: ");
  // Serial.println(h);
  // Serial.print("Concentracion de Co2: ");
  Serial.println(val);
  // Serial.print("Temperatura: ");
  // Serial.println(t);
  //LDR
  ldrValor = analogRead(ldrPin);
  // Serial.print("LDR: ");
  // Serial.println(ldrValor );
  //Pantalla
  Heltec.display->clear();
  Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
  Heltec.display->setFont(ArialMT_Plain_10);
  Heltec.display->drawString(0, 0, "Temperatura: ");
  Heltec.display->drawString(70, 0, String(t));
  Heltec.display->drawString(0, 10, "Humedad: ");
  Heltec.display->drawString(70, 10, String(h));
  Heltec.display->drawString(0, 20, "LDR: ");
  Heltec.display->drawString(70, 20, String(ldrValor));
  Heltec.display->drawString(0, 30, "ppm CO2: ");
  Heltec.display->drawString(70, 30, String(val));
  Heltec.display->display();
  //Enviar Datos

  if (millis() - lastSendTime > interval)
  {
    sendMessage();
    lastSendTime = millis(); // timestamp the message
    interval = random(2000) + 2000; // 2-3 seconds
  }
}

```

```

}

void sendMessage()
{
    LoRa.beginPacket();           // start packet
    LoRa.write(destination);     // add destination address
    LoRa.write(localAddress);    // add sender
    address
    LoRa.print(t);
    LoRa.print(",");
    LoRa.print(h);
    LoRa.print(",");
    LoRa.print(ldrValor);
    LoRa.print(",");
    LoRa.print(val);
    LoRa.endPacket();           // finish packet and send
}
it
}

```

Anexo B

Programación correspondiente al Nodo 2 (Esclavo) del sistema de monitoreo inteligente, desarrollado en el software Arduino IDE.

```
#include "heltec.h"
#include<DHT.h>
//Variables de humedad y temperatura
#define DHTPIN 13
#define DHTTYPE DHT22
DHT dht(DHTPIN,DHTTYPE);
// LDR
const int ldrPin = A0;

#define BAND      433E6 //you can set band here directly,e.g.
868E6,915E6

String outgoing;           // outgoing message

byte localAddress = 0xBD;  // address of this device
byte destination = 0xFD;  // destination to send to

byte msgCount = 0;        // count of outgoing messages
long lastSendTime = 0;    // last send time
int interval = 2000;      // interval between sends

//Variables de los sensores
float h,t;
int ldrValor;

void setup()
{
  Heltec.begin(true /*DisplayEnable Enable*/, true
/*Heltec.Heltec.Heltec.LoRa Disable*/, true /*Serial Enable*/, true
/*PABOOST Enable*/, BAND /*long BAND*/);
  Heltec.display->init();
  Heltec.display->flipScreenVertically();
  Heltec.display->setFont(ArialMT_Plain_10);
  Heltec.display->clear();
  Heltec.display->drawString(0, 0, "Iniciando!");
  Heltec.display->display();
  delay(1000);
  Serial.begin(115200);
  pinMode(ldrPin, INPUT);
  dht.begin();
}

void loop()
{
  //Sensor de humedad
  h = dht.readHumidity();
  t = dht.readTemperature();
  Serial.print("Humedad: ");
  Serial.println(h);
  Serial.print("Temperatura: ");
  Serial.println(t);
}
```

```

//LDR
ldrValor = analogRead(ldrPin);
Serial.print("LDR: ");
Serial.println(ldrValor );
//Pantalla
Heltec.display->clear();
Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
Heltec.display->setFont(ArialMT_Plain_10);
Heltec.display->drawString(0, 0, "Temperatura: ");
Heltec.display->drawString(70, 0, String(t));
Heltec.display->drawString(0, 10, "Humedad: ");
Heltec.display->drawString(70, 10, String(h));
Heltec.display->drawString(0, 20, "LDR: ");
Heltec.display->drawString(70, 20, String(ldrValor));
Heltec.display->display();
//Enviar Datos

if (millis() - lastSendTime > interval)
{
    sendMessage();
    lastSendTime = millis();           // timestamp the message
    interval = random(2000) + 1000;   // 2-3 seconds
}
}

void sendMessage()
{
    LoRa.beginPacket();                // start packet
    LoRa.write(destination);           // add destination address
    LoRa.write(localAddress);          // add sender
address
    LoRa.print(t);
    LoRa.print(",");
    LoRa.print(h);
    LoRa.print(",");
    LoRa.print(ldrValor);
    LoRa.endPacket();                  // finish packet and send
it
}

```

Anexo C

Programación correspondiente al Nodo 3 (Maestro) del sistema de monitoreo inteligente, desarrollado en el software Arduino IDE.

```
#define REMOTEXY_MODE __ESP32CORE_WIFI_CLOUD

#include "heltec.h"
#include<Separador.h>
// WIFI IoT
#include <ThingerESP32.h>
#include <Wire.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>

//Aplicacion de celular
#include <WiFi.h>
#include <RemoteXY.h>

//Variables configuración telegram
#include "CTBot.h"

//Variables para el monitoreo en thinger.io
#define THINGER_SERIAL_DEBUG
#define USERNAME "Leonardo2994"
#define DEVICE_ID "ESP32LORA"
#define DEVICE_CREDENTIAL "IJTP_2ozdT_r?%E+"

//Variables para conectarse a la red WiFi
#define REMOTEXY_WIFI_SSID "xxxxxxxxxx"
#define REMOTEXY_WIFI_PASSWORD "xxxxxxxxxx"

//Variables para subir los datos al servidor móvil
#define REMOTEXY_CLOUD_SERVER "cloud.remotexy.com"
#define REMOTEXY_CLOUD_PORT 6376
#define REMOTEXY_CLOUD_TOKEN "309d6b1c775fb0aaf71d3da1768ab80c"

// RemoteXY configuración
#pragma pack(push, 1)
uint8_t RemoteXY_CONF[] =
{ 255, 0, 0, 105, 0, 237, 1, 15, 195, 0,
  130, 1, -11, 11, 34, 51, 179, 66, 1, -3,
  19, 4, 8, 1, 26, 71, 56, 11, 43, 9,
  9, 0, 6, 24, 165, 0, 0, 0, 0, 0,
  128, 187, 68, 0, 0, 150, 67, 0, 0, 200,
  66, 0, 0, 200, 66, 24, 0, 129, 0, 11,
  14, 9, 2, 8, 72, 117, 109, 101, 100, 97,
  100, 0, 66, 1, 14, 19, 4, 8, 190, 26,
  66, 130, -7, 43, 10, 8, 31, 8, 129, 0,
  -8, 14, 12, 2, 24, 84, 101, 109, 112, 101,
  114, 97, 116, 117, 114, 97, 0, 68, 1, 71,
  12, 27, 20, 8, 36, 129, 0, -7, 39, 10,
  2, 242, 73, 110, 116, 101, 110, 115, 105, 100,
  97, 100, 0, 129, 0, 14, 39, 4, 2, 8,
  67, 79, 50, 0, 68, 1, 99, 12, 27, 20,
  8, 16, 68, 1, 71, 37, 27, 20, 8, 94,
```

```

68, 1, 99, 37, 27, 20, 8, 135, 67, 4,
-5, 30, 8, 3, 2, 26, 11, 67, 4, 12,
30, 8, 3, 2, 26, 11, 67, 4, -7, 57,
10, 3, 2, 26, 11, 67, 4, 12, 57, 9,
3, 2, 26, 11, 130, 1, 28, 11, 34, 51,
179, 66, 1, 35, 19, 4, 8, 1, 26, 129,
0, 50, 14, 9, 2, 8, 72, 117, 109, 101,
100, 97, 100, 0, 66, 1, 53, 19, 4, 8,
190, 26, 66, 130, 32, 43, 10, 8, 31, 8,
129, 0, 31, 14, 12, 2, 24, 84, 101, 109,
112, 101, 114, 97, 116, 117, 114, 97, 0, 129,
0, 32, 39, 10, 2, 242, 73, 110, 116, 101,
110, 115, 105, 100, 97, 100, 0, 67, 4, 33,
30, 8, 3, 2, 26, 11, 67, 4, 51, 30,
8, 3, 2, 26, 11, 67, 4, 32, 57, 10,
3, 2, 26, 11, 129, 0, -10, 8, 13, 3,
190, 77, 195, 179, 100, 117, 108, 111, 32, 49,
0, 129, 0, 28, 8, 13, 3, 190, 77, 195,
179, 100, 117, 108, 111, 32, 50, 0, 129, 0,
-20, 1, 40, 4, 50, 65, 118, 195, 173, 99,
111, 108, 97, 32, 34, 83, 97, 110, 32, 77,
105, 103, 117, 101, 108, 34, 0, 129, 0, 97,
2, 19, 3, 8, 195, 129, 110, 103, 101, 108,
32, 79, 114, 111, 115, 99, 111, 0, 69, 0,
51, 44, 8, 8, 1, 129, 0, 71, 8, 12,
3, 8, 71, 114, 195, 161, 102, 105, 99, 97,
115, 0, 129, 0, 71, 33, 12, 2, 8, 84,
101, 109, 112, 101, 114, 97, 116, 117, 114, 97,
0, 129, 0, 99, 33, 9, 2, 8, 72, 117,
109, 101, 100, 97, 100, 0, 129, 0, 71, 59,
12, 2, 8, 84, 101, 109, 112, 101, 114, 97,
116, 117, 114, 97, 0, 129, 0, 99, 59, 4,
2, 8, 67, 79, 50, 0, 129, 0, 49, 39,
9, 2, 1, 65, 76, 65, 82, 77, 65, 0
};

// this structure defines all the variables and events of your
control interface
struct {

    // output variables
    int8_t M1T; // =0..100 level position
    float M1CO2;
    int8_t M1H; // =0..100 level position
    int8_t M1I; // =0..100 level position
    float TEMP_G;
    float HUM_G;
    float LUM_G;
    float CO2_G;
    char text_1[11]; // string UTF8 end zero
    char text_2[11]; // string UTF8 end zero
    char text_3[11]; // string UTF8 end zero
    char text_4[11]; // string UTF8 end zero
    int8_t M2T; // =0..100 level position
    int8_t M2H; // =0..100 level position
    int8_t M2I; // =0..100 level position
    char text_5[11]; // string UTF8 end zero
    char text_6[11]; // string UTF8 end zero
    char text_7[11]; // string UTF8 end zero

```

```

    int16_t sound_1; // =0 no sound, else ID of sound, =1001 for
example, look sound list in app

    // other variable
    uint8_t connect_flag; // =1 if wire connected, else =0

} RemoteXY;
#pragma pack(pop)

ThingyESP32 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);
//LORA
#define BAND    433E6 //you can set band here directly,e.g.
868E6,915E6
byte localAddress = 0xFD; // address of this device

// Datos Nodo 1
float hNd1 = 0;
float tNd1 = 0;
int ldrNd1 = 0;
int CO2Nd1 = 0;
// Datos Nodo 2
float hNd2 = 0;
float tNd2 = 0;
int ldrNd2 = 0;

Separador s; //Separador de datos

//Variables entrada
float min_temp = 18;
float max_temp = 25;
float ppmco2 = 700;

//Salida relees
int foco1 = 22;
int foco2 = 23;

//Variables histerecis
int his_temp = 5;
int his_co2 = 200;

//Variables para el bot
CTBot myBot;
String token = "5157337735:AAGmJB9PAQNeR39RlKpUd3bcoF6Eadza5l8";

void ImpresionNodo1() {

    Heltec.display->clear();
    Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
    Heltec.display->setFont(ArialMT_Plain_10);
    Heltec.display->drawString(0, 0, "Recibido Nodo 1");
    Heltec.display->drawString(0, 10, "Temperatura (°C): ");
    Heltec.display->drawString(85, 10, String(tNd1));
    Heltec.display->drawString(0, 20, "Humedad (%): ");
    Heltec.display->drawString(85, 20, String(hNd1));
    Heltec.display->drawString(0, 30, "LDR: ");
    Heltec.display->drawString(85, 30, String(ldrNd1));
    Heltec.display->drawString(0, 40, "ppm CO2: ");
    Heltec.display->drawString(85, 40, String(CO2Nd1));
}

```

```

    Heltec.display->display();
}
void ImpresionNodo2() {

    Heltec.display->clear();
    Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
    Heltec.display->setFont(ArialMT_Plain_10);
    Heltec.display->drawString(0, 0, "Recibido Nodo 2");
    Heltec.display->drawString(0, 10, "Temperatura (°C): ");
    Heltec.display->drawString(85, 10, String(tNd2));
    Heltec.display->drawString(0, 20, "Humedad (%): ");
    Heltec.display->drawString(85, 20, String(hNd2));
    Heltec.display->drawString(0, 30, "LDR: ");
    Heltec.display->drawString(85, 30, String(ldrNd2));
    Heltec.display->display();
}

int ayuda = 0;
int modulo1 = 0;
int modulo2 = 0;
#define BOTtoken
"5157337735:AAGmJB9PAQNeR39RlKpUd3bc0F6Eadza5l8" // your Bot Token
(Get from Botfather)
#define BOT_ID "5312870617"
WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);

void setup()
{

    //Iniciar un servidor en el celular
    RemoteXY_Init ();

    //Configuracion de pantalla LCD
    Heltec.begin(true /*DisplayEnable Enable*/, true
/*Heltec.Heltec.Heltec.LoRa Disable*/, true /*Serial Enable*/, true
/*PABOOST Enable*/, BAND /*long BAND*/);
    Heltec.display->init();
    Heltec.display->flipScreenVertically();
    Heltec.display->setFont(ArialMT_Plain_10);
    Heltec.display->clear();
    Heltec.display->drawString(0, 0, "Iniciando!");
    Heltec.display->display();
    delay(1000);
    Serial.begin(115200);

    //Configuracion de variables con el servidor IoT
    thing["Temp_N1"] >> outputValue(tNd1);
    thing["Hum_N1"] >> outputValue(hNd1);
    thing["Ldr_N1"] >> outputValue(ldrNd1);
    thing["CO2_N1"] >> outputValue(CO2Nd1);

    thing["Temp_N2"] >> outputValue(tNd2);
    thing["Hum_N2"] >> outputValue(hNd2);
    thing["Ldr_N2"] >> outputValue(ldrNd2);

    thing["min_temp"] << inputValue(min_temp);
    thing["max_temp"] << inputValue(max_temp);

```

```

thing["ppmco2"] << inputValue(ppmco2);

//Configuracion del bot en telegram
myBot.setTelegramToken(token);

pinMode(foco1, OUTPUT);
pinMode(foco2, OUTPUT);

}

void loop()
{
  RemoteXY_Handler ();
  RemoteXY.M1T = tNd1; // =0..100 level position
  RemoteXY.M1H = hNd1; // =0..100 level position

  RemoteXY.M1I = map(ldrNd1, 0, 5000, 0, 100); // =0..100 level
position
  RemoteXY.M1CO2 = CO2Nd1;

  RemoteXY.M2T = tNd2; // =0..100 level position
  RemoteXY.M2H = hNd2; // =0..100 level position
  RemoteXY.M2I = map(ldrNd2, 0, 5000, 0, 100); // =0..100 level
position

  RemoteXY.TEMP_G = (tNd1 + tNd2) / 2;
  RemoteXY.HUM_G = (hNd1 + hNd2) / 2;
  RemoteXY.LUM_G = (ldrNd1 + ldrNd2) / 2;
  RemoteXY.CO2_G = CO2Nd1;

  /*RemoteXY.text_1[11] = tNd1; // string UTF8 end zero
  RemoteXY.text_2[11] = hNd1; // string UTF8 end zero
  RemoteXY.text_3[11] = ldrNd1; // string UTF8 end zero
  RemoteXY.text_4[11] = CO2Nd1; // string UTF8 end zero
  RemoteXY.text_5[11] = tNd2; // string UTF8 end zero
  RemoteXY.text_6[11] = hNd2; // string UTF8 end zero
  RemoteXY.text_7[11] = ldrNd2; // string UTF8 end zero
  */
  dtostrf(tNd1, 0, 2, RemoteXY.text_1);
  dtostrf(hNd1, 0, 2, RemoteXY.text_2);
  dtostrf(ldrNd1, 0, 2, RemoteXY.text_3);
  dtostrf(CO2Nd1, 0, 2, RemoteXY.text_4);
  dtostrf(tNd2, 0, 2, RemoteXY.text_5);
  dtostrf(hNd2, 0, 2, RemoteXY.text_6);
  dtostrf(ldrNd2, 0, 2, RemoteXY.text_7);

  RemoteXY.sound_1 = 0;

  //LoRa
  onReceive(LoRa.parsePacket());
  //Wifi IoT
  thing.handle();

  if (tNd1 < min_temp ) {
    TBMessage msg;
    Serial.print("encender foco1");
  }
}

```

```

    digitalWrite(foco1, LOW);
    RemoteXY.sound_1 = 1001;
    bot.sendMessage(BOT_ID, "Temperatura Modulo1 Muy Fria\n", "");
    delay(1000);
}

if (tNd1 > max_temp ) {
    Serial.print("parpadear foco1");
    bot.sendMessage(BOT_ID, "Temperatura Modulo1 Muy Alta\n", "");
    digitalWrite(foco1, LOW);
    delay(1000);
    digitalWrite(foco1, HIGH);
    delay(1000);
    RemoteXY.sound_1 = 1001;
    delay(1000);
}

if ((tNd1 > min_temp + his_temp) && (tNd1 < max_temp - his_temp))
{
    Serial.print("apagar foco1");
    digitalWrite(foco1, HIGH);
    RemoteXY.sound_1 = 1001;
    delay(1000);
}

if (tNd2 < min_temp ) {
    bot.sendMessage(BOT_ID, "Temperatura Modulo2 Muy Fria\n", "");
    Serial.print("encender foco2 ");
    digitalWrite(foco2, LOW);
    RemoteXY.sound_1 = 1001;
    delay(1000);
}

if (tNd2 > max_temp ) {
    bot.sendMessage(BOT_ID, "Temperatura Modulo2 Muy Alta\n", "");
    Serial.print("parpadear foco2 ");
    digitalWrite(foco2, LOW);
    delay(1000);
    digitalWrite(foco2, HIGH);
    delay(1000);
    RemoteXY.sound_1 = 1001;
    delay(1000);
}

if ((tNd2 > min_temp + his_temp) && (tNd2 < max_temp - his_temp))
{
    Serial.print("apagar foco2");
    digitalWrite(foco2, HIGH);
    RemoteXY.sound_1 = 1001;
    delay(1000);
}

if (CO2Nd1 >= ppmco2) {
    bot.sendMessage(BOT_ID, "CO2 Muy Alto\n", "");
}

Serial.print("temperatura minima");
Serial.print(min_temp);
Serial.print("temperatura maximo");

```

```

    Serial.print(max_temp);
    Serial.print("ppm CO2");
    Serial.println(ppmco2);
    iot_bot();

    delay(1000);
}

void onReceive(int paqueteLong)
{
    if (paqueteLong == 0) return; //Si no existe un paquete retorna.

    // Lectura del paquete:
    int receptor = LoRa.read(); // Dirección del Receptor
    int emisor = LoRa.read(); // Dirección del emisor
    // Si el receptor no es este dispositivo retorna
    if (receptor != localAddress && receptor != 0xFF) {
        Serial.println("Este mensaje no es para este dispositivo.");
        return;
    }
    String mensaje_rcbNd1 = "";
    if (emisor == 0xBB) {
        Serial.println("NODO 1");
        while (LoRa.available())
        {
            mensaje_rcbNd1 += (char)LoRa.read();
        }

        String tN1 = s.separa(mensaje_rcbNd1, ',', 0);
        String hN1 = s.separa(mensaje_rcbNd1, ',', 1);
        String ldrN1 = s.separa(mensaje_rcbNd1, ',', 2);
        String CO2N1 = s.separa(mensaje_rcbNd1, ',', 3);

        tNd1 = tN1.toFloat();
        hNd1 = hN1.toFloat();
        ldrNd1 = ldrN1.toInt();
        CO2Nd1 = CO2N1.toInt();
        ImpresionNodo1();
    }
    String mensaje_rcbNd2 = "";
    if (emisor == 0xBD) {
        Serial.println("NODO 2");
        while (LoRa.available())
        {
            mensaje_rcbNd2 += (char)LoRa.read();
        }
        String tN2 = s.separa(mensaje_rcbNd2, ',', 0);
        String hN2 = s.separa(mensaje_rcbNd2, ',', 1);
        String ldrN2 = s.separa(mensaje_rcbNd2, ',', 2);
        tNd2 = tN2.toFloat();
        hNd2 = hN2.toFloat();
        ldrNd2 = ldrN2.toInt();
        ImpresionNodo2();
    }
    Serial.println("Recibido de: 0x" + String(emisor, HEX));
    Serial.println("Enviado a: 0x" + String(receptor, HEX));
    Serial.println("Mensaje:" + String(mensaje_rcbNd1) );
    Serial.println("Mensaje:" + String(mensaje_rcbNd2) );
    Serial.println("RSSI: " + String(LoRa.packetRssi()));
    Serial.println("Snr: " + String(LoRa.packetSnr()));
}

```

```

    Serial.println();
}

void iot_bot() {
    TBMMessage msg;
    if (CTBotMessageText == myBot.getNewMessage(msg)) {
        if (msg.text.equalsIgnoreCase("/Ayuda") && ayuda == 0) {
            myBot.sendMessage(msg.sender.id, "Bienvenido al Sistema de
Monitoreo de la Avícola San Miguel \nEn que puedo ayudarte?
\n/Modulo1 \n/Modulo2 \n/Alarmas");
            ayuda = 1;
        } else if (ayuda == 0)
        {
            myBot.sendMessage(msg.sender.id, "Envie la palabra \nAyuda
para información \n/Salir para volver al inicio ");
        }
        if (msg.text.equalsIgnoreCase("/Salir")) {
            myBot.sendMessage(msg.sender.id, "Gracias por usar nuestros
servicios");
            ayuda = 0;
        }

        if (msg.text.equalsIgnoreCase("/Cerrar")) {
            myBot.sendMessage(msg.sender.id, "Gracias por usar nuestros
servicios");
            ayuda = 0;
        }

        if (msg.text.equalsIgnoreCase("/Modulo1")) {
            String info_modulo1 = "Los datos obtenidos del Modulo1
son:\nTemperatura: ";
            info_modulo1 += String(tNd1);
            info_modulo1 += "°C\nHumedad: ";
            info_modulo1 += String(hNd1);
            info_modulo1 += "%\nIluminación: ";
            info_modulo1 += String(ldrNd1);
            info_modulo1 += "Lumens\nContaminación: ";
            info_modulo1 += String(CO2Nd1);
            info_modulo1 += "ppm\n\n/Modulo2";
            info_modulo1 += "\n/Alarmas";
            info_modulo1 += "\n/Salir";
            myBot.sendMessage(msg.sender.id, info_modulo1);
        }

        if (msg.text.equalsIgnoreCase("/Modulo2")) {
            String info_modulo2 = "Los datos obtenidos del Modulo2
son:\nTemperatura: ";
            info_modulo2 += String(tNd2);
            info_modulo2 += "°C\nHumedad: ";
            info_modulo2 += String(hNd2);
            info_modulo2 += "%\nIluminación: ";
            info_modulo2 += String(ldrNd2);
            info_modulo2 += "Lumens\n\n/Modulo1";
            info_modulo2 += "\n/Alarmas";
            info_modulo2 += "\n/Salir";
            myBot.sendMessage(msg.sender.id, info_modulo2);
        }

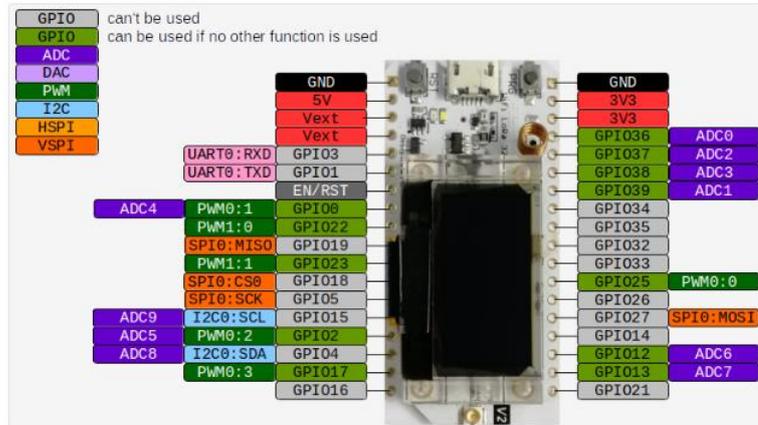
        if (msg.text.equalsIgnoreCase("/Alarmas")) {

```

```
String alarma = "Las alarmas programadas son:\nTemperatura Min:
";
alarma += String(min_temp);
alarma += "°C\nTemperatura Max: ";
alarma += String(max_temp);
alarma += "°C\nCo2 Max: ";
alarma += String(ppmco2);
alarma += "ppm\n\n/Modulo1";
alarma += "\n/Modulo2";
alarma += "\n/Salir";
myBot.sendMessage(msg.sender.id, alarma);
}
}
}
```

Anexo D

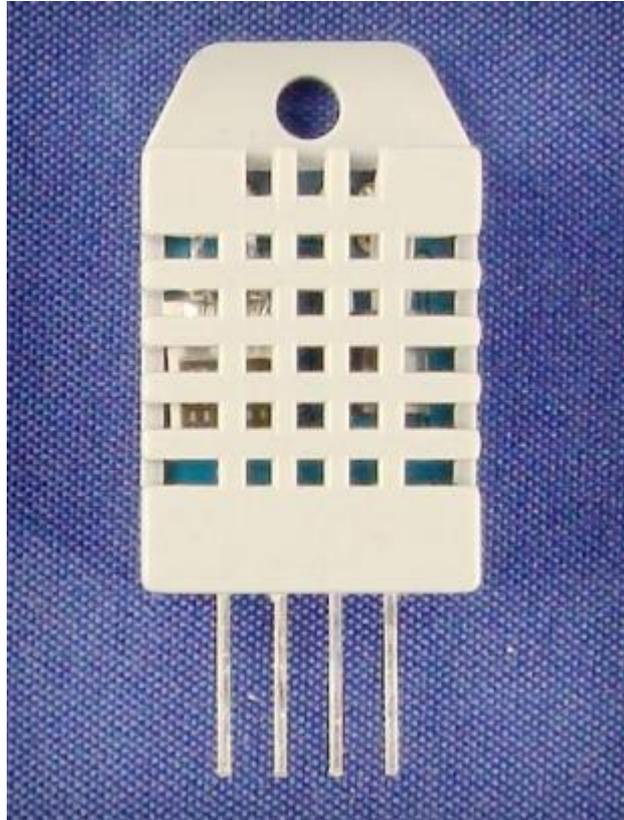
Shield Lora ESP32



Function	GPIOs	Remarks	Configuration
BTN0	GPIO0	low active	
LED0	GPIO25	high active	
ADC	GPIO36, GPIO39, GPIO37, GPIO38, GPIO0, GPIO2, GPIO12, GPIO13, GPIO4, GPIO15		ADC Channels
DAC			DAC Channels
PWM_DEV(0)	GPIO25, GPIO0, GPIO2, GPIO17		PWM Channels
PWM_DEV(1)	GPIO22, GPIO23		PWM Channels
I2C_DEV(0):SDA	GPIO4		I2C Interfaces
I2C_DEV(0):SCL	GPIO15	I2C_SPEED_FAST is used	I2C Interfaces
SPI_DEV(0):CLK	GPIO5	VSPI is used	SPI Interfaces
SPI_DEV(0):MISO	GPIO19	VSPI is used	SPI Interfaces
SPI_DEV(0):MOSI	GPIO27	VSPI is used	SPI Interfaces
SPI_DEV(0):CS0	GPIO18	VSPI is used	SPI Interfaces
UART_DEV(0):TxD	GPIO1	Console (configuration is fixed)	UART interfaces
UART_DEV(0):RxD	GPIO3	Console (configuration is fixed)	UART interfaces
UART_DEV(1):TxD	GPIO10	not available in qout and qio flash mode	UART interfaces
UART_DEV(1):RxD	GPIO9	not available in qout and qio flash mode	UART interfaces
OLED RESET	GPIO16		

Anexo E

Sensor DHT22



Technical Specification:

Model	DHT22
Power supply	3.3-6V DC
Output signal	digital signal via single-bus
Sensing element	Polymer capacitor
Operating range	humidity 0-100%RH; temperature -40~80Celsius
Accuracy	humidity +-2%RH(Max +-5%RH); temperature <+/-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH; temperature 0.1Celsius
Repeatability	humidity +-1%RH; temperature +-0.2Celsius
Humidity hysteresis	+/-0.3%RH
Long-term Stability	+/-0.5%RH/year
Sensing period	Average: 2s
Interchangeability	fully interchangeable
Dimensions	small size 14*18*5.5mm; big size 22*28*5mm

Anexo F

Sensor LDR

RS Data Sheet

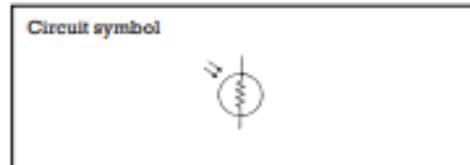
Light dependent resistors

NORP12 RS stock number 651-507
NSL19-M51 RS stock number 596-141

Two cadmium sulphide (cdS) photoconductive cells with spectral responses similar to that of the human eye. The cell resistance falls with increasing light intensity. Applications include smoke detection, automatic lighting control, batch counting and burglar alarm systems.

Guide to source illuminations

Light source	Illumination (Lux)
Moonlight	0.1
60W bulb at 1m	50
1W MES bulb at 0.1m	100
Fluorescent lighting	500
Bright sunlight	30,000



Light memory characteristics

Light dependent resistors have a particular property in that they remember the lighting conditions in which they have been stored. This memory effect can be minimised by storing the LDRs in light prior to use. Light storage reduces equilibrium time to reach steady resistance values.

NORP12 (RS stock no. 651-507)

Absolute maximum ratings

Voltage, ac or dc peak	320V
Current	75mA
Power dissipation at 30°C	250mW
Operating temperature range	-60°C to +75°C

Electrical characteristics

$T_A = 25^\circ\text{C}$. 2854°K tungsten light source

Parameter	Conditions	Min.	Typ.	Max.	Units
Cell resistance	1000 lux	-	400	-	Ω
	10 lux	-	9	-	k Ω
Dark resistance	-	1.0	-	-	M Ω
Dark capacitance	-	-	3.5	-	μF
Rise time 1	1000 lux	-	2.8	-	ms
	10 lux	-	18	-	ms
Fall time 2	1000 lux	-	48	-	ms
	10 lux	-	120	-	ms

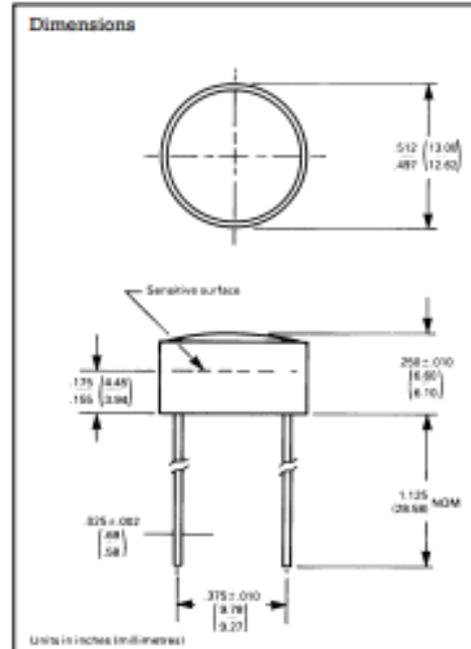
1. Dark to 110% R_L

2. To $10 \times R_L$

R_L = photocell resistance under given illumination.

Features

- Wide spectral response
- Low cost
- Wide ambient temperature range.



Anexo G

Sensor MG 811

MG811

CO₂ Sensor

Features

- Good sensitivity and selectivity to CO₂
- Low humidity and temperature dependency
- Long stability and reproducibility

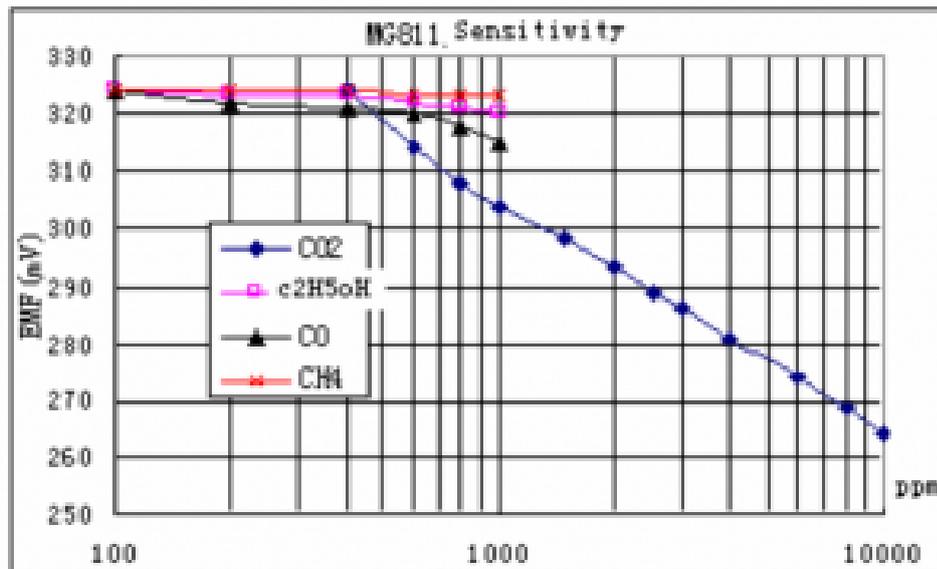
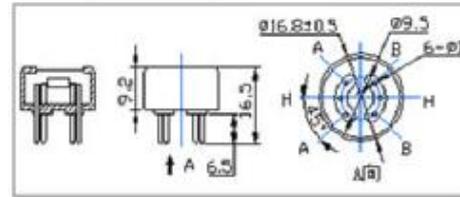
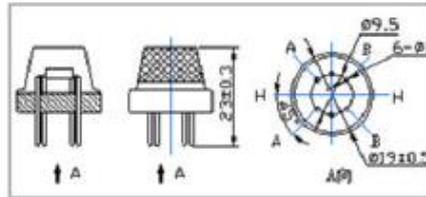
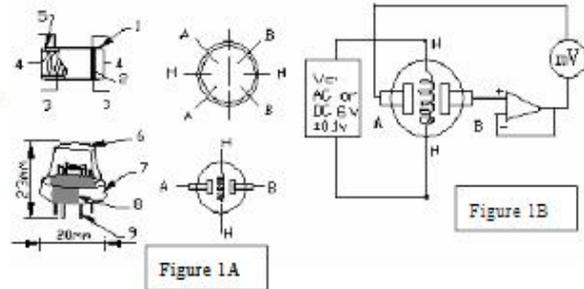
Application

- Air Quality Control
- Ferment Process Control
- Room Temperature CO₂ concentration Detection



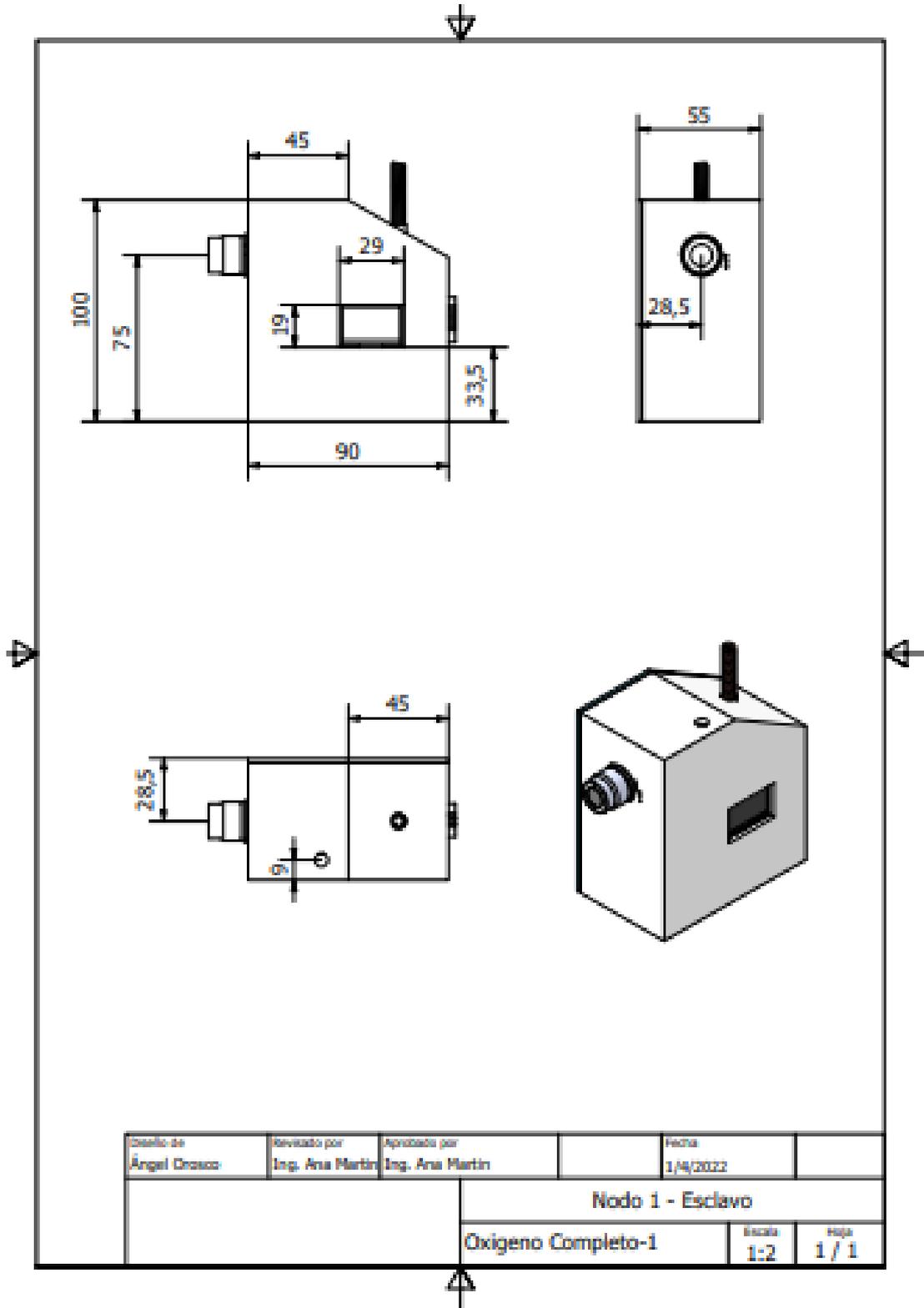
Structure and Testing Circuit

Sensor Structure and Testing Circuit as Figure. It composed by solid electrolyte layer (1) , Gold electrodes (2) , Platinum Lead (3) , Heater (4) , Porcelain Tube (5) , 100m double-layer stainless net (6) , Nickel and copper plated ring (7) , Bakelite (8) , Nickel and copper plated pin (9) ,



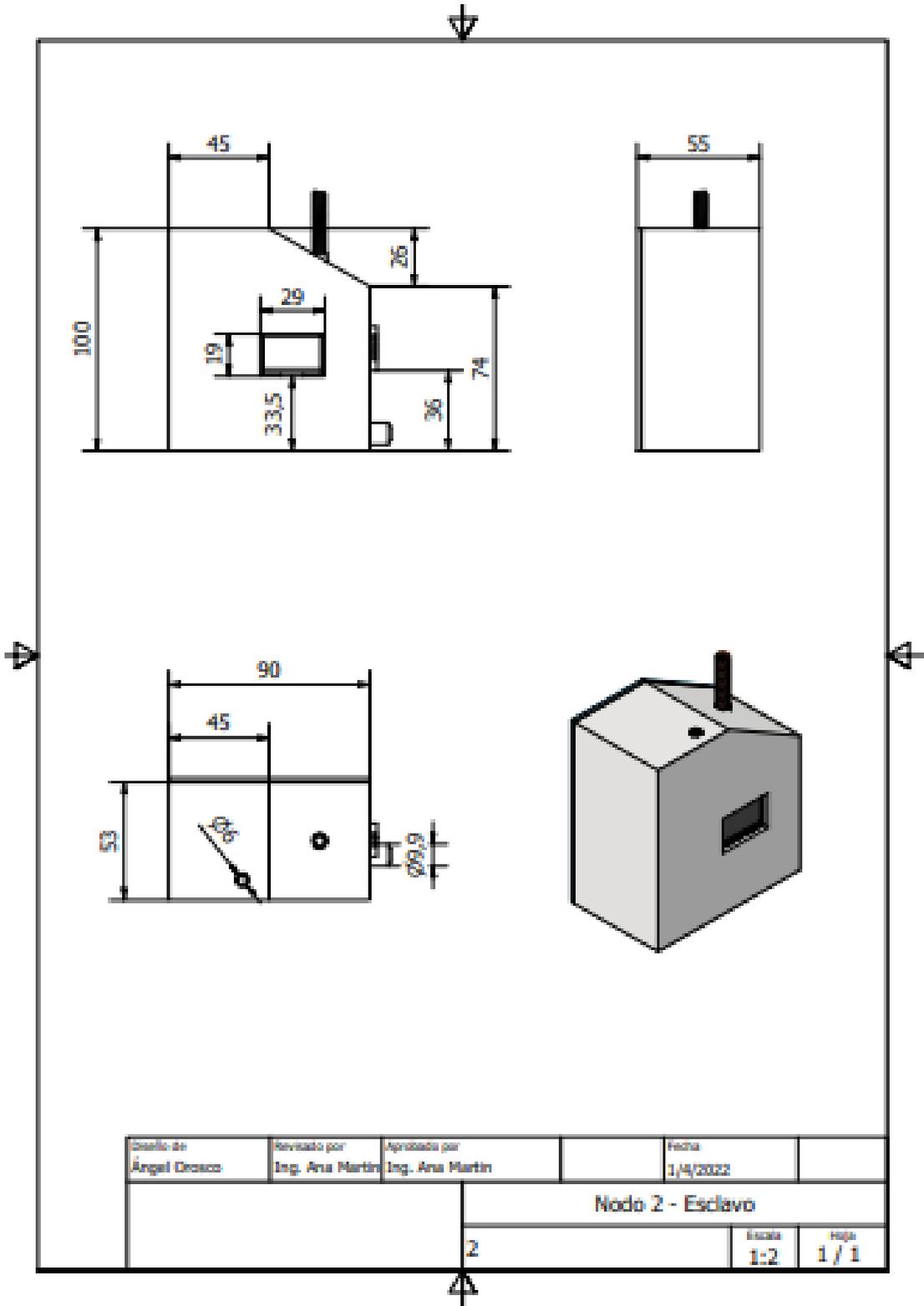
Anexo H

Planos del Nodo 1 – Esclavo



Anexo I

Planos del Nodo 2 – Esclavo



Anexo J

Planos del Nodo 3 – Maestro

