



UNIVERSIDAD TÉCNICA DE AMBATO

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

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y COMUNICACIONES

Tema:

**SISTEMA DE CONTROL Y MONITOREO INALÁMBRICO PARA LA
GRANJA DAMIANCITO DEL CANTÓN PELILEO**

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

AREA: Electrónica

LINEA DE INVESTIGACIÓN: Tecnología de la Información y Sistemas de Control

AUTOR: Kevin Sebastián Zúñiga Sánchez

TUTOR: Ing. Santiago Altamirano Meléndez, Mg

AMBATO – ECUADOR

septiembre – 2022

APROBACIÓN DEL TUTOR

En calidad de tutor de Trabajo de Titulación con el tema: SISTEMA DE CONTROL Y MONITOREO INALÁMBRICO PARA LA GRANJA DAMIANCITO DEL CANTÓN PELILEO, desarrollado bajo la modalidad Proyecto de Investigación por el señor Kevin Sebastián Zúñiga Sánchez , estudiante de la Carrera de Ingeniería en 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



Firmado electrónicamente por:
**SANTIAGO MAURICIO
ALTAMIRANO
MELENDEZ**

Ing. Santiago Altamirano Meléndez, Mg

TUTOR

AUTORÍA

El presente Proyecto de Investigación titulado: SISTEMA DE CONTROL Y MONITOREO INALÁMBRICO PARA LA GRANJA DAMIANCITO DEL CANTÓN PELILEO, es absolutamente original, autentico y personal. En tal virtud el contenido, efectos legales y académicos que se desprendan del mismo son de exclusiva responsabilidad del autor.

Ambato, septiembre 2022



Firmado electrónicamente por:
**KEVIN SEBASTIAN
ZUNIGA SANCHEZ**

Kevin Sebastián Zúñiga Sánchez

CC: 1804787214

AUTOR

APROBACIÓN TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Kevin Sebastián Zúñiga Sánchez, estudiante de la Carrera de Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA DE CONTROL Y MONITOREO INALÁMBRICO PARA LA GRANJA DAMIANCITO DEL CANTÓN PELILEO, 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 la señora Presidenta del Tribunal.

Ambato, septiembre 2022.



Firmado electrónicamente por:
**ELSA PILAR
URRUTIA**

Ing. Pilar Urrutia, Mg.
PRESIDENTA DEL TRIBUNAL



Firmado electrónicamente por:
**MARCO ANTONIO
JURADO LOZADA**

Ing. Marco Jurado, Mg
PROFESOR CALIFICADOR



Firmado electrónicamente por:
**FABIAN RODRIGO
SALAZAR ESCOBAR**

Ing. Fabian Salazar, Mg
PROFESOR CALIFICADOR

DERECHOS DE AUTOR

Autorizo 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, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, septiembre 2022



Firmado electrónicamente por:
**KEVIN SEBASTIAN
ZUNIGA SANCHEZ**

Kevin Sebastián Zúñiga Sánchez

CC: 1804787214

AUTOR

DEDICATORIA

Este proyecto de investigación lo dedico en primer lugar a Dios, por protegerme, iluminarme y brindarme el don de la vida, a mi padre Juan Nepalí que desde el cielo me brinda su apoyo y amor incondicional.

A mi madre Rosa Savina quien es el pilar fundamental de mi vida, y el motivo por el cual día a día me esfuerzo para salir adelante, a mis hermanos; Juan Diego por ser mi ejemplo de perseverancia, dedicación y constancia para cumplir mis metas, Mateo Leonel por alentarme y apoyarme emocionalmente en los momentos que la carrera se ponía cuesta arriba.

A mis amigos, compañeros que de una u otra manera me apoyaron a lo largo de este camino llamado universidad.

Kevin Sebastián Zúñiga Sánchez

AGRADECIMIENTO

Un agradecimiento a la Universidad Técnica de Ambato en especial a la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, por abrirme las puertas de sus aulas desde el primer día que inicie mi carrera. A los ingenieros quienes compartieron a lo largo de mis estudios sus conocimientos, experiencias y de su don de gente.

A mi familia que de una u otra manera me apoyaron con consejos y sus palabras de aliento, a mis primos que me guiaron a lo largo de esta carrera, siendo el ejemplo a seguir.

A todos mis compañeros, en especial a mi grupo de trabajo con quienes compartí buenas y malas experiencias, donde nunca faltó las palabras de aliento y el apoyo incondicional.

A la granja Avícola Damiancito por creer en mí y darme la oportunidad de desarrollar este proyecto bajo la tutoría del Ing. Santiago Altamirano.

Kevin Sebastián Zúñiga Sánchez

ÍNDICE GENERAL

APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
APROBACIÓN TRIBUNAL DE GRADO	iv
DERECHOS DE AUTOR.....	v
DEDICATORIA	vi
AGRADECIMIENTO.....	vii
ÍNDICE GENERAL.....	viii
ÍNDICE DE TABLAS	xii
ÍNDICE DE FIGURAS.....	xiv
CAPÍTULO I.....	1
MARCO TEÓRICO.....	1
1.1 Temas de investigación	1
1.2 Antecedentes investigativos	1
1.3 Contextualización del problema	4
1.4 Fundamentación teórica.....	5
1.4.1 Avicultura.....	5
1.4.1.1 Avicultura de traspatio	5
1.4.1.2 Avicultura deportiva o recreativa	6
1.4.1.3 Avicultura rural (Avicultura tradicional).....	6
1.4.1.4 Avicultura intensiva (Avicultura industrial).....	7
1.4.2 Crianza de las gallinas de postura	9
1.4.3 Galpones según la etapa de crianza.....	12
1.4.3.1 Galpón de cría y recría	12
1.4.3.2 Galpón de postura y engorde	13

1.4.4	Parámetros técnicos a controlar en un galpón de levante de gallinas de postura	14
1.4.4.1	Temperatura.....	15
1.4.4.2	Humedad relativa.....	16
1.4.4.3	Calidad del aire.....	16
1.4.4.4	Nivel de iluminación	17
1.4.5	Sistemas de control	18
1.4.5.1	Actuadores.....	19
1.4.5.2	Sensores.....	20
1.4.5.3	Equipos de control y maniobra.....	24
1.4.6	Placa de desarrollo	26
1.4.7	Interfaz Hombre Maquina (HMI).....	27
1.4.8	Comunicación inalámbrica.....	27
1.4.8.1	Redes LPWAN	28
1.4.8.2	LoRa	30
1.4.9	Plan de frecuencias.....	35
1.4.10	Protocolo de comunicación	36
1.4.10.1	CoAP.....	37
1.4.10.2	HTTP	37
1.4.10.3	MQTT	38
1.4.11	Base de datos.....	39
1.5	Objetivos.....	40
1.5.1	Objetivo general.....	40
1.5.2	Objetivos específicos	40
CAPITULO II		41
METODOLOGÍA		41
2.1	Materiales	41

2.2	Métodos	41
2.2.1	Modalidad de la investigación	41
2.2.2	Recolección de la información.....	42
2.2.3	Procesamiento y Análisis de Datos	42
2.2.4	Desarrollo del proyecto	42
CAPITULO III.....		44
RESULTADOS Y DISCUSIONES		44
3.1	Análisis y discusión de los resultados	44
3.1.1	Análisis de factibilidad.....	44
3.1.2	Desarrollo de la propuesta.....	44
3.1.2.1	Análisis del sistema existente en el galpón de levante de la granja Damiancito	44
3.1.2.2	Características técnicas de los diferentes equipos	46
3.1.2.3	Requerimientos del sistema.....	53
3.1.2.4	Requerimientos de las condiciones físicas del galpón de levante de gallinas de postura LOHMANN BROWN.	56
3.1.2.5	Etapas del sistema de control y monitoreo inalámbrico	58
3.1.2.6	Diagrama de bloques del sistema	60
3.1.2.7	Selección de los dispositivos para la implementación del prototipo	60
3.1.2.8	Diseño del sistema control y monitoreo inalámbrico	81
3.1.2.9	Implementación del sistema	117
3.1.3	Resultados y discusión	128
3.1.3.1	Verificación inicial del galpón	128
3.1.3.2	Comunicación nodo central nodo sensorial.....	128
3.1.3.3	Nodo central y control manual del sistema	129
3.1.3.4	Comunicación nodo receptor y emisor.....	130
3.1.3.5	Funcionamiento de la HMI.....	135

3.1.3.6 Almacenamiento de los datos	140
3.1.3.7 Comparación de la eficiencia del sistema	141
3.1.4 Presupuesto	150
CAPÍTULO IV	154
CONCLUSIONES Y RECOMENDACIONES.....	154
4.1 Conclusiones.....	154
4.2 Recomendaciones	155
REFERENCIAS BIBLIOGRAFÍAS	156
ANEXOS.....	162
Anexo 1: Código de programación de los nodos sensoriales.....	162
Anexo 2: Conexiones del nodo central o emisor en el software proteus	168
Anexo 3: Código de programación del nodo central o emisor.....	169
Anexo 4: Conexiones eléctricas del gabinete.....	200
Anexo 5: Código de programación del nodo receptor	201
Anexo 6: programación de la HMI en Node-RED.....	227
Anexo 7: Instalación del nodo central en el galpón	229
Anexo 8: Registro de la etapa de crianza del lote 20 datos proporcionados por la granja	230
Anexo 9: Registro de la etapa de crianza del lote 21 datos proporcionados por la granja	231
Anexo 10: Planos de las carcasas del nodo sensorial base y tapa, las unidades de medidas están en milímetros.....	232
Anexo 11: Instalación de los nodos sensoriales.....	234
Anexo 12: Pruebas de funcionamiento del nodo central.....	234

ÍNDICE DE TABLAS

Tabla 1 Temperatura deseable del medio ambiente dependiendo la edad de las aves	15
Tabla 2 Requerimientos mínimos de la calidad del aire.....	16
Tabla 3 Programa de iluminación para galpones cerrados para pollas ponedoras LOHMANN BROWN.....	17
Tabla 4 Características de los termopares según el tipo.....	21
Tabla 5 Código de colores de las luces de señalización.....	25
Tabla 6 Comparación entre tecnologías LPWAN.....	29
Tabla 7 Factores de ensanchamiento LoRa	33
Tabla 8 Niveles de radiación para bandas ICM según UIT-R SM. 1015-1	35
Tabla 9 Parámetros según la ubicación de la región.	36
Tabla 10 Descripción de los equipos existentes en el galpón de levante	45
Tabla 11 Características del motor WEG W22	47
Tabla 12 Características técnicas del calefactor ECOFLAM MAX 12 TC O 50-60Hz	49
Tabla 13 Características técnicas del sistema de ventoleras Megamat 2 TF 6KN - D53582	50
Tabla 14 características técnicas del Termostato TS9501.....	51
Tabla 15 Características del Dimmer AviLamp220	52
Tabla 16 Características del termómetro digital HAKUSA	52
Tabla 17 Características de la lámpara Led Avilamp WY-03.....	53
Tabla 18 Elementos de hardware que se utilizó en la elaboración del prototipo .	54
Tabla 19 Elementos de software que se utilizó en la elaboración del prototipo ..	55
Tabla 20 Temperatura deseable del medio ambiente dependiendo la edad de las aves	56
Tabla 21 Requerimientos mínimos de la calidad del aire.....	57
Tabla 22 Programa de iluminación para galpones cerrados para pollas ponedoras LOHMANN BROWN	58
Tabla 23 Etapas del sistemas	59
Tabla 24 Comparativa de las diferentes placas de desarrollo del nodo.....	61
Tabla 25 Distribución de pines de la tarjeta Lilygo LoRA32 V1.3	64
Tabla 26 Comparativa de los diferentes sensores de temperatura.....	66

Tabla 27 Comparativa de los diferentes sensores de calidad del aire	68
Tabla 28 Comparativa de los diferentes sensores de Humedad Relativa	70
Tabla 29 Características de comunicación inalámbrica de las placas con tecnología LoRa	72
Tabla 30 Comparativa de fuentes de alimentación conmutadas	78
Tabla 31 Comparativa de los proveedores de servicios en la Nube	79
Tabla 32 Componentes electrónicos de los nodos sensoriales	87
Tabla 33 Componentes eléctricos y electrónicos del circuito de mando y control del sistema.	95
Tabla 34 Ubicación de las estaciones emisora y receptora	98
Tabla 35 Tiempo de envío y recepción de los datos entre el nodo emisor y el receptor.....	132
Tabla 36 Tiempo de envío y recepción de los datos entre el nodo emisor y el receptor pruebas iniciales	134
Tabla 37 Tiempo de respuesta en la activación y desactivación de los actuadores desde la HMI	138
Tabla 38 Consumo de alimento sin el sistema de control y monitoreo.....	141
Tabla 39 Inventario de aves existentes en el galpón sin el sistema control y monitoreo	142
Tabla 40 Uniformidad de las aves sin el sistema de control y monitoreo.....	143
Tabla 41 Consumo de alimento con el sistema de control y monitoreo.....	146
Tabla 42 Inventario de aves existentes en el galpón con el sistema control y monitoreo	147
Tabla 43 Uniformidad de las aves con el sistema de control y monitoreo.....	147
Tabla 44 Presupuesto del diseño e implementación del sistema de control y monitoreo	150

ÍNDICE DE FIGURAS

Figura 1 Galpón semi-tecnificado.....	8
Figura 2 Galpón de postura para gallinas ponedoras	9
Figura 3 Pollitas entre los días 1 y 5 de vida	10
Figura 4 Pollitas entre las semanas 8 y 12	10
Figura 5 Gallina cursando la semana 15 de vida	11
Figura 6 Gallina en la semana 18 de vida lista para iniciar la etapa de producción.....	11
Figura 7 Gallinas en la semana 45 de producción.	12
Figura 8 Galpón de levante al suelo con control de parámetros técnicos de manera automática	13
Figura 9 Galpón de levante en jaulas con control de parámetros técnicos de manera automática	13
Figura 10 Galpón inteligente CAVENCO para gallinas de postura al piso..	14
Figura 11 Galpón de producción con jaulas [9].....	14
Figura 12 Esquema general de un sistema	18
Figura 13 Motor eléctrico	19
Figura 14 Calefactor eléctrico MET MANN	20
Figura 15 Funcionamiento del sensor por deformación.....	22
Figura 16 Sensor de humedad de tipo bulbo húmedo y seco.....	23
Figura 17 Diagrama esquemático de un sensor NH3-B1	24
Figura 18 Sensor de iluminación de tipo mini-fotocélula.....	24
Figura 19 Arquitectura de un Microcontrolador	26
Figura 20 Arquitectura LPWAN	29
Figura 21 LoRa en las capas del modelo OSI.....	31
Figura 22 Cambios en la señal del transmisor de fase portadora del sistema DSSS	32
Figura 23 Espectro ensanchado de la modulación LoRa con chirp [29]	32
Figura 24 Estructura del protocolo LoRaWAN [29]	33
Figura 25 Elementos de una red LoRaWAN	34
Figura 26 Estructura del protocolo HTTP.....	38
Figura 27 Estructura del protocolo MQTT	38
Figura 28 Extractor de aire.....	47

Figura 29	Diagrama de bloques del sistema de control y monitoreo inalámbrico.....	60
Figura 30	Distribución de pines de la tarjeta Lilygo LoRa32 V1.3 (TTGO)	63
Figura 31	Sensor de temperatura M12FTH4Q [46].....	67
Figura 32	<i>Conexión del sensor MQ137</i>	69
Figura 33	Conexión del sensor DHT22	72
Figura 34	Diagrama de bloques del nodo sensor	82
Figura 35	Diagrama de flujo del nodo sensorial	82
Figura 36	Diagrama de conexiones de los sensores, y placa de ESP8266-1	82
Figura 37	Capas del Modelo OSI.....	83
Figura 38	Formato de cabecera del protocolo UDP.....	85
Figura 39	Envío de paquetes en el protocolo UDP	85
Figura 40	Diagrama de flujo del código de programación del nodo Sensor.	86
Figura 41	Diagrama de bloques de los circuitos de mando y control	87
Figura 42	Entradas de datos	89
Figura 43	Salidas analógicas para conectar al circuito eléctrico	89
Figura 44	Tarjeta de desarrollo ESP32 LoRa Lilygo TTGO	90
Figura 45	Pantalla LCD	90
Figura 46	Teclado matricial 4x4	90
Figura 47	Regulador de voltaje	91
Figura 48	Diagrama de flujo de programación del nodo central	92
Figura 49	Circuito eléctrico de control	93
Figura 50	Circuito de potencia de los extractores y los calefactores	95
Figura 51	Ubicación de la granja y de la oficina	97
Figura 52	Ubicación de estaciones emisora y receptora en Radio Mobile .	102
Figura 53	Ubicación de estaciones emisora y receptora en Google Earth..	102
Figura 54	Configuración del sistema en el software Radio Mobile.....	103
Figura 55	Configuración de la frecuencia a trabajar.....	104
Figura 56	Perfil del enlace de comunicación inalámbrica entre la base emisora y receptora.....	105
Figura 57	Diagrama de flujos del nodo emisor con configuración Lora	106
Figura 58	Diagrama de flujos del nodo receptor con configuración Lora ..	107
Figura 59	Selección de la opción EC2	108

Figura 60	Selección del sistema operativo UBUNTU	108
Figura 61	Instalación dl protocolo MQTT	109
Figura 62	Verificación del estado del protocolo MQTT.....	110
Figura 63	Configuración del puerto externo en el protocolo mqtt.....	110
Figura 64	Instalación del Bróker EMQX.....	111
Figura 65	Inicio del bróker EMQX.....	111
Figura 66	Configuración de la dirección IP 127.0.0.1	111
Figura 67	Declaración de los tópicos que utilizaran el protocolo MQTT ..	112
Figura 68	Instalaciones de requisitos para Node Red	113
Figura 69	Instalación de NODE RED.....	113
Figura 70	Verificación que Node-RED este inicializado.....	114
Figura 71	Configuración de la adquisición de datos de los nodos sensores y publicación en la HMI.....	115
Figura 72	Configuración de las señales de activación desde la HMI a los actuadores.....	115
Figura 73	Programación del botón de activación o activación del modo automático desde la HMI	116
Figura 74	Programación de las entradas de los datos de temperatura máxima y mínima desde la HMI.....	116
Figura 75	Programación del Dimmer desde la HMI.....	117
Figura 76	Ubicación de las ventoleras y los extractores	118
Figura 77	Ubicación de los calefactores en el galpón.....	118
Figura 78	Ubicación de los nodos sensoriales	119
Figura 79	Ubicación del tablero de control.....	119
Figura 80	Ubicación de la antena Emisora	120
Figura 81	Diseño del Ruteo de pistas del nodo sensor	120
Figura 82	Ruteado transferido a la baquelita nodo sensor	121
Figura 83	Corrosión de la baquelita es decir eliminación del exceso de cobre del nodo sensor.....	121
Figura 84	Perforación de los agujeros para cada uno de los pines a conectar del nodo sensor.....	121
Figura 85	Soldadura de componentes y pruebas de funcionamiento del nodo sensor.....	121

Figura 86 Implantación del nodo sensor 1	122
Figura 87 Implementación del nodo sensor 2	122
Figura 88 Base de la carcasa del nodo sensorial.....	122
Figura 89 Tapa superior de la carcasa del nodo sensorial.....	122
Figura 90 Diseño del Ruteo de pistas del circuito de mando y control	123
Figura 91 Pistas transferidas a baquelita del circuito de mando y control..	124
Figura 92 Corrosión de la baquelita es decir eliminación del exceso de cobre del circuito de mando y control.....	124
Figura 93 Perforación de los agujeros para cada uno de los pines a conectar del circuito de mando y control.....	124
Figura 94 Soldadura de componentes y pruebas de funcionamiento del circuito de mando y control.....	124
Figura 95 Instalación de componentes eléctricos en el gabinete	125
Figura 96 Exterior del gabinete de mando y control manual.....	126
Figura 97 Representacion de las instalaciones de los diferentes componentes del sistema.....	127
Figura 98 Envío de datos desde los nodos sensoriales al nodo central.....	129
Figura 99 Funcionamiento del sistema en modo manual.....	130
Figura 100 Instalación del nodo receptor en el elevador de 25m de altura.	130
Figura 101 Recepción y envío de datos en el nodo emisor en el elevador .	131
Figura 102 Etapa de enfriamiento.....	135
Figura 103 Etapa de calentamiento.....	135
Figura 104 Etapa de habilitación del acceso a al HMI	136
Figura 105 etapa de ingreso de los rangos de temperaturas.....	136
Figura 106 Datos de los sensores de temperatura	136
Figura 107 Datos de los sensores de humedad	137
Figura 108 Etapa de variación de la intensidad luminosa.....	137
Figura 109 Datos del sensor de amoniaco	138
Figura 110 Visualización de datos en el tablero del galpón	140
Figura 111 Visualización de datos la HMI	140
Figura 112 Datos promedios de temperatura almacenados en la base de datos INFLUXDB.....	140

Figura 113 Datos promedios de humedad almacenados en la base de datos INFLUXDB.....	140
Figura 114 Representación de los datos de peso y consumo de alimento de las aves en la etapa de crecimiento	144
Figura 115 Porcentaje de mortandad de las aves sin el sistema de control y monitoreo	145
Figura 116 Representación de los datos de peso y consumo de alimento de las aves en la etapa de crecimiento	149
Figura 117 Porcentaje de mortandad de las aves sin el sistema de control y monitoreo	149

RESUMEN EJECUTIVO

Los avances tecnológicos en el ámbito de la avicultura permiten obtener mejores resultados en la producción, por lo que la granja Damiancito requiere implementar un proceso automático en la etapa de crianza de gallinas de postura, por ende, este proyecto de investigación tiene como finalidad implementar un sistema de control y monitoreo inalámbrico, para la supervisión de los parámetros técnicos del galpón de crianza de gallinas de postura, utilizando tecnología LoRa y MQTT.

En el presente proyecto se desarrolla el diseño y la implementación del sistema de control y monitoreo inalámbrico, para la supervisión de los parámetros técnicos del galpón de crianza de gallinas de postura en la granja Damiancito. Para el cumplimiento de este proyecto se procede al diseño eléctrico y electrónico, para lo cual se requiere dispositivos que se ajusten a las necesidades del sistema; por consiguiente, los elementos electrónicos encargados de generar instrucciones y adquirir datos, y los elementos eléctricos encargados de la activación o desactivación de los actuadores.

Para el funcionamiento del sistema control y monitoreo inalámbrico del galpón se ha implementado tres nodos los que son: nodo sensorial encargado de adquirir los datos de temperatura, humedad y amoníaco, el nodo central encargado de generar las instrucciones, procesamiento de datos y la comunicación LoRa, el nodo receptor encargado de conectarse al servidor y permitir la visualización de los datos y el control de los parámetros en la HMI. Como resultado se tendrá un sistema confiable y eficiente mejorando los índices de uniformidad de las aves, la mortandad y el índice de aves descriadas al final de la etapa de crianza.

Palabras clave: LoRa, comunicación inalámbrica, espectro de frecuencias, nodos, protocolos, paquete de datos.

ABSTRACT

Technological advances in the field of poultry farming allow better results in production, so the Damiancito farm needs to implement an automatic process at the stage of raising laying hens, therefore, this research project intention to implement a wireless control and monitoring system for the supervision of the technical parameters of the laying hens rearing house, using LoRa and MQTT technology.

This project develops the design and implementation of a wireless control and monitoring system for the supervision of the technical parameters of the laying hens rearing house at Damiancito farm. For the fulfillment of this project, we start with the electronic and electrical design., for which devices that fit the needs of the shed are required; consequently, the electronic elements in charge of generating instructions and acquiring data, and the electrical elements in charge of the activation or deactivation of the actuators.

For the operation of the wireless control and monitoring system of the poultry shed, three nodes have been implemented: the sensor node in charge of acquiring the temperature, humidity and ammonia data, the central node in charge of generating the instructions, data processing and LoRa communication, the receiver node in charge of connecting to the server and allowing the visualization of the data and the control of the parameters in the HMI. The result will be a reliable and efficient system that will improve the uniformity indexes of the hens, the mortality and the index of hens descried at the end of the rearing stage.

Keywords: LoRa, wireless communication, frequency spectrum, nodes, protocols, data packe

CAPÍTULO I

MARCO TEÓRICO

1.1 Temas de investigación

Sistema de control y monitoreo inalámbrico para la granja Damiancito del cantón Pelileo

1.2 Antecedentes investigativos

En base a las investigaciones realizadas en la biblioteca virtual de la Universidad técnica de Ambato en especial de la Facultad de Ingeniería en Sistemas Electrónica e Industrial y posterior a ello en las bases de datos de las Universidades y Escuelas Politécnicas del Ecuador, además en Internet, se logró recolectar la información relacionada al tema de investigación de las cuales se encontró los siguientes documentos:

En el año 2014, en Barranquilla, Colombia, J. Herrera, M. Barrios, S. Pérez, en su investigación “Design and implementation of a wireless scada system by means of zigbee and arduino technology”, desarrollaron un sistema SCADA inalámbrico capaz de supervisar y controlar procesos ubicados a largas distancias. En el sistema utilizaron una placa arduino para la adquisición de datos, la adquisición de la temperatura lo realizan con el sensor LM35, para la comunicación inalámbrica han utilizado el módulo Xbee y el procesamiento de datos en LabView. El resultado obtenido de la investigación fue un dispositivo versátil capaz de funcionar con diferentes sensores, para adquisición de datos de forma remota y almacenarlos en un base de datos y generar tablas e histogramas. [1]

En el año 2018, en Santander, Colombia, O Peña y G Osorio en su trabajo “Impact of new technologies of industry 4.0 in Colombia” de tipo experimental desarrollaron un dispositivo electrónico basado en la tecnología IOT, este tiene la capacidad de adquirir datos de tipo analógico como la temperatura, los datos obtenidos por los sensores se almacenan en la nube permitiendo su visualización en una App. Para el diseño del dispositivo se utilizó un

dispositivo lógico-programable como es la ESP-32, un módulo de conexión a la nube por medio de GPRS con una antena de exterior, para el desarrollo de la App se utilizó APPINVENTOR y a la base de datos MySQL. Al finalizar el diseño e implementarlo las empresas beneficiarias como Axum Gropu, Tecnoparque Nodo Socorro concluyeron que el equipo realiza las tareas de muestreo de temperatura de forma precisa y una duración de baterías de alrededor de 4 días. Además, menciona que se consiguió un dispositivo robusto, con una relación preciso beneficio satisfactorio. [2]

A. Osorio, C. González, en Cali, Colombia en el año 2018 en su trabajo “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”, han presentado un prototipo capaz de adquirir, controlar y visualizar los parámetros de temperatura, humedad relativa y presión en los galpones de aves de engorde. Este prototipo cuenta con comunicación inalámbrica utilizando el protocolo Zigbee él cuenta con un módulo Xbee compatible con arduino. Para la adquisición de datos los sensores de utilizados son BMP180 que es capaz de medir la presión y la temperatura de los galpones y el sensor DTH22 para medir la humedad. Para la visualización de los datos en la web se ha utilizado ASP.NET. Una vez finalizado el prototipo se ha concluido que la comunicación inalámbrica que brinda mayor confianza en este tipo de aplicaciones es la Zigbee, por lo que el prototipo cumple con la adquisición de los datos de temperatura, humedad y presión en los galpones. [3]

En Chimborazo, en la Escuela Superior Politécnica de Chimborazo en el año 2019 C. Santiana, D. Guamán y J. Flores en su trabajo “Application of a remote control protocol for a farm house and a greenhouse according to zonal environmental parameters” de tipo experimental han desarrollado un sistema de control y monitoreo inalámbrico de temperatura, humedad, luminosidad y CO2 en invernaderos y galpones de crianza de pollos. Para el desarrollo del prototipo han utilizado una NI DAQ y en el procesamiento de datos LabView. En lo que respecta a los sensores se ha utilizado, LM35, FC-01 para humedad en los invernaderos y un HS-02 en los galpones, y el sensor de luminosidad.

En la parte de comunicación inalámbrica se utilizó un enlace punto a punto con antenas de comunicación bidireccional que funcionan a una frecuencia de 5.8Ghz y una potencia de 15dBi. El desarrollo del prototipo resulto amigable en lo que se refiere a la adquisición y procesamiento de datos, la forma de graficar todos los parámetros de cada proceso, la interface gráfica utiliza elementos de control e ingreso de datos de configuración que nos son complejos en lo que corresponde a la operación de cada sistema. [4]

En el año 2019, Miranda Jennifer, de la Universidad Técnica de Ambato, Ecuador, en su trabajo “Sistema electrónico de control y monitoreo remoto para el orquideario de la junta parroquial río verde en la provincia de Tungurahua” de tipo experimental ha desarrollado un sistema electrónico utilizando un microcontrolador arduino, con la finalidad de monitorear los parámetros técnicos del suelo y del ambiente en el invernadero, además utiliza una base de datos como MySQL para el almacenamiento de la información. Los resultados obtenidos han sido los deseados puesto que con el sistema ha logrado el control de los parámetros como: temperatura y humedad ambiental del invernadero, lo que facilita las labores de monitoreo realizadas por el personal encargado del orquideario. [5]

M. Ortiz, en Quito, en la Universidad Politécnica Salesiana en el año 2020 en su trabajo “Desarrollo de una red de sensores inalámbricos utilizando tecnología LORA para el monitoreo de un sistema” de tipo experimental ha desarrollado una red de sensores inalámbrico utilizando la con tecnología LORA. El prototipo está compuesto por un nodo cliente y un nodo servidor. Para el nodo cliente se ha utiliza una Raspberry Pi 3 Modelo B+ y un módulo LoRa/GPS HAT que admite comunicación inalámbrica por LoRa y opera en las bandas 433 MHz, 868 MHz y 915 MHz. Para la visualización y control de los datos se ha utilizado Ubidots. Los resultados obtenidos fueron un prototipo capaz de comunicarse con sus tres estaciones cliente al servidor, enviando la información en tiempos de 6.2 y 6,3 segundos y con un porcentaje de 98.75 de recepción de la información entre el servidor y el cliente. Además, la visualización en Ubidots fue la correcta observando el análisis de los datos adquiridos. [6]

La revisión bibliográfica permite conocer proyectos relacionados al tema de investigación, en los que se describe las tecnologías utilizadas y los parámetros técnicos controlados por los investigadores, además, se concluye que la ubicación de los nodos sensoriales debe ser en los lugares que se presenten los niveles más altos de temperatura, humedad y amoníaco, y para la comunicación LoRa colocar el nodo emisor y receptor con una línea de vista directa.

1.3 Contextualización del problema

La industria avícola en los países desarrollados utiliza nuevas tecnologías como: la industria 4.0, las comunicaciones inalámbricas basadas en el Servicio General de Paquetes vía Radio (GPRS), el Sistema Global de Comunicaciones Móviles (GSM) y la conexión a la nube mediante el internet, esto permite la integración y el monitoreo de los procesos y el incremento de la productividad y la calidad de los productos. Según el INEC, en el Ecuador el 5.4% de las aves son criadas en el campo y el 94.6% son de granjas avícolas, la región Sierra tiene un 85.5%, mientras que la Costa cuenta con un 11.5% y la Amazonia está representada por el 2.9%. La provincia de Tungurahua cuenta con un total de 149 granjas de aves de postura, las cuales el 30% son semi-tecnificadas, un 8.5% son granjas cien por ciento tecnificadas y un 61.5% son granjas artesanales o familiares. La granja Damiancito se encuentra en el grupo de las 149 granjas de aves de postura y se ubica entre las granjas semi-tecnificadas, por lo que desea la implementación de un sistema de control y monitoreo en el área de crianza de aves de postura, debido a que, en la etapa de crecimiento que contemplan las 12 semanas de vida se obtiene un 73.7% de aves idóneas, un 20.83% de descriadas y un 5.47% de mortandad.

El principal inconveniente en la crianza de las gallinas es la alta presencia de recursos humanos, debido a que, el personal debe permanecer en constante monitoreo y control de los diferentes parámetros como: temperatura, humedad, gases tóxicos, luminosidad y velocidad del viento en los galpones de levante, donde las gallinas permanecen desde el día 1 hasta el día 48 cumpliendo su etapa de crecimiento, esto que obliga al avicultor a invertir gran cantidad de recursos humanos en el área de crianza. Otro de los inconvenientes es la ubicación geográfica, los galpones de crianza están ubicados en lugares

alejados a los galpones postura con la finalidad de evitar la transmisión de enfermedades de las gallinas de mayor edad. Con el objetivo de evitar el uso excesivo de recursos humanos en la etapa de crianza y mejorar la comunicación con la etapa de manutención de las gallinas, se plantea un sistema de control y monitoreo inalámbrico de los diferentes sensores y actuadores en el galpón de crianza de la granja Damiancito.

El diseño del sistema de control y monitoreo inalámbrico de los diferentes parámetros técnicos es factible realizar, ya que se cuenta con los dispositivos tecnológicos en el mercado nacional, en la etapa de diseño existen los programas computacionales las referencias bibliográficas y la tecnología requerida para la implementación del sistema de control y monitoreo. Esto permitirá desarrollar un producto que cumpla con su correcto funcionamiento y apegado a las características técnicas de diseño e implementación mejorando la etapa de crecimiento de las gallinas en el galpón de crianza de la Granja Damiancito.

1.4 Fundamentación teórica

1.4.1 Avicultura

La avicultura es una rama de la zootecnia, que se encarga de la crianza, mejora y explotación comercial de las aves domésticas, según el sector en el que se desarrolle pueden ser diferentes tipos de aves como pavos, gallinas, pollos, patos o codornices. La avicultura en el Ecuador está relacionada con la comercialización de carne de pollo y huevos, por lo que se relaciona el termino avicultura con la crianza y explotación de los pollos de engorde (pollos de carne) y las gallinas de postura (gallinas para la producción de huevos). [1]

Actualmente la producción avícola tiene diferentes variantes o modalidades de explotación, algunas de las que se describen a continuación:

1.4.1.1 Avicultura de traspatio

La avicultura de traspatio es la producción de gallinas de postura o pollos de engorde en pequeñas cantidades, con la finalidad de obtener productos como huevos o carne para el consumo familiar, además los espacios de crianza son

reducidos y casi siempre carecen de tecnologías y recursos de mejora porque esta actividad se da en las zonas rurales o suburbanas. [1]

1.4.1.2 Avicultura deportiva o recreativa

La avicultura deportiva o recreativa es la crianza de aves con finalidades de competencia y ornamentales. La crianza de palomas mensajeras para competencias de resistencia, gallos de combate y aves exóticas para la decoración de parques o zoológicos, este tipo de avicultura es más desarrollada que la de traspatio, porque se utiliza conocimientos de zootecnia y nutrición para un mejor rendimiento de las aves. [1]

1.4.1.3 Avicultura rural (Avicultura tradicional)

La avicultura rural o también conocida como avicultura tradicional es la que se dedica a la crianza de gallinas de postura o pollos de carne en medianas o pequeñas cantidades en espacios extensos, con la finalidad de que las gallinas busquen su propio alimento tipo pastoreo, y realizar un pequeño aporte de alimento procesado. Si bien este tipo de avicultura tiene una finalidad comercial la producción de huevos y carne es mínima, ya que solamente puede abastecer al mercado local. [2]

Ventajas de la avicultura tradicional

Las ventajas de la avicultura tradicional son:

- Inversión en la crianza y manutención de las gallinas o pollos es baja.
- Aprovechamiento de los pastos y granos producidos por los granjeros.
- El uso de recursos usados en mínimo.
- Aportan de abono o fertilización a los espacios de pastoreo.
- Aporta mayor resistencia y rusticidad a las aves.

Desventajas de la avicultura tradicional

Las desventajas de la avicultura tradicional son:

- Disponibilidad de grandes extensiones de terreno.

- La producción de huevo o carne es menor poca rentabilidad.
- Las aves son más propensas a contraer enfermedades

1.4.1.4 Avicultura intensiva (Avicultura industrial)

La avicultura intensiva o también conocida como avicultura industrial es la que en las etapas de crianza y producción se utiliza al máximo el espacio disponible mientras mayor sea el área de productividad mayor será la rentabilidad, por ende, este método de crianza se puede realizar de dos formas al piso o en jaula. La avicultura intensiva utiliza diferentes dispositivos eléctricos y electrónicos para el control de los parámetros técnicos como: humedad, calidad del aire, temperatura, luminosidad y alimentación en cada uno de sus galpones, dependiendo de la etapa de producción de. [1, 2]

Ventajas de la avicultura intensiva

Las ventajas de la avicultura tradicional son:

- Mayor producción de huevos o carne.
- Aprovechamiento del alimento del 95%
- Mayor número de aves por metro cuadrado.
- Control absoluto del proceso de producción.
- Mayor control de enfermedades.
- Facilidad y eficiencia en el manejo de cada una de las etapas de producción.

Desventajas de la avicultura intensiva

Las desventajas de la avicultura tradicional son:

- Mayor capital de inversión para la construcción de infraestructura.
- Se requiere de capacitación constante al personal.
- Alta demanda de alimentación para las aves.
- Se requiere la división de la producción por etapas: etapa de crianza, etapa de producción y la etapa de engorde cada una con su respectiva infraestructura y su personal.

En la avicultura intensiva o industrial el uso de tecnologías en las diferentes etapas como alimentación, manejo de ambientes y el área de crianza ha permitido el aumento de la producción, por lo que las granjas industrializadas se dividen en las siguientes categorías:

- **Granjas semi-tecnificadas**

Las granjas semi-tecnificadas son aquellas en la que los procesos de producción poseen un grado mínimo de tecnificación o automatización industrial, porque existe el uso de recursos humanos y recursos tecnológicos al mismo tiempo, estas representan el control de parámetros como la temperatura y la humedad de manera electrónica, mientras que la alimentación y la recolección de huevos son de manera manual, por lo tanto los productos son muy similares a la de una granja tecnificada, pero con un costo de producción mayor. La figura 1 representa un galpón semi-tecnificado donde se controla la temperatura, luminosidad y humedad mediante un panel de control y la alimentación se realiza de manera manual. [1]



***Figura 1** Galpón semi-tecnificado [1]*

- **Granjas tecnificadas**

Las granjas tecnificadas son las que utilizan los avances tecnológicos disponibles a nivel mundial, para mejorar los procesos de crianza y manutención de las aves, en este tipo de granjas la integración vertical permite tener un mayor crecimiento y control de los procesos permitiendo la obtención del aumento en la producción huevos en el caso de las gallinas de postura y de carne en los pollos de engorde. En la figura 2, se indica la estructura y los

elementos tecnológicos utilizados en los galpones pertenecientes a las granjas tecnificadas. [3]

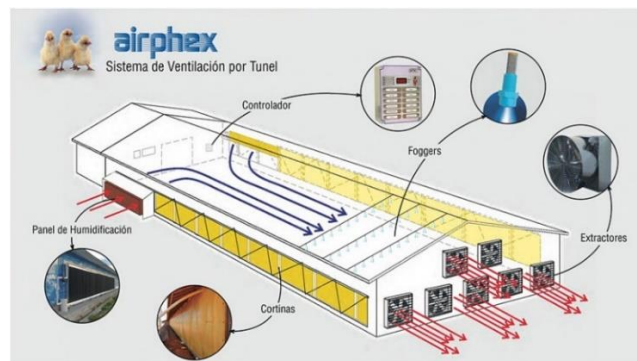


Figura 2 Galpón de postura para gallinas ponedoras [4]

1.4.2 Crianza de las gallinas de postura

- **Etapas de crianza de las gallinas de postura**

Las etapas de crianza de las gallinas de postura inician desde el día 1 hasta el día 136, en las que las gallinas cumplen con la etapa de crianza y pasan a formar parte de la etapa de producción. Las etapas que cumplen las gallinas ponedoras son:

- Iniciación
- Levante
- Prepostura
- Producción

- **Etapas de iniciación**

La etapa de iniciación es la que está comprendida entre el periodo del día 1 hasta la doceava semana de vida, la cual se divide en dos etapas: la etapa de cría (desde el día 1 hasta la cuarta semana) y recria (desde la cuarta semana hasta la doceava), en estas semanas las pollitas desarrollan la totalidad de sus órganos, y por ende el consumo de alimento debe ser alto en proteína. En la figura 3 se muestra pollitas que se encuentran en los primeros 5 días de vida. [5, 6]



Figura 3 Pollitas entre los días 1 y 5 de vida

Elaborado por: El investigador

La figura 4 indica las pollitas entre las octava y doceava semana de vida, correspondiente a la etapa de iniciación.



Figura 4 Pollitas entre las semanas 8 y 12

Elaborado por: El investigador

- **Etapas de levante**

La etapa de levante está comprendida en el periodo de las semanas 12 y 16 de vida, aquí se concentra el desarrollo esquelético y muscular de las gallinas, por ende, su alimentación esta alto en proteína. En la figura 5 se muestra una gallina en la etapa de levante. [5]



Figura 5 Gallina cursando la semana 15 de vida

Elaborado por: El investigador

- **Etapas de prepostura**

La etapa de prepostura está comprendida en el periodo de las semanas 17 y 18 donde las gallinas empiezan a poner huevos, desarrollando los órganos reproductores como son los ovarios, en esta etapa las gallinas deberán alimentarse con balanceados que aporten calcio y grasa corporal. En la figura 6, se muestra una gallina en la etapa de prepostura. [6]



Figura 6 Gallina en la semana 18 de vida lista para iniciar la etapa de producción.

Elaborado por: El investigador

- **Etapas de producción**

La etapa de producción está comprendida en el periodo de las semanas 18 y 90 donde las gallinas llegan a su punto más alto de producción de huevos. Esta etapa se divide en la fase 1 la cual comprende desde el día que sueltan su primer huevo y alcanza el 85% de producción y la fase 2 la cual comprende desde el día que alcanza el 85% hasta el final del ciclo de producción es decir las 90 semanas. En la figura 7 se muestra una gallinas en la etapa de producción. [5, 6]



Figura 7 Gallinas en la semana 45 de producción.

Elaborado por: El investigador

1.4.3 Galpones según la etapa de crianza

Cada una de las etapas por las que pasan las gallinas de postura requieren de infraestructura adecuada según su edad. Para lo cual los avicultores han desarrollado dos tipos de galpones los que son:

- ✓ Galpón de cría y recría.
- ✓ Galpón de postura y engorde.

1.4.3.1 Galpón de cría y recría

Los galpones de cría y recría o también conocidos como galpones de levante son aquellos en los que permanecen las pollitas desde el día 1 hasta la semana 16 de vida. En los galpones se controla los parámetros técnicos que las empresas incubadoras recomiendan como son: temperatura, humedad, velocidad del viento y calidad del aire es decir gases tóxicos como: Oxígeno, dióxido de carbono, monóxido de carbono, amoníaco y el ácido sulfhídrico, estos parámetros son los que cada galpón debe cumplir basándose en las especificaciones técnicas proporcionas por el manual LOMAN BROWN. Los galpones de levante pueden ser de dos tipos: galpones para la crianza en el suelo o galpones con jaulas automáticos o manuales. En la figura 8 se muestra un galpón de crianza al suelo. [5, 6]



Figura 8 Galpón de levante al suelo con control de parámetros técnicos de manera automática [7]

En la figura 9 se indica un galpón de levante industrial o automatizado con jaulas.



Figura 9 Galpón de levante en jaulas con control de parámetros técnicos de manera automática [8]

1.4.3.2 Galpón de postura y engorde

El galpón de postura y engorde es el que alberga las gallinas en las etapas de prepostura y producción que van desde la semana 16 a la 90, estos están adecuados para la recolección de los huevos y una óptima alimentación, el control de parámetros técnicos no es muy rigurosa ya que en esta etapa las gallinas ya han alcanzado su maduración y están dedicadas a la producción de huevos y en las semanas finales al engorde para el recambio. Los galpones de postura y engorde pueden ser al suelo o con jaulas, estos pueden ser

automáticos o manuales. En la figura 10, se indica un galpón adecuado para la producción al suelo.



Figura 10 Galpón inteligente CAVENCO para gallinas de postura al piso [9]

En la figura 11, se indica un galpón con jaulas apropiadas a la producción de huevos.



Figura 11 Galpón de producción con jaulas [9]

1.4.4 Parámetros técnicos a controlar en un galpón de levante de gallinas de postura

Los parámetros a controlar en los galpones de levante de gallinas de postura son los siguientes: [10]

- ✓ Temperatura
- ✓ Humedad relativa
- ✓ Calidad del aire
- ✓ Nivel de iluminación

1.4.4.1 Temperatura

La temperatura es el nivel térmico que posee un cuerpo o un sistema. En lo que se refiere a la crianza de aves se trata de la sensación térmica existente en los galpones, por lo que el nivel térmico debe ser controlado de manera continua, con el uso de criadoras convencionales o generadores de calor de tipo industrial la temperatura en los galpones depende de la edad de las aves. [11]

En la crianza de aves la temperatura corporal es uno de los indicadores más importantes que debido a que indican si la temperatura del galpón es la correcta y así permite ajustar de manera óptima la temperatura ambiente del galpón.

Tabla 1 Temperatura deseable del medio ambiente dependiendo la edad de las aves [11]

Edad	Temperatura del galpón °C	Temperatura corporal °C
Día 1-2	35-36	40-41
Día 3-4	33-34	40-41
Día 5-7	30-32	40-41
Semana 2	28-29	40-41
Semana 3	26-27	40-41
Semana 4	22-25	40-41
Desde la semana 5	18-22	40-41

La tabla 1 muestra el nivel medio de temperatura que deben poseer el galpón en las diferentes semanas de vida de las aves, se evidencia que la temperatura inicial debe ser de 35-36 grados centígrados y va disminuyendo un grado conforme transcurren las 4 primeras semanas hasta alcanzar una temperatura de 18-22 grados centígrados en la semana 5, manteniendo esta temperatura hasta el final de la etapa de crianza, mientras que la temperatura corporal de las aves siempre se mantendrá entre los 40-41 grados centígrados.

1.4.4.2 Humedad relativa

La humedad relativa es la medida de cuanta humedad expuesta existe como vapor de agua presente en el aire, además, representa el porcentaje saturación de agua en el aire a una temperatura determinada, comúnmente la capacidad de mantener la humedad en el aire depende de la temperatura.

En los galpones de crianza de gallinas la humedad relativa debe permanecer entre los 60 y 70 por ciento durante la etapa de crecimiento contemplada entre el día 1 y la semana 12 de vida de las aves, mientras que para las etapas de producción el nivel de humedad dependerá del lugar y las condiciones climáticas, por lo este parámetro no es necesario ser controlado porque las aves ya desarrollan la capacidad de adaptabilidad al medio ambiente. Este rango de humedad relativa permite el correcto desarrollo de las pollitas y el control de los gases tóxicos como el amoníaco, previniendo así el desarrollo de enfermedades inmunológicas y respiratorias. [11]

1.4.4.3 Calidad del aire

La calidad del aire es la representación de la cantidad de contaminantes existentes en el medio ambiente, en los galpones de crianza de gallinas se refiere al manejo de los siguientes gases: Oxígeno (O_2), dióxido de carbono (CO_2), monóxido de carbono (CO), amoníaco (NH_3) y el ácido sulfhídrico (H_2S) los que son producidos por las gallinas y sus desechos.

Tabla 2 Requerimientos mínimos de la calidad del aire [11]

Gases	Relación	Nivel
O_2	>	20%
CO_2	<	0,3%
CO	<	40ppm
NH_3	<	20ppm
H_2S	<	5ppm

En la tabla 2 se indica los niveles de los gases tóxicos que deben poseer los galpones de levante, como se muestra el nivel de Oxígeno (O_2) debe ser mayor al 20% , el dióxido de carbono (CO_2) menor que el 0,3%, el monóxido de carbono (CO) debe contener menos de 40 partes por millón en el aire, el amoníaco (NH_3) debe contener menos de 20 partes por millón en el aire y el ácido sulfhídrico (H_2S) debe contener menos de 40 partes por millón en el aire.

1.4.4.4 Nivel de iluminación

El nivel de iluminación o iluminancia es la cantidad de luz o flujo luminoso que incide sobre una superficie, la unidad de medida es el lux el cual está dado por:

$$E = \frac{\Phi}{S} = \frac{lumen}{metros^2}$$

Donde:

Φ = Flujo luminoso que llega a la superficie [Lumen]

S = Superficie o area a iluminar [m^2]

E = Iluminación en la superficie S [Lux]

En lo que se refiere a los galpones de crianza de gallinas el nivel de iluminación hace referencia a las horas luz que deben poseer las aves desde su primer día de vida hasta la etapa de producción. Para el correcto control del nivel de iluminación es importante conocer las horas luz en el exterior de los galpones, y así establecer un programa de iluminación. La tabla 3 indica el programa de iluminación que se debe implementar en los galpones según su etapa.

Tabla 3 Programa de iluminación para galpones cerrados para pollas ponedoras LOHMANN BROWN [11]

Edad (semanas)	Horas luz (Estándar)	Iluminancia (Lux)
Día 1-2	24	20-40
Día 3-6	18	20-30

2	16	10-20
3	14	10-20
4	12	4-6
5	11	4-6
6	10	4-6
7	9	4-6
8	9	4-6
9	9	4-6
10	9	4-6
11	9	4-6
12	9	4-6

1.4.5 Sistemas de control

Un sistema de control es el proceso que tiene un conjunto de elementos relacionados entre sí para alcanzar un determinado fin; generan valores en las señales de salida en función de las señales de entrada, lo que permite seleccionar la acción de control adecuada para mejorar las respuestas. La regulación y control de procesos industriales permiten la automatización de la producción, mejorando la calidad del producto y la optimización del costo de producción. En la figura 12 se muestra la representación básica de un sistema de control [12]

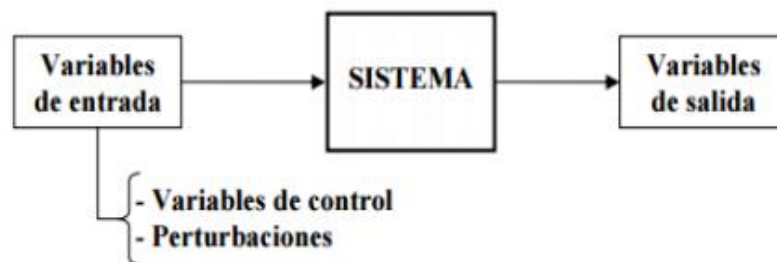


Figura 12 Esquema general de un sistema [12]

- **Sistemas de control industrial**

Un sistema de control industrial es la integración de hardware y software dentro de una red industrial, existen diferentes elementos que conforman un sistema como: actuadores, sensores y equipos de mando y control.

1.4.5.1 Actuadores

Un actuador es un dispositivo con la capacidad de generar una fuerza que ejerce un cambio de posición, velocidad o estado de algún tipo sobre un elemento mecánico, partir de la transformación de energía. Los actuadores se dividen en neumáticos, hidráulicos y eléctricos. [13]

Actuadores eléctricos

Los actuadores eléctricos son dispositivos que funcionan directamente con alimentación eléctrica, la cual va desde los 24 voltios en corriente directa hasta los 440 voltios en corriente alterna. Los actuadores eléctricos más comunes son los motores, estos dispositivos convierten la energía eléctrica en energía mecánica para realizar un trabajo ya sea de tipo rotacional o lineal por medio de la acción de los campos magnéticos variables. En la figura 13 se muestra la representación de un motor eléctrico. [14]



Figura 13 Motor eléctrico [14]

- **Calefactor**

Un calefactor eléctrico es una aplicación de las resistencias donde el calor generado por la corriente que fluye por un elemento resistivo se emplea para efectuar una función útil como la de propagar calor. En la figura 14 se puede observar un calefactor eléctrico de uso industrial. [15]



Figura 14 Calefactor eléctrico MET MANN [15]

1.4.5.2 Sensores

Un sensor es un dispositivo electrónico que adquiere una señal de entrada de un entorno físico y provee a su salida una señal manipulable de tipo eléctrica, además son los encargados de captar las magnitudes físicas y convertirlas en señales eléctricas capaces de ser entendidas y procesadas por un microcontrolador. [13]

Los sensores se pueden clasificar según la variable física a medir: [13]

- De posición, velocidad y aceleración
- De nivel y proximidad
- De humedad y temperatura
- De fuerza y deformación
- De flujo y presión
- De color, luz y visión
- De gas y pH
- Biométricos
- De corriente
- **Sensor de temperatura**

Un sensor de temperatura es un dispositivo electrónico capaz de detectar la variación en la temperatura del ambiente, a partir de una señal eléctrica la que es enviada a un sistema electrónico para su acondicionamiento es decir el filtrado y amplificado de la señal enviada por el detector. Los sensores o transductores pueden ser de diferente tipo dependiendo su aplicación, estos pueden ser: [16]

- **Detector de temperatura Resistivo (RTD):** Un RTD es un sensor de temperatura resistivo basado en la variación de resistencia del conductor, con respecto al nivel de temperatura expuesto.
- **Termocuplas o termopares:** Un termopar es un sensor de temperatura activo, debido a que genera una tensión eléctrica entre sus extremos en el rango de los milivolt esta tensión es proporcional a la temperatura a la cual es sometido el termo par. Las termocuplas pueden ser de diferentes tipos según el componente metálico que estén fabricadas, en la tabla 4 se indica los diferentes tipos de termopares según su composición química:

Tabla 4 Características de los termopares según el tipo.

Tipo de termopar	Composición química	Rango de temperatura	Sensibilidad $\mu\text{V}/^{\circ}\text{C}$
Tipo K	Cromo y Aluminio	-200°C a +1200°C	41 a 42
Tipo E	Cromo y Constantán	-200°C a +900°C	68 a 79
Tipo J	Hierro y Constantán	-40°C a +750°C	53 a 54
Tipo N	Nicrosil y Nisil	-270°C a +1300°C	28
Tipo B	Platino y Radio	0°C a 1800°C	3
Tipo R	Platino y Radio	-50°C a 1600°C	10 a 12
Tipo S	Hierro y Constantán	-500°C a 1600°C	10 a 11

Elaborada por: El investigador

- **Termistores:** Un termistor es un sensor de temperatura resistivo basado en la variación de resistencia del semiconductor, con respecto al nivel de temperatura expuesto. Estos pueden ser de dos tipos de coeficiente negativo de temperatura (NTC) y de coeficiente positivo de temperatura (PTC), estos permiten el paso parcial de la corriente. Los termistores son dispositivos fabricados a partir de óxidos metálicos como: manganeso, cobalto, cobre y níquel.

✓ Sensor de humedad

Un sensor de humedad es un dispositivo electrónico capaz de medir la cantidad de agua o la humedad relativa existente en el aire en un área determinada, este dispositivo convierte la magnitud física presente en el aire en una señal eléctrica con una intensidad entre 4 y 20 mA. Los sensores de humedad pueden ser:

- Sensor por deformación.
 - Sensor de bulbos húmedos y secos.
 - Sensor de humedad por contacto
-
- **Sensor por deformación:** es un sensor que se fundamenta en los cambios que experimentan las propiedades físicas del medio ambiente por efecto de la humedad, este sensor posee fibras sensitivas las que presentan una variación al contacto con la humedad que es representada por Δx como se indica en figura 17, donde el filamento resistivo (resorte) modificada su valor con respecto a la variación Δx . En la figura 15 se muestra el funcionamiento del sensor por deformación. [17]

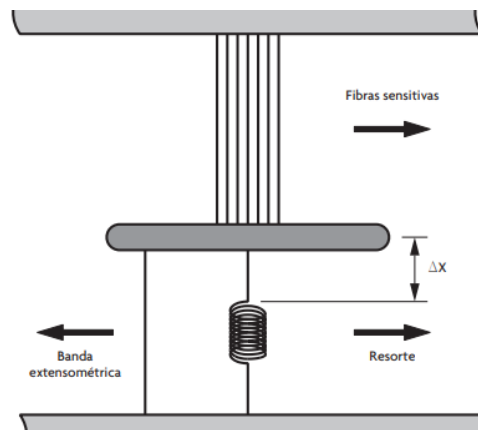


Figura 15 Funcionamiento del sensor por deformación [17]

- **Sensor de bulbos húmedos y secos:** es aquel que a partir de la temperatura cuantifica la cantidad de moléculas de vapor de agua que existe en el aire, posee dos bulbos, el uno seco y el otro húmedo como se muestra en la figura 16.

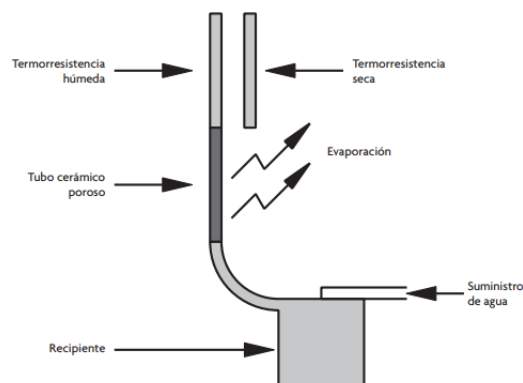


Figura 16 Sensor de humedad de tipo bulbo húmedo y seco [17].

- **Sensor de humedad por contacto:** es aquel que posee dos electrodos, los que permiten identificar la cantidad de humedad existente en la superficie de contacto, funciona en forma de un circuito abierto mientras más humedad exista en la superficie de contacto mayor será la conductividad y producirá un mayor flujo de corriente entre los dos electrodos.

La humedad relativa es la cantidad de agua existente en un aire expresada en un tanto por ciento de la cantidad que el aire tendría antes de llegar punto de saturación. Si estas cantidades son iguales se dice que la humedad relativa es del 100 por ciento [17].

✓ **Sensor de amoniaco**

Un sensor de amoniaco es un dispositivo capaz de medir los niveles de amoniaco en cantidades de partículas por millón (ppm), funcionan según el principio electroquímico de redox. Los sensores de amoniaco están compuestos por tres electrodos catalíticos, electrolito líquido y orificios de difusión de gas, el electrolito líquido al tener contacto con el amoniaco u otro gas realiza la transferencia de electrones generando una señal de corriente como salida En la figura 17 se muestra la estructura de una sensor de amoniaco NH₃-B1 [18].

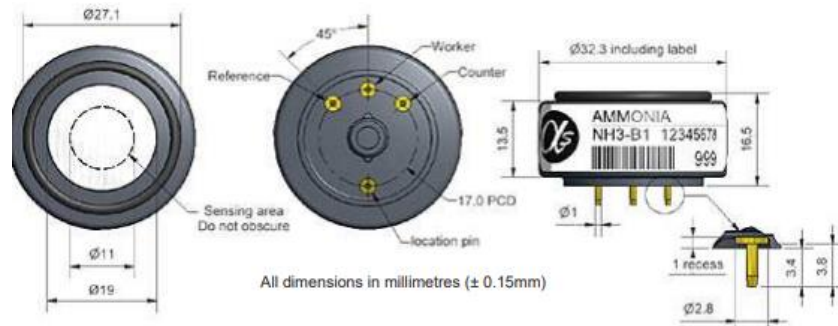


Figura 17 Diagrama esquemático de un sensor NH3-B1 [18]

✓ Sensor del nivel de iluminación

Un sensor de iluminancia es un equipo capaz de detectar la intensidad de luz existente en un espacio cerrado o abierto, esta intensidad o brillo de luz es medida en Lux, están compuestos por un transistor, un receptor y un circuito de detección. Los sensores de iluminancia pueden ser fotodiodos que responden a longitudes de onda similares a las del ojo humano y generan una señal de salida en micro amperios o en voltios, la que será proporcional a la cantidad de luz existente mientras mayor sea el nivel de iluminancia mayor será el nivel de corriente en la salida del sensor. En la figura 18 se indica un sensor de iluminancia con una salida variable. [19]



Figura 18 Sensor de iluminación de tipo mini-fotocélula [19]

1.4.5.3 Equipos de control y maniobra

Los equipos de control y maniobra son elementos que permiten realizar las acciones de abrir o cerrar el circuito, estos pueden ser:

- Pulsadores
- Relés
- Interruptores y conmutadores

- Contactores
- Relés temporizados
- Relés de protección
- Detectores y lámparas de señalización

✓ **Interruptores**

Los interruptores son dispositivos de control comúnmente de accionamiento manual, el cual permite abrir o cerrar un circuito, una vez accionado permanece en su posición de activación o desactivación hasta que se vuelve a accionar.

✓ **Contactor**

Un contactor es dispositivo eléctrico accionado a través de un electroimán, este electroimán se activa o desactiva cuando por el circula una corriente eléctrica, los contactores pueden ser de tipo abierto o cerrado. El contactor normalmente abierto (NA) permite el paso de la corriente cuando se encuentra accionado el electroimán, el normalmente cerrado (NC) impide el paso de la corriente cuando se acciona el electroimán. [20]

✓ **Relés de protección**

Los relés de protección son equipos capaces de proteger las instalaciones eléctricas sustituyendo a los fusibles, tienen la capacidad de estar preparados para su accionamiento en cualquier momento que la intensidad de corriente aumente de una manera no deseada. Los relés de protección pueden ser de tres tipos térmicos, electromagnéticos y magnetotérmicos. [21]

✓ **Lámparas de señalización**

Las lámparas de señalización son las encargadas de indicar al operador el funcionamiento del sistema y alertar en caso existir averías, para la utilización se debe seguir el código de colores que se indica en la tabla 5.

Tabla 5 Código de colores de las luces de señalización

COLOR	FUNCIÓN
Rojo	Emergencia (problemas o daños en el sistema)
Amarillo	Anomalía (advertencia)

Verde	En marcha (activación del sistema)
Blanco	Circuito en condiciones normales o sin accionar
Azul	Acción de mando

Elaborado por: El investigador

1.4.6 Placa de desarrollo

La placa de desarrollo es un dispositivo electrónico compuesto por un microcontrolador (microchip) reprogramable, el que puede ejecutar una serie de instrucciones, estas cuentan con puertos de comunicación, entrada y salida análogas o digitales para la comunicación entre los actuadores y sensores. Los microcontroladores utilizan diferentes lenguajes de programación como: C, C++, Python, Javascript, etc. En el caso de placas de desarrollo más desarrolladas se utiliza el sistema operativo Linux.

Un microcontrolador está conformado por una unidad central de procesamiento (CPU), bus de comunicación de datos, puertos de entradas y salidas analógicas y/o digitales y memorias Flash, EEPROM Y RAM como se muestra en la figura 19. [22]

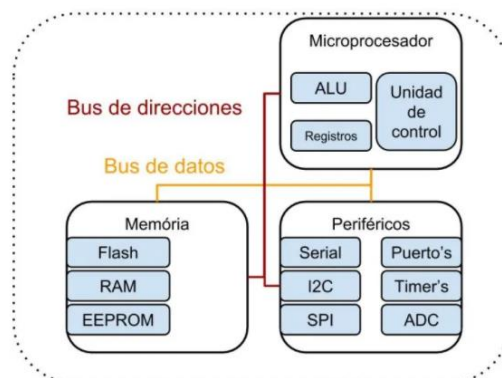


Figura 19 Arquitectura de un Microcontrolador [22]

Entre las placas de desarrollo más comunes se encuentran:

- ✓ **Arduino:** es definido como una plataforma de hardware y software libre basado en los microcontroladores AVR, por su facilidad al momento del uso, son muy conocidos por la gran acogida en el desarrollo de proyectos y aplicaciones domóticas, su lenguaje de programación está basada en

wirings y con su software de programación (IDE) basado en processing. [23]

- ✓ **Raspberry Pi:** es definido como un ordenador simple SoC (System on Chip), de bajo costo y de tamaño reducido, este dispositivo es conocido por su gran utilidad en los proyectos electrónicos, teniendo la capacidad de ejecutar las mismas acciones que una computadora común, debido a que cuenta con, memoria RAM, CPU, puertos de conectividad a la red y entradas y salidas de datos, en este ordenador se puede instalar todo tipo de sistema operativo libre el cual es ejecutado desde una tarjeta SD. [24]

1.4.7 Interfaz Hombre Maquina (HMI)

Una Interfaz Hombre Maquina (HMI) es la representación gráfica de un proceso industrial, también es conocida como panel de control que permite mantener una relación entre el operador y la maquinaria del proceso. La HMI permite mostrar la información en tiempo real con la que permite controlar y monitorear el proceso. La HMI se comunica con los controladores lógicos programables (PLC) y microcontroladores o tarjetas de desarrollo para mostrar la información. [25].

Las funciones más comunes de las HMI son las siguientes [25]:

- Visualización de datos
- Seguimiento del tiempo de producción
- Supervisar los KPI
- Supervisar entradas y salidas de las máquinas

1.4.8 Comunicación inalámbrica

La comunicación inalámbrica es la que se da entre dispositivos (móviles o no), engloba desde una comunicación Bluetooth entre un teléfono móvil y un ordenador portátil hasta una comunicación de dos terminales de telefonía móvil GSM. Incluso la comunicación verbal entre dos personas sería una comunicación inalámbrica: utilizan el aire como un canal para el intercambio de información. [26]

- **Clasificación de la comunicación inalámbrica**

Se pueden encontrar diferentes clasificaciones de las comunicaciones inalámbricas estas pueden ser por su alcance y la manera de controlar el acceso a la red.

Según el alcance las comunicaciones inalámbricas se dividen en tres grandes grupos: [26]

- ✓ Redes de área local inalámbrica (WLAN).
- ✓ Redes de área personal inalámbrica (WPAN).
- ✓ Redes de área extendida inalámbrica (WWAN)
- ✓ Redes de área extensa a baja potencia (LPWAN)

1.4.8.1 Redes LPWAN

La red de área extensa a baja potencia (LPWAN) es una tecnología inalámbrica creada para las comunicaciones entre dispositivos IOT y aplicaciones Maquina a Maquina (M2M), la red proporciona conectividad a un sin número de dispositivos con consumo de energía mínimo, además que emplea una comunicación de mediana y larga distancia, con baja tasa de información y baja potencia de operación. Las redes LPWAN poseen características como: [27]

- ✓ Bajo consumo eléctrico por parte de los dispositivos.
- ✓ Bajo coste de conectividad.
- ✓ Cobertura de 10-40km en zonas rurales y 1-5km en zonas urbanas.
- ✓ Bajo costo de fabricación en los dispositivos.
- ✓ Uso de un bajo ancho de frecuencia.

- **Arquitectura de la red LPWAN**

Las redes LPWAN poseen una arquitectura de tipo estrella, la que está conformado por tres elementos: los nodos finales, el Gateway o puerta de enlace y el servidor, en la figura 20 se representa la arquitectura de la red LPWAN. Los nodos finales adquieren la información para conectarse con la puerta de enlace la cual dirige la información a los nodos finales y hacia el servidor donde se manipulará y gestionará dicha información recibida.

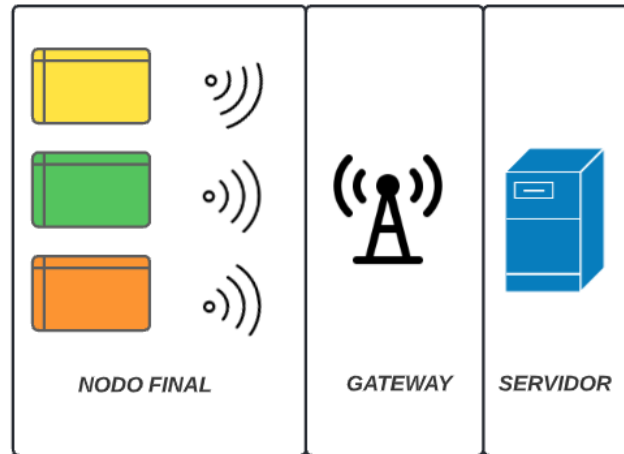


Figura 20 Arquitectura LPWAN

Elaborado por: El investigador

- ✓ **Nodo final:** está conformado por elementos direccionados a la adquisición de datos, la manipulación de actuadores y la comunicación con el Gateway.
- ✓ **Gateway:** está conformado por elementos que agrupan la información adquirida de los nodos finales y los comunica con el servidor a través de protocolos TCP/IP, comprobando el estado de la red.
- ✓ **Servidor:** es un conjunto de recursos informáticos donde se procesa y manipula la información recibida.

• Tecnologías LPWAN

Entre las tecnologías LPWAN más conocidas están LoRa, Sigfox, LTE-M y NB-IoT, las que son aplicadas en las redes de sensores y soluciones IoT. Presentan características como las que se indican en la tabla 6. [28]

Tabla 6 Comparación entre tecnologías LPWAN. [28]

Tecnología	LoRa	Sigfox	NB-IoT	LTE-M
Alcance máximo	5km (urbano)	10km (urbano)	1km (urbano)	1km (urbano)
	10km (rural)	40km (rural)	10km (rural)	5km (rural)

Tipo de espectro	No licenciado	No licenciado	Licenciado	Licenciado
Latencia	10 segundos	10 segundos	1,5-10 segundos	50-100 mili segundos
Velocidad de datos	50 kps	32 bytes/dia	250 kps	1 Mps
Modulación	CSS	BPSK	QPSK	OFDM
Frecuencias de operación	433 MHz 868 MHz 915 MHz	433 MHz 868 MHz 915 MHz	De 700 a 900 MHz	De 700 a 900 MHz
Ancha de banda	<500KHz	100Hz	1.4MHz o compartido	200KHz o compartido
Mensajes/dia (max)	Ilimitado	140 (UL), 4(DL)	Ilimitado	Ilimitado
Tasa de bits adaptables	SI	SI	NO	NO
Topología	Estrella	Estrella	Estrella	Estrella
Geografías	Europa, Asia, América	Europa, América	Europa, Asia, América	Europa, Corea del Sur, América

1.4.8.2 LoRa

LoRa es una tecnología de comunicación inalámbrica basada en la modulación de radio frecuencia (RF), dentro de las redes de amplio alcance de baja potencia LPWAN, proporciona una comunicación de largo alcance: de hasta 5km en áreas urbanas y 10km en áreas rurales, los requisitos de energía requeridas por LoRa son bajos, por lo que permite operar con dispositivos con baterías que

pueden durar hasta los 10 años. Las principales características son: baja transmisión de datos, largo alcance, bajo costo de los dispositivos, larga duración de las baterías en los dispositivos y gran capacidad de conexión. [29]

- **Modulación LoRa**

LoRa utiliza la modulación de espectro ensanchado basada en la técnica de Chirp Spread Spectrum (CSS), que consiste en utilizar pulsos de tipo chirp modulados a una frecuencia lineal de banda ancha para codificar la información, además utiliza factores de dispersión ortogonales, lo que permite la conservación de la batería de los nodos finales, LoRa opera en los rangos de frecuencia de 125KHz a 500KHz para canales ascendentes y solo de 500KHz para canales descendentes [29, 30].

LoRa trabaja dentro de la capa física del modelo OSI como se indica en la figura 21, en la cual en lugar de cableado utiliza al aire para transportar las ondas de RF desde el transmisor RF hasta el receptor de RF.

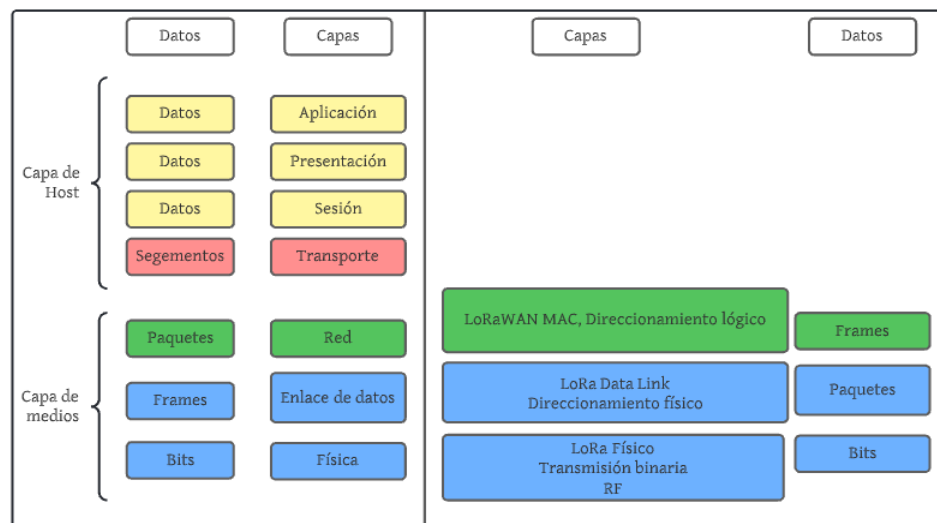


Figura 21 LoRa en las capas del modelo OSI

Elaborado por: El investigador

Para modulación en un sistema de espectro ensanchado de secuencia directa (DSSS), la fase de la señal portadora del transmisor varía de acuerdo a la secuencia del código como se indica en la Figura 22, al momento de multiplicar la señal de datos con un patrón de bits predefinido a una velocidad superior o conocida como código de expansión, se crea una señal más rápido con

componentes de frecuencia más alta que la señal original. Cuando la señal transmitida llega hacia el receptor de radio frecuencia RF, se realiza el proceso de multiplicación por una copia idéntica del código de dispersión utilizada en el transmisor de RF, lo que da como resultado una réplica de la señal de datos original. [29]

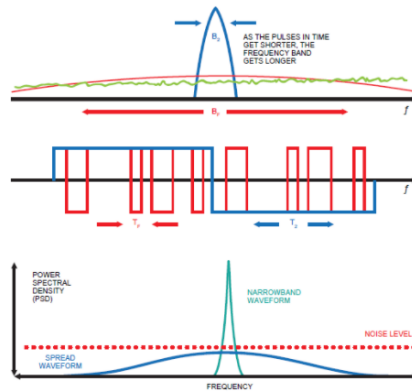


Figura 22 Cambios en la señal del transmisor de fase portadora del sistema DSSS [29]

Una desventaja que presenta un sistema DSSS es la necesidad de un reloj de referencia de alta precisión, mientras que la tecnología LoRa CSS ofrece una alternativa a los sistemas DSSS de bajo costo y bajo consumo de energía que no requiere un reloj de referencia de alta precisión. En la modulación LoRa la expansión del espectro de la señal se obtiene generando una señal chirp que varía de forma continua en frecuencia como se indica en la figura 23.

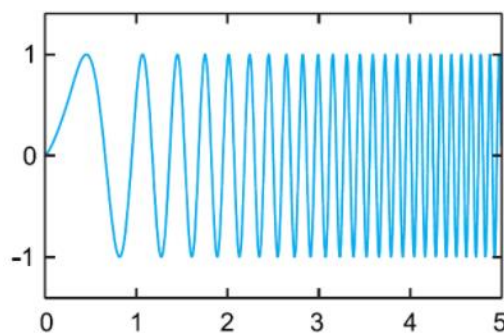


Figura 23 Espectro ensanchado de la modulación LoRa con chirp [29]

En LoRa la cantidad de código de dispersión aplicado a la señal de datos original es conocida como factor de dispersión (SF), mientras la modulación

LoRa tiene un total de seis factores de dispersión (SF7 a SF12), lo que representa a mayor SF mayor alcance de la señal al receptor de RF, la tabla 7 indica el factor de ensanchamiento en la modulación LoRa. [29, 30]

Tabla 7 Factores de ensanchamiento LoRa [29, 30]

Factor de ensanchamiento	Bit Rate	Alcance	Tiempo en el aire para 64-byte de payload
SF11	537 bps	10km	1320 ms
SF10	980 bps	8 km	700 ms
SF9	1760 bps	6 km	390 ms
SF8	3125 bps	4 km	220 ms
SF7	5470 bps	2 km	120 ms

- **Protocolo LoRaWAN**

El protocolo LoRaWAN es el estándar de LoRa Alliance, que define la comunicación y de la arquitectura de la red, para facilitar servicios seguros de comunicación bidireccional, localización y la movilidad. LoRa está presente en la capa física del modelo OSI por lo que permite crear enlaces de largo alcance siempre que se encuentre dentro del protocolo, en la figura 24 se indica la estructura del protocolo LoRaWAN. [30]

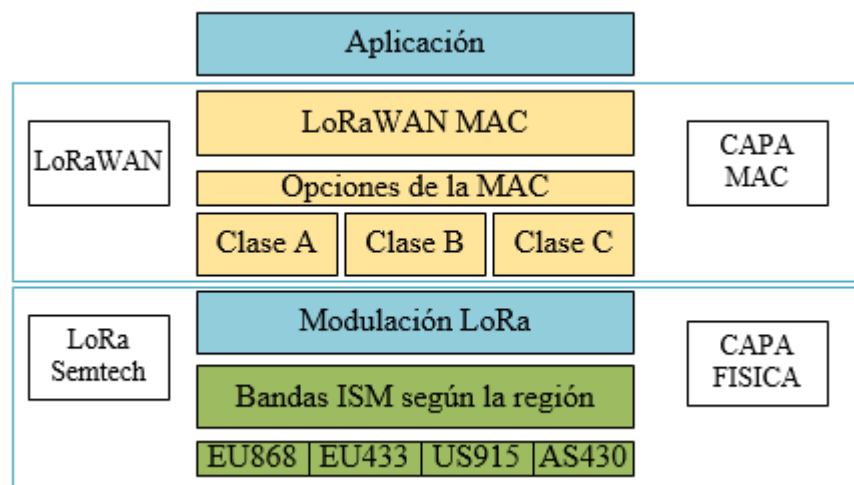


Figura 24 Estructura del protocolo LoRaWAN [29]

- **Elementos de red LoRaWAN**

Para la identificar los elementos de una red LoRaWAN, hay que tomar en cuenta la estructura del protocolo en el cual dice que las redes LoRa se encuentran basadas en LPWAN, por lo que se los elementos de dicha red son: dispositivos finales basados en LoRa, gateways, servidores de red y servidores de aplicación como se indica en la figura 25.

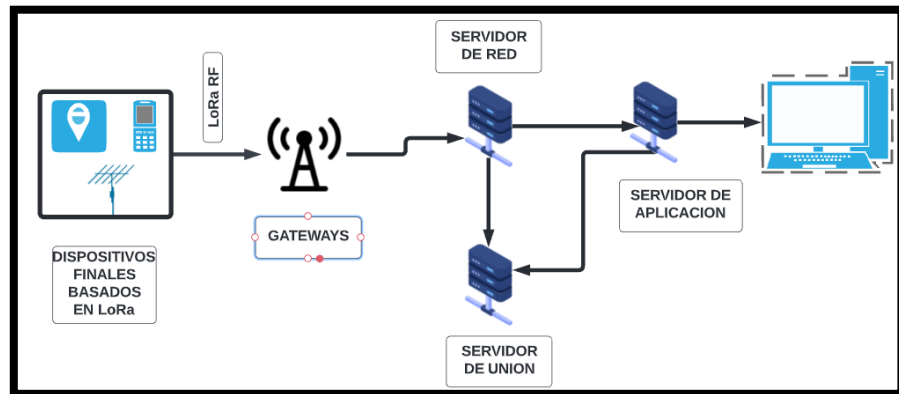


Figura 25 Elementos de una red LoRaWAN

Elaborado por: El investigador

Dispositivos finales basados en LoRa: los dispositivos finales basados en LoRa se consideran a los sensores, actuadores que se encuentran conectados de forma inalámbrica a una red LoRaWAN, a través de puertas de enlace de modulación RF.

Gateways: es una puerta de enlace entre los dispositivos finales, por la cual se recibe los mensajes enviados a través del medio de transporte.

Servidor de red LoraWAN: es aquel que administra toda la red, controla los diferentes parámetros de la red para adaptar el sistema a las condiciones establecidas de 128bits para el transporte de datos de extremo a extremo, además de controlar el tráfico de la red.

Las características que todos los servidores LoRaWAN deben cumplir son [29, 30]:

- ✓ Autenticación de tramas.
- ✓ Reconocimiento de mensajes.

- ✓ Responder a las solicitudes hechas por la capa MAC.
- ✓ Reenvió de mensajes de solicitud de unión y aceptación.
- ✓ Comprobación de la dirección de los dispositivos.

Servidor de aplicación: es el encargado de manejar, gestionar e interpretar los datos adquiridos por los sensores ubicados en la etapa de nodos finales, al igual que genera las señales de accionamiento a los actuadores.

Servidor de unión: es aquel que contiene la información para procesar las tramas de solicitud de unión al enlace ascendente y generar tramas de aceptación de unión al enlace descendente.

Los servidores de unión requieren cierta información de los dispositivos tal como: [29, 30]

- ✓ DevEUI: identificador único de serie del dispositivo final.
- ✓ AppKey: clave de cifrado de la aplicación.
- ✓ NwkKey: clave de cifrado de red.
- ✓ Identificador del servidor de aplicaciones.
- ✓ Perfil de servicio del dispositivo final.

1.4.9 Plan de frecuencias

La tecnología LoRa opera en las bandas de frecuencia asignadas por la UIT para aplicaciones militares, industriales y médicas (ICM), las cuales están consideradas como frecuencias sin licencia, por lo que en el Ecuador el ARCOTEL se acoge a los lineamientos establecidos por la UIT, en la Tabla 8 se muestra la banda de frecuencias, y los niveles de radiación permitidos en el país. [31]

Tabla 8 Niveles de radiación para bandas ICM según UIT-R SM. 1015-1 [31]

Banda de frecuencia	Frecuencia central	Intensidad del campo eléctrico (dBuV/m)
6765-6795 KHz	6780 KHz	80-100

13553-13567 KHz	13560 KHz	80-120
26957-27283 KHz	27120 KHz	70-120
40.60-40.70 MHz	40.68 MHz	60-120
902-928 MHz	915 MHz	60-120
2400-2500 MHz	2450 MHz	20-120

Lora utiliza las frecuencias libres de acuerdo a la región que se encuentre, la tabla 9 muestras los planes de frecuencias de cada una de las regiones.

Tabla 9 *Parámetros según la ubicación de la región. [31]*

Banda	Nombre común	Región
EU863-870	EU868	Región 1 Europa
US902-928	US915	Región 2 América
CN779-787	CN779	Región 3 China
EU433	EU433	Región 1 Europa
AU915-928	AU915	Región 3 Australia
CN470-510	CN470	Región 3 China
AS923	AS923	Región 3 Asia
KR920-923	KR920	Región 3 Corea
IN865-867	IN865	Región 3 India
RU864-870	RU864	Región 1 Rusia

1.4.10 Protocolo de comunicación

Un protocolo de comunicación es un conjunto de normas establecidas para que un servidor pueda comunicarse y entenderse con los diferentes dispositivos, mejorando la transmisión de la información de manera rápida y ligera, entre los

protocolos más utilizados en la industria 4.0 y en las IoT están: MQTT, CoAP y HTTP.

1.4.10.1 CoAP

El protocolo de aplicación restringida (CoAP) está dirigido a la comunicación de dispositivos de baja frecuencia utilizando el modelo REST perteneciente a HTTP, este fue diseñado para la transferencia de información máquina a máquina (M2M). Entre sus características están [32]:

- ✓ Modelo de reposo para dispositivos pequeños.
- ✓ Creados para la comunicación entre millones de nodos.
- ✓ Transferencia de habilidades del protocolo HTTP COAP.
- ✓ Seguridad equivalente a llaves RSA de 3072-bits.
- ✓ Compatibilidad con el protocolo HTTP
- ✓ Escoger el formato de los datos.

1.4.10.2 HTTP

El protocolo de transferencia de hipertexto (HTTP) es un protocolo cliente-servidor que realiza la comunicación entre un cliente web y el servidor HTTP, su diseño está basado en la estructura de conexión TCP/IP y el puerto de comunicación es el 80. Entre las características generales están: [32]

- ✓ Conexión única.
- ✓ Eliminación de información repetida o redundante.
- ✓ Comprensión de cabeceras para enviar menos información.
- ✓ Usado para transmitir documentos con extensión HTML.

La estructura del protocolo HTTP se muestra en la Figura 26, la que está conformada por el cliente, el servidor y el proxy, cada uno cumpliendo con una tarea específica, el cliente es el encargado de enviar la información y realizar la petición al servidor, el proxy verifica la información cumpliendo la función de gateway, el servidor se encarga de ejecutar y administrar la información.

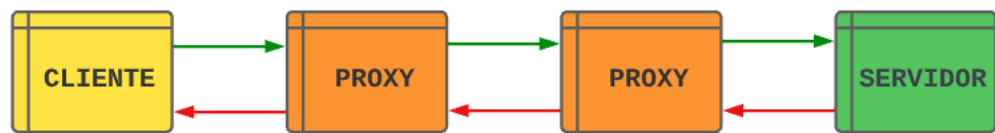


Figura 26 Estructura del protocolo HTTP

Elaborado por: El investigador

1.4.10.3 MQTT

El protocolo Transporte de Telemetría de Cola de Mensajes (MQTT) se basa en el modelo de publicación-suscripción de mensajes, esto permite la comunicación entre dos máquinas en tiempo real, por lo que requiere poco ancho de banda y un bajo consumo de energía, además actúa sobre la pila del protocolo TCP/IP para la comunicación. [33]

✓ Arquitectura de MQTT

La arquitectura del protocolo MQTT contiene dos entidades de red: el bróker y los clientes (publicador y subscriptor) como se observa en la Figura 27. El bróker es el servidor al cual se suscriben los clientes para la establecer la comunicación, mientras que los clientes son los encargados de proporcionar la información.

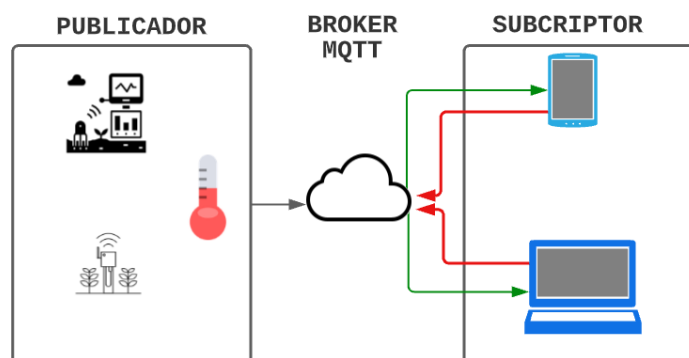


Figura 27 Estructura del protocolo MQTT

Elaborado por: El investigador

La información que se envía desde el publicador es de tipo mensaje, el mensaje está compuesto por dos elementos importantes: el topic o tema que es el

nombre con el cual se identifica el mensaje enviado, la carga o payload que es la información que contiene el mensaje. [33]

La comunicación entre el bróker y los clientes se ejecuta a través de los puertos de comunicación MQTT, generalmente son dos:

- ✓ **Puerto 1883:** es el que establece la conexión bróker MQTT-cliente sin encriptación, solo para la realizar pruebas rápidas.
- ✓ **Puerto 8883:** es el que establece la conexión bróker MQTT-cliente con encriptación basado en la Seguridad de la Capa de transporte (TLS) y la Seguridad de los puertos (SSL).

1.4.11 Base de datos

Una base de datos es un sistema formado por un conjunto de información, la cual puede ser manipulada y organizada de una manera estructurada, además estas permiten consultar, actualizar y modificar los datos almacenados en una tabla. Existen bases de datos como:

- **MySQL:** Es un sistema de gestión de bases de datos relacional, donde los datos se encuentran organizados en tablas de valores, donde las acciones de la base de datos se las realiza sobre las tablas. Donde las tablas están compuestas por filas y columnas, las que almacenan la información relativa a una característica. [34]
- **MariaDB:** es un gestor de base datos basado en código libre las que se encuentran incorporadas en los sistemas operativos Linux como Debian y Ubuntu.
- **Oracle Database:** es un sistema de administración de base datos multimodelo ejecutas con JavaScript y tablas blockchain.
- **InfluxDB:** es un sistema que se encarga de la gestión de las bases de dato para el almacenamiento de datos pertenecientes a series temporales, relacionadas a las IoT.

1.5 Objetivos

1.5.1 Objetivo general

Implementar un sistema de control y monitoreo inalámbrico, para la supervisión de los parámetros técnicos del galpón de crianza de gallinas de postura, en la granja Damiancito del Cantón Pelileo utilizando tecnología Lora y MQTT.

1.5.2 Objetivos específicos

- Analizar el sistema de control y monitoreo existente en el galpón de crianza de aves de postura en la granja Damiancito.
- Diseñar el sistema electrónico para el control y monitoreo de los parámetros técnicos del galpón de crianza de aves de postura.
- Implementar la Interfaz Hombre Maquina (HMI) del sistema de control y monitoreo inalámbrico de los parámetros técnicos del galpón de crianza de aves de postura.

CAPITULO II

METODOLOGÍA

2.1 Materiales

Para el desarrollo e implementación del proyecto de investigación se utilizaron materiales como: libros, artículos, tesis, revistas tecnologías, manuales de crianza de gallinas de postura, instructivos para el diseño de galpones de crianza y documentación web, además se requirió el galpón con pollitas desde el día 1 de hasta la semana 12, para la elaboración de las pruebas de funcionamiento del prototipo.

2.2 Métodos

2.2.1 Modalidad de la investigación

Este proyecto de titulación implemento la investigación aplicada, debido a que el propósito principal es poner en práctica los conocimientos de comunicación inalámbrica, electrónica y programación, los que se han adquirido durante la formación académica.

Se realizó investigación bibliográfica, con el fin de obtener información confiable acerca de temas relacionados con el monitoreo y control de los parámetros como temperatura y humedad en los galpones de aves de postura, se basará en fuentes documentales de libros, revistas, artículos, proyectos y páginas oficiales de Universidades de donde se obtendrá información necesaria para fundamentar el trabajo.

Durante el desarrollo del proyecto de investigación se realizó una serie de pruebas, en las que se analizó las falencias que puede llegar a tener el sistema de control y monitoreo de los galpones de aves de postura, así como el funcionamiento de la comunicación entre la estación emisora y receptora de datos.

Para el presente proyecto de investigación se utilizó la investigación de campo debido a que se levantó datos de forma directa en los galpones de levante de la granja Damiancito.

2.2.2 Recolección de la información

Para la recolección de la información en el presente trabajo de titulación se analizó artículos científicos de revistas indexadas, libros, artículos académicos y proyectos de investigación de las bases de datos de repositorios de las Universidades del país relacionados a diseño e implementación de sistemas de control y monitoreo inalámbrico de procesos industriales aplicados a la agroindustria.

2.2.3 Procesamiento y Análisis de Datos

El procesamiento y análisis de datos se desarrolló de acuerdo a los siguientes pasos:

- Análisis de la información recolectada en fuentes bibliográficas.
- Análisis del sistema existente en el galpón.
- Determinar los parámetros a medir en un galpón de aves de postura.
- Determinar los elementos principales utilizados para el control y monitoreo de un galpón de aves de postura.
- Planteamiento de la propuesta de solución.

2.2.4 Desarrollo del proyecto

Para el desarrollo del proyecto se realizó las siguientes actividades:

- Análisis del funcionamiento del sistema electrónico existente y la forma de operación.
- Identificación de los diferentes equipos de control que posee el galpón de levante de la granja.
- Determinar los parámetros a controlar en el galpón de levante de la granja Damiancito.

- Establecer los protocolos de comunicación inalámbrica utilizados en la industria avícola.
- Análisis del funcionamiento de la tecnología Lora y MQTT en la transmisión de datos en tiempo real
- Determinación de elementos electrónicos que intervienen en la comunicación inalámbrica.
- Selección de los equipos electrónicos como sensores y actuadores a utilizar para el control de los parámetros técnicos.
- Análisis de la distribución de los sensores y actuadores en el galpón de levante.
- Diseño el sistema electrónico para la medición y el control de los parámetros en el galpón de aves de postura.
- Desarrollo de la HMI para el control y monitoreo de los parámetros técnicos.
- Elaboración las placas PCB.
- Implementación del prototipo
- Pruebas de funcionamiento.
- Identificación y corrección de errores.
- Elaboración del informe final.

CAPITULO III

RESULTADOS Y DISCUSIONES

3.1 Análisis y discusión de los resultados

3.1.1 Análisis de factibilidad

Factibilidad técnica

Este proyecto es técnicamente factible, debido a que todos los dispositivos electrónicos y eléctricos empleados en el sistema han sido investigados previamente, además existen en mercado nacional por lo que se puede adquirirlos en el país.

Factibilidad económica

Este proyecto es económicamente factible debido a que todos los gastos correrán por cuenta del investigador y una vez finalizado el proyecto se desembolsara por parte de la granja. También tomando en cuenta que los actuadores no serán adquiridos ya que la granja cuenta con estos.

Factibilidad bibliográfica

La información requerida para este proyecto es de libre acceso, debido a que la encontramos en libros, artículos científicos, tesis referenciales y los conocimientos adquiridos a lo largo de la carrera.

3.1.2 Desarrollo de la propuesta

3.1.2.1 Análisis del sistema existente en el galpón de levante de la granja Damiancito

Para el análisis del sistema se acudió a la granja avícola Damiancito con el objetivo de realizar una revisión técnica al galpón de levante, en el cual se identificó los parámetros técnicos a controlar y los diferentes equipos existentes en dicho galpón. En la actualidad el galpón consta con el control de forma manual de los siguientes parámetros: temperatura y luminosidad y el

control de los actuadores. En la tabla 10 se indica los equipos existentes la función que cumplen cada uno de ellos.

Tabla 10 Descripción de los equipos existentes en el galpón de levante

Equipos	Cantidad	Función
Motor	3	<p>Encargados de extraer el aire caliente dentro del galpón, cuando la temperatura sobrepasa los niveles permitidos. Para su activación o desactivación se acude al tablero de control y se acciona el interruptor del motor que se desea activar.</p> <ul style="list-style-type: none"> • Motor 1: nivel bajo de velocidad de extracción. • Motor 2: nivel medio de velocidad de extracción • Motor 3: nivel alto de velocidad de extracción <p>Cada motor se encuentra conectado con su circuito de mando compuesto por relés, equipos de maniobra e indicadores.</p>
Calefactor	2	<p>Encargado enviar aire caliente cuando la temperatura se encuentra en niveles inferiores a los permitidos. Para su activación recibe una señal directa del termostato. Cada calefactor está conectado a un termostato.</p>
Termostato	2	<p>Encargo de medir la temperatura en el ambiente y dar la señal de activación al calefactor cuando la temperatura es inferior a la fijada en el termostato. Cada termostato se encuentra conectado a un calefactor.</p>

Ventoleras con actuador lineal	2	Encargo de abrir y cerrar las ventoleras del galpón, por donde ingresa el aire frío del exterior. Para su activación o desactivación se acude al tablero de control y se acciona el interruptor permanente a ventoleras. Cada ventolera lineal tiene su circuito de mando compuesto por relés, equipos de maniobra e indicadores
Dimmer	1	Encargado de controlar los niveles de luminosidad dentro del galpón, este se encuentra conectado de manera directa a los 60 leds distribuidos en el galpón.
Luces led	60	Encargados de proporcionar la energía lumínica en el galpón, su nivel de luminosidad depende del ajuste realizado en el dimmer.
Termómetro digital	2	Encargado de medir la temperatura dentro del galpón, se encuentran ubicados en la mitad del galpón junto al termostato.
Tablero de control	1	Lugar donde se realiza las conexiones de los circuitos de mando y control de los diferentes actuadores, y la distribución de energía eléctrica a los dispositivos.

Elaborado por: El investigador

3.1.2.2 Características técnicas de los diferentes equipos

Los motores que se describe a continuación son los que conjuntamente con el ventilador centrífugo forman un extractor como el que se indica en la figura 28.




Figura 28 Extractor de aire

Elaborado por: El investigador

- **Motor 1**

En la visita técnica realizada se identificó un motor de la marca WEQ, el cual este encargado de extraer el aire caliente del galpón y su velocidad y trabaja a una velocidad mima, para lo cual este motor cumple con las especificaciones técnicas indicadas en la tabla 11.

Tabla 11 Características del motor WEG W22 [35]

Equipo	Especificaciones técnicas	
 <p>Motor WEG W22</p>	Potencia	1.5 hp (1.1kW)
	Numero de polos	4
	Frecuencia	60 Hz
	Rotación nominal	1755rpm
	Tensión nominal	230/460V
	Corriente nominal	4.04/2.02A
	Corriente de arranque	33.9/17A
	Corriente en vacío	2.20/1.10A
	Torque animal	4.49 ft.lb

	Torque de arranque	290%
	Eficiencia	50%= 82.5
		75%= 85.5
		100%= 86.5
	Factor de potencia	50%= 0.60
		75%= 0.70
		100%= 0.79

- **Motor 2**

En la visita técnica realizada se identificó un motor de la marca WEQ, el cual está encargado de extraer el aire caliente del galpón a una velocidad media, este es utilizado cuando la temperatura dentro del galpón de levante es alta durante un tiempo superior a los 5 minutos, pero inferior a los 10 minutos, para lo cual este motor cumple con las especificaciones técnicas indicadas en la tabla 11.

- **Motor 3**


En la visita técnica realizada se identificó un motor de la marca WEQ, el cual está encargado de extraer el aire caliente del galpón a una velocidad alta, es utilizado cuando la temperatura dentro del galpón de levante es alta durante un tiempo superior a los 10 minutos, pero inferior a los 10 minutos, para lo cual este motor cumple con las especificaciones técnicas indicadas en la tabla 11.

Nota: Los tres motores pueden activarse de manera simultánea, esto dependerá del operador y si la temperatura que indica el termómetro digital es demasiado alta. Cuando se activan los 3 motores simultáneamente el galpón se enfría rápidamente, y la temperatura llega a niveles inferior a los deseados en muy poco tiempo y de una manera abrupta, lo que puede generar malestar o estrés térmico en las aves, por lo que se debe activar el calefactor de manera secuencial.

- **Calefactor**

El calefactor es el encargado de enviar aire caliente a todo el galpón, este dispositivo posee un ventilador y una resistencia para generar el calor. Las características de este dispositivo son las que se indica en la tabla 12.


Tabla 12 Características técnicas del calefactor ECOFLAM MAX 12 TC O 50-60Hz [36]

Equipo	Especificaciones técnicas	
<p>Calefactor ECOFLAM MAX 12</p> 	Potencia máxima del quemador	130 kW
	Potencia mínima del quemador	60 kW
	Caudal de alimentación del quemador	Max= 5kg/h Min= 11 kg/h
	Temperatura	Max= 112.200 kcal/h Min= 52.000 kcal/h
	Alimentación eléctrica	1/230 V a 50-60 Hz
	Potencia del Motor	130 W
	Clases NOx	Clase 2
	Alimentación del quemador	Diésel
	Revoluciones por minuto rpm (motor)	Max= 2800 Min= 3400

- **Ventolera**

La ventolera es la encargada de permitir el ingreso de aire frío a la parte superior del galpón, estas ventoleras se encuentran ubicadas en la parte de las cortinas del galpón y son manipuladas por el brazo Megamat MCZ, la cual mueve un total de 10 ventoleras con el accionamiento su brazo incorporado en su sistema. La tabla 13 muestra las características del equipo existente en el galpón.

Tabla 13 Características técnicas del sistema de ventoleras Megamat 2 TF 6KN -D53582 [37]

Equipo	Especificaciones técnicas	
<p>Motor con brazo Megamat 2 TF 6KN -D53582</p> 	Alimentación	24 Vcc
	Capacidad	Compresión=6kN Tracción=3kN
	Potencia	95 W
	Velocidad	8 mm/s
	Ciclo de funcionamiento	2/18 min
	Desplazamiento	300 mm
	Desconexión rápida para	-Versión GQR: Liberación rápida -Versión ER: liberación de Emergencia
	Descripción	MCZ 300

El sistema de ventoleras funciona con dos sensores de final de carrera, estos indican la distancia recorrida por el brazo del Megamat 2 TF 6KN -D53582, el cual establece el nivel apertura de las ventoleras. Este es accionado desde el tablero de control mediante un circuito de mando.

- **Termostato**

El termostato es el encargado de medir la temperatura y dar la señal de activación los calefactores, la tabla 14 muestra las características técnicas de este equipo.

Tabla 14 características técnicas del Termostato TS9501 [38]

Equipo	Especificaciones técnicas	
Termostato TS9501 	Rango de medición máximo	60 C°
	Rango de medición máximo	5 C°
	Voltaje máximo de contactos	C1 = 250 Vca C2 = 250 Vca
	Corriente de contacto	C1 = 16 A C2 = 6 A
	Valor diferencial	1K-2K
	Versión detectora o sonda	Capilar
	Temperatura máxima	70 C°
	Grado de protección	IP5
	Diferencia ajustable	N0

- **Dimmer**

El dimmer es el encargado de controlar el nivel de luminosidad en el galpón, este posee una memoria programable en la que se establece el porcentaje de luminosidad del galpón la cual puede ir desde el 0% hasta el 100% de luminosidad, y las características de este equipo se indica en la tabla 15.

Tabla 15 Características del Dimmer AviLamp220 [39]

Equipo	Especificaciones técnicas	
Dimmer AviLamp220 	Alimentación eléctrica	220 Vca
	Temperatura de operación	0-60C°
	Corriente	20 A
	Potencia máxima de trabajo	3000W
	Control de entrada analógica	0-10 V
	Ajuste de brillo	0-100%
	Frecuencia	60/50Hz

- **Termómetro digital**

El termómetro se utiliza para verificar si la temperatura ambiente del galpón es la adecuada, y comprar con la temperatura ajustada en el termostato. La tabla 16 indica las características del equipo existente en el galpón.

Tabla 16 Características del termómetro digital HAKUSA


Equipo	Especificaciones técnicas	
Termómetro digital HAKUSA 	Rango de mida temperatura	Min= 0 C° Max= 50 C°
	Rango de mida humedad relativa	Min= 10% Max= 99%
	Precisión	Temperatura $\pm 1\%$ Humedad $\pm 5\%$
	Fuente de alimentación	3 Vcc
	Ciclo de medición	Cada 10 segundos

Elaborado por: El investigador

- **Luces Led**

En el galpón de levante existen un total de 60 lámparas Led, encargadas de mantener las horas luz necesarias según la etapa de crianza. Estas se encuentran controladas por el dimmer, el que otorga el nivel de luminosidad en el galpón, además este es controlado de forma manual y su activación y desactivación la realiza el operador de acuerdo al número de horas luz que se necesite. En la tabla 17 se indica las características técnicas de la lámpara leds.

Tabla 17 Características de la lámpara Led Avilamp WY-03

Equipo	Especificaciones técnicas	
Lámpara Led Avilamp WY-03 	Tensión	220 Vca
	Potencia	10 W
	Frecuencia	60 Hz
	Corriente	0.048A
	Flujo luminoso	1200 lm
	Vida útil	25000 horas
	IRC	>80
	Temperatura de corte	4500 K°

Elaborado por: El investigador

3.1.2.3 Requerimientos del sistema

Para el diseño del sistema de control y monitoreo inalámbrico es necesario determinar los requerimientos tanto en la parte de hardware, software y las condiciones físicas del galpón de levante de las gallinas de postura en la granja Damiancito.

- **Requerimiento de hardware**

Para el desarrollo del prototipo se requiere de los siguientes dispositivos que se indican en la tabla 18.

Tabla 18 Elementos de hardware que se utilizó en la elaboración del prototipo

Dispositivo	Descripción	Requerimientos técnicos
Placa de desarrollo	La placa de desarrollo ESP32 para la adquisición de datos.	<ul style="list-style-type: none"> ▪ Entradas analógicas 1 ▪ Salidas analógicas 1 ▪ Salidas digitales 5 ▪ Entradas digitales 7 ▪ Voltaje de alimentación de 5 a 12 Vdc. ▪ Corriente de alimentación de 0.5 a 2A.
Módulos de comunicación inalámbrica LoRa	Modulo emisor y receptor LoRa para la comunicación inalámbrica entre la estación de control y el galpón de levante.	<ul style="list-style-type: none"> ▪ Comunicación LoRa a una distancia mínima de 2Km
Actuadores	Los actuadores a utilizar son: calefactor, ventiladores con motores AC. Ventolera y bomba de agua.	<ul style="list-style-type: none"> ▪ Voltaje de alimentación de 220 Vac. ▪ Corriente de alimentación de 2 a 5A.
Sensores	Los sensores adquieren los datos de las variables a medir como son: temperatura, humedad. Amoniaco o calidad del aire y luminosidad.	<ul style="list-style-type: none"> ▪ Sensor de humedad con capacidad de trabajo en un ambiente con temperaturas de 0 a 60 C°. ▪ Sensor de temperatura con capacidad de trabajo en un ambiente con temperaturas de 0 a 60 C°. ▪ Sensor de amoniaco con capacidad de trabajo en un ambiente con temperaturas de 0 a 60 C°. ▪ Voltaje de alimentación de 5 a 12 Vdc

		<ul style="list-style-type: none"> ▪ Corriente de alimentación de 0.5 a 2A.
Elementos de maniobra y control	Los elementos de control y maniobra se utilizaron en el tablero de control y estos son: Contactores, pulsadores y luces piloto.	<ul style="list-style-type: none"> ▪ Selectores que trabajen a 0 a 220V. ▪ Indicadores que trabajen a 0 a 220V. ▪ Contactores que trabajen a 220V
Fuente de alimentación	Para energizar a la placa ubicada en la etapa de adquisición de datos se procedió al desarrollo de una fuente DC.	<ul style="list-style-type: none"> ▪ Voltaje de alimentación 220V. ▪ Salidas de voltaje en DC de: 5V, 12V y 24V.

Elaborado por: El investigador

• **Requerimiento de software**

Para el desarrollo del prototipo se requiere además de los elementos de hardware los elementos de software, ya que estos permiten el procesamiento de la información y la creación de instrucciones para los actuadores, los requerimientos de software se indican en la tabla 19.

Tabla 19 Elementos de software que se utilizó en la elaboración del prototipo

Elementos	Descripción	Requerimientos técnicos
Servidor web	El servidor web se utiliza para ejecutar los archivos de programación y el almacenamiento de la información.	<ul style="list-style-type: none"> ▪ Procesador de 2,5 GHz ▪ Memoria RAM de 4 GB ▪ Almacenamiento mínimo de 40GB. ▪ Ubuntu server.

Software de programación	el software de programación permite el desarrollo de las instrucciones y el procesamiento de los datos adquiridos además la publicación de los mismo en una HMI.	<ul style="list-style-type: none"> ▪ Acceso libre sin licencias.
Base de datos	La base de datos almacena la información adquirida por el nodo de sensores.	<ul style="list-style-type: none"> ▪ Almacenamiento mínimo de 10GB. ▪ Contenga la habilitación de los puertos de comunicación.

Elaborado por: El investigador

3.1.2.4 Requerimientos de las condiciones físicas del galpón de levante de gallinas de postura LOHMANN BROWN.

Las condiciones físicas del galpón de levante son importantes para el desarrollo del prototipo, debido a que, indican los parámetros técnicos a controlar dentro del galpón y estos son: temperatura, humedad relativa, nivel de iluminación y calidad del o amoniaco.

• Temperatura

Los niveles de temperatura que se requieren en el galpón de levante de postura son los que se indica en la tabla 20 , estos representan a las primeras 12 semanas de vida de las aves. Con dicha información se procede al diseño e implementación del sistema inalámbrico.

Tabla 20 *Temperatura deseable del medio ambiente dependiendo la edad de las aves [11]*

Edad	Temperatura del galpón °C	Temperatura corporal °C
Día 1-2	35-36	40-41

Día 3-4	33-34	40-41
Día 5-7	30-32	40-41
Semana 2	28-29	40-41
Semana 3	26-27	40-41
Semana 4	22-25	40-41
Desde la semana 5	18-22	40-41

- **Humedad relativa**

La humedad relativa establecida por el proveedor (LOHMANN BROWN) de gallinas dentro del galpón de levante de 60 y 70 porciento, esto debe permanecer constante desde el día 1 de vida hasta la semana 10.

- **Calidad del aire**

para el control de la calidad del aire hay que tener en cuenta los diferentes gases que se generan dentro del galpón como son.: Oxígeno (O_2), dióxido de carbono (CO_2), monóxido de carbono (CO), amoníaco (NH_3) y el ácido sulfhídrico (H_2S) los que son producidos por las gallinas y sus desechos. En la tabla 21 se indica los niveles que debe poseer cada uno de los gases en el galpón de crianza según el proveedor.

Tabla 21 Requerimientos mínimos de la calidad del aire [11]

Gases	Relación	Nivel
O_2	>	20%
CO_2	<	0,3%
CO	<	40ppm
NH_3	<	20ppm
H_2S	<	5ppm

- **Nivel de iluminación**

El nivel de iluminación del galpón depende de la etapa en la que se encuentren las aves, y se hace referencia en horas luz al día, por lo que el proveedor recomienda un programa de iluminación el cual se indica en la tabla 22.

Tabla 22 Programa de iluminación para galpones cerrados para pollas ponedoras LOHMANN BROWN [11]

Edad (semanas)	Horas luz (Estándar)	Iluminancia (Lux)
Día 1-2	24	20-40
Día 3-6	18	20-30
2	16	10-20
3	14	10-20
4	12	4-6
5	11	4-6
6	10	4-6
7	9	4-6
8	9	4-6
9	9	4-6
10	9	4-6
11	9	4-6
12	9	4-6

3.1.2.5 Etapas del sistema de control y monitoreo inalámbrico

El sistema está constituido por las siguientes etapas: nodo de sensores y nodo de actuadores, comunicación inalámbrica, procesamiento de datos y la publicación o visualización de la información en una pantalla HMI. En la etapa

del nodo sensorial se utiliza una placa de desarrollo, esta permite adquirir los datos enviados por los sensores de temperatura, humedad relativa y la calidad del aire, en la etapa de comunicación inalámbrica se utilizó los módulos LoRa tanto en el receptor como emisor, en la etapa de procesamiento de información se utiliza un servidor web el permite tratar la información y generar las instrucciones hacia la etapa de actuadores, además permite conectar con la etapa de publicación en la HMI al cual tendrá acceso a través del servicio de internet. En la tabla 23 se indica las etapas del sistema de control y monitoreo inalámbrico.

Tabla 23 Etapas del sistemas

Etapas	Descripción
Adquisición de datos y control de actuadores	<ul style="list-style-type: none"> • Comprende de sensores de temperatura, humedad, y calidad del aire. • Controla los actuadores como: ventiladores con motores AC, ventoleras y la bomba de agua. • Envía y recibe los datos a la etapa de comunicación inalámbrica mediante una placa de desarrollo.
Procesamiento de datos y visualización de datos	<ul style="list-style-type: none"> • Recibe los datos de la etapa de Adquisición de datos. • Procesa la información con la tarjeta de desarrollo • Envía los datos a la etapa de control de actuadores. • Envía los datos a etapa de visualización • Manejo y control de los actuadores y sensores mediante una HMI.
Comunicación inalámbrica	<ul style="list-style-type: none"> • Está constituida por los módulos de comunicación inalámbrica LoRa y el

	servidor WEB para el procesamiento de información.
--	--

Elaborado por: El investigador

3.1.2.6 Diagrama de bloques del sistema

El esquema del sistema de control y monitoreo está constituido por 3 etapas la etapa de adquisición de datos, la comunicación inalámbrica con tecnología LoRa y la etapa de monitoreo y control estas dos últimas se encuentran integradas en una sola estación la receptora de los datos por parte de la adquisición de datos. En la figura 29 se muestra el esquema del sistema implementado.

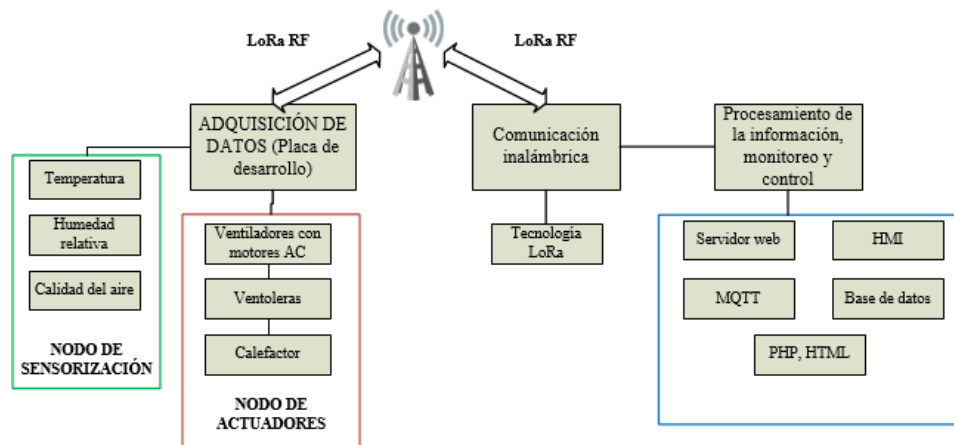


Figura 29 Diagrama de bloques del sistema de control y monitoreo inalámbrico

Elaborado por: El investigador

3.1.2.7 Selección de los dispositivos para la implementación del prototipo

El análisis técnico desarrollado a los componentes eléctricos y electrónicos pertenecientes al sistema, permitió establecer sus principales características y especificaciones de funcionamiento, para esto se realizó la búsqueda de dispositivos existentes en el mercado nacional. Para lo cual se desarrolló tablas comparativas con cada uno de los elementos a utilizar en las diferentes etapas del sistema.

- **Tarjeta de desarrollo**

La tarjeta de desarrollo permite la adquisición de los datos, la manipulación y programación de las instrucciones para cada uno de los dispositivos pertenecientes al sistema, con la ayuda de un lenguaje de programación el que ejecutara dichas instrucciones mediante su microprocesador. En la tabla 24 se muestra la comparativa entre las diferentes placas de desarrollo existentes en el mercado.

Tabla 24 Comparativa de las diferentes placas de desarrollo del nodo [40, 41, 42]

Características	Tarjeta de desarrollo del nodo		
Dispositivo	Arduino uno WiFi Rev2	Lilygo LoRa32 (TTGO)	WiFi LoRa 32 (HELTEC LoRa)
Tipo	Microcontrolador	Microprocesador	Microprocesador
Memoria	- Flash 48KB - SRAM 1KB - EEPROM 256B	- Flash 4MB - SRAM 520KB -EEPROM 384KB	- Flash 4MB -SRAM 520KB -EEPROM 384KB
Procesador	ATmega4809	SX1276	LX6 de 32 bits
WiFi	Si 2.4 GHz-2.5 GHz IEEE802.11b/g/n	Si 2.4 GHz-2.5 GHz IEEE802.11b/g/n	Si 2.4GHz-2.5 GHz IEEE802.11b/g/n
LoRa	No	Si	Si
Frecuencia de reloj	16HHz	Ajustable de 80 a 240MHZ	Ajustable de 80 a 240MHZ
Voltaje de operación	5v	5v	5v

Voltaje de entrada máximo	12v	12v	12v
Voltaje de entrada mínimo	7v	7v	7v
Corriente	20mA	60mA	50mA
Rango de temperatura de trabajo	En la industria desde -40 a +85C°	En la industria desde -40 a +85C°	En la industria desde -40 a +85C°
Números de pines digitales	E/S 14	Poseen un total de 48 pines los que se distribuyen de la siguiente manera:	
Números de pines analógicos	Entrada 6	18 canales convertidores de analógico a digital (ADC) 3 interfaces SPI 3 interfaces UART 2 interfaces I2C 16 canales de salida PWM 2 convertidores de digital a analógico (DAC) 2 interfaces I2S 10 GPIO de detección capacitiva	
Disponibilidad en el país	Si	Si	Si
Costo en dólares	30.66\$ sin IVA	45.23\$ sin IVA	50.75\$ sin IVA

Elaborado por: El investigador

Una vez realizado el análisis comparativo entre las placas de desarrollo en la tabla 21, se selecciona la Lilygo LoRa32 (TTGO) debido a que se puede realizar la adquisición de datos y la comunicación inalámbrica con la misma

tarjeta, gracias a su microprocesador ESP32 que permite la configuración del protocolo LoRa, además posee el número de entradas y salidas analógicas y digitales que requiere el sistema, por lo que cumple con los requerimientos técnicos.

- Configuración de pines de la tarjeta Lilygo LoRa32 (TTGO)**

La disposición de pines de la tarjeta Lilygo LoRa32 (TTGO) está dado por su chip ESP32, el cual cuenta con un total de 48 pines, los se encuentran distribuidos en pines de múltiples funciones y en pines que no se pueden utilizar, para lo que en la figura 30 se muestra la distribución de los pines en la placa.

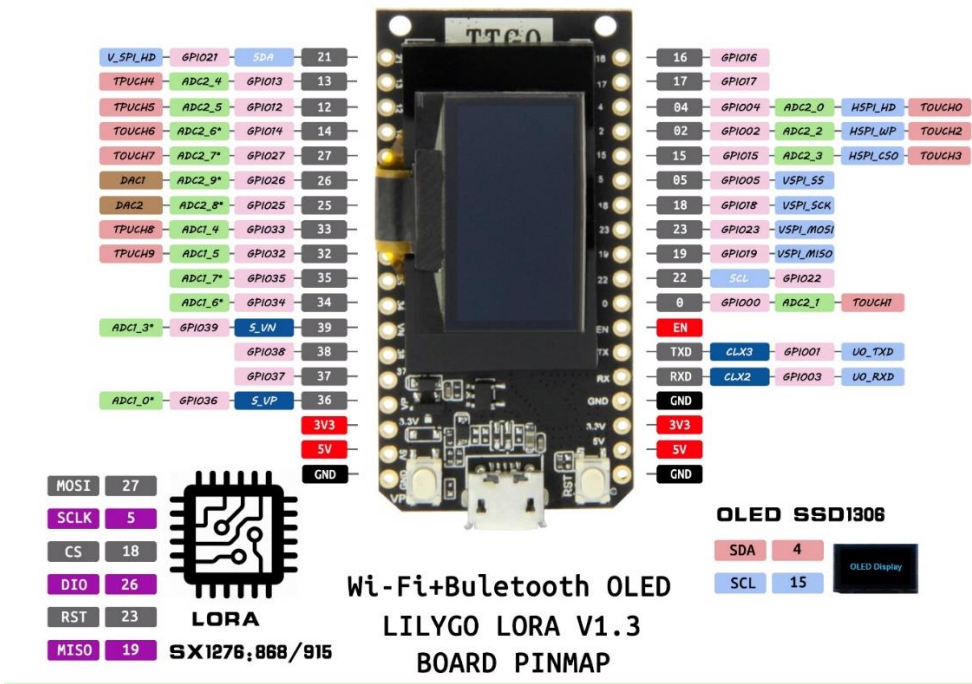


Figura 30 Distribución de pines de la tarjeta Lilygo LoRa32 V1.3 (TTGO)
[43]

Para determinar las funciones de cada uno de los pines se ha desarrollado la tabla 25 en la que se describe la función de cada uno estos.

Tabla 25 Distribución de pines de la tarjeta Lilygo LoRA32 V1.3 [43]

Pines	Función	Observación
GPIO34 GPIO35 GPIO36 GPIO39	Entradas sin resistencias internas de pull-up y pull-down	No se pueden usar como salidas porque presentan interrupciones al arranque
GPIO 6 GPIO 7 GPIO 8 GPIO 9 GPIO 10 GPIO 11	Pines conectados al flash SPI.	No se pueden conectar, uso específico del integrado EPS-WROOM-32
T0 (GPIO 4) T1 (GPIO 0) T2 (GPIO 2) T3 (GPIO 15) T4 (GPIO13) T5 (GPIO12) T6 (GPIO14) T7 (GPIO27) T8 (GPIO33) T9 (GPIO32)	GPIO táctiles capacitivos, también son configurables como salidas.	Pines capaces de detectar cambios de en alguna carga eléctrica.

ADC1_CH0 (GPIO36) ADC1_CH1 (GPIO37) ADC1_CH2 (GPIO38) ADC1_CH3 (GPIO 39) ADC1_CH4 (GPIO32) ADC1_CH5 (GPIO33) ADC1_CH6 (GPIO34) ADC1_CH7 (GPIO35) ADC2_CH0 (GPIO 4) ADC2_CH1 (GPIO 0) ADC2_CH2 (GPIO2) ADC2_CH3 (GPIO15) ADC2_CH4 (GPIO13) ADC2_CH5 (GPIO12) ADC2_CH6 (GPIO14) ADC2_CH7 (GPIO27) ADC2_CH8 (GPIO25)	Pines convertidores de analógico a digital (ADC), también son configurables como salidas.	Posee canales de entrada ADC de 18 X 12 Bits
DAC1 (GPIO25) DAC2 (GPIO26)	Pines convertidores de Digital a analógico (DAC)	Posee canales de entrada ADC de 2 X 8 Bits
PWM	Todos los pines que se comportan como salidas pueden ser configurables con PWM excepto desde el GPIO 34 al 39.	Hay que tener en cuenta los siguientes parámetros para su configuración:

		<ul style="list-style-type: none"> • Frecuencia del reloj • Ciclo de trabajo • Canal PWM • CPIO donde se va configurar
--	--	--

- **Temperatura**

El sensor de temperatura permitirá determinar el nivel de calor existente en el galpón para lo cual este debe estar en ubicado en el interior del mismo, para lo cual se realizó la comparación entre diferentes sensores existente en el mercado, en la tabla 26 se muestra las características de cada uno de estos dispositivos.

Tabla 26 Comparativa de los diferentes sensores de temperatura [44, 45, 46]

Características	Sensores de temperatura		
Dispositivo	DS18B20	M18TB14Q	M12FTH4Q
Alimentación	Max: 5Vcd Min: 3Vcd	Max: 20Vcd Min: 10Vcd	Max: 20Vcd Min: 10Vcd
Numero de puertos de datos a utilizar	1 puerto analógico	1 puerto analógico	1 puerto analógico
Rango de temperatura	Max: +125C° Min: -55C°	Max: +70C° Min: -20C°	Max: +85C° Min: -40C°
Precisión	± 0,5C°	± 0,1C°	± 0,1C°
Resolución en bits	De 9 a 12 bits	-	-

Tiempo de conversión	Los 12 bits en 750ms máximo	<25ms	<25ms
Alcance o cobertura	10m Ø	5m Ø	12m Ø
Costo en dólares	15,12\$ sin IVA	20,17\$ sin IVA	33,60\$ sin IVA

Una vez realizado el análisis comparativo entre los sensores temperatura en la tabla 26, se selecciona el sensor M12FTH4Q porque permite la manipulación de los datos en forma de bits, además el rango de temperatura que soporta es mayor que los demás sensores, y está en el rango de operación de los requerimientos técnicos del sistema, también se lo encuentra en el mercado nacional a un costo accesible para el usuario. En la figura 31 se puede observar al sensor M12FTH4Q.



Figura 31 Sensor de temperatura M12FTH4Q [46]

- **Calidad del aire**

El sensor de calidad del aire nos permitirá medir el nivel de amónica existente en el galpón de levante, debido a que este gas es el más abundante en el galpón, ya que es generado por la defecación de las aves, y este es el que se requiere controlar. La tabla 27 indica los diferentes sensores de gases existentes en el mercado.

Tabla 27 Comparativa de los diferentes sensores de calidad del aire [47, 48, 49]

Características	Sensores de calidad del aire y humedad		
Dispositivo	ZPHS01B	MQ137	Air Monitoring HAT
Alimentación	Max: 5Vcd Min: 3Vcd	Max: 5Vcd Min: 1Vcd	Max: 5Vcd Min: 3Vcd
Corriente pico	< 500mA	-	100mA
Corriente promedio	< 300mA	-	200μA
Señal de salida	UART (TTL)	Analógica en voltios (TTL logic)	UART (TTL)
Rango del tamaño de partículas	Max: 10μm Min: 0,3 μm	5 ~ 500 ppm	Max: 0.5μm Min: 0,3 μm
Precisión	± 15%	± 0,5%	± 10%
Resolución en bits	De 8 bits	-	De 8 bits
Tiempo de respuesta	T90<45s	<3s	<10s
Alcance o cobertura	12m ³	6m ³	15m ³
Rango de humedad relativa	0 – 99%	No cumple	0 – 99%

Tiempo de funcionamiento	De 4-5 años	De 1-3 años	De 4-5 años
Costo en dólares	160,37\$ sin IVA	56,56\$ sin IVA	67,15\$ sin IVA

El sensor seleccionado es el MQ137 debido a que cumple con la función de medir el gas con mayor presencia en el galpón que es el amoníaco (NH_3), además contamos con la disponibilidad en el mercado nacional y su tiempo de funcionamiento es de 3 años, también permite ser controlado por la placa de desarrollo seleccionada sin la necesidad de adquirir dispositivos para su comunicación.

- **Funcionamiento del sensor MQ137**

El sensor MQ137 es sensor de tipo electroquímico que varía su resistencia al momento de ser expuesto con determinados gases en este caso con el amoníaco, para esto posee en su interior un calentador que reacciona con el gas y aumenta la temperatura interna de sensor y directamente la variación de la resistencia. Este sensor debe ser conectado en serie con una resistencia de carga (R_L), con la que se cierra el circuito y así obtener un valor analógico (voltios) el que se controla desde una tarjeta de desarrollo. En la figura 32 se indica la conexión del sensor y la resistencia en serie, obteniendo un divisor de voltaje. [48]

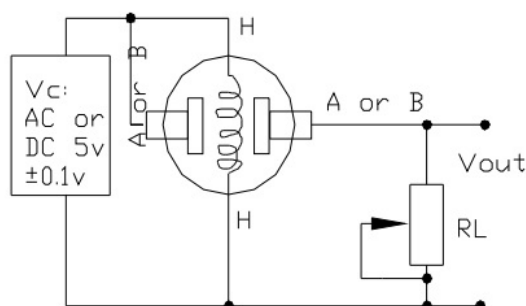


Figura 32 Conexión del sensor MQ137 [48]

- **Humedad**

El sensor del nivel de humedad permite control el porcentaje de humedad relativa existente en un ambiente, estos sensores deben ser capaces de medir entre un 0 y 100% de HR, para lo cual se indica en la tabla 28 las características de 3 sensores de Humedad, los que se pueden utilizar en un galpón de levante de gallinas de postura.

Tabla 28 Comparativa de los diferentes sensores de Humedad Relativa [49, 48, 47]

Características	Sensores de calidad del aire y humedad		
Dispositivo	ZPHS01B	DHT22	Air Monitoring HAT
Elemento a detectar	Humedad	Humedad	Humedad
Alimentación	Max: 5Vcd Min: 3Vcd	Max: 6Vcd Min: 3.3Vcd	Max: 5Vcd Min: 3Vcd
Corriente pico	< 500mA	2,5mA	100mA
Corriente promedio	< 300mA	1.5mA	200µA
Señal de salida	UART (TTL)	Señal digital mediante bus único	UART (TTL)
Rango del tamaño de partículas	Max: 10µm Min: 0,3 µm	-	Max: 0.5µm Min: 0,3 µm
Precisión	± 15%	± 2% humedad relativa	± 10%

Resolución en bits	De 8 bits	De 8 bits	De 8 bits
Tiempo de respuesta	T90<45s	<2s	<10s
Alcance o cobertura	12m ³	10m ³	15m ³
Rango de humedad relativa	0 – 99%	0 – 99%	0 – 99%
Tiempo de funcionamiento	De 4-5 años	De 1-3 años	De 4-5 años
Costo en dólares	160,37\$ sin IVA	16,80\$ sin IVA	67,15\$ sin IVA

El sensor seleccionado para medir la humedad relativa en el galpón de levante es el DHT22, debido a que presta las características requeridas por el sistema que es de una cobertura mínima de 10m³, también es un sensor que existente en el mercado nacional y su durabilidad de 1 a 3 años, mientras que los demás sensores indicados en la tabla 25 no se encuentran disponibles en el mercado nacional y su costo económico elevado.

✓ **Funcionamiento del sensor DHT22**

El sensor DHT22 es un sensor digital con la capacidad de medir la temperatura con un termistor y un sensor capacitivo para medir la humedad. Para la comunicación entre sensor y placa de desarrollo utiliza el protocolo de un solo bus es decir un solo hilo por el cual envía la señal digital, el dispositivo cuenta con 4 pines enumerado del 1 al 4 los cuales son: pin 1 alimentación, pin 2 señal digital pin 3 nulo y pin 4 conexión a tierra, además posee una calibración realizada en la fábrica y almacenado en la memoria OPT, lo que asegura alta estabilidad y fiabilidad en el funcionamiento. En la figura 33 se indica la estructura del sensor y su conexión a la placa de desarrollo. [50]

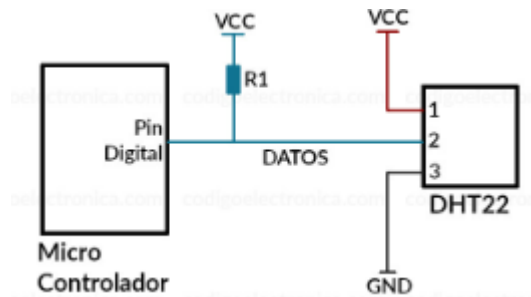


Figura 33 Conexión del sensor DHT22 [50]

Los actuadores que se utilizaron para el control de los diferentes parámetros técnicos son los existentes en la granja, y descritos anteriormente. Tanto los motores que trabajan en la parte de extracción de aire del galpón y el calentador en la etapa de calentamiento del galpón, de igual manera para el ingreso de aire las ventoleras, todos estos dispositivos siendo activados por el operador.

- **Selección de los elementos de comunicación inalámbrica LoRa**

- ✓ **Nodo- Gateway**

El nodo sensor posee una tarjeta de desarrollo con tecnología LoRa la cual posee características de comunicación inalámbrica, las que se pueden verificar en la tabla 29 y la comparación con tarjetas similares.

Tabla 29 Características de comunicación inalámbrica de las placas con tecnología LoRa [41, 43, 51]

Característica	Heltect ESP8266	Lilygo LoRa32	Shield Lora
Comunicación inalámbrica	<ul style="list-style-type: none"> • Lora • WiFi • Bluetooth 	<ul style="list-style-type: none"> • Lora • WiFi • Bluetooth 	<ul style="list-style-type: none"> • LoRa
Chip RF	SX1276/SX1278	SX1276/SX1278	SX1276/SX1278
Frecuencia de operación	868/915 MHz	868/915 MHz	868/915 MHz

Potencia de transmisión	19,5dBm	20dBm	20dBm
Sensibilidad de recepción	-98dBm	-130dBm	-148dBm
Modulación	FSK	FSK;GFSK,MSK, LoRa TM	FSK/MSK/OOK/ FSK
Ganancia de la antena	1.5dBi	2dBi	1.5dBi-3dBi
Tipo de antena	IPEX	IPEX	Coaxial
Costo \$	39,20\$ sin IVA	45,23\$ sin IVA	28,70\$ sin IVA

Al culminar el análisis de las características técnicas en la tabla 29 se escogió el módulo Lilygo LoRa32 V1.3 Nodo, debido a que presta los requerimientos necesarios para el diseño del enlace inalámbrico, como el protocolo LoRa a una frecuencia de 902-928 MHz, y una potencia de transmisión de 20dBm y una sensibilidad de -130dBm. Esta tarjeta Lilygo LoRa32 V1.3. permite su configuración en modo emisor y en modo receptor lo que facilita la creación del enlace inalámbrica, además este dispositivo se encuentra disponible en el mercado nacional.

- **Selección de los elementos de mando y control**

Para la selección de los elementos que intervienen en los circuitos de mando y control es necesario la elaboración de los cálculos matemáticos correspondientes a los requerimientos eléctricos de cada uno de los componentes como son los actuadores.

- ✓ **Requerimientos de los calefactores**

Según la tabla 12 los requerimientos del extractor son:

Voltaje= 230 Vca

Frecuencia= 50-60Hz

Potencia eléctrica del motor (Pe)= 250W

Factor de potencia (Fp)= 0.79

Revoluciones por minuto del motor = 2800-3400 rpm

Para la selección del contactor hay que realizar el cálculo de la corriente nominal del motor para lo que se utiliza la siguiente ecuación:

$$In = \frac{Pe}{V * Fp} \quad \text{Ecuación 1}$$

Donde:

Pe= potencia eléctrica del motor en watts

V= Voltaje

Fp= factor de potencia

$$In = \frac{250W}{230V * 0.79}$$

$$In = 1.38 A$$

✓ **Requerimientos de los extractores (Motor WEG)**

Según la tabla 11 los requerimientos del extractor son:

Voltaje= 230 Vca

Frecuencia= 60Hz

Potencia mecánica del motor= 1.5hp

Factor de potencia (Fp)= 0,94

Eficiencia nominal(η)= 78%

Revoluciones por minuto del motor= 1710rpm

Para la selección del contactor hay que realizar el cálculo de la corriente nominal del motor para lo que se utiliza la ecuación 1:

$$In = \frac{Pe}{V * Fp}$$

Donde:

Pe= potencia eléctrica del motor en watts

V= Voltaje

Fp= factor de potencia

Encontrar la potencia eléctrica con la siguiente ecuación:

$$Pe = \frac{Pm \text{ en Watts}}{\eta} \quad \text{Ecuación 2}$$

Donde:

Pe=Potencia eléctrica

Pm= Potencia mecánica

η = Eficiencia nominal

$$Pe = \frac{1.5Hp * 745.7W}{0.78}$$

$$Pe = \frac{1118.55 W}{0.78}$$

$$Pe = 1434.039 W$$

Una vez realizado el cálculo de la potencia eléctrica requerida por el extractor se procede a realizar el cálculo de la corriente nominal con la ecuación 1.

$$In = \frac{1434.039W}{230V * 0,94}$$

$$In = 6.63 A$$

Una vez calculado la corriente nominal que es de 6.63A se seleccionó los siguientes equipos:

- Contactores
- Disyuntores
- Fuente de alimentación

- **Selección de los contactores**

Para un correcto funcionamiento de los extractores y los calentadores se procedió a seleccionar los contactores previa elaboración de los cálculos de la corriente nominal y nos dice que para seleccionar el contactor tiene que cumplir la siguiente ecuación:

$$I_c > I_n \quad \text{Ecuación 3}$$

Donde:

I_c = corriente del contactor

I_n = corriente nominal

Para los calefactores

$$I_c > I_n$$

$$I_c > 1.38A$$

$$I_c > 2A$$

Para nuestro caso elegimos un contactor de 5A, con una bobina de 220V/60Hz y con su respectivo contacto auxiliar normalmente abierto.

Para los extractores

$$I_c > I_n$$

$$I_c > 6.63A$$

$$I_c > 7A$$

Se seleccionó un contactor de 12A, con una bobina de 220V/60Hz y con su respectivo contacto auxiliar normalmente abierto.

- **Selección de los disyuntores**

De igual manera que los contactores la selección del disyuntor dependen de la corriente nominal de los motores y para seleccionar el contactor se tiene que cumplir la siguiente ecuación:

$$I_d > I_n \quad \text{Ecuación 4}$$

Donde:

I_d = corriente del disyuntor

I_n = corriente nominal

Para los calefactores

$$I_d > I_n$$

$$I_d > 1.38A$$

$$I_d > 2A$$

Se selecciona un contactor de 5A, de 220V/60Hz.

Para los extractores

$$I_d > I_n$$

$$I_d > 6.63A$$

$$I_d > 7A$$

Para nuestro caso elegimos un contactor de 12A, de 220V/60Hz.

- **Selección de la fuente**

Para la selección de la fuente hay que tener en cuenta la corriente, el voltaje y potencia que requieren los diferentes dispositivos, también la fuente debe soportar las condiciones de la red eléctrica que alimenta el sistema, por lo que conoce la red eléctrica principal es bifásica de 220V a 60Hz, además hay que tener en cuenta la corriente a la que trabajan los dispositivos eléctricos y electrónicos, siendo esta de 2A y 7A por lo que se selecciona una fuente conmutada con las especificaciones dicho anteriormente y las expuestas en los requerimientos técnicos del sistema. La tabla 30 indica las características de algunas de las fuentes existentes en el mercado nacional.

Tabla 30 Comparativa de fuentes de alimentación conmutadas

Características	Fuente conmutada S8FS (OMRON)	Fuente conmutada AC/DC 250W 24V 10A (WODE)	Fuente conmutada 24VDC 10A S-250-24 (Genérica)
Tipo de fuente	Conmutada	Conmutada	Conmutada
Voltaje de entrada	110/220Vca	110/220Vca	110/220Vca
Corriente de entrada	4.5 A / 2.5A	4.5 A / 2.5A	4.5 A / 2.5A
Frecuencia	50/60Hz	50/60Hz	50/60Hz
Potencia de salida	240W	15W a 350W	240W
Voltajes de salida	4Vdc 12Vdc 15Vdc 24Vdc	12Vdc 24Vdc	24Vdc
Corriente de salida	0~5A 0~45A 0~20A 0~10A	0~20A 0~10A	0~10A
Temperatura de operación	-10°C a 60°C	-10°C a 60°C	-10°C a 60°C
Costo en dólares	23,75\$ sin IVA	19,85\$ sin IVA	15,50\$ sin IVA

Elaborado por: El investigador

Al culminar con el análisis la fuente seleccionada es la S8FS (OMRON) por el cumplimiento de los requisitos técnicos del sistema como son: voltajes a la

salida de 0 a 12 Vdc, y una corriente de 0 a 20A, lo que permite tener una mayor escalabilidad en el desarrollo del sistema. Este dispositivo existe en mercado nacional a un precio accesible.

- **Selección del proveedor de un servidor de nube**

La selección de un servidor de nube permitirá utilizar una infraestructura tecnológica alojada en el internet, esta permitirá la ejecución de un sistema operativo el cual se comporta como un servidor, el que tendrá como función el almacenamiento de datos y la ejecución de órdenes a través de los diferentes lenguajes de programación los que permitirán la creación de una HMI, para lo que se ha desarrollado la tabla 31 con la comparativa entre proveedores de servidores de nube.

Tabla 31 Comparativa de los proveedores de servicios en la Nube

Característica	AWS (Amazon)	Microsoft Azure	Google Cloud Platform
Máquinas virtuales	Instancias	Máquinas virtuales	Máquinas virtuales
Imágenes	Imagen de la maquina propia de Amazon	Imagen VM	Imagen VM
Plantillas VM	Formación de la Nube AWS	Administrador de los recursos de Azure	No posee
Formatos de importación de VM admitidos	RAW, OVA, VMDK y VHD	VHD	VMDK y VHD
Acceso a máquinas virtuales	- Incluye claves SSH	- Incluye claves SSH	- Incluye claves SSH

	- Acceso a través de dirección IPV6 y Session Manager	- Acceso a través de Microsoft PowerShell	- funciones IAM
Escalamiento de las maquinas	De 448vCPUs y 24.576 GB de RAM	De 416vCPUs y 11 GB de RAM	De 448vCPUs y 24.576 GB de RAM

Elaborado por: El investigador

El servidor web seleccionado es el de AWS, debido a que permite el escalamiento de las maquinas con capacidades de De 448vCPUs y 24.576 GB de RAM, también permite la instanciación de máquinas virtuales con diferentes sistemas operativos.

- Selección de software de programación y procesamiento de datos
 - **Arduino IDE:** el entorno de desarrollo integrado (IDE) de Arduino permitirá la configuración de los diferentes elementos como los actuadores y los sensores, esta plataforma permite la descripción de todas las instrucciones a desarrollar a lo largo de la programación, además facilita la configuración de los elementos de comunicación inalámbrica para el sistema.
 - **Python:** es un lenguaje de programación de alto nivel, que permite la creación de aplicaciones dinámicas para el desarrollo web y el desarrollo de aplicaciones informáticas, además permite desarrollar códigos de programación que facilitan el tratamiento de datos e información.
 - **Node-Red:** es una herramienta de programación visual basada en flujo, para la comunicación de hardware y servicios de una manera sencilla, esta orienta a las IoT.
 - **INFLUXDB:** es un sistema que se encarga de la gestión de bases de datos para el almacenamiento de datos pertenecientes a series temporales, relacionadas a las IoT.

3.1.2.8 Diseño del sistema control y monitoreo inalámbrico

Para el diseño del sistema se ha procedido a dividir el sistema en 3 etapas, la etapa de adquisición de datos y control de actuadores, la comunicación inalámbrica, procesamiento de datos y la visualización de datos.

- **Adquisición de datos y control de actuadores**

Para la adquisición de los datos y el control de los actuadores se ha utilizado una tarjeta de desarrollo ESP8266-1 la cual cuenta con un chip ESP-01 el cual permite adquirir las señales de cada uno de los sensores y realizar la comunicación WiFi con la placa Lilygo LoRa32 la cual realizara la comunicación inalámbrica entre en el nodo y el Gateway y el procesamiento de los datos adquiridos.

- **Circuito de adquisición de datos**

- ✓ **Nodo sensor**

El nodo sensor está compuesto por el sensor dos estaciones sensoriales las que contienen los siguientes elementos: un sensor MQ-137 que detecta los niveles de amónico, un sensor DHT22 que mide la humedad relativa y la temperatura ambiente del galpón, una placa de desarrollo ESP8266-1 la que adquiere los datos de los sensores, para posteriormente realizar la comunicación WiFi entre la placa ESP8266-1 y la placa Lilygo LoRa32, para el procesamiento de la información y posteriormente realizar la comunicación LoRa entre el emisor y el receptor del sistema. En la siguiente figura 34 se indica el diagrama de bloques del nodo sensor y en la figura 36 se muestra el diagrama de conexión de estos elementos. El número total de nodos sensoriales de 2 los que se ubicaran en los lugares apropiados del galpón.

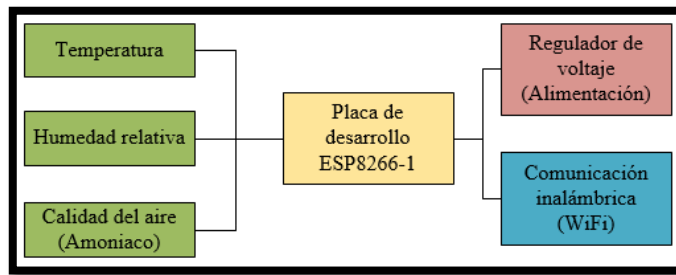


Figura 34 Diagrama de bloques del nodo sensor

Elaborado por: El investigador

La figura 35 muestra el diagrama de flujo del nodo sensorial, donde se explica el funcionamiento.

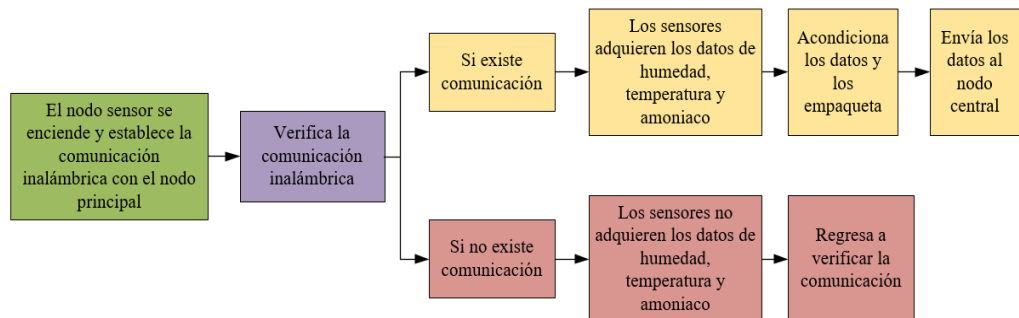


Figura 35 Diagrama de flujo del nodo sensorial

Elaborado por: El investigador

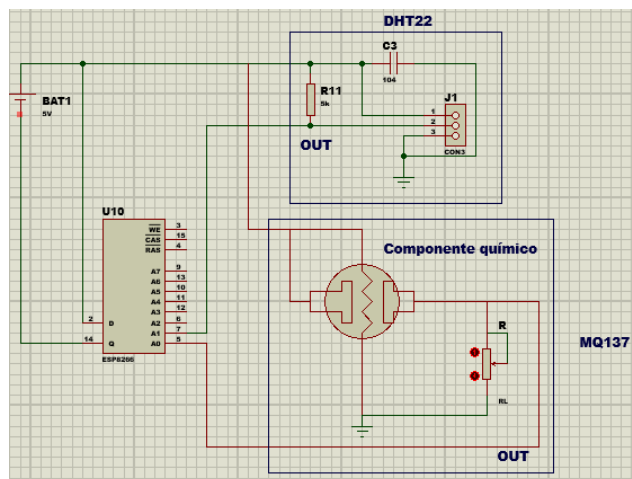


Figura 36 Diagrama de conexiones entre los sensores y la placa de ESP8266-1

Elaborado por: El investigador

✓ Programación

Para la programación del nodo sensor se sigue una serie de pasos como son:

✓ Seleccionar los pines a conectar los sensores.

Los pines seleccionados en la ESP8266 son: A0 (pin Analógico) para recibir los datos de tipo analógico es decir en función del voltaje del sensor de amoniac MQ-137, el pin GPIO 04 que puede ser configurado como salida o entrada de tipo digital, en nuestro caso será utilizado para adquirir los datos digitales de la humedad.

• Selección del protocolo de comunicación entre nodo sensor y nodo central

Para la sección del protocolo de comunicación entre los nodos hay que tener muy en cuenta el funcionamiento del modelo OSI (Interconexión de Sistemas Abiertos), ya que este modelo de comunicación estandarizado permite garantizar la interoperabilidad entre los dispositivos de comunicación.

✓ Modelo OSI

El modelo OSI es un modelo de referencia para los protocolos de comunicación de red, este se divide en siete capas donde cada una de estas es encargada de realizar una parte del proceso. Estas capas son las que se indica en la figura 37.

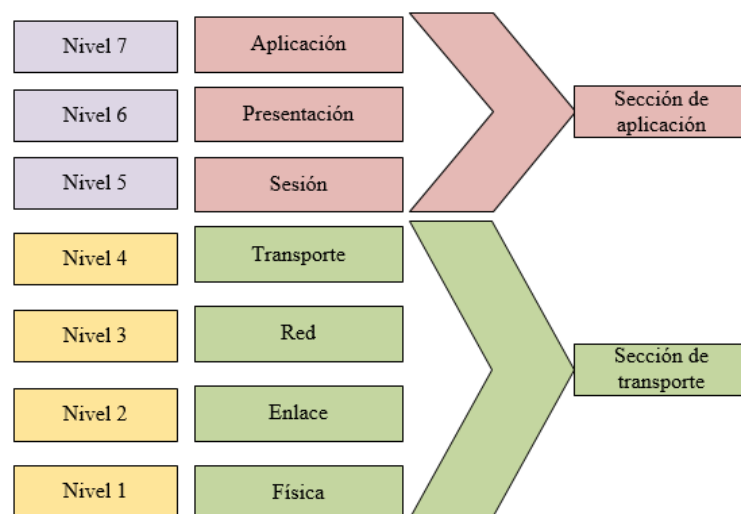


Figura 37 Capas del Modelo OSI

Elaborado por: El investigador

Capa física: se considera a la capa más baja del modelo OSI, está encargada de definir el medio o medios físicos por los que se va a realizar la conexión pueden ser alámbricos o inalámbricos, además regula los niveles de tensión a los que se va trabajar por ultimo garantiza la conexión mas no la fiabilidad de la conexión.

Capa de enlace: la capa de enlace es la encargada del direccionamiento físico de los datos mediante un enlace de comunicación entre el receptor y el emisor. En esta capa se controla que los datos se encuentren en tramas para tener una correcta transmisión e interpretación de la información entre nodos. La capa de enlace verifica que el envío de las tramas a través del enlace lleguen al receptor sin errores, para esto se utiliza el protocolo CRC (Chequeo de Redundancia Cíclica).

Capa de red: la capa red es la encargada de encaminar y direccionar los paquetes hasta entregarlos al destinatario, también en esta capa es donde se convierte las direcciones lógicas en dirección físicas.

Capa de transporte: es la encargada de controlar el flujo de datos entre el emisor y el receptor, también verifica el tamaño y los errores de los paquetes que llegan, para esto utilizan los protocolos TCP/IP, UDP y el SCTP.

Capa de sesión: es la encargada de establecer el enlace de comunicación o sesión entre el emisor y el receptor, sincronizar la comunicación.

Capa de presentación: es la encargada de convertir los paquetes en un lenguaje entendible para todos los elementos de la red.

Capa de aplicación: es la encargada de presentar la información en la interfaz pueden ser servidores web o aplicaciones.

Para la comunicación entre el nodo sensor y el nodo central se ha procedido a seleccionar el protocolo de comunicación UDP (Protocolo de Datagrama de Usuario), debido a que la comunicación entre nodos será inalámbrica y lo importante es la eficiencia de la comunicación en función del tiempo.

✓ **Protocolo UDP (Protocolo de Datagrama de Usuario).**

El protocolo de comunicación UDP se considera un protocolo sin conexión, debido que no necesita establecer una conexión con el receptor previo al envío de datagramas o paquetes, solamente necesita conocer el puerto de destino y el puerto de origen, por lo que se considera un protocolo poco fiable al no disponer de un control de congestión, flujo y errores, para garantizar la correcta transmisión de los paquetes se consulta con los protocolos de capas superiores. El formato de un paquete UDP se indica en la figura 38.

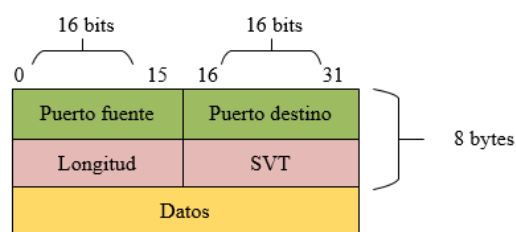


Figura 38 Formato de cabecera del protocolo UDP

Elaborado por: El investigador

La cabecera de un paquete en el protocolo UDP es muy sencilla ya que consta de 8bytes, en la que se encuentra detallada el puerto fuente de 16 bits, el puerto destino de 16bits, la longitud del paquete y el SVT que es la suma de verificación de errores del paquete. Los puertos tanto de destino y fuente son los más importantes de conocer en este protocolo. En la figura 39 se puede verificar cómo funciona el envío de paquetes a través del protocolo UDP, en el que se evidencia que solo se necesita una respuesta de conexión por parte de emisor, y el receptor realizara las peticiones de conexión las veces que sean necesarias sin importar el número de respuestas.

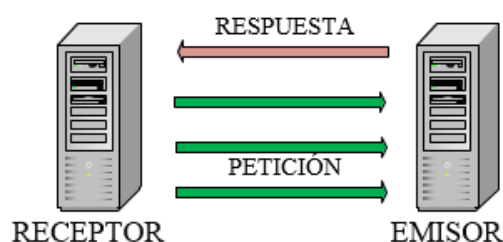


Figura 39 Envío de paquetes en el protocolo UDP

Elaborado por: El investigador

✓ **Configuración y manejo de datos recibidos de los sensores.**

Para la programación del nodo sensor se realizó un diagrama de flujo, en el que se representan las diferentes etapas del código esto se muestra en la figura 40.

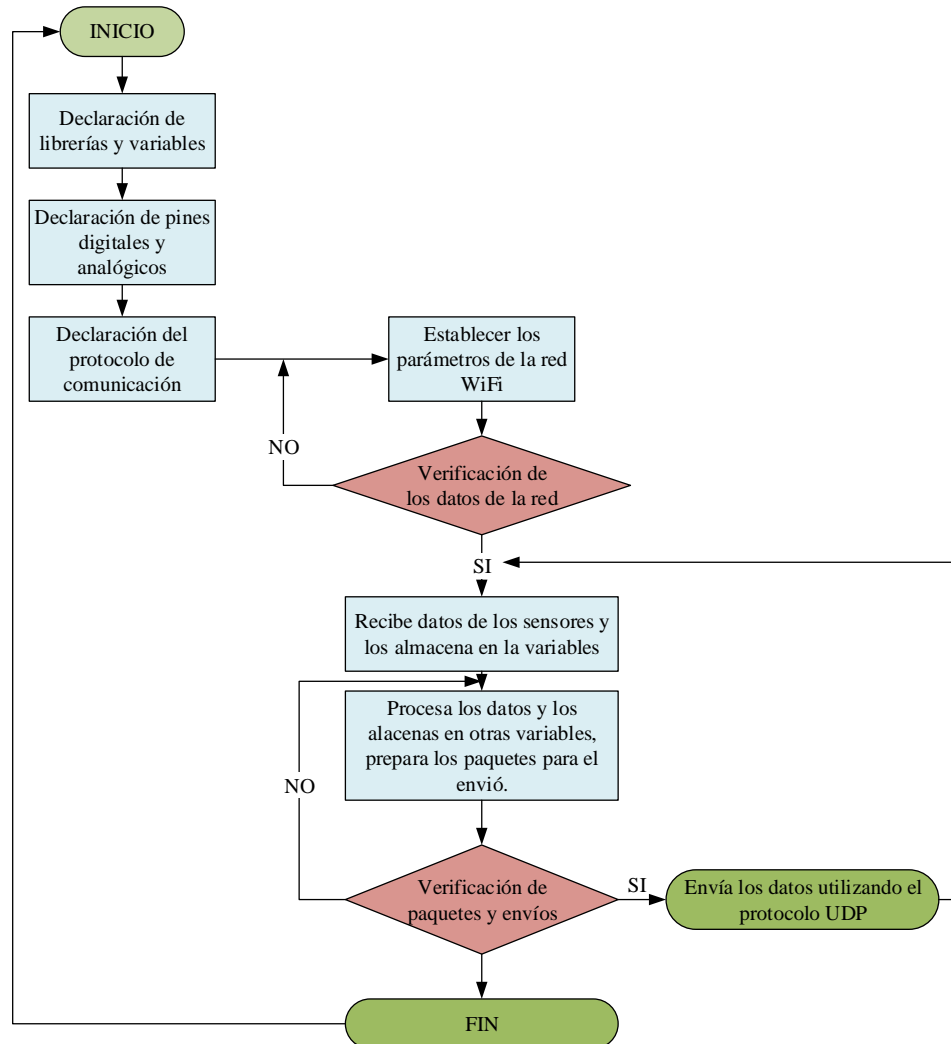


Figura 40 Diagrama de flujo del código de programación del nodo Sensor

Elaborado por: El investigador

El código de programación del nodo sensor se indica en el anexo 1.

✓ **Componentes electrónicos**

Los componentes electrónicos utilizados en el nodo sensorial son los que se indica en la tabla 32, en lo que tiene que ver con la cantidad de elementos están dos de cada componente porque el número total de nodos sensoriales es dos.

Tabla 32 Componentes electrónicos de los nodos sensoriales

Componentes electrónicos	Cantidad
ESP8266-1	2
MQ-137	2
DHT22	2
Regulador de voltaje LM2595	2

Elaborado por: El investigador

- **Circuito de mando y control**

Para el correcto accionamiento de los extractores y de los calefactores se procedió a realizar el diseño de un circuito de mando compuesto por contactores eléctricos, luces pilotos, las que reciben una señal de activación de la placa de desarrollo Lilygo LoRa32, la cual una vez procesado los datos de los sensores envía señales de activación o desactivación a los contactores, los que se encargan de activar los extractores y los calefactores según sea el requerimiento. A continuación, se indica el diagrama de bloques de las etapas que conforman los circuitos de mando y control en la figura 41.

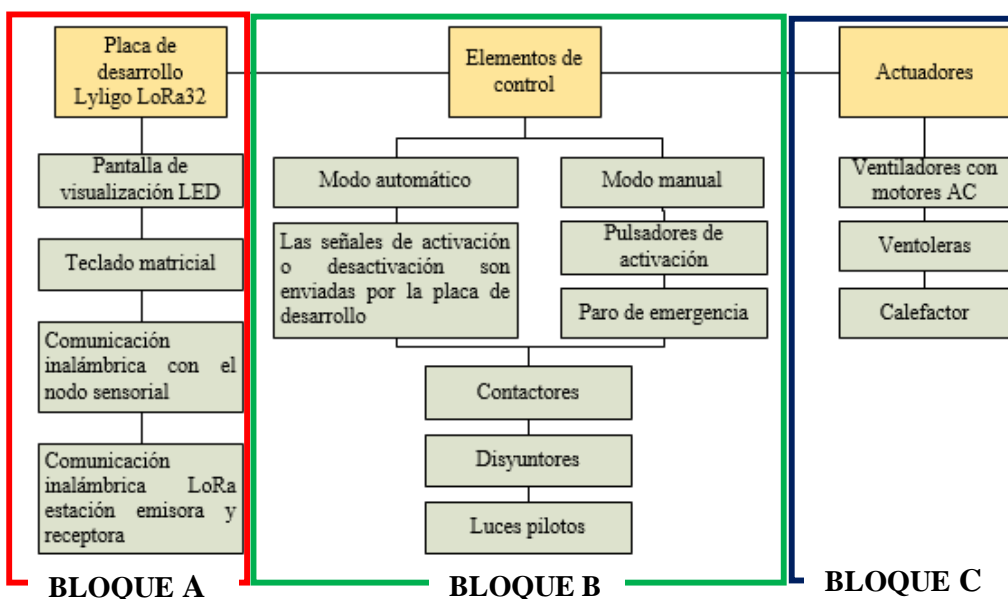


Figura 41 Diagrama de bloques de los circuitos de mando y control

Elaborado por: El investigador

Bloque A.- En el bloque A de la figura 41 se indica la parte central de los circuitos de mando y control, debido a que está conformado por la placa de desarrollo Lilygo LoRa32, en la cual se ha desarrollado todas las instrucciones del sistema, además recibe la señal del nodo sensorial mediante una comunicación inalámbrica WiFi y procesa dichos datos, para generar las instrucciones de activación o desactivación de los actuadores, la pantalla led permite la visualización de valores proporcionados por los sensores correspondientes a los parámetros técnicos. Por ultimo en este bloque se realiza la comunicación inalámbrica con tecnología LoRa con el servidor en internet, para posteriormente la elaboración de la HMI y así poder controlar de manera remota el sistema.

Bloque B.- En el bloque B de la figura 41 se indica los elementos que pertenecen al circuito de control, el que está constituido por contactores que permiten el accionamiento de los actuadores al recibir una señal de la placa de desarrollo, o ya sea de los pulsadores que conforman la parte manual del sistema, también existe un pulsador de emergencia el que se utilizara en situaciones de avería del sistema, los disyuntores existentes en el circuito permiten el paso de la energía eléctrica siempre y cuando no haya sobrecargas, además las luces piloto se activaran de manera simultánea a los actuadores. Cabe mencionar que el sistema puede ser manejado de manera manual o automática.

Bloque C.- En el bloque C de la figura 41 se indica la etapa de actuadores la cual está conformado por los extractores, los calefactores y las ventoleras, estos se activarán o desactivarán según las instrucciones enviadas desde el bloque hasta A el bloque C.

✓ **Circuito de control**

Los circuitos de control se encuentran divididos en dos partes la parte electrónica y la parte eléctrica como se detalla a continuación:

✓ **Circuito electrónico de control**

El circuito electrónico esta constituidos por las siguientes partes, entradas de datos de los selectores correspondientes al control manual de los actuadores

figura 42, las salidas analógicas las cuales constituyen señales de activación o desactivación de los contactores figura 43, estas salidas representan a los interruptores del circuito eléctrico de control correspondientes a los bloques B y C, la tarjeta de desarrollo ESP32 LoRa Lilygo TTGO encargada del procesamiento de datos y la comunicación LoRa figura 44, una pantalla LCD con su decodificador para la visualización de los datos figura 45, un teclado matricial 4x4 con un decodificador figura 46 y un regulador de voltaje LM2496 figura 47. En cada una de los circuitos detallados anteriormente se encuentras sus respectivas conexiones para mejor entendimiento en el anexo 2.

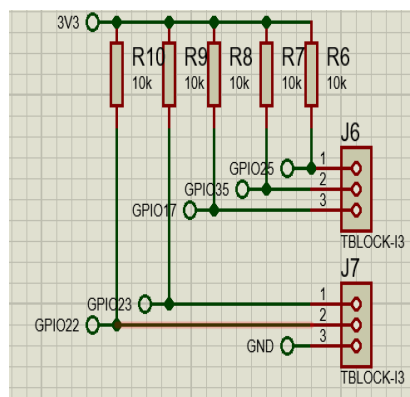


Figura 42 Entradas de datos

Elaborado por: El investigador

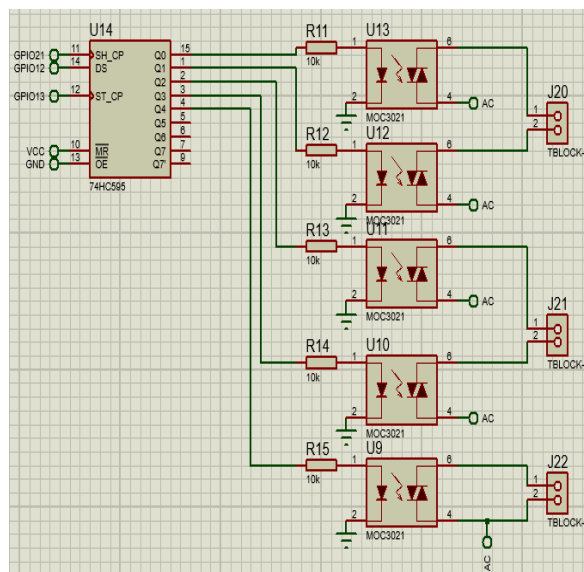
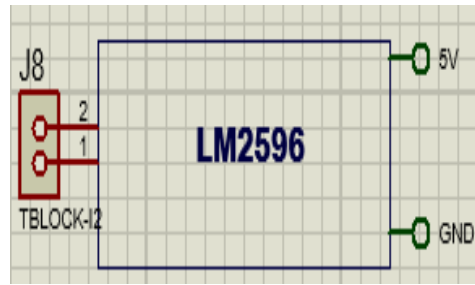


Figura 43 Salidas analógicas para conectar al circuito electrico

Elaborado por: El investigador



***Figura 47** Regulador de voltaje*

***Elaborado por:** El investigador*

✓ **Configuración del circuito de control electrónico**

Para la programación del nodo central se realizó un diagrama de flujo, en el que representa las diferentes etapas del código esto se muestra en la figura 48.

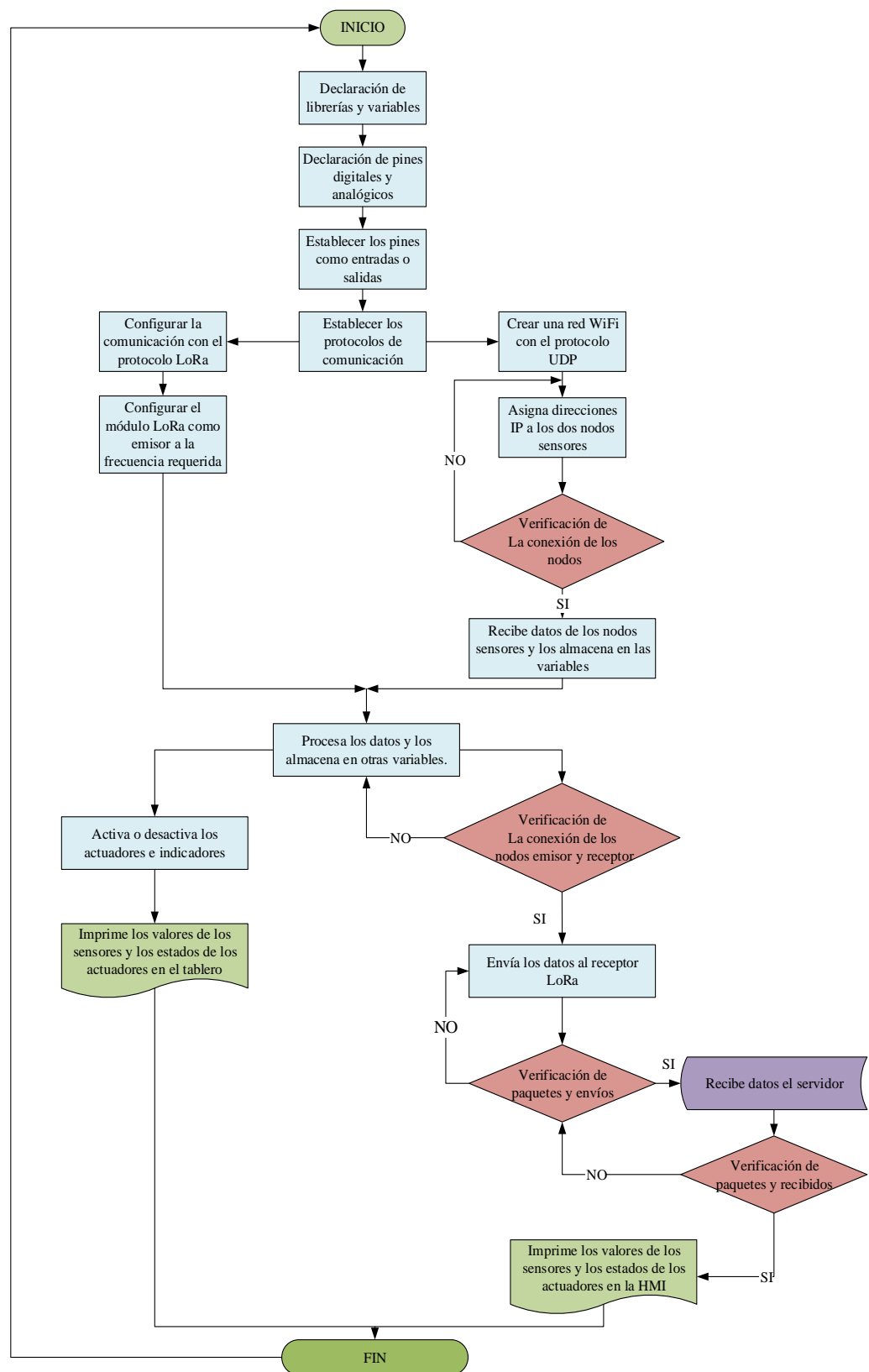


Figura 48 Diagrama de flujo de programación del nodo central

Elaborado por: El investigador

El código de programación del nodo central se lo presenta en el anexo 3, este código está basado en el diagrama de flujo de la figura 48.

- **Circuito eléctrico de control**

El circuito de control eléctrico está compuesto por los contactores, disyuntores, paro de emergencia y las luces piloto, en la figura 48 se puede visualizar el diagrama eléctrico de control el cual está representado con la simbología de los componentes, pero este sistema los elementos correspondientes a los pulsadores están solamente representados por sus símbolos, debido a que estos están ubicados en el circuito electrónico y a partir de estos se enviarán señales de activación o desactivación cumpliendo las funciones de activación. Mientras que los contactores darán paso a la activación de los actuadores una vez recibida las señales del circuito electrónico.

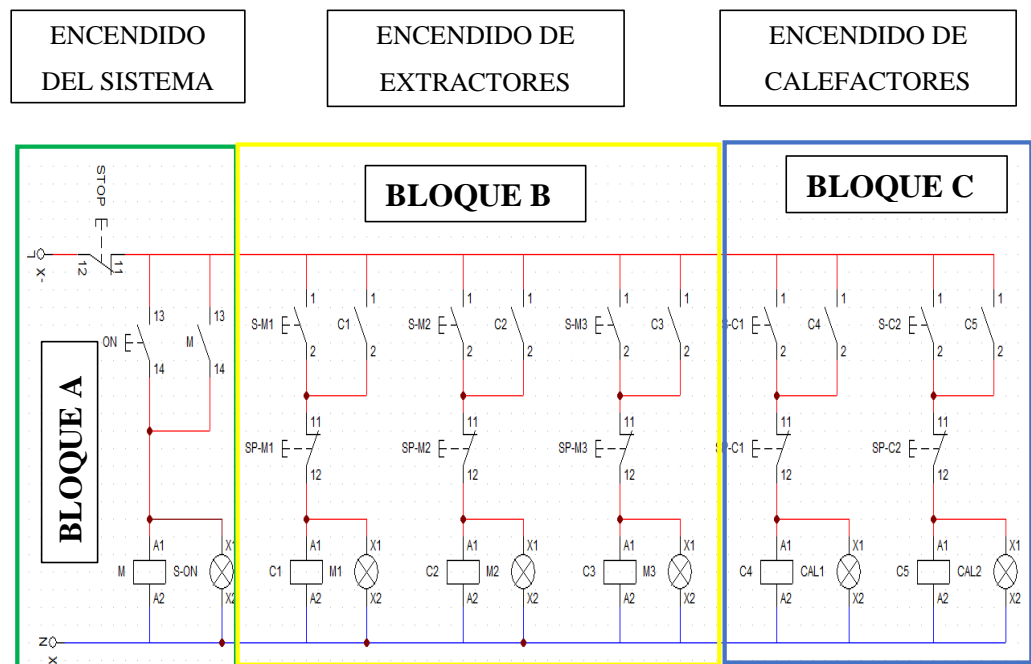


Figura 49 Circuito eléctrico de control

Elaborado por: El investigador

Bloque A.- En bloque A de la figura 49 se representa el encendido del sistema es decir la puerta en marcha, por lo que está constituido por un pulsador de paro de emergencia represando con el “STOP”. La señal de encendido está representada por el pulsador “ON” que recibe la señal de activación desde

circuito electrónico de control. También cuenta con un contactor que activará su bobina “M” al recibir la energía eléctrica una vez activado el botón “ON”, esto formará un circuito de enclavamiento con su contacto normalmente abierto el que mantendrá activado el sistema hasta recibir una nueva orden.

Bloque B.- En el bloque B de la figura 49 se representa la activación y desactivación de los extractores y ventoleras, por lo que está constituido por seis pulsadores los que representan las entradas de las señales enviadas por el circuito electrónico, el pulsador S-M1 representa a la señal de activación del extractor de baja velocidad y de la ventolera uno, SP-M1 representa a la señal de desactivación del extractor de baja velocidad de la ventolera uno, el pulsador S-M2 representa a la señal de activación del extractor de media velocidad y de la ventolera dos, SP-M2 representa a la señal de desactivación del extractor de media velocidad de la ventolera dos, por último el pulsador S-M3 representa a la señal de activación del extractor de alta velocidad, SP-M3 representa a la señal de desactivación del extractor de alta velocidad. Estos circuitos cumplen estas funciones con los tres contactores C1 representa al extractor de baja potencia y ventolera uno, C2 representa al extractor de media potencia y ventolera dos, por último C3 representa al extractor de alta potencia, cada uno de estos con sus respectivos contactos normalmente abiertos para formar el circuito de enclavamiento.

Bloque C.- En el bloque C de la figura 49 se representa la activación y desactivación de calefactores, por lo que está constituido por cuatro pulsadores los que representan las entradas de las señales enviadas por el circuito electrónico, el pulsador S-C1 representa a la señal de activación del calefactor uno, SP-C1 representa a la señal de desactivación del calefactor uno, el pulsador S-C2 representa a la señal de activación del calefactor dos, SP-C2 representa a la señal de desactivación del calefactor dos. Estos circuitos cumplen estas funciones con los dos contactores C4 representa al calefactor uno, C5 representa al calefactor dos, cada uno de estos con sus respectivos contactos normalmente abiertos para formar el circuito de enclavamiento.

- **Circuito de potencia**

El circuito de potencia pertenece a la activación y desactivación de los actuadores tanto de los extractores y los calefactores por lo que en la figura 50 se muestra las conexiones eléctricas de estos elementos, como se puede ver en la parte de los extractores está conformada por sus tres motores mientras que los calefactores por sus dos motores, estos se activaran o desactivaran según sea la señal envía por el circuito eléctrico de mando.

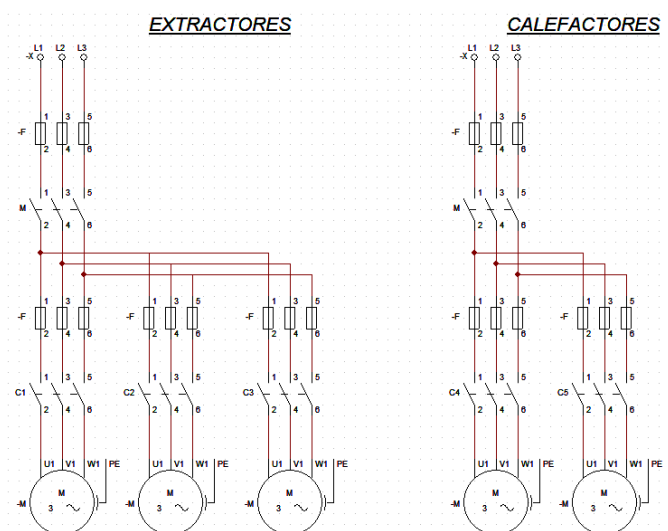


Figura 50 Circuito de potencia de los extractores y los calefactores

Elaborado por: El investigador

Las conexiones de los circuitos de mando y control de la parte eléctrica se indica en el anexo 4.

- ✓ **Componentes eléctricos y electrónicos**

Los componentes eléctricos y electrónicos utilizados en la etapa de mando y control son los que se indica la tabla 33.

Tabla 33 Componentes eléctricos y electrónicos del circuito de mando y control del sistema.

Componentes electrónicos	Cantidad
Lilygo LoRa32 (TTGO)	1

Regulador de voltaje LM2596	1
Pantalla LED	1
Circuito integrado PCF8574(expansor de entradas y salidas digitales para comunicación I2C)	1
Teclado matricial 4x4	1
Opto acoplador MOC3021	5
Decodificador MM74C922	1
Circuito integrado 74HC595 (registro de desplazamiento)	1
Componentes eléctricos	
Contactores	6
Pulsadores	2
Selectores	6
Paro de emergencia	1
Disyuntores	6
Luces pilotos verde	5
Luz piloto roja	1

Elaborado por: El investigador

Para el diseño de la comunicación inalámbrica lo primero que se realizó fue la identificación de los lugares donde se ubicarían los nodos, el nodo central que está en el galpón y el nodo receptor en la zona administrativa de la granja a 1100 metros de distancia. Esta representación se indica en la figura 51. Para conocer las coordenadas geográficas de la ubicación de los nodos verificar la tabla 34.

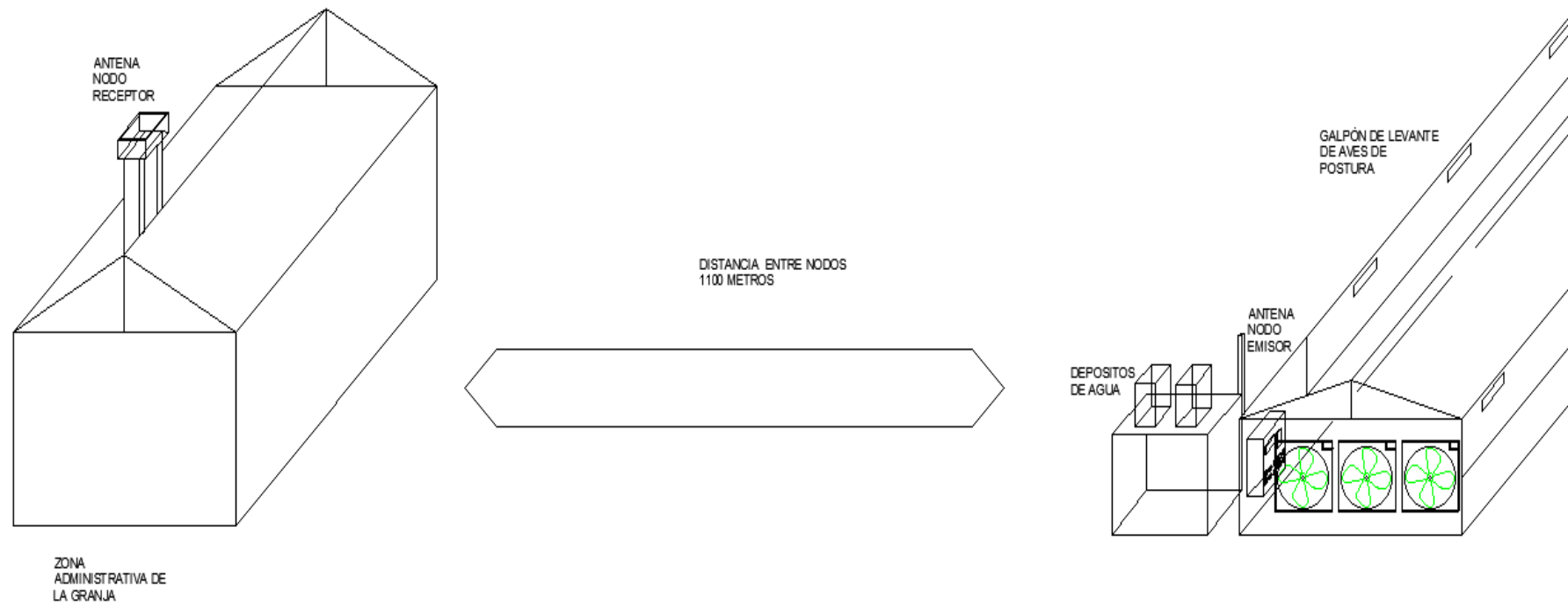




Figura 51 Ubicacion de la granja y de la oficina

Elaborado por: El investigador

- **Comunicación inalámbrica**

Para el diseño de la comunicación inalámbrica entre las estaciones emisora y receptora se procede a su ubicación en la granja, con la ayuda de la herramienta Google Earth se tomará las coordenadas de ubicación. En la tabla 34 se indica las coordenadas geográficas de las estaciones, además se puede observar la ubicación de las misma en las imágenes expuestas en la tabla 34 las que han sido adquiridas de Google Earth, tanto de la estación receptora como la emisora.

Tabla 34 Ubicación de las estaciones emisora y receptora

	Coordenadas geográficas	Fotografía
Estación emisora	Latitud: $1^{\circ}22'21.00''\text{S}$ Longitud: $78^{\circ}31'30.36''\text{O}$	
Estación receptora	Latitud: $1^{\circ}22'9.26''\text{S}$ Longitud: $78^{\circ}32'4.73''\text{O}$	

Elaborado por: El investigador

Para el diseño de la red inalámbrica se procedió a realizar los cálculos matemáticos correspondientes que son:

- Frecuencia necesaria para el enlace.
- Zona de Fresnel

- Perdidas en el espacio libre
- Margen de desvanecimiento
- Potencia de recepción disponible
- Margen umbral

✓ **Cálculos**

Frecuencia necesaria para el enlace (Comunicación LoRa)

$$f = \frac{f_{min} + f_{max}}{2} [52] \quad \text{Ecuación 5}$$

Donde:

Fmin= Frecuencia mínima (MHz)

Fmax= Frecuencia máxima (MHz)

$$f = \frac{(902 + 928)MHz}{2}$$

$$f = 915MHz$$

- **Zona de Fresnel**

$$r = 17.32 \sqrt{\frac{d}{4f}}$$

Donde:

d= distancia (km)

f= frecuencia (GHz)

$$r = 17.32 \sqrt{\frac{1.106Km}{4(0.915GHz)}}$$

$$r = 17.32 \sqrt{\frac{1.106Km}{3.660GHz}}$$

$$r = 17.32(0.5497)$$

$$r = 9.5210$$

$$r_1 = 0.6 * r$$

$$r_1 = 0.6 * 9.5210m$$

$$r_1 = 5.7126m$$

- **Perdidas en la trayectoria por el espacio libre (FSL)**

$$F_{SL}(dB) = 20\log(d) + 20\log(f) + 32.4 [52] \quad \text{Ecuación 6}$$

Donde:

d= distancia (km)

f= frecuencia (MHz)

$$F_{SL}(dB) = 20\log(1.106km) + 20\log(915MHz) + 32.4$$

$$F_{SL}(dB) = 0.8751 + 59.2284 + 32.4$$

$$F_{SL}(dB) = 92.5035dB$$

- **Margen de desvanecimiento (FM)**

$$FM = 30\log(d) + 10\log(6 * a * b * f) - 10\log(1 - R) - 70 [52] \quad \text{Ecuación 7}$$

Donde:

d= distancia (km)

f= frecuencia (GHz)

a= factor de rugosidad (4 sobre agua o terreno liso, 1 sobre terreno promedio, 0.25 sobre terreno muy áspero y montañoso)

b= factor para convertir la peor probabilidad mensual en una probabilidad anual (1 para una disponibilidad anual a la peor mensual, 0.5 para áreas calientes y húmedas, 0.25 para áreas continentales promedio, 0.125 para áreas muy secas y montañosas)

R = confiabilidad en tanto por uno 99.999% es decir 0.9999

$$FM = 30\log(1.106km) + 10\log(6 * 1 * 0.25 * 0.915GHz) - 10\log(1 - 0.9999) - 70$$

$$FM = 1.3126 + 3.8291 - (-40) - 70$$

$$FM = -27.3122 \text{ dB}$$

- **Potencia de recepción disponible**

$$P_{R_x} = P_{T_x} - AL_{T_x} + G_{T_x} - F_{SL} + G_{R_x} - AL_{R_x} [52] \quad \text{Ecuación 8}$$

Donde:

P_{T_x} = potencia de la antena transmisora.

AL_{T_x} = pérdidas por cables y conectores en el transmisor.

G_{T_x} = ganancia en el transmisor.

F_{SL} = pérdidas del espacio libre.

G_{R_x} = ganancia en el receptor.

AL_{R_x} = pérdidas por cables y conectores en el receptor

$$P_{R_x} = 20\text{dB} - 0.5\text{dB} + 2\text{dB} - 92.5035\text{dB} + 2\text{dB} - 0.5\text{dB}$$

$$P_{R_x} = -69.5035\text{dB}$$

- **Margen de umbral**

$$MU = P_{R_x} - S_r [52] \quad \text{Ecuación 9}$$

Donde:

S_r : sensibilidad del receptor

Nota: El margen de umbral entre el receptor y el emisor debe ser igual o mayor que el margen de desvanecimiento.

$$MU \geq FM$$

$$MU = 69.5035\text{dB} - (-138\text{dB})$$

$$MU = 68.4965$$

Por lo tanto, el margen umbral es:

$$MU \geq FM$$

$$68.4965 \geq -27.3122$$

Una vez realizado los cálculos matemáticos se procedió a la verificación de los resultados obtenidos matemáticamente con los resultados encontrados en las simulaciones en el software Radio Mobile, en el cual se realizó la simulación del enlace inalámbrico entre la estación emisora y la estación base de la siguiente manera.

Selección de mapa en el software Radio Mobile y la ubicación de las estaciones, para lo que acudimos a la tabla 34 para establecer las coordenadas de las estaciones. En la figura 52 se indica la ubicación de las estaciones en el software Radio Mobile.



Figura 52 Ubicación de las estaciones emisora y receptora en Radio Mobile

Elaborado por: El investigador

La figura 53 indica la ubicación de las estaciones en Google Earth.

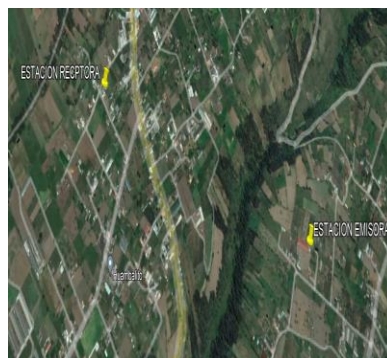


Figura 53 Ubicación de las estaciones emisora y receptora en Google Earth

Elaborado por: El investigador

✓ **Configuración de los diferentes parámetros en Radio Mobile.**

Para la configuración del sistema hay que tomar en cuenta los diferentes parámetros como son:

- Nombre del sistema = RED DAMIANCITO
- Potencia del transmisor = 20dBm
- Umbral del receptor = -138dBm
- Pérdidas de la línea = 0.5dB
- Tipo de antena = Omnidireccional
- Ganancia de la antena = 2dBi
- Altura de la antena = 15m
- Frecuencia máxima= 928MHz
- Frecuencia mínima= 902MHz

Para la adquisición de estos datos se ha extraído la información de la tabla 26 en la que se indica las características técnicas de los componentes electrónicos de las estaciones emisora y receptora, para lo que se procede a configurar en la opción “Propiedades de las redes” en el ítem “Sistemas” dichos datos como se puede observar en la figura 54 la configuración de las estaciones en el software Radio Mobile.

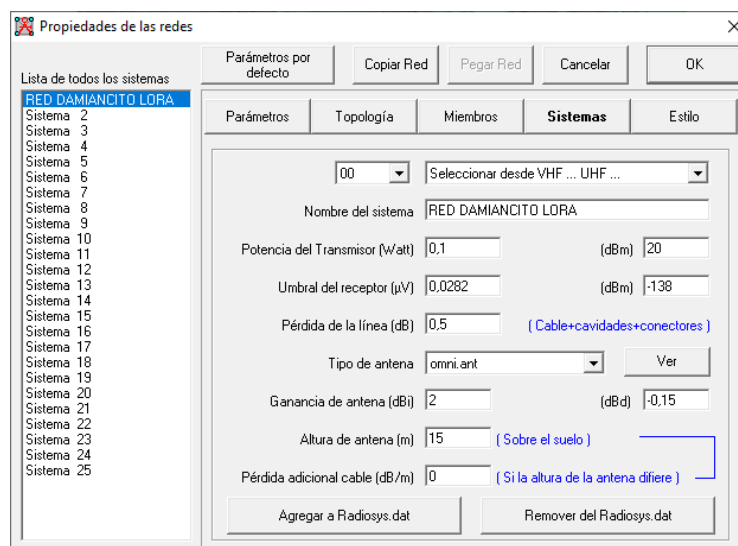


Figura 54 Configuración del sistema en el software Radio Mobile

Elaborado por: El investigador

Una vez configurada los parámetros de las estaciones se procede a establecer las frecuencias a las que se va a trabajar en la red, en nuestro caso la frecuencia mínima es de 902MHz y la frecuencia máxima es de 928Mhz, para lo que se procede a configurar en las opciones “propiedades de las redes” en el ítem “parámetros” dichas frecuencias como se indica en la figura 55.

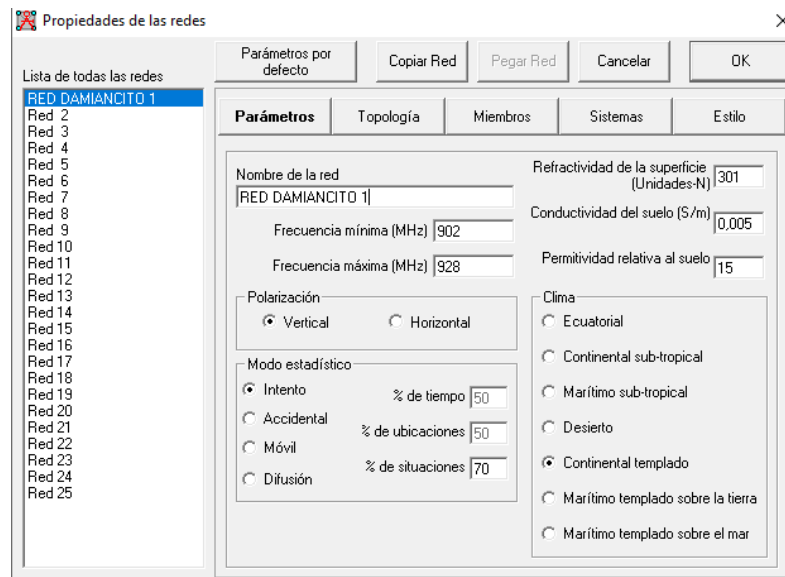


Figura 55 Configuración de la frecuencia a trabajar

Elaborado por: El investigador

Una vez configurado el sistema se procede a la simulación en la que se visualiza el comportamiento de la red, además se verifica los resultados de la simulación con los resultados calculados de manera matemática.

La figura 56 indica el primer resultado alcanzado en la simulación que corresponde al perfil de la red, la zona de Fresnel a lo largo de la trayectoria del enlace, además se observa una línea de color verde a lo largo del perfil la que significa que la señal será envía correctamente a lo largo del enlace sin obtener un número elevado de perdidas, también se indica en la parte inferior los datos configurados previa la simulación y en la parte superior se observa los datos numéricos más importantes como son: perdidas en el espacio libre, el valor del primera zona de Fresnel, potencias de transmisión y recepción y la distancia del enlace.

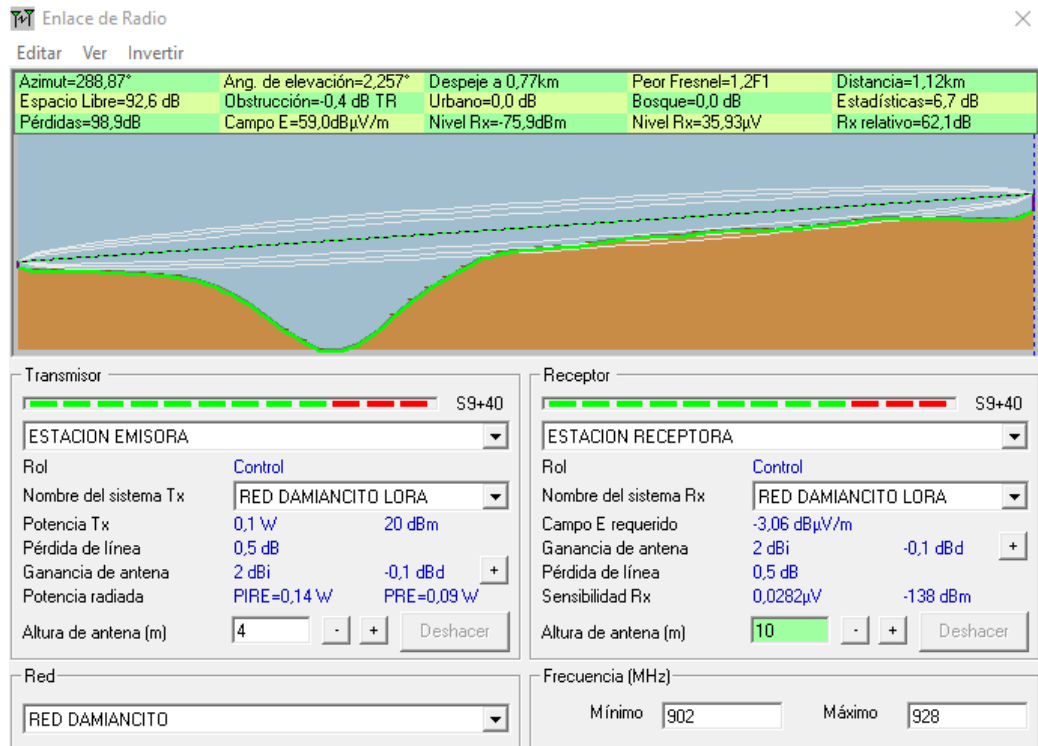


Figura 56 Perfil del enlace de comunicación inalámbrica entre la base emisora y receptora

Elaborado por: El investigador

• Configuración de los módulos LoRa

Para la configuración de los módulos LoRa hay que tomar en cuenta el plan de frecuencias de la ICM; por lo tanto, Ecuador que se encuentra en la región dos le corresponde la banda de frecuencias de 902Mhz a 928Mhz, la que está asignada para el uso industrial, esto se especifica en la tabla 9 para el proyecto se ha utilizado la frecuencia 915Mhz.

✓ Nodo emisor

Para la programación se ha desarrollado un diagrama de flujo en el que se representa el análisis del código del nodo emisor esto se indica en la figura 57. La configuración del nodo emisor está dentro del nodo central, debido a que este enviará al nodo receptor todos los datos adquiridos en el ambiente a controlar, de igual manera este recibirá las instrucciones enviadas desde la HMI remota. La programación se la detalla en el anexo 4.

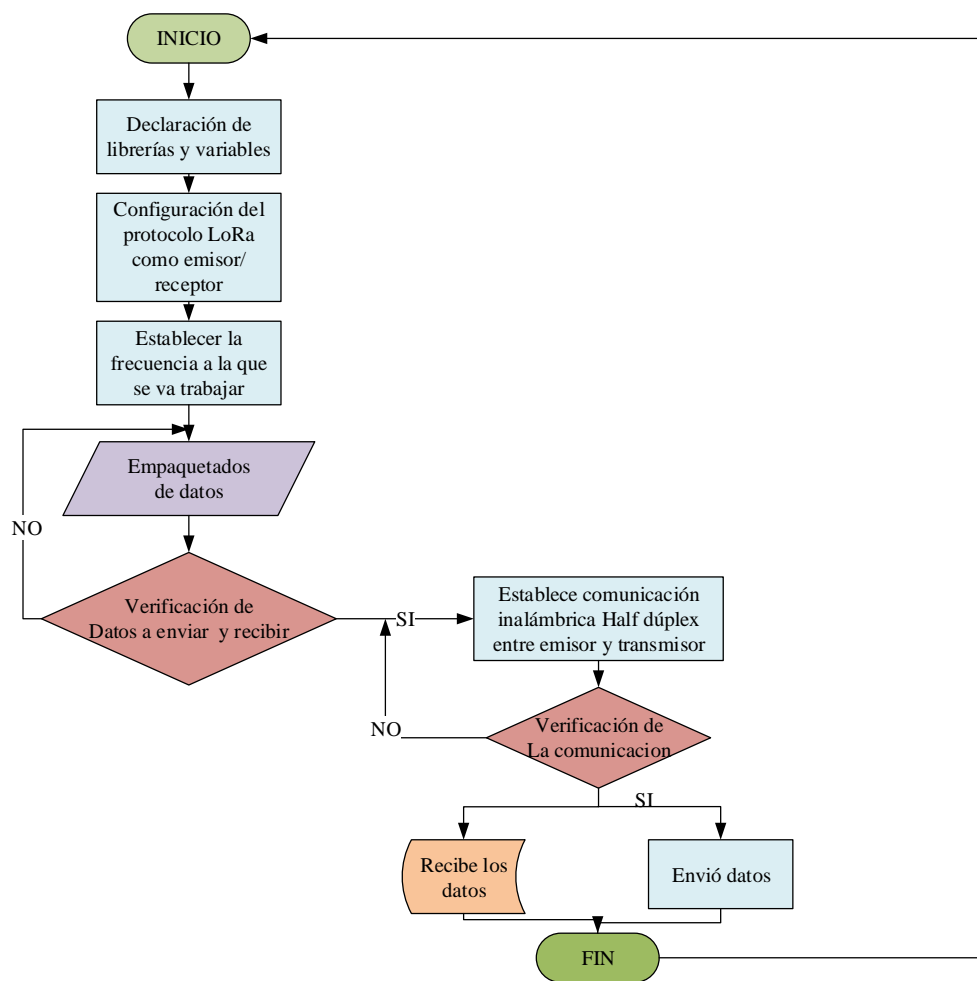


Figura 57 Diagrama de flujos del nodo emisor con configuración Lora

Elaborado por: El investigador

✓ **Nodo receptor**

Para el entendimiento de la programación se ha desarrollado un diagrama de flujo en el que se representa el análisis del código del nodo receptor esto se indica en la figura 58. En este nodo también se encuentra la programación para el envío de los datos al servidor web para su posterior publicación en la HMI, y de igual para esta la configuración del envío de datos perteneciente a las instrucciones de la HMI al nodo central. La programación se la detalla en el anexo 4.

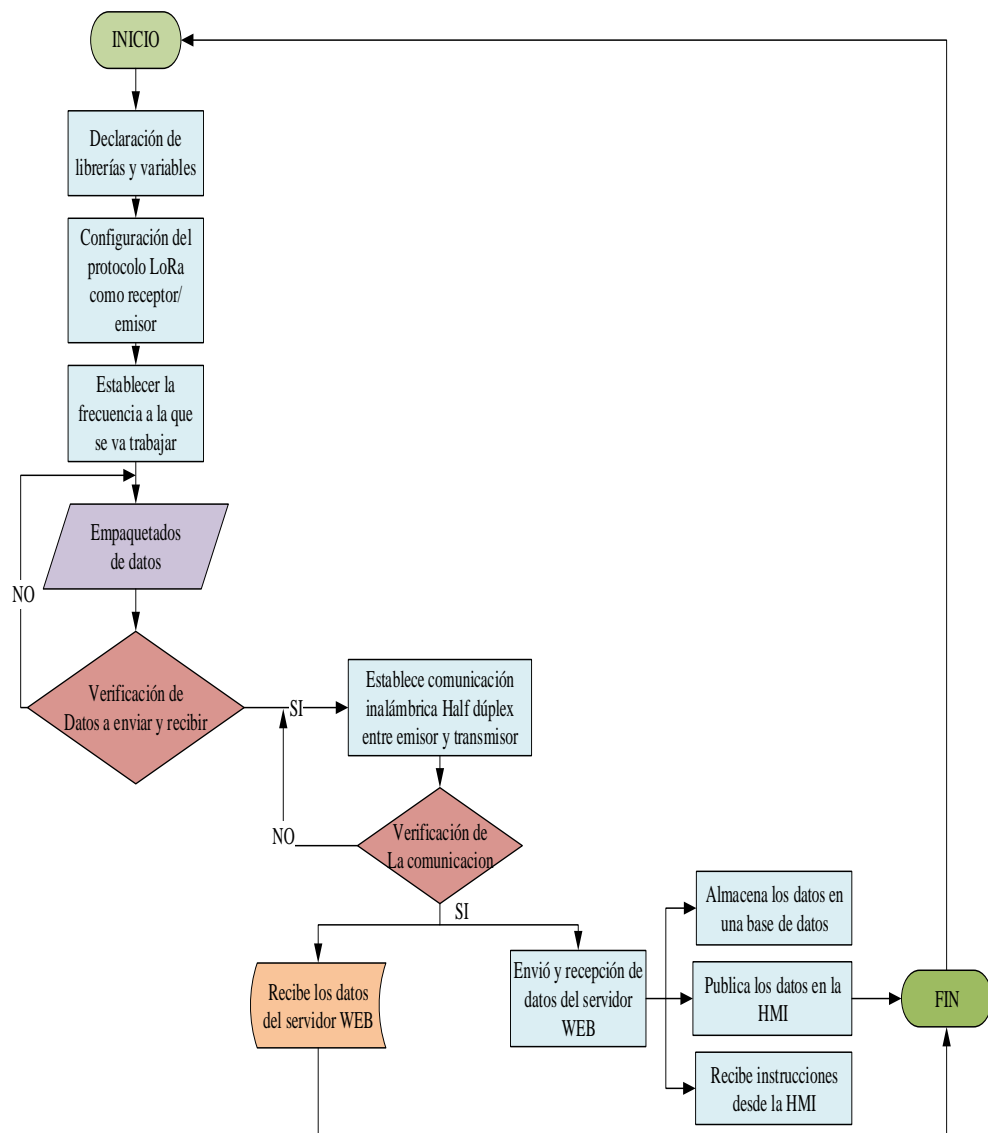


Figura 58 Diagrama de flujos del nodo receptor con configuración Lora

Elaborado por: El investigador

- **Configuración del servidor**

Para la configuración del servidor se ha procedido a utilizar el prestador de servicios en la nube AWS seleccionado anteriormente. Para esto se ha seguido una serie de pasos como son:

- ✓ Registrarse o crear una cuenta en la página oficial de AWS con sus datos personales solicitados.
- ✓ Verificar su correo electrónico.
- ✓ Crear una contraseña para el acceso a la cuenta AWS.

- ✓ Completar la información personal.
- ✓ Agregar método de pago de la cuenta.
- ✓ Verificar el número de teléfono ingresado en el punto de información personal.
- ✓ Elegir el plan a contratar en AWS.
- ✓ Esperar que se le active la cuenta.

Una vez realizado todos estos pasos se procedió a crear el servidor virtual en AWS para lo que siguió los siguientes pasos: Ingresar a la página principal de AWS y buscar en la paleta de servicios EC2 correspondiente a servidores virtuales como se indica en la figura 59 e ingresar a esta opción.



Figura 59 Selección de la opción EC2

Elaborado por: El investigador

Al ingresar en la opción EC2 se muestra un panel como el que se muestra en la figura 60, donde se coloca nombre del servidor en este caso “Avícola Damiancito”, y por último se selecciona el sistema operativo Ubuntu.

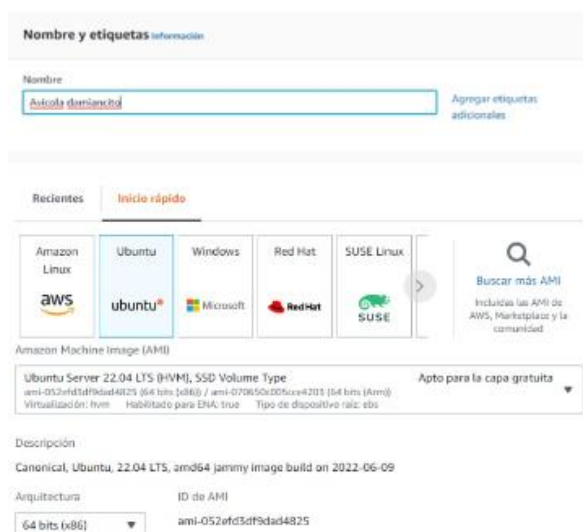


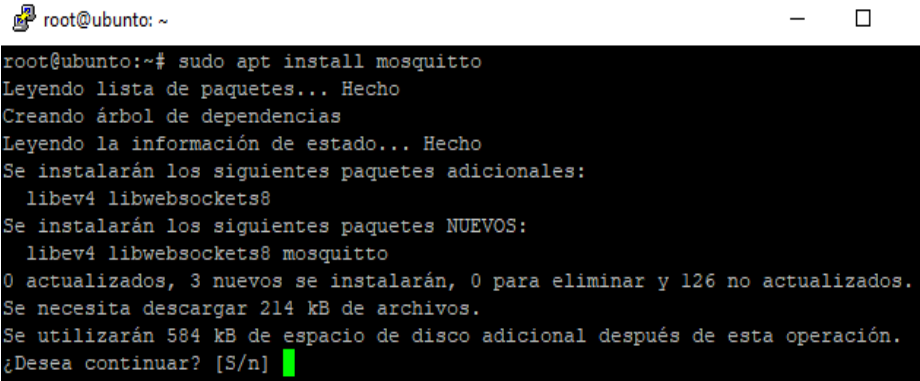
Figura 60 Selección del sistema operativo UBUNTU

Elaborado por: El investigador

Al seleccionar el sistema operativo automáticamente al presionar en finalizar se iniciará el servidor en la plataforma AWS, con esto se procede a la instalación de los demás componentes dentro del servidor.

Una vez completa la instalación del sistema operativo en el servidor AWS se procede a realizar las diferentes configuraciones e instalaciones. Lo primero que se hace es la instalación del protocolo de comunicación MQTT para eso se procede a realizar de la siguiente manera.

Para la instalación del protocolo MQTT se utiliza el siguiente comando: “*sudo apt install mosquitto*”, esto se lo hace una vez ingresado en modo root, ejecutado el comando saldrá lo que se indica en la figura 61.

A terminal window titled 'root@ubuntu: ~' with standard window controls. The terminal shows the execution of 'sudo apt install mosquitto'. The output includes: 'Leyendo lista de paquetes... Hecho', 'Creando árbol de dependencias', 'Leyendo la información de estado... Hecho', 'Se instalarán los siguientes paquetes adicionales: libev4 libwebsockets8', 'Se instalarán los siguientes paquetes NUEVOS: libev4 libwebsockets8 mosquitto', '0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 126 no actualizados.', 'Se necesita descargar 214 kB de archivos.', 'Se utilizarán 584 kB de espacio de disco adicional después de esta operación.', and a prompt '¿Desea continuar? [S/n]' with a green cursor.

```
root@ubuntu:~# sudo apt install mosquitto
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libev4 libwebsockets8
Se instalarán los siguientes paquetes NUEVOS:
  libev4 libwebsockets8 mosquitto
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 126 no actualizados.
Se necesita descargar 214 kB de archivos.
Se utilizarán 584 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Figura 61 Instalación dl protocolo MQTT

Elaborado por: El investigador

Una vez instalado se verifica si se encuentra activado el protocolo con el siguiente comando “*sudo systemctl status mosquitto*”, como se muestra la figura 62 se encuentra activado el servicio lo que nos indica que podemos hacer uso de este protocolo. En las letras de color verde se puede verificar que se encuentra activado. En caso de estar desactivado ingresa el comando “*sudo systemctl start mosquitto*”.

```
root@ubuntu: ~  
Preparando para desempaquetar .../mosquitto_1.4.15-2ubuntu0.18.04.3_amd64.deb ...  
Desempaquetando mosquitto (1.4.15-2ubuntu0.18.04.3) ...  
Configurando libev4 (1:4.22-1) ...  
Configurando libwebsockets8:amd64 (2.0.3-3build1) ...  
Configurando mosquitto (1.4.15-2ubuntu0.18.04.3) ...  
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...  
Procesando disparadores para ureadahead (0.100.0-21) ...  
Procesando disparadores para libc-bin (2.27-3ubuntu1.4) ...  
Procesando disparadores para systemd (237-3ubuntu10.52) ...  
root@ubuntu:~# sudo systemctl status mosquitto  
● mosquitto.service - LSB: mosquitto MQTT v3.1 message broker  
   Loaded: loaded (/etc/init.d/mosquitto; generated)  
   Active: active (running) since Thu 2022-06-23 03:03:28 UTC; 1min 52s ago  
     Docs: man:systemd-sysv-generator(8)  
    Tasks: 1 (limit: 1104)  
   CGroup: /system.slice/mosquitto.service  
           └─2228 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf  
  
jun 23 03:03:27 ubuntu systemd[1]: Starting LSB: mosquitto MQTT v3.1 message bro  
jun 23 03:03:28 ubuntu mosquitto[2214]: * Starting network daemon: mosquitto  
jun 23 03:03:28 ubuntu mosquitto[2214]: ...done.  
jun 23 03:03:28 ubuntu systemd[1]: Started LSB: mosquitto MQTT v3.1 message brok  
lines 1-12/12 (END)
```

Figura 62 Verificación del estado del protocolo MQTT

Elaborado por: El investigador

Por último, se realizó la configuración del puerto de la comunicación con el bróker, esto se realiza ingresando a la siguiente dirección “*sudo nano /etc/mosquitto/mosquitto.conf*” y colocando el puerto 8094 como se indica en la figura 63.

```
## External WebSocket/SSL listener for MQTT Protocol  
  
## listener.wss.$name is the IP address and port that the MQTT/WebSocket/SSL  
## listener will bind.  
##  
## Value: IP:Port | Port  
##  
## Examples: 8084, 127.0.0.1:8084, :::1:8084  
listener.wss.external = 8094  
  
## The path of WebSocket MQTT endpoint
```

Figura 63 Configuración del puerto externo en el protocolo mqtt

Una vez instalado el servicio de mosquito se procede a instalar el bróker llamado EMQX, el que facilita la configuración del publicador y subscriptor en MQTT, para esta instalación se usó los siguientes comandos: “*sudo apt install curl vim unzip*” para la habilitación de paquetes, luego se instaló el bróker EMQX desde el repositorio con el comando “*curl https://repos.emqx.io/install_emqx.sh | sudo bash*” como se indica en la figura 64, y por último se inicializa el bróker EMQX con el comando “*sudo systemctl*

`start emqx && sudo systemctl enable emqx`” con este comando se iniciara el bróker cada vez que se encienda el servidor.

```

root@ubuntu: ~
root@ubuntu:~# curl https://repos.emqx.io/install_emqx.sh | sudo bash
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left    Speed
100 1827 100 1827    0     0  468      0  0:00:03  0:00:03 --:--:--  468
Obj:1 http://ec.archive.ubuntu.com/ubuntu bionic InRelease
Des:2 http://ec.archive.ubuntu.com/ubuntu bionic-updates InRelease [88,7 kB]
Des:3 http://ec.archive.ubuntu.com/ubuntu bionic-backports InRelease [74,6 kB]
Des:4 http://ec.archive.ubuntu.com/ubuntu bionic-security InRelease [88,7 kB]
Descargados 252 kB en 19s (13,5 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se pueden actualizar 120 paquetes. Ejecute «apt list --upgradable» para verlos.
Leyendo lista de paquetes... Hecho

```

Figura 64 Instalación del Bróker EMQX

Elaborado por: El investigador

Una vez instalado inicializamos el bróker con el comando “`sudo systemctl start emqx`” o “`emqx start`”, esto se ejecuta dentro del directorio bin como indica en la figura 65.

```

root@ubuntu: /bin
root@ubuntu:/bin# emqx start

```

Figura 65 Inicio del bróker EMQX

Elaborado por: El investigador

Al finalizar con la instalación se procede a acceder a la cuenta del EMQX en la cual se configura los publicadores y suscriptores, para esto solo se utiliza la dirección IP del servidor y el nombre de la variable que enviará los datos, también la variable que recibirá el dato como se indica en la figura 66.

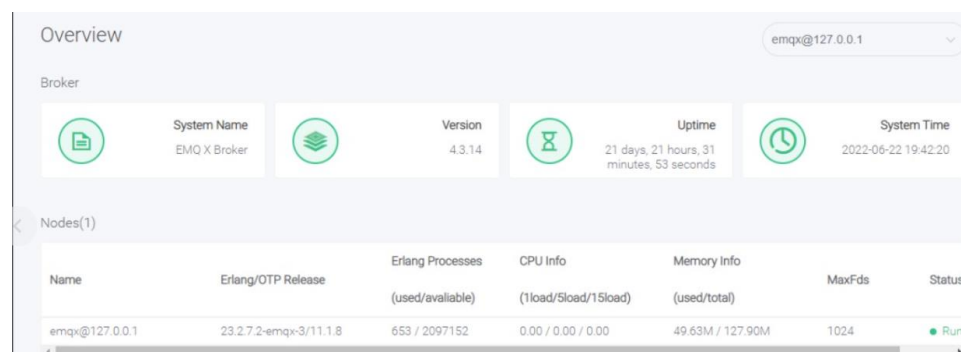
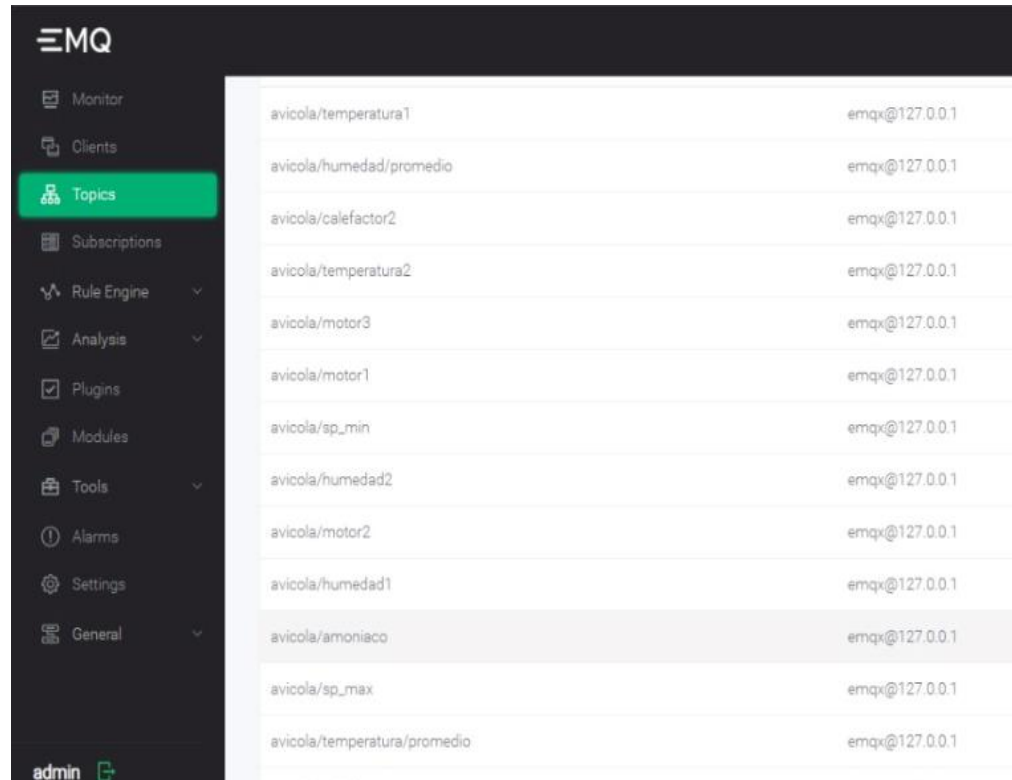


Figura 66 Configuración de la dirección IP 127.0.0.1

Elaborado por: El investigador

Una vez configurado la dirección IP del servidor se procede a declarar los tópicos a utilizar, para esto se utiliza la dirección IP 127.0.0.0 anteponiendo emqx, como muestra la figura 67.

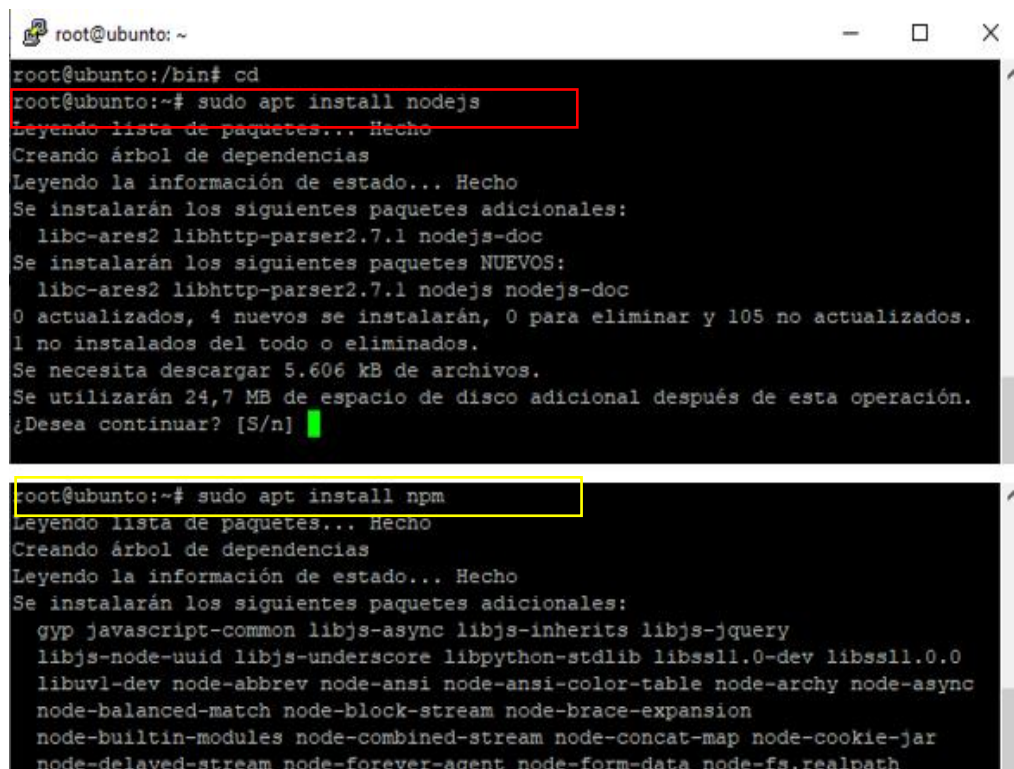


Topic	Client
avicola/temperatura1	emqx@127.0.0.1
avicola/humedad/promedio	emqx@127.0.0.1
avicola/calefactor2	emqx@127.0.0.1
avicola/temperatura2	emqx@127.0.0.1
avicola/motor3	emqx@127.0.0.1
avicola/motor1	emqx@127.0.0.1
avicola/sp_min	emqx@127.0.0.1
avicola/humedad2	emqx@127.0.0.1
avicola/motor2	emqx@127.0.0.1
avicola/humedad1	emqx@127.0.0.1
avicola/amoniaco	emqx@127.0.0.1
avicola/sp_max	emqx@127.0.0.1
avicola/temperatura/promedio	emqx@127.0.0.1

Figura 67 Declaración de los tópicos que utilizaran el protocolo MQTT

Elaborado por: El investigador

Con la obtención de los tópicos en el servidor se procede a la elaboración de la HMI, para lo que se debe instalar los diferentes softwares en el servidor, estos son Node.js y npm; por lo tanto, estos nos permiten la instalación del archivo node-red el que nos permitirá el diseño de la HMI. Para esto se ejecutan los siguientes comandos “*sudo apt install nodejs*”, para instalar los archivos Node.js como se en rece urdo rojo de la figura 68, “*sudo apt install npm*” para instalar los archivos npm, como se indica en el recuerdo amarillo de la figura 68



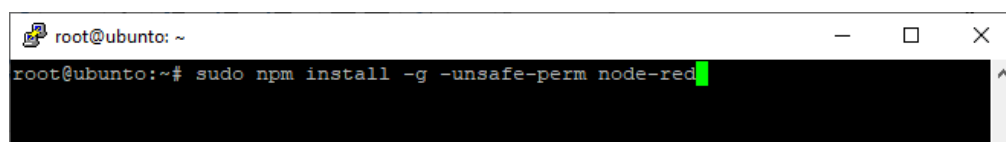
```
root@ubuntu:~/bin# cd
root@ubuntu:~# sudo apt install nodejs
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libc-ares2 libhttp-parser2.7.1 nodejs-doc
Se instalarán los siguientes paquetes NUEVOS:
  libc-ares2 libhttp-parser2.7.1 nodejs nodejs-doc
0 actualizados, 4 nuevos se instalarán, 0 para eliminar y 105 no actualizados.
1 no instalados del todo o eliminados.
Se necesita descargar 5.606 kB de archivos.
Se utilizarán 24,7 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

```
root@ubuntu:~# sudo apt install npm
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  gyp javascript-common libjs-async libjs-inherits libjs-jquery
  libjs-node-uuid libjs-underscore libpython-stdlib libssl1.0-dev libssl1.0.0
  libuv1-dev node-abbrev node-ansi node-ansi-color-table node-archy node-async
  node-balanced-match node-block-stream node-brace-expansion
  node-builtin-modules node-combined-stream node-concat-map node-cookie-jar
  node-delayed-stream node-forever-agent node-form-data node-fs.realpath
```

Figura 68 Instalaciones de requisitos para Node Red

Elaborado por: El investigador

Con la instalación de los prerrequisitos se procede a instalar Node Red con el comando “*sudo npm install -g -unsafe-perm node-red*” como indica la figura 69.



```
root@ubuntu:~# sudo npm install -g -unsafe-perm node-red
```

Figura 69 Instalación de NODE RED

Elaborado por: El investigador

Al culminar con las instalaciones de todas las herramientas a utilizar en la programación de la HMI se puede proceder con el desarrollo de la HMI. Verificando que Node-RED se encuentre inicializado ingresando al directorio */var/log/Node.js*, donde mostrara lo que se indica en la figura 70.

```
Stop Node-RED ✓
Remove old version of Node-RED ✓
Remove old version of Node.js ✓
Install Node.js LTS ✓ Node v12.18.2 Npm 6.14.6
Clean npm cache ✓
Install Node-RED core ✓ 1.1.1
Move global nodes to local -
Install extra Pi nodes ✓
Npm rebuild existing nodes -
Add shortcut commands ✓
Update systemd script ✓

Any errors will be logged to /var/log/nodered-install.log
All done.
You can now start Node-RED with the command node-red-start
or using the icon under Menu / Programming / Node-RED
Then point your browser to localhost:1880 or http://{your_pi_ip-address}
:1880

Started Wed 8 Jul 23:52:45 BST 2020 - Finished Thu 9 Jul 00:00:10 BS
T 2020
```

Figura 70 Verificación que Node-RED este inicializado

Elaborado por: El investigador

Programación de la HMI

Para la programación de la HMI se utiliza Node-RED en la cual se programa los diferentes componentes que se utilizaran para la visualización de los parámetros técnicos y de los actuadores.

Para la visualización de los parámetros técnicos se utiliza los diferentes componentes de programación, en el sistema se utiliza el protocolo de comunicación mqtt; por ende, se utiliza el nodo de comunicación “*mqtt in*” para recibir los datos desde el servidor, en este nodo se configurara la dirección del servidor que es “*http:// http://3.235.79.216/*” y la variable de la que se recibe los datos “*avicola/temperatura1*”, en lo que respecta a la visualización se utiliza el nodo indicador “*gauge*”, por ultimo para almacenar los datos en la base de datos utiliza el nodo “*influxdb in*” en el que se configura la dirección del mismo “*http:// http://3.235.79.216/*”, y la variable que envía los datos “*avicola/temperatura1*”. En la figura 71 se indica la programación de la adquisición de los datos los nodos sensores, esta información se visualizará en la HMI.

Para todos los parámetros a controlar se realizó esta programación, estos parámetros son: temperatura, humedad y amoniaco. La programación completa de cada uno de los parámetros técnicos se muestra en anexo 6.

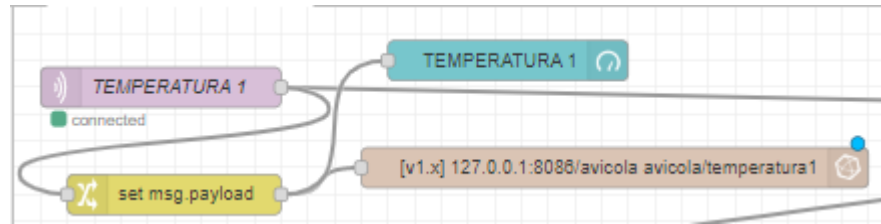


Figura 71 Configuración de la adquisición de datos de los nodos sensores y publicación en la HMI

Elaborado por: El investigador

Para la programación de los actuadores se la realiza de la siguiente manera: el comando “mqtt-out” para él envío de datos desde la HMI: por ende, la configuración será la dirección del servidor “*http:// http://3.235.79.216/*” y el nombre de la variable como es “*avicola/motor1*”, además se realiza la programación del nodo “button” en el que se configura los siguientes datos: “*http:// http://3.235.79.216/*” dirección del servidor y en nombre de las variables “*avicola/actuadores*”, en la que se enviara la instrucción y su respectivo indicador. En la figura 72 se indica la programación de los nodos actuadores correspondientes a motor 1 (extractor 1), motor 2 (extractor 2), motor 3 (extractor 3), calefactor 1 y calefactor 2, en caso de los extractores también se activa las ventores. La programación completa de cada uno de los actuadores se muestra en anexo 6.

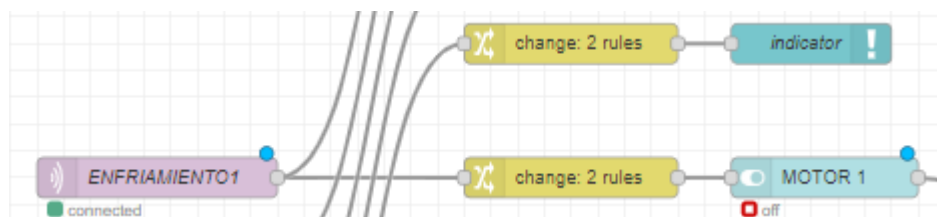


Figura 72 Configuración de las señales de activación desde la HMI a los actuadores

Elaborado por: El investigador

Para la configuración del botón de activación o desactivación del modo automático o modo manual, se ha realizado con los siguientes comandos: nodo “mqtt in” para recibir la señal desde el nodo central en el que se refleja el modo de opracion, en este nodo se configura la dirección del servidor “*http:// http://3.235.79.216/*” y el nombre de la variable “*avicola/on/off*”, se configura el nodo “button” para enviar la señal desde la HMI y se pone el nombre de este

nodo “ON-OFF” y para el envío de la señal desde la HMI se usa el nodo “*mqtt out*” con los mismo datos del nodo “*mqtt in*”, además la colocación del indicador. En la figura 73 se indica esta programación.

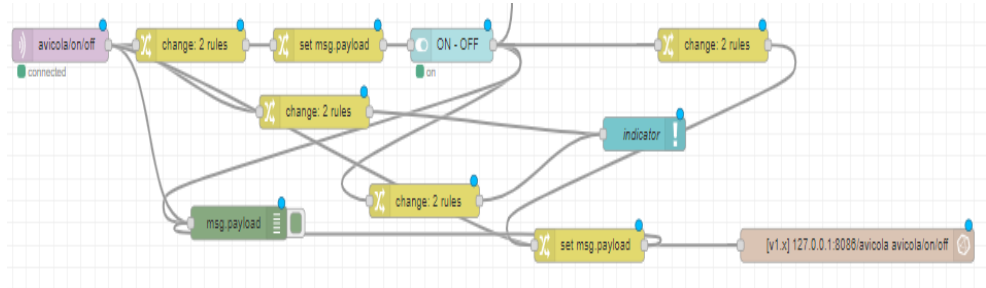


Figura 73 Programación del botón de activación o desactivación del modo automático desde la HMI

Elaborado por: El investigador

Para la programación del ingreso de temperaturas mínima y máxima desde el HMI se procede a la configuración de los siguientes nodos nodo “*mqtt in*” para recibir los datos desde el nodo central y el nodo “*mqtt out*” para enviar los datos desde el HMI, para la lo que se establece los parámetros de dirección del servidor “*http:// http://3.235.79.216/*” y el nombre de las variables “*avicola/sp_min*” para el estado alto y “*avicola/sp_max*” para el estado alto en lo que respecta a recibir datos y para enviar datos “*avicola/temperatura/alta*” para estado alto y “*avicola/temperatura/baja*” para el estado bajo. En la figura 74 se indica la programación de estos botones.

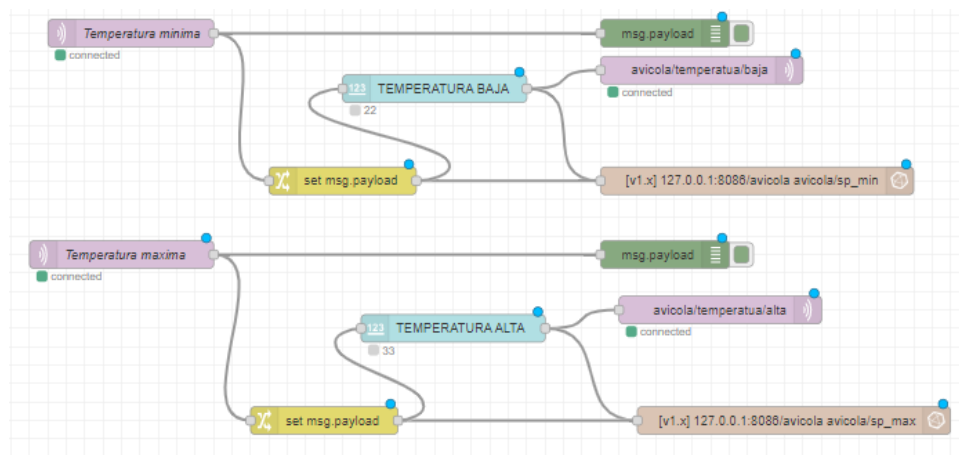


Figura 74 Programación de las entradas de los datos de temperatura máxima y mínima desde la HMI

Elaborado por: El investigador

Para la programación del ingreso del porcentaje de luminosidad desde el HMI se procede a la configuración de los siguientes nodos “mqtt in” para recibir los datos desde el nodo central y el nodo “mqtt out” para enviar los datos desde el HMI, para lo que se establece los parámetros de dirección del servidor “http:// http://3.235.79.216/” y el nombre de las variables “avicola/dimmer” en lo que respecta a recibir datos y para enviar datos “avicola/luminosidad”, para la variación del porcentaje de luminosidad de utiliza el nodo “silder”, este permite variar el valor desde 0 a 100 por ciento y enviar este valor al nodo central. En la figura 75 se indica a la programación del Dimmer.

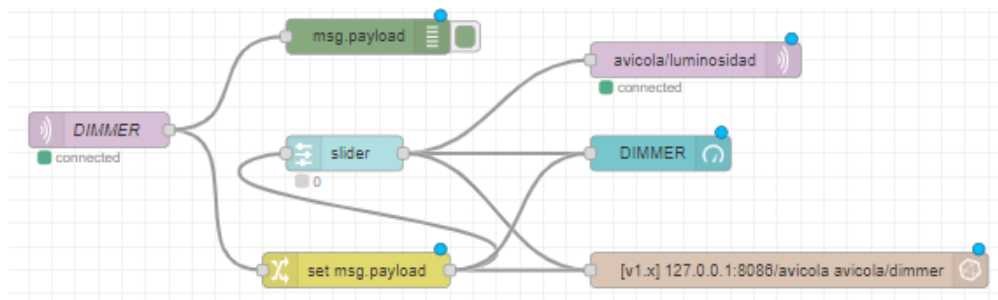


Figura 75 Programación del Dimmer desde la HMI

Elaborado por: El investigador

3.1.2.9 Implementación del sistema

- **Identificación de la ubicación de los actuadores y los nodos sensores.**

Los diferentes componentes que intervienen en el sistema se encuentran ubicados de la siguiente manera:

Los actuadores como las ventoleras se encuentran ubicados en las paredes correspondientes a la izquierdo y derecho del galpón, estas en la parte superior para permitir el ingreso del aire de una manera uniforme y técnicamente correcta, esto ayudara a disipar el aire ingresado de una mejor manera y así mantener la presión atmosférica dentro del galpón, esto se puede observar en la figura 76. Los extractores se encuentran ubicados en la pared frontal junto a la puerta del galpón el cómo se puede observar en la figura 76

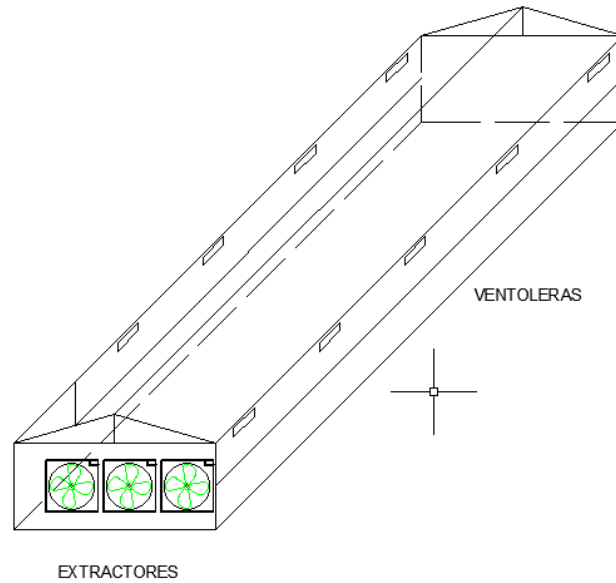


Figura 76 *Ubicación de las ventoleras y los extractores*
Elaborado por: *El investigador*

Los calefactores se encuentran ubicados en la parte posterior del galpón, para poder emitir el aire caliente a través de sus mangas dispersadas por el galpón, estos calefactores también se encuentran ubicados en ese lugar debido a que ahí se encuentra el suministro de combustible para quemado y generara así el calor y gracias a los ventiladores que poseen estos calefactores emitir el aire caliente generado a todo el galpón, esto se puede observar en la figura 77

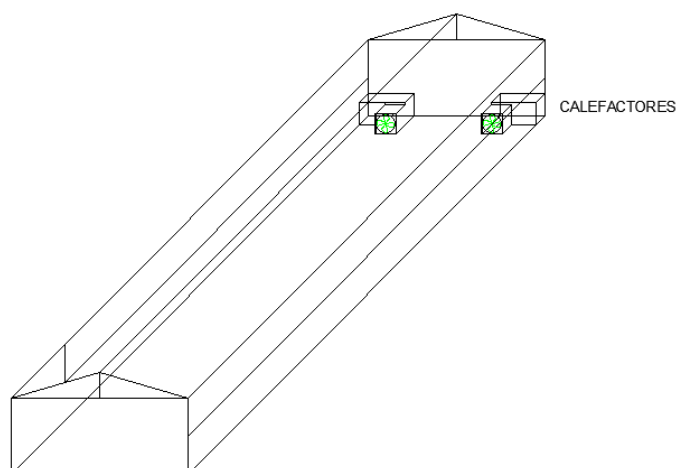


Figura 77 *Ubicación de los calefactores en el galpón*
Elaborado por: *El investigador*

La ubicación de los sensores es en la parte media del galpón, donde se genera una mayor alteración de los parámetros técnicos como la temperatura, humedad relativa y los gases tóxicos como el amoníaco, por lo que el lugar óptimo para los sensores es el indicado y así poder tomar los datos con mayor exactitud. La ubicación de los nodos sensores es como se indica en la figura 78.

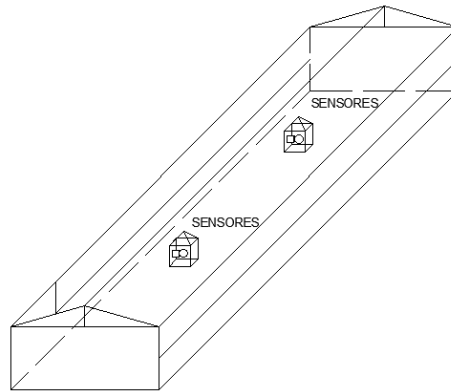


Figura 78 Ubicación de los nodos sensoriales

Elaborado por: El investigador

El tablero de control debe estar ubicado lo más cercano posible a la puerta de acceso por eso se ha colocado en la pared junto a la puerta, además está en este lugar porque las líneas de alimentación eléctrica llegan hasta este punto del galpón. Por lo que se ha colocado el tablero en este sitio como indica la figura 79.

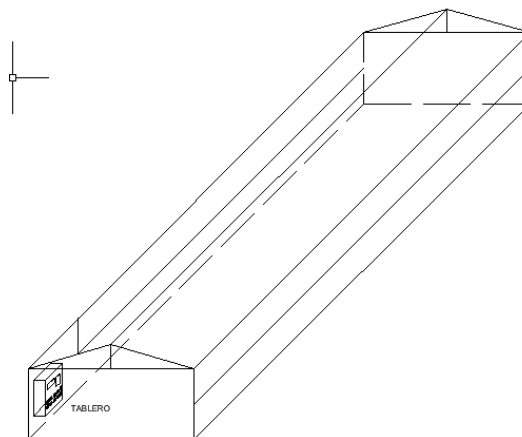


Figura 79 Ubicación del tablero de control.

Elaborado por: El investigador

La ubicación de la antena de transmisión es en la parte más alta del galpón, la cual está junto a los depósitos de agua, la antena se colocó con ayuda de un mástil a una altura de 4 metros desde la superficie como se indica en la figura 80.

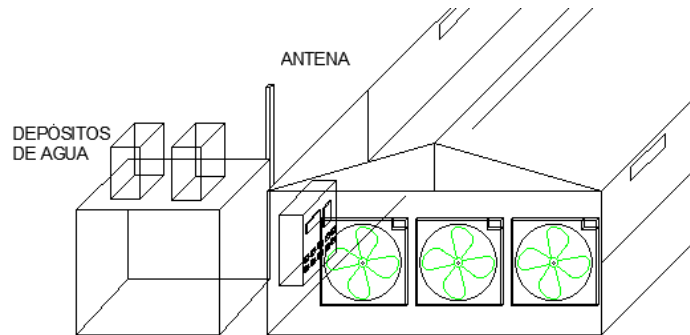


Figura 80 Ubicación de la antena Emisora

Elaborado por: El investigador

- **Implementación de los circuitos electrónico**

- ✓ **Nodo sensor**

Con diseño electrónico del circuito del nodo sensor se ha procedió a la implementación, para lo cual se ha realizado el diseño del ruteado es decir las conexiones de los componentes electrónicos. La figura 81 muestra el diseño PCB o ruteo de las conexiones, esto se lo ha realizado en el software Proteus, para luego implementar en la baquelita.

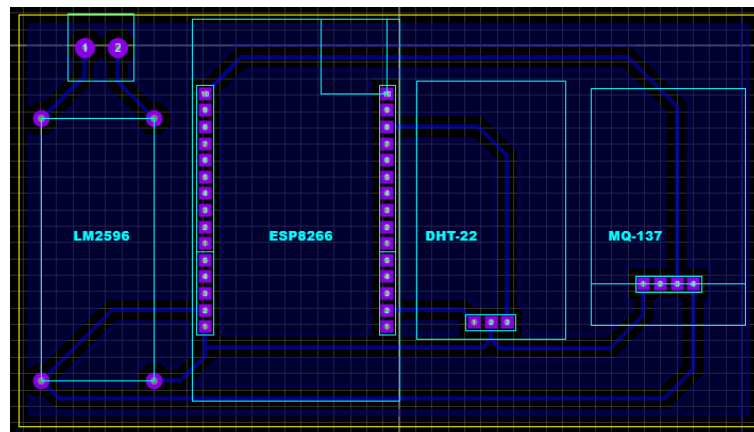


Figura 81 Diseño del Ruteo de pistas del nodo sensor

Elaborado por: El investigador

Una vez realizado el ruteo de pistas se procede con el proceso de elaboración de la baquelita, para lo cual hay que tener listo los materiales como son: baquelita, cautín, taladro, estaño y los componentes electrónicos. El paso número uno es transferir el ruteo de las pistas a la baquelita como indica la figura 82, luego se procede a corroer el cobre innecesario quedando solo las pistas de conexión como indica la figura 83, una vez corroído se procede a perforar la baquelita según el número de pines a soldar como indica la figura 84, y por último se procede a soldar los componentes quedando el resultado final como la figura 85, además de realizar pruebas de funcionamiento.



Figura 82 Ruteado transferido a la baquelita nodo sensor

Elaborado por: El investigador

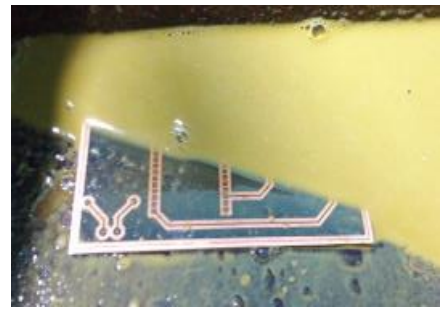


Figura 83 Corrosión de la baquelita es decir eliminación del exceso de cobre del nodo sensor.

Elaborado por: El investigador

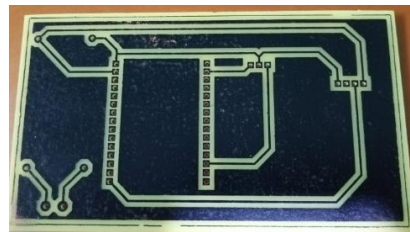


Figura 84 Perforación de los agujeros para cada uno de los pines a conectar del nodo sensor

Elaborado por: El investigador



Figura 85 Soldadura de componentes y pruebas de funcionamiento del nodo sensor

Elaborado por: El investigador

La implementación de los nodos sensoriales en el galpón de levante de la granja Damiancito se la realizo en base al estudio de ubicación de los sensores, en la figura 86 se indica el nodo sensor 1 ubicado en el galpón y en la figura 87 se muestra el nodo sensor 2.



Figura 86 Implantación del nodo sensor 1

Elaborado por: El investigador



Figura 87 Implementación del nodo sensor 2

Elaborado por: El investigador

Para realizar la conexión de los nodos sensoriales se utilizó 35 metros de cable gemelo AWG número 16, el que alimenta los nodos con un voltaje de 5 voltios y una corriente de 6.5 amperios.

El diseño de las carcasas de los nodos sensoriales se utilizó el software Fusion 360, en el que se tomó en consideración las dimensiones de las placas electrónicas, de los sensores y la dimensión del cable de alimentación, en la figura 88 se muestra el diseño de la base de la carcasa, y en la figura 89 se muestra la tapa superior de la carcasa.

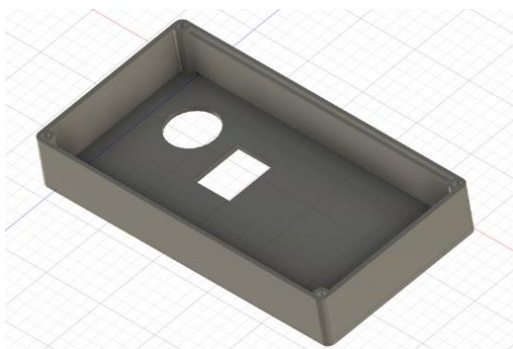


Figura 88 Base de la carcasa del nodo sensorial.

Elaborado por: El investigador

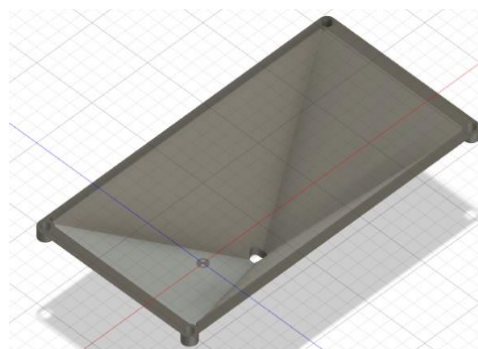


Figura 89 Tapa superior de la carcasa del nodo sensorial.

Elaborado por: El investigador

Nodo de mando y control electrónico

Una vez realizado el diseño electrónico del circuito de mando y control se ha procedido a la implementación, para lo cual se ha realizado el diseño del ruteado es decir las conexiones de los componentes electrónicos. La figura 90 muestra el diseño PCB o ruteo de las conexiones, esto se lo ha realizado en el software Proteus, para luego implementar en la baquelita.

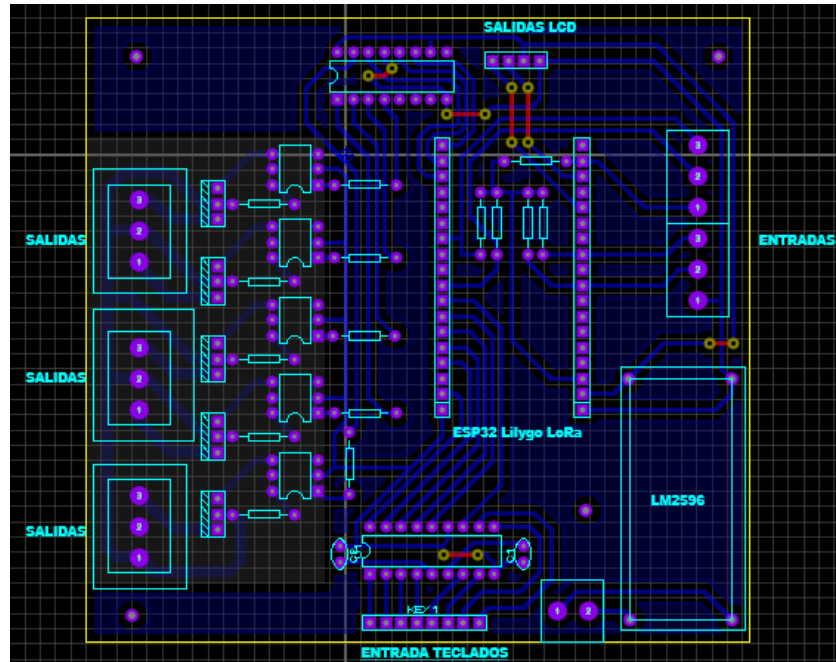


Figura 90 Diseño del Ruteo de pistas del circuito de mando y control

Elaborado por: El investigador

Para la implementación del nodo central se sigue los siguientes pasos: paso número uno es transferir el ruteo de las pistas a la baquelita como indica la figura 91, luego se procede a corroer el cobre innecesario quedando solo las pistas de conexión como indica la figura 92, una vez corroído se procede a perforar la baquelita según el número de pines a soldar como indica la figura 93, y por último se procede a soldar los componentes quedando el resultado final como la figura 94, además de realizar pruebas de funcionamiento. La instalación en el galpón se indica en el anexo 7.

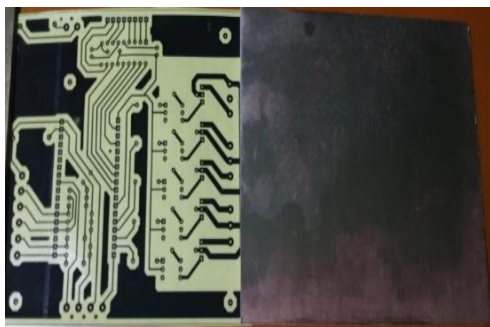


Figura 91 Ruteado transferido a la baquelita del circuito de mando y control

Elaborado por: El investigador

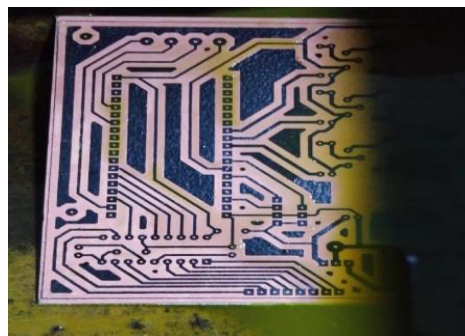


Figura 92 Corrosión de la baquelita es decir eliminación del exceso de cobre del circuito de mando y control.

Elaborado por: El investigador

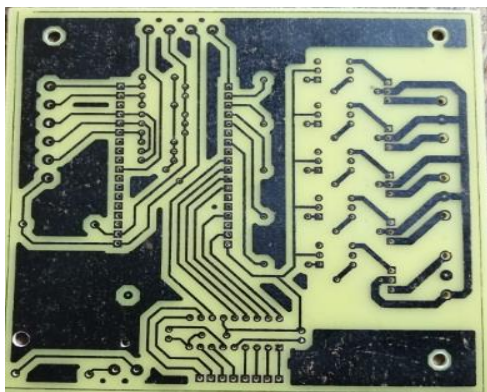


Figura 93 Perforación de los agujeros para cada uno de los pines a conectar del circuito de mando y control

Elaborado por: El investigador

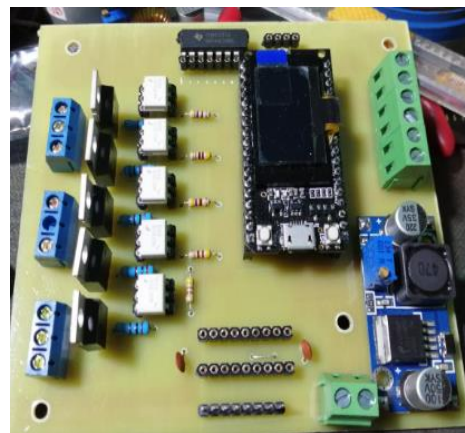


Figura 94 Soldadura de componentes y pruebas de funcionamiento del circuito de mando y control

Elaborado por: El investigador

✓ Nodo receptor

El nodo receptor está constituido por una tarjeta de desarrollo ESP32 LoRa Lilygo TTGO la cual se conecta al servidor en la plataforma AWS.

Implementación de los circuitos eléctricos

Para la implementación del circuito eléctrico se utilizó una caja metálica de 60x100x40cm, en la que se encuentran ubicaron los componentes como: los disyuntores, los contactores, las salidas y la fuente de poder en la parte interior de esta caja, mientras que los selectores, el botón de emergencia y los diferentes

actuadores se encuentran en la parte exterior de la caja, de igual manera en la parte exterior se encuentra ubicado el circuito de mando y control en una caja aislada del material metálico de la caja principal.

Parte interior

En la parte interior del gabinete se han ubicado tres regletas metálicas y tres plásticas, las metálicas para la colocación de los disyuntores, contactores y las salidas mientras que las plásticas para los cables de conexión, además se colocó un total de seis disyuntores dobles y un disyuntor triple, los dobles corresponden a los contactores de los actuadores mientras que el triple a la alimentación, el número de contactores colocados son síes, uno para la activación del sistemas, tres para los actuadores y ventoleras y 2 para los calefactores, además se colocó las salidas para los actuadores. En la figura 95 se indica los diferentes componentes instalados en el gabinete.



Figura 95 *Instalación de componentes eléctricos en el gabinete*

Elaborado por: *El investigador*

Parte exterior

En la parte exterior del gabinete figura 96 se colocó un total de 5 selectores correspondientes al modo de operación manual, cada uno de estos son su respectivo indicador lumínico, dos pulsadores uno correspondiente a la activación del sistema el de color verde y el de color rojo de desactivación y

de igual manera con sus indicadores con el color identifico al del pulsador, además un pulsador de emergencia en la parte inferior del gabinete. La caja de color blanco pertenece al circuito de mando y control electrónico en esta se encuentra un teclado matricial y una pantalla LCD para la visualización de los datos.



Figura 96 Exterior del gabinete de mando y control manual

Elaborado por: El investigador

Al finalizar la implementación de cada uno de los compontes del sistema se ha obtenido como resultado el funcionamiento de los circuitos de una manera correcta, por ultimo se ha desarrollado un esquema general de las conexiones eléctricas y eletronicas del sistema como se indica en la figura 97. La que esta comprendida por el nodo central con las conexiones a la etapa de potencia, la conexión de los nodos sensoriales al nodo central y la comunicación Lora.

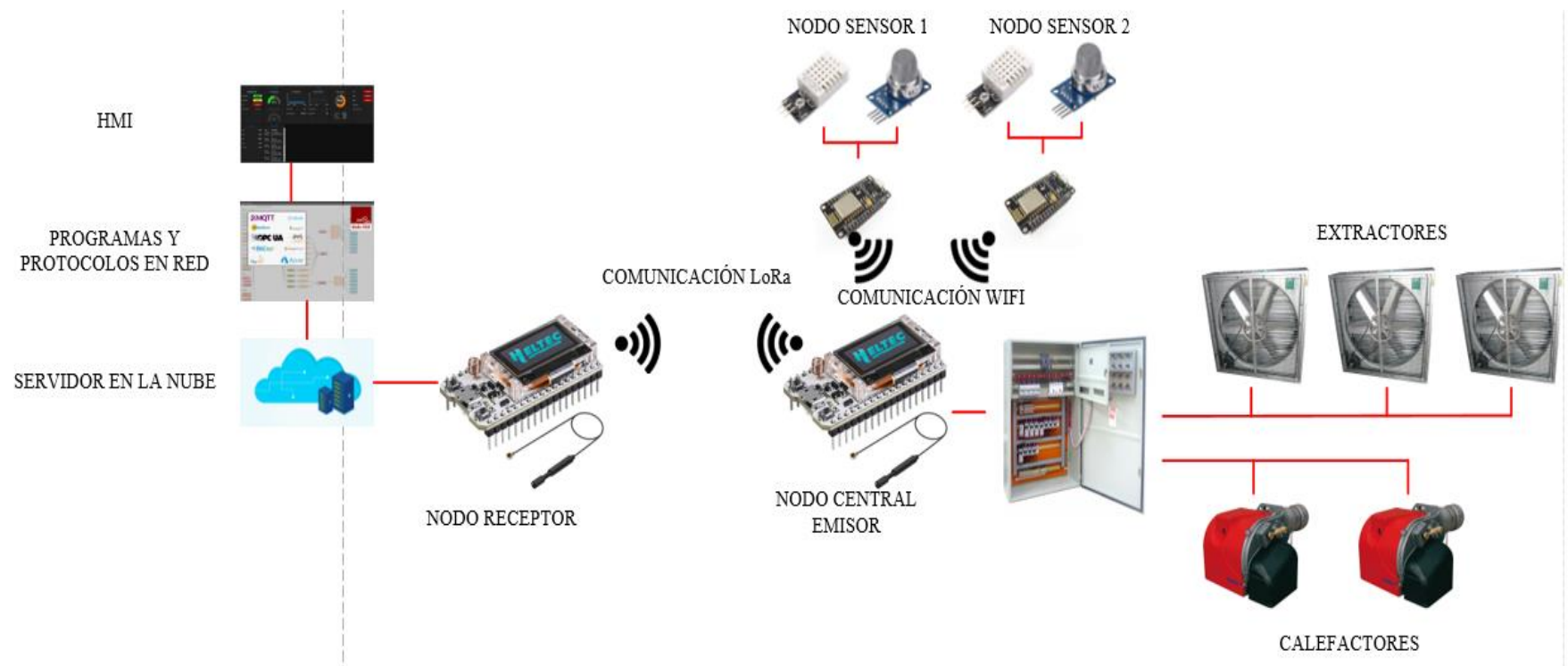


Figura 97 Representacion de las instalaciones de los diferentes componentes del sistema

Elaborado por: El investigador

3.1.3 Resultados y discusión

Una vez culminada la implementación y el diseño de los diferentes nodos del sistema de control y monitoreo inalámbrico para la granja Damiancito, se ha procedido a realizar las pruebas de funcionamiento en las que se ha obtenido los siguientes resultados:

3.1.3.1 Verificación inicial del galpón

En la verificación del estado inicial del galpón se determinó la existencia de los siguientes elementos eléctricos: tres extractores con un motor WEG W22 cada uno, las especificaciones técnicas se indica en la tabla 11 página 47. Dos calefactores ECOFLAM MAX12 los que trabajan con diésel y solamente necesita la señal eléctrica de activación, las especificaciones técnicas se indican en la tabla 12 página 49. Dos ventoleras MEGAMAT 2TF6KN las que se activan con la misma señal de los extractores, las especificaciones técnicas se indican en la tabla 13 página 50. Dos termostatos TS9501 encargados de medir la temperatura del galpón las especificaciones técnicas se indican en la tabla 14 página 51. Un dimmer AviLamp220 encargado de establecer el nivel de luminosidad las especificaciones técnicas se indican en la tabla 15 páginas 51-52.

3.1.3.2 Comunicación nodo central nodo sensorial

La comunicación entre el nodo central y los nodos sensoriales es la adecuada, para que esto sucediera se utilizó el protocolo de comunicación inalámbrica UDP, esto ha permitido conectar y enviar los datos desde los sensores hacia el nodo central, para la verificación del correcto funcionamiento se puede observar la figura 98, esta corresponde al puerto serial, en el que se puede indicar las temperaturas enviadas por los nodos sensoriales al nodo central.

```

COM5
192.168.10.126 humedad = 35.00 temperatura = 27.60
-----VENTILACION
192.168.10.126 humedad = 35.00 temperatura = 27.60
192.168.10.126 humedad = 35.00 temperatura = 27.60
192.168.10.126 humedad = 35.00 temperatura = 27.60
192.168.10.126 humedad = 35.00 temperatura = 27.50
192.168.10.126 humedad = 35.00 temperatura = 27.50
192.168.10.126 humedad = 35.00 temperatura = 27.50
-----VENTILACION
192.168.10.126 humedad = 35.00 temperatura = 27.50
192.168.10.126 humedad = 35.00 temperatura = 27.50
Sending HeLoRa World !numero de paquete enviado
192.168.10.126 humedad = 34.90 temperatura = 27.60
192.168.10.126 humedad = 34.90 temperatura = 27.60
192.168.10.126 humedad = 34.90 temperatura = 27.60
192.168.10.126 humedad = 34.90 temperatura = 27.60
-----VENTILACION
192.168.10.126 humedad = 34.90 temperatura = 27.60
192.168.10.126 humedad = 35.30 temperatura = 27.60
192.168.10.126 humedad = 35.30 temperatura = 27.60
192.168.10.126 humedad = 35.30 temperatura = 27.60
192.168.10.126 humedad = 35.30 temperatura = 27.60
192.168.10.126 humedad = 35.30 temperatura = 27.60
Sending HeLoRa World !numero de paquete enviado
-----VENTILACION

```

Figura 98 *Envío de datos desde los nodos sensoriales al nodo central*

Elaborado por: El investigador

3.1.3.3 Nodo central y control manual del sistema

La implementación del nodo central da como resultados el control y monitoreo de los parámetros técnicos como temperatura, humedad, luminosidad y amoniac, para la parte de control se lo puede realizar de manera manual desde los selectores ubicados en el gabinete, y de manera automática. Para la activación de la ventilación se manipula los selectores con el nombre de ventilación como se observa en la figura 99, ventilación 1 corresponde al extractor 1 y a la ventolera 1, ventilación 2 corresponde al extractor 2 y a la ventolera 2, ventilación 3 corresponde al extractor 3, para la activación de los calefactores se debe manipular los selectores con el nombre de calefacción como se observa la figura 99, calefacción 1 corresponde al actuador calentador 1 y calefacción 2 al calentador 2. Además, en la pantalla LCD se muestra los datos enviados por los nodos sensores. También se indica el rango de temperatura a la cual debe estar el galpón, las temperaturas máximas y mínimas deben ser ingresadas por el operario según el día de vida de las aves. Esto se observa en la figura 99.



Figura 99 *Funcionamiento del sistema en modo manual*

Elaborado por: *El investigador*

3.1.3.4 Comunicación nodo receptor y emisor

La implementación del nodo emisor se realizó en el mismo gabinete del nodo central dentro de la caja de color blanco mientras que en el nodo receptor se realizó en un elevador de una altura de 25 metros de altura desde la superficie terrestre, esto se indica en la figura 100.



Figura 100 *Instalación del nodo receptor en el elevador de 25 metros de altura*

Elaborado por: *El investigador*

Para la verificación del envío de datos desde el nodo emisor se ha accedido al puerto serial de este nodo receptor, la figura 101 indica el envío y recepción de datos en el nodo emisor.

```

COM5
-----
HUMEDAD PROMEDIO=0.00
AUT_MAN=1
ACTIVACION WEB=0
Received from: 0xaa
Sent to: 0xbb
Message ID:1299
Message length: 56
Message: 1299a26.90bc29.90de0.23f29g30h0i0j0kl1lm0n26.80o0.00plq0
RSSI: -109
Snr: -3.25

1300a26.90bc29.60de0.21f29g30h0i0j0kl1lm0n26.90o0.00plq0Packet No. = 1300
Temperatura 1 = 26.90
Temperatura 2 =
Humedad 1=29.60
Humedad 2=
Amoniaco=0.21
Temperatura minima=29
Temperatura maxima=30
MOTOR 1=0
MOTOR 2=0
MOTOR 3=0
CAL 1=1
CAL 2=1
DIMMER=0
TEMPERATURA PROMEDIO=26.90
HUMEDAD PROMEDIO=0.00
AUT_MAN=1
ACTIVACION WEB=0
Received from: 0xaa
Sent to: 0xbb

```

Figura 101 Recepción y envío de datos en el nodo emisor en el elevador

Elaborado por: El investigador

Además, para la verificación de los datos enviados por el nodo emisor al nodo receptor se ha realizado la tabla 35, en la que se indica el tiempo que tarde en llegar el dato de temperatura, humedad y amoniaco de un nodo al otro; por ende, los datos que se envían y reciben llegan en orden específico el que es temperatura, humedad y por último amoniaco. Los datos que se indican en la tabla corresponden a la semana 4 de vida de las pollitas, donde la temperatura debe oscilar entre los 22 °C y los 24 °C, y la muestra extraída es desde las 14:00 hasta las 14:01, como se puede observar en la tabla 35 el tiempo entre dato y dato es de 5 segundos.

Tabla 35 *Tiempo de envío y recepción de los datos entre el nodo emisor y el receptor*

Tiempo que tarde en llegar los datos del nodo emisor al receptor					
Nodo sensor 1					
Temperatura 1	Tiempo (S)	Humedad 1	Tiempo (S)	Amoniaco 1	Tiempo (S)
24.7	14:00:00	45%	14:00:01	16ppm	14:00:02
24.7	14:00:05	45%	14:00:06	16ppm	14:00:07
24.7	14:00:10	45%	14:00:11	16ppm	14:00:12
24.6	14:00:15	45%	14:00:16	16ppm	14:00:17
24.6	14:00:20	45%	14:00:21	16ppm	14:00:22
24.6	14:00:25	45%	14:00:26	16ppm	14:00:27
24.6	14:00:30	45%	14:00:31	16ppm	14:00:32
24.6	14:00:35	45%	14:00:36	16ppm	14:00:37
24.6	14:00:40	45%	14:00:41	16ppm	14:00:42
24.7	14:00:45	45%	14:00:46	16ppm	14:00:47
24.7	14:00:50	45%	14:00:51	16ppm	14:00:52
24.7	14:00:55	45%	14:00:56	16ppm	14:00:57
24.7	14:00:60	-	14:00:60	-	14:00:60
Nodo sensor 2					
Temperatura 2	Tiempo (S)	Humedad 2	Tiempo (S)	Amoniaco 2	Tiempo (S)
23.4	14:00:03	46%	14:00:04	18ppm	14:00:05
23.4	14:00:08	46%	14:00:09	18ppm	14:00:10

23.4	14:00:13	46%	14:00:14	18ppm	14:00:15
23.4	14:00:18	46%	14:00:19	18ppm	14:00:20
23.5	14:00:23	46%	14:00:24	18ppm	14:00:25
23.5	14:00:28	46%	14:00:29	18ppm	14:00:30
23.5	14:00:33	46%	14:00:34	18ppm	14:00:35
23.5	14:00:38	46%	14:00:39	18ppm	14:00:40
23.5	14:00:43	46%	14:00:44	18ppm	14:00:45
23.4	14:00:48	46%	14:00:49	18ppm	14:00:50
23.4	14:00:53	46%	14:00:54	18ppm	14:00:55
23.4	14:00:58	46%	14:00:59	18ppm	14:00:60
-	14:01:03	-	14:01:04	-	14:01:05

Elaborado por: El investigador

En la tabla 35 se indica los tiempos que tardan en llegar los datos de un nodo al otro, en esta se evidencia que el dato de la temperatura solamente varia en 0.1 grados debido a que la muestra elegida es de un minuto y los cambios son insignificantes, a pesar de que la capacidad de muestreo del sensor de temperatura es de 2, de igual manera los datos del sensor de humedad y amoniaco permanecen constante a lo largo del minuto. Cabe recalcar que la tabla 35 tiene como finalidad indicar el tiempo de respuestas entre nodos.

Los erros obtenidos en la adquisición de datos fueron al momento de poner en funcionamiento los nodos sensoriales, en lo que respecta al sensor de temperatura fue de un error de más menos 2 grados centígrados, el sensor de humedad tuvo un error inicial de más 10%. Los errores obtenidos fueron los indicados en las hojas técnicas de los sensores, donde indican la necesitan de 1 día a 2 días de estar en funcionamiento constante para lograr la estabilización en la adquisición de datos. Además, el tiempo de retardo en el envío de los datos desde el emisor al receptor fue de 25 a 30 segundos como se muestra en la tabla 36.

Tabla 36 *Tiempo de envío y recepción de los datos entre el nodo emisor y el receptor pruebas iniciales*

Tiempo que tarde en llegar los datos del nodo emisor al receptor pruebas iniciales			
Nodo sensor 1			
Temperatura 1	Tiempo (S)	Humedad 1	Tiempo (S)
15.6	10:00:00	19%	14:00:03
13.2	14:00:20	23%	14:00:23
14.8	14:00:45	20%	14:00:48
16.6	14:01:05	24%	14:01:08
13.3	14:01:30	22%	14:01:33
15.4	14:01:50	20%	14:01:53
17.6	14:02:15	24%	14:02:18
14.3	14:02:35	25%	14:02:36
12.1	14:03:00	26%	14:03:03
17.7	14:03:30	23%	14:03:33
Nodo sensor 2			
Temperatura 2	Tiempo (S)	Humedad 2	Tiempo (S)
14.6	10:00:05	20%	14:00:08
12.2	14:00:25	24%	14:00:28
13.8	14:00:50	22%	14:00:53
15.6	14:01:10	26%	14:01:13
12.3	14:01:35	23%	14:01:38

14.4	14:01:55	22%	14:01:58
16.6	14:02:20	25%	14:02:23
13.3	14:02:40	26%	14:02:41
11.1	14:03:05	28%	14:03:08
16.7	14:03:30	25%	14:03:33

Elaborado por: El investigador

3.1.3.5 Funcionamiento de la HMI

La HMI permite el control y monitoreo de forma inalámbrica y remota a la vez, está conformada por las siguientes etapas: enfriamiento compuesta por 3 interruptores y 3 indicadores como se indica en la figura 102.

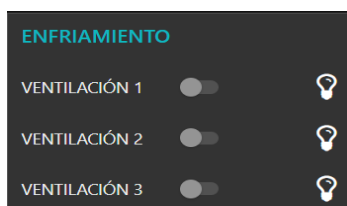


Figura 102 Etapa de enfriamiento

Elaborado por: El investigador

La etapa de calentamiento está compuesta por 2 interruptores y dos indicadores como se indica en la figura 103.



Figura 103 Etapa de calentamiento

Elaborado por: El investigador

La etapa de activación del modo automático se indica en la figura 104, cuando esta se encuentra en modo ON no se podrá activar y desactivar los actuadores desde la HMI, cuando se encuentre en modo OFF si se puede activar y desactivar los actuadores desde la HMI.

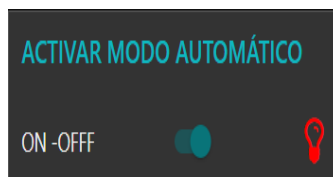


Figura 104 Etapa de habilitación del acceso a al HMI

Elaborado por: El investigador

Para el ingreso del rango de temperatura desde la HMI se lo hace en la etapa de parámetros como se ve en la figura 105.

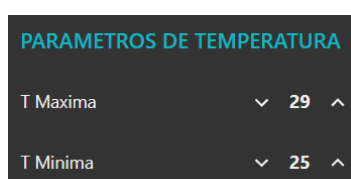


Figura 105 etapa de ingreso de los rangos de temperaturas

Elaborado por: El investigador

La etapa de sensores de temperatura correspondiente al nodo sensor 1 y nodo sensor 2, además cuenta con un indicador del promedio de las temperaturas y un histograma donde se muestra en comportamiento de la temperatura en el galpón figura 106.



Figura 106 Datos de los sensores de temperatura

Elaborado por: El investigador

La etapa de sensores de humedad correspondiente al nodo sensor 1 y nodo sensor 2, además cuenta con un indicador del promedio de humedades y un histograma donde se muestra en comportamiento de la humedad en el galpón figura 107.

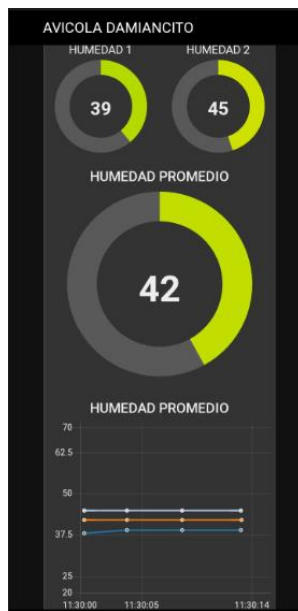


Figura 107 Datos de los sensores de humedad

Elaborado por: El investigador

La etapa del dimmer cuenta con un slider para modificar el porcentaje de luminosidad y un indicador como se indica en la figura 108.



Figura 108 Etapa de variación de la intensidad luminosa

Elaborado por: El investigador

Para la visualización del amónico se ha utilizado dos indicadores como se muestra en la figura 109.

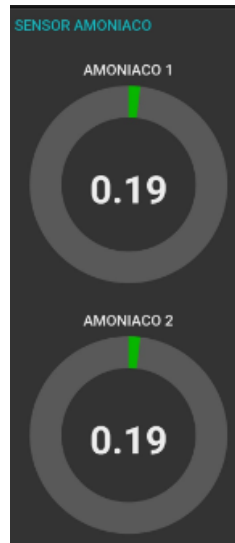


Figura 109 Datos del sensor de amoniaco

Elaborado por: El investigador

Con la HMI finalizada se realizó las pruebas de envío y recepción de las instrucciones hacia el tablero principal ubicado en el galpón de levante de la granja Damiancito, para constar el correcto funcionamiento se ha verificado el tiempo de respuesta de los actuadores una vez envía la señal desde la HMI. Para la toma de los resultados hay constatar que la comunicación LoRa entre el nodo emisor y receptor se esté ejecutando en la frecuencia seleccionada de 915Mhz. Para el tiempo de envío según el factor de potencia se tomó en cuenta el facto de dispersión SF11 siendo utilizado por LoRa en distancias de 2km con una tase bist de 24bytes.

Tabla 37 Tiempo de respuesta en la activación y desactivación de los actuadores desde la HMI

Dato		Tiempo estimado según el factor de dispersión	Tiempo obtenido en las pruebas
Botón de acceso al control remoto	Botón ON-OFF	700 ms	400ms

Activación desactivación	Calefactor 1	700 ms	500 ms
	Calefactor 1	700 ms	500 ms
	Ventilación 1	700 ms	500 ms
	Ventilación 2	700 ms	500 ms
	Ventilación 3	700 ms	500 ms
Escritura de datos	Temperatura máxima	700 ms	400 ms
	Temperatura mínima	700 ms	400 ms
	Porcentaje Intensidad lumínica	700 ms	400 ms

***Elaborado por:** El investigador*

Según la tabla 37 el tiempo que se demora en recibir los datos el tablero desde la HMI es inferior a la estimada por el factor de dispersión. El tiempo que tarda los datos en ser enviados esta entre los 400ms y los 500ms, debido a que, la línea de vista entre los nodos es directa y sin obstáculos. Además, con las pruebas realizadas se puede evidenciar que el envío de datos desde la HMI es más rápido que el envío desde el tablero principal hacia la HMI, esto se debe a que los datos son enviados individualmente y el tamaño del paquete es menor, debido a que solo se envía señales binarias ya sea un 1 lógico o un cero lógico, para la activación o desactivación de los actuadores, por otra parte, al enviar los datos de la temperatura máxima o mínima tarda 500ms, esto se debe al tipo de dato enviado en este caso es un dato entero y de igual manera el dimmer. Las figuras 110 y 111 indica que el valor de la HMI y el del tablero es el mismo, con esto se comprueba el funcionamiento.



Figura 110 Visualización de datos en el tablero del galpón

Elaborado por: El investigador



Figura 111 Visualización de datos la HMI

Elaborado por: El investigador

3.1.3.6 Almacenamiento de los datos

La base de datos INFLUXDB permite ser configurada con NODE-RED y Grafana; por consiguiente, los datos enviados desde el galpón hacia el servidor mediante el sistema de comunicación inalámbrica se almacenaran en la base de datos INFLUXDB, teniendo como resultado lo que se indica en la figura 112 datos de los sensores correspondientes a los promedios de temperatura y los datos correspondiente al promedio de humedad en la figura 113, siendo estos los requeridos por la granja para el registro en el documento de control de los lotes, el dato del sensor de amónico no requiere ser almacenado por la granja.

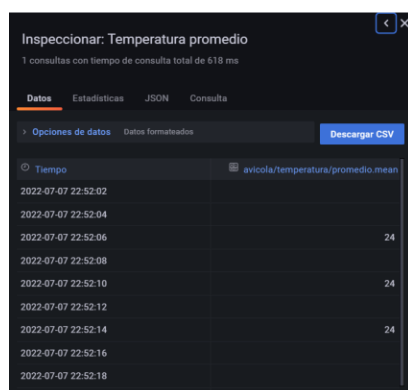


Figura 112 Datos promedios de temperatura almacenados en la base de datos INFLUXDB

Elaborado por: El investigador

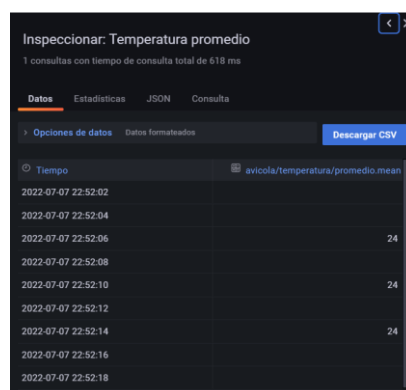


Figura 113 Datos promedios de humedad almacenados en la base de datos INFLUXDB

Elaborado por: El investigador

3.1.3.7 Comparación de la eficiencia del sistema

Para el análisis de la incidencia del sistema de control y monitoreo inalámbrico en el galpón de levante de las gallinas de postura de la granja Damiancito, se procede a realizar la comparación de los resultados obtenidos al final de las 10 semanas de vida de las gallinas, para esto se realiza dicha comparación entre el lote 20 en que no se utilizaba el sistema, y el lote 21 donde se utilizó el sistema de control y monitoreo inalámbrico. Para el análisis de los resultados se ha utilizado la información proporcionada por el operador y el inspector, la información proporcionada por el operador se obtiene del anexo 8. A continuación, se indica los resultados obtenidos en cada uno de los lotes.

- **Crianza de las gallinas de postura sin el sistema de control y monitoreo inalámbrico lote 20**

En el anexo 8 se muestra la tabla que represente el libro diario del operador del galpón de levante, en la que debe registrar los datos como: consumo de alimento, inventario de las aves, uniformidad de las aves; por ende, peso de cada una de las aves, y la ganancia de peso de cada ave por semana. Estos datos son corroborados con los proporcionados en el manual de crianza de gallinas de postura del proveedor.

Para el análisis del consumo de alimento de las aves se ha extraído un segmento de la tabla del anexo 8, la tabla 38 muestra el consumo de alimento que han realizado las aves y el que indica en manual de crianza.

Tabla 38 Consumo de alimento sin el sistema de control y monitoreo

Sem a.	CONSUMO DE ALIMENTO					Gr Ave Tab	Gr Ave Ac	Tem Pro C°	Hu m Pro m %
	TA B	REA L	BULTO S	KILO S	K.ACU M				
1	11	13,0	31,0	1.395	1.395	105	91	31.5	50
2	17	16,0	38,0	1.710	3.105	252	204	30.4	50
3	22	25,6	60,5	2.723	5.828	427	384	28.3	50
4	28	35,0	82,5	3.713	9.540	630	630	25.2	50
5	35	37,5	88,0	3.960	13.500	882	894	20.5	50
6	41	48,9	114,5	5.153	18.653	1.16	1.23	21.5	60
7	47	47,4	111,0	4.995	23.648	1.46	1.57	22.3	60
8	51	52,1	122,0	5.490	29.138	1.79	1.93	23.4	60
9	55	58,6	137,0	6.165	35.303	2.16	2.34	22.6	60

10	58	57,7	135,0	6.075	41.378	2.55	2.75	23.4	60
11	60	65,0	152,0	6.840	48.218	2.98	3.20	22.4	60

Elaborado por: El investigador

Según la tabla 38 el consumo de alimento de las aves es un total de 425 quintales hasta cumplir las 11 semanas de vida según el manual de crianza, mientras que en la realidad se consumió un total de 457 quintales. También se indica el peso que cada ave debe tener en la semana, la columna de color amarillo indica el peso que debe poseer según el manual de crianza, mientras que la columna verde representa el peso obtenido en el muestreo realizado en el galpón. La diferencia de pesos de las aves se debe a que la temperatura promedio en cada una de las etapas no es la adecuada, en este caso en cada semana se tiene una diferencia de 2 grados centígrados respecto a lo que indica en manual de crianza LOHMANN BROWN, de igual manera el porcentaje de humedad esa por debajo del indicado en el manual que es de 60 a 70 porciento de humedad. Al no cumplir con las especificaciones brindadas por el manual, las aves tienen a alimentarse demasiado si la temperatura es inferior a la indica y si la temperatura es mayor deja de alimentarse y solo consumen agua lo que conlleva a la deshidratación.

En la tabla 39 se muestra el inventario de aves, donde se indica la cantidad de aves muertas en la semana, el porcentaje de mortandad semanal y por último el número de aves existentes en el galpón al final de cada semana.

Tabla 39 *Inventario de aves existentes en el galpón sin el sistema control y monitoreo*

Semana	AVES - INVENTARIO				%Mort Sem	% Mort Acm.	% M+S Acu.	SALDO
	Mortandad (Mort)	Sel	Otros	ACU				AVES
1	554	-	-	554	3,48	3,48	3,48	15.346
2	125	-	-	679	0,81	4,27	4,27	15.221
3	42	-	-	721	0,28	4,53	4,53	15.179
4	41	-	-	762	0,27	4,79	4,79	15.138
5	33	-	-	795	0,22	5,00	5,00	15.105
6	48	-	-	843	0,32	5,30	5,30	15.057
7	13	-	-	856	0,09	5,38	5,38	15.044
8	3	-	-	859	0,02	5,40	5,40	15.041

9	2	-	-	861	0,01	5,42	5,42	15.039
10	5	-	-	866	0,03	5,45	5,45	15.034
11	4	-	-	870	0,03	5,47	5,47	15.030

Elaborado por: El investigador

Según la tabla 39 la columna de color celeste indica el número de aves muertas por semana, la columna de color amarillo el número de aves muertas acumuladas y la columna de color verde el número de aves sobrantes en el galpón, la columna de color naranja representa el porcentaje de mortandad a la semana. Según estos datos obtenidos el porcentaje de mortandad en el galpón es de un 5.47% esto se indica en la celda marcada de color rojo en la tabla 39.

Para conocer la uniformidad de las aves se procede a revisar el anexo 8 en el que se muestra los datos obtenidos a lo largo de la etapa de crianza. Para esto la tabla 40 indica todos los datos como son: peso corporal de las aves según el manual y el peso obtenido en el muestro de los pesos en el galpón.

Tabla 40 Uniformidad de las aves sin el sistema de control y monitoreo

PESO CORPORAL		UNIFORMIDAD			
TAB	REAL	10%-	%Uniformidad	10%+	C.V.
75	76,7		82,0		7,30
130	123,0		79,2		7,40
195	187,6		76,4		8,00
275	264,7		74,4		8,30
367	370,7		70,0		9,00
475	481,7		66,8		10,30
583	593,2		70,8		9,70
685	671,6		74,0		8,10
782	778,6		67,6		10,20
874	868,0		76,4		8,40
961	1.001,5		73,6		-

Uniformidad promedio	73.7		
----------------------	------	--	--

Elaborado por: El investigador

Según la tabla 40 la columna de color amarillo indica el peso corporal que cada ave debe poseer según la semana de vida, en la columna adyacente se muestra el peso real de las aves que se encuentran en el galpón, las celdas de color rojo de esta tabla indica que el peso no es el adecuado. Según estos datos expuestos en la tabla 40 el porcentaje de uniformidad de las aves en el galpón es de un 73.7%, esto indica que el porcentaje de uniformidad está por debajo del 85% que indica el manual de crianza del proveedor.

Con los resultados expuestos en las tablas 38,39 y 40 se ha realizado la gráfica 114; por lo que, en esta se visualiza el comportamiento de las aves en el galpón de levante sin el sistema de control y monitoreo. La línea de color verde indica el peso corporal real de las aves en el galpón, la línea de color rojo el peso corporal de las aves según el manual de crianza, la línea de color rosado el consumo de alimento real y la línea de color azul el consumo de alimento según el manual de crianza. Las unidades utilizadas son los gramos.

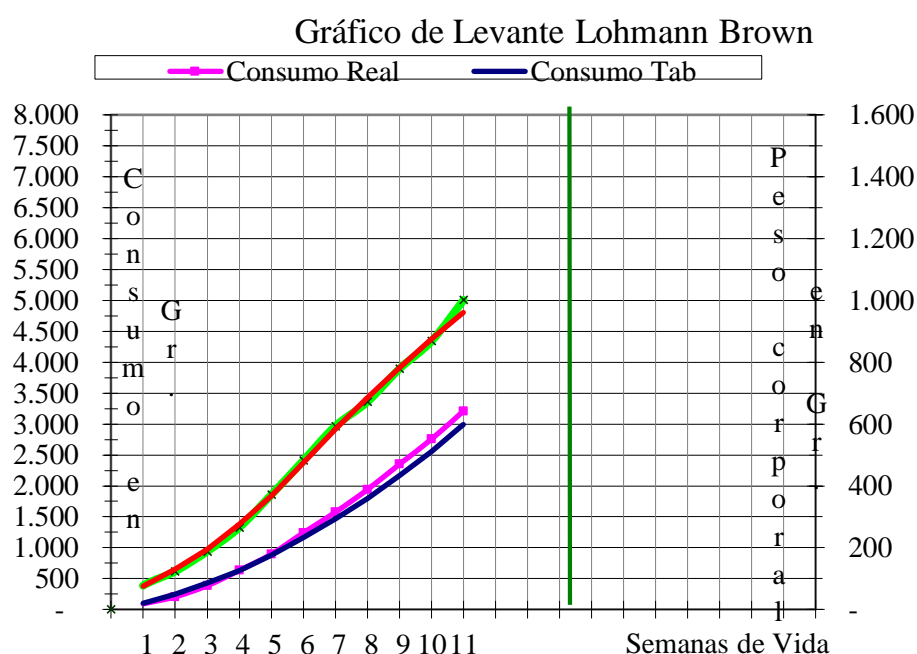


Figura 114 Representación de los datos de peso y consumo de alimento de las aves en la etapa de crecimiento

Elaborado por: El investigador

La figura 115 indica el porcentaje de mortandad de las aves a lo largo de la etapa de crianza. Para esto se ha utilizado los datos de la tabla 39.

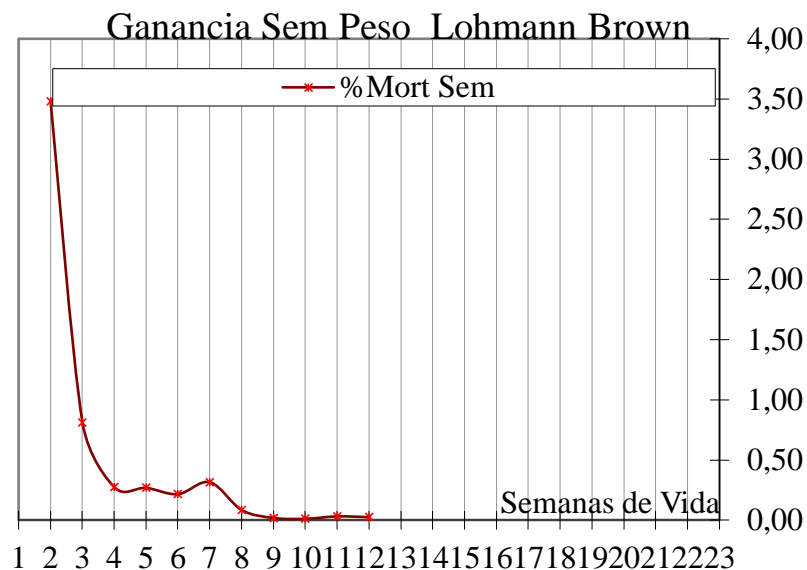


Figura 115 Porcentaje de mortandad de las aves sin el sistema de control y monitoreo

Elaborado por: El investigador

Los datos obtenidos en las tablas 38,39 y 40 conjuntamente con la presentación de los mismos en las figuras 114 y 115 han dado como resultado lo siguiente: el porcentaje de mortandad es de un 5.47, de uniformidad de 73.7%, mientras que el 20.83% representan a las gallinas que requieren unos días más de cuidado para salir de la etapa de crianza. En la gráfica de 111 se aprecia que la línea de color verde permanece junto a la línea de color rojo lo que representa que el peso de las aves es el correcto según, pero si observamos la línea de color rosado verificamos que sobrepasa la línea de color azul, esto indica que la cantidad de alimento colocada es superior a la indica en el manual de crianza.

- **Crianza de las gallinas de postura con el sistema de control y monitoreo inalámbrico lote 21**

En el anexo 9 se muestra la tabla que represente el libro diario del operador del galpón de levante del lote 21, en la que debe registrar los datos como: consumo de alimento, inventario de las aves, uniformidad de las aves; por ende, peso de cada una de las aves, y la ganancia de peso de cada ave por semana. Estos

datos son corroborados con los proporcionados en el manual de crianza de gallinas de postura del proveedor.

Para el análisis del consumo de alimento de las aves se ha extraído un segmento de la tabla del anexo 9, la tabla 41 muestra el consumo de alimento que han realizado las aves y el que indica en manual de crianza.

Tabla 41 Consumo de alimento con el sistema de control y monitoreo

Sema	CONSUMO DE ALIMENTO					Gr Ave Tab	Gr Ave Ac	Tem Pro C°	Hum Pro m %
	TAB	REAL	BULTOS	KILOS	K.ACU M				
1	11	11,5	33,0	1.485	1.485	105	100	35.4	58
2	17	13,8	39,5	1.778	3.263	252	237	29.6	61
3	22	22,3	64,0	2.880	6.143	427	413	28.3	64
4	28	34,2	98,0	4.410	10.553	630	617	25.7	66
5	35	41,7	119,5	5.378	15.930	882	864	22.8	63
6	41	35,7	102,5	4.613	20.543	1.16	1.11	23.4	62
7	47	43,8	125,5	5.648	26.190	1.46	1.42	22.8	63
8	51	42,9	123,0	5.535	31.725	1.79	1.72	21.6	62
9	55	57,8	165,5	7.448	39.173	2.16	2.12	22.5	63
10	58	55,9	160,0	7.200	46.373	2.55	2.52	20.3	64
11	60	53,1	152,0	6.840	53.213	2.98	2.89	22.3	63

Elaborado por: El investigador

Según la tabla 41 el consumo de alimento de las aves es un total de 425 quintales hasta cumplir las 11 semanas de vida según el manual de crianza, mientras que en la realidad se consumió un total de 412,6 quintales. También se indica el peso que cada ave debe tener en la semana, la columna de color amarillo indica el peso que debe poseer según el manual de crianza, y la columna de color verde representa el peso obtenido en el muestreo realizado en el galpón. Los pesos de las aves en el muestreo es lo más similar al de la tabla esto se debe a que la temperatura y humedad están dentro de los parámetros técnicos que indica el manual, por ende, la cantidad alimentación balanceado es la correcta al igual que el consumo de agua. Cabe mencionar que la temperatura promedio y humedad promedio se han extraído de la base de datos INFLUXDB una hora en específico todos los días en este caso a las 2pm.

En la tabla 42 se muestra el inventario de aves, donde se indica la cantidad de aves muertas en la semana, el porcentaje de mortandad semanal y por último el número de aves existentes en el galpón al final de cada semana.

Tabla 42 Inventario de aves existentes en el galpón con el sistema control y monitoreo

Seman a	AVES - INVENTARIO				%Mort Sem	% Mort Acm.	% M+S Acu.	SALD O
	Mortandad	Sel	Otros	ACU				AVES
1	35	-	-	35	0,19	0,19	0,19	18.465
2	8	-	-	43	0,04	0,23	0,23	18.457
3	7	-	-	50	0,04	0,27	0,27	18.450
4	5	-	-	55	0,03	0,30	0,30	18.445
5	8	-	-	63	0,04	0,34	0,34	18.437
6	5	-	-	68	0,03	0,37	0,37	18.432
7	5	-	-	73	0,03	0,39	0,39	18.427
8	8	-	-	81	0,04	0,44	0,44	18.419
9	11	-	-	92	0,06	0,50	0,50	18.408
10	4	-	-	96	0,02	0,52	0,52	18.404
11	4	-	-	100	0,02	0,54	0,54	18.400

Elaborado por: El investigador

Según la tabla 42 la columna de color celeste indica el número de aves muertas por semana, la columna de color amarillo el número de aves muertas acumuladas y la columna de color verde el número de aves sobrantes en el galpón, la columna de color naranja representa el porcentaje de mortandad a la semana. Según estos datos obtenidos el porcentaje de mortandad en el galpón es de un 0.54% esto se indica en la celda marcada de color rojo en la tabla 42.

Para conocer la uniformidad de las aves se procede a revisar el anexo 9 en el que se muestra los datos obtenidos a lo largo de la etapa de crianza. Para esto la tabla 43 indica todos los datos como son: peso corporal de las aves según el manual y el peso obtenido en el muestro de los pesos en el galpón.

Tabla 43 Uniformidad de las aves con el sistema de control y monitoreo

PESO CORPORAL		UNIFORMIDAD			
TAB	REAL	10%-	%Uniformidad	10%+	C.V.
75	70,4		92,0		7,30
130	139,4		80,2		7,40
195	200,5		79,4		8,00

275	300,0		75,4		8,30
367	423,3		77,0		9,00
475	532,9		77,8		10,30
583	596,3		78,8		9,70
685	666,2		78,9		8,10
782	786,1		79,6		0,20
874	881,4		79,4		8,40
961	1.001,5		79,6		
Uniformidad promedio			79,8		

Elaborado por: El investigador

Según la tabla 43 la columna de color amarillo indica el peso corporal que cada ave debe poseer según la semana de vida, en la columna adyacente se muestra el peso real de las aves que se encuentran en el galpón, las celdas de color rojo de esta tabla indican que el peso no es el adecuado. Según estos datos expuestos en la tabla 43 el porcentaje de uniformidad de las aves en el galpón es de un 79.8%, esto indica que el porcentaje de uniformidad está por debajo del 85% que indica el manual de crianza del proveedor.

Con los resultados expuestos en las tablas 41,42 y 43 se ha realizado la gráfica 116; por lo que, en esta se visualiza el comportamiento de las aves en el galpón de levante con el sistema de control y monitoreo. La línea de color verde indica el peso corporal real de las aves en el galpón, la línea de color rojo el peso corporal de las aves según el manual de crianza, la línea de color rosado el consumo de alimento real y la línea de color azul el consumo de alimento según el manual de crianza. Las unidades utilizadas son los gramos.

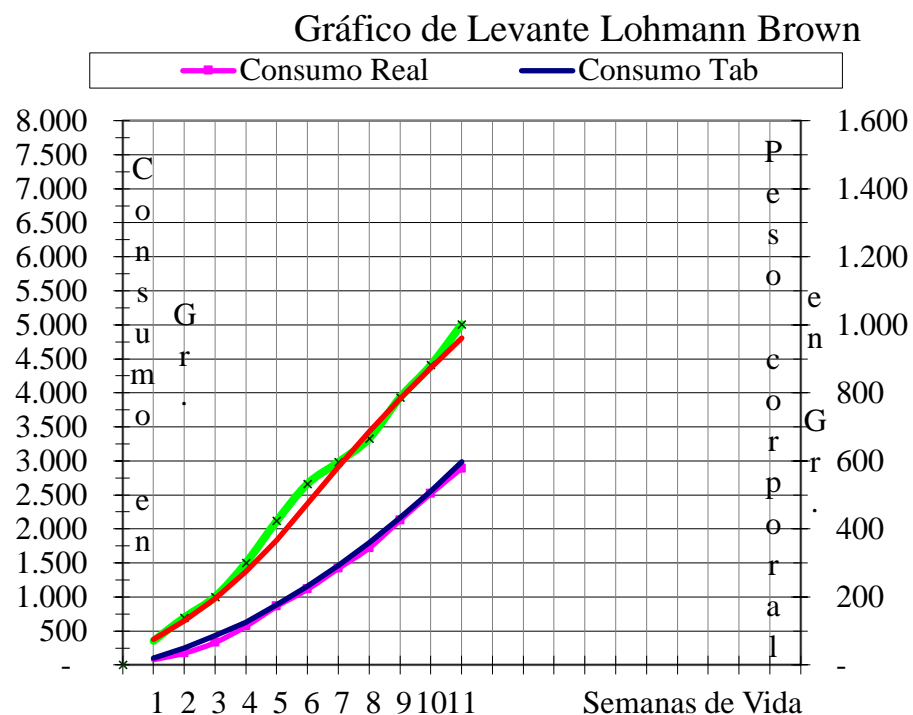


Figura 116 Representación de los datos de peso y consumo de alimento de las aves en la etapa de crecimiento

Elaborado por: El investigador

La figura 117 indica el porcentaje de mortandad de las aves a lo largo de la etapa de crianza. Para esto se ha utilizado los datos de la tabla 42.

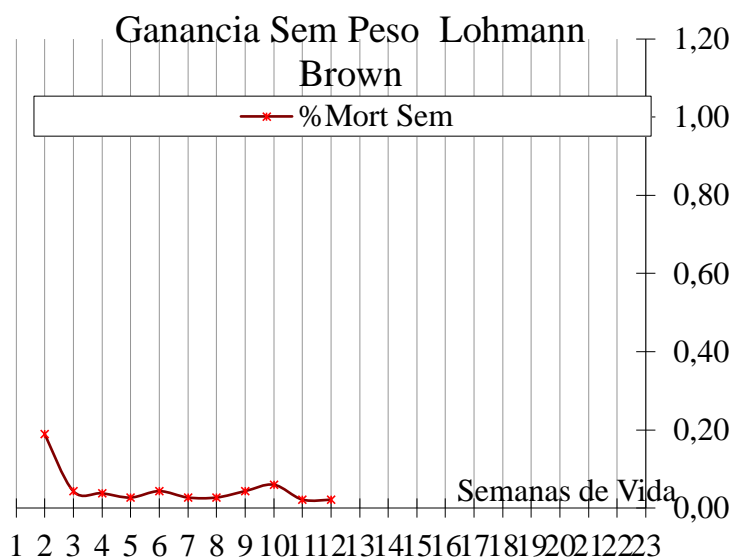


Figura 117 Porcentaje de mortandad de las aves sin el sistema de control y monitoreo

Elaborado por: El investigador

Los datos obtenidos en las tablas 41,42 y 43 conjuntamente con la presentación de los mismos en las figuras 116 y 117

ha dado como resultado lo siguiente: el porcentaje de mortandad es de un 0.54, de uniformidad de 79.8%, mientras que el 20.66% representan a las gallinas que requieren unos días más de cuidado para salir de la etapa de crianza. En la gráfica de 113 se aprecia que la línea de color verde permanece junto a la línea de color rojo en las primeras 3 semanas, desde la semana 4 a la 7 sobrepasa a la roja y desde la semana 8 a la 11 se mantiene junto a la línea roja, esto indica que el peso de las aves es el correcto según el manual de crianza, pero si observamos la línea de color rosado verificamos que el consumo real de alimento es inferior a la proporcionada por el manual desde la semana 1 a la 4, pero el peso es correcto, desde la semana 5 hasta la 11 el consumo de alimento es igual o inferior al que indica el manual de crianza, y así logrando el peso deseado en las aves.

3.1.4 Presupuesto

En la tabla 44 se detalla el costo de los materiales utilizados en la implementación del sistema de control y monitoreo inalámbrico; por ende, este tiene un valor de 996,50 dólares incluido IVA. En la tabla se está indicando el precio unitario y el total sin IVA de cada uno de los elementos y al final de esta se indica el subtotal y total.

Tabla 44 *Presupuesto del diseño e implementación del sistema de control y monitoreo*

Item	Componentes electrónicos del nodo central	Unidad	Precio unitario (dólares)	Precio total (dólares)
1	Lilygo LoRa32 (TTGO)	2	45,23	90,46
2	Regulador de voltaje LM2596	1	4,28	4,28
3	Pantalla LED 20x40	1	12,15	12,15

4	Circuito integrado PCF8574(expansor de entradas y salidas digitales para comunicación I2C)	1	5,40	5,40
5	Teclado matricial 4x4	1	2,19	2,19
6	Opto acoplador MOC3021	5	5,30	26,5
7	TRIAC BT436	5	4,50	22,5
8	Circuito integrado 74HC595 (registro de desplazamiento)	1	5,40	5,40
9	Borneras	13	0,35	4,55
Componentes eléctricos del nodo central				
10	Contactores 220V, 10A	6	25,83	154,98
11	Pulsadores tipo hongo	2	2,13	4,26
12	Selectores CNC 3 posiciones	6	2,30	13,8
13	Botón paro de emergencia	1	2,19	2,19
14	Disyuntores dos posiciones 25A, 10K	6	6,47	38,82
15	Luces pilotos verde 12V-540V	5	1,07	5,35
16	Luz piloto roja 12V-540V	1	1,07	1,07

17	Cable de alimentación de numero 16 color verde	10m	0,16	1,6
18	Cable de alimentación de numero 16 color amarillo	10m	0,16	1,6
19	Cable de alimentación para 220 18AWG	6m	0,43	2,58
20	Fuente de poder 24V,10A	1	19,85	19,85
21	Disyuntores tres posiciones 25A, 10K	1	15,43	15,43
Elementos del gabinete metálico				
22	Gabinete metálico 100x60x40	1	184,42	184,42
23	Riel perforado	4m	1,64	6,54
24	Bornera LEG riel 4mm 30A, 10AWG	20	1,26	25,2
25	Canaleta ranurada 25x25	3m	3,71	11,13
Componentes electrónicos de los nodos sensoriales				
26	ESP8266	2	7,99	15,98
27	MQ-137	2	56,56	113,12
28	DHT22	2	16,8	33,6
29	Regulador de voltaje LM2595	2	4,28	8,56
30	Cable gemelo AWG número 16	30m	0,50	15
31	Carcasas 3D de los nodos	2	20,60	41,20

	SUBTOTAL	889,73
	IVA 12%	106,77
	TOTAL	996,50

Elaborado por: El investigador

Además, al costo de implementación se debe añadir los gastos de transporte que corresponde a 150 dólares, gastos de materiales de oficina que son 40 dólares y por ultimo se debe añadir un sueldo básico de ingeniería que está en 800 dolares, el costo de ingeniería se añade porque se toma el encuentra el tiempo dedicado por el investigador en el desarrollo e implementación del sistema, dando como resultado un valor total de 1986,5 dólares.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- En este proyecto de investigación se implementó un sistema de control y monitoreo inalámbrico, para la supervisión de los parámetros técnicos del galpón de crianza de gallinas de postura, en la granja Damiancito del Cantón Pelileo utilizando tecnología Lora y MQTT, obteniendo como resultados el mejoramiento de este proceso en los aspectos de uniformidad, aves descriadas y mortandad; donde, los índices de uniformidad son de un 79.8%, un 20.66% de aves descriadas, y por ultimo un 0.54% de mortandad y así acercándonos al 0.02% de mortandad que indica el manual LOHMANN BROWN-CLASSIC.
- En este proyecto de investigación se analizó el sistema de control y monitoreo existente en el galpón de crianza de aves de postura en la granja Damiancito, porque en el proceso de crianza se implementaba altos recursos humanos y económicos, debido al bajo control de los parámetros técnicos del galpón, generando índices de uniformidad de un 73.7% de aves, un 20.83% de aves descriadas, y por ultimo un 5.47% de mortandad siendo este el indicador más alejado al que indica el manual de crianza LOHMANN BROWN-CLASSIC que corresponde a un 0.02%.
- Se diseñó el sistema electrónico para el control y monitoreo de los parámetros técnicos del galpón de crianza de aves de postura, el que está construido por los dos nodos sensoriales para a la adquisición de datos, el nodo central encargado de la generación de instrucciones y la manipulación de datos, además de la comunicaron inalámbrica LoRa entre el nodo central o emisor y el nodo receptor ubicado a 1100m de

distancia. Cada uno de los nodos constituidos por sus tarjetas de desarrollo y sus elementos de control y visualización.

- La Implementación de la Interfaz Hombre Maquina (HMI) del sistema de control y monitoreo inalámbrico de los parámetros técnicos del galpón de crianza de aves de postura, se realizó con la implementación de un servidor en AWS el que permite la comunicación con el nodo receptor; por consiguiente, la adquisición de los datos de los sensores y de los estados en el que se encuentra los actuadores, para posteriormente visualizar en la pantalla HMI elaborada en Node-Red, y almacenamiento de los datos promedios de temperatura, humedad y amoníaco en la base de datos INFLUXDB.

4.2 Recomendaciones

- Para el diseño e implementación del sistema se recomienda acudir con el encargado del proceso, para tomar apunte de las necesidades existentes en el lugar a implementar, y ajustarse a los requerimientos del operador.
- Para un correcto funcionamiento del sistema de control y monitoreo inalámbrico se recomienda tener en cuenta los requerimientos eléctricos, como son voltajes y corrientes en la fuente de energía, también cada vez que se inicie el sistema tomar en cuenta que las conexiones de los nodos se encuentren en perfectas condiciones.
- Para la comunicación inalámbrica LoRa se recomienda tener una línea de vista directa. Para la comunicación del nodo emisor con el servidor se debe tener acceso a internet estable de preferencia por cable.
- Para la comunicación Wi-Fi en lugares cerrados tomar en cuenta el tipo de protocolo a utilizar ya sea UDP o TCP IP para el control de datos, dependiendo el requerimiento del sistema el que puede ser un mayor número de datos en corto tiempo correctos o incorrectos; o por ende, pocos datos pero solo correctos.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Universidad Nacional Autónoma de México "Facultad de Medicina Veterinaria y Zootecnia, «Introducción a la producción avícola,» de *Introducción a la zootecnia del pollo y la gallina*, México, LDCVF. Abril Braulio Ortiz, 2018, pp. 15-39.
- [2] J. W. y. A. S. Javier Pedroza, «Manual de producción avícola,» de *Sistemas de explotación avícola*, Tuluá, SENA, 2005, pp. 9-13.
- [3] Ministerio de Agroindustria Presidencia de la Nación INTA, «Sistemas de producción Avícolas,» de *Manual de Avicultura*, Buenos Aires, 2016, pp. 69-79.
- [4] Airphex, «Condiciones de Producción de gallinas ponedoras,» 14 03 2017. [En línea]. Available: <https://sites.google.com/site/razasdegallinasponedoras/condiciones-de-produccion/ventilacion>. [Último acceso: 16 02 2022].
- [5] K. Gonzalez, «Zootecnia y veterinaria es mi pasión,» Cría y levante de la gallina ponedora, 12 10 2018. [En línea]. Available: <https://zoovetesmipasion.com/avicultura/gallinas-ponedoras/pasos-para-una-adeuada-cria-y-levante-de-la-pollona-etapa-critica-para-la-postura/>. [Último acceso: 17 02 2022].
- [6] Solla Nutrición Animal, «SOLLA,» Avicultura gallinas de postura, 16 05 2020. [En línea]. Available: <https://www.solla.com/productos/avicultura/ciclo>. [Último acceso: 17 02 2022].
- [7] Cátedra avícola, «Puntos clave para mejorar la ventilación en los túneles con galpones de aves,» Cátedra avícola & Agropecuaria, 19 01 2022. [En línea]. Available: <https://www.catedraavicola.com.ar/puntos-clave-para-mejorar-la-ventilacion-en-los-tuneles-con-galpones-de-aves/>. [Último acceso: 17 02 2022].

- [8] INSAVI, «Galpón de pollitas INSAVI,» AVINEWS America Latina, 18 08 2017. [En línea]. Available: <https://avicultura.info/insavi-mejores-companeros-de-viaje-hacia-rentabilidad-de-su-negocio/>. [Último acceso: 17 02 2022].
- [9] CAVENCO, «GRANJAS INTELIGENTES CAVENCO EN BRASIL,» Avicultura.com, 17 08 2017. [En línea]. Available: <https://avicultura.com/granjas-inteligentes-cavenco-en-brasil/>. [Último acceso: 17 02 2022].
- [10] L. Pablo, «Medición de Parámetros Productivos en Avicultura,» 13 10 2017. [En línea]. Available: <https://avicultura.info/medicion-de-parametros-productivos-en-avicultura/>. [Último acceso: 12 0 2021].
- [11] LOHMANN TIERZUCHT, «LOHMANN BROWN-CLASSIC PONEDORAS,» LOHMANN TIERZUCHT, Cuxhaven-Germany, 2020.
- [12] S. I. Marllelis Gutierrez, «Sistemas de Control,» de *Fundamentos básicos de instrumentación y control*, Santa Elena, Universidad Estatal Península de Santa Elena, 2017, pp. 100-110.
- [13] G. A. y. J. M. Leonel Corona, Sensores y actuadores, Mexico D. F: Ebook, 2014.
- [14] P. S. S. y. R. H. Alberto Brunete, «Actuadores Electricos,» de *Introducción a la automatización Industrial*, Madrid, Universidad Pontificia de Madrid, 2020.
- [15] R. Boylestad, «Calefactor electrico,» de *Introducción al análisis de circuitos*, México, Pearson Educación, 2004, pp. 85-86.
- [16] M. G. y. S. Iturralde, «Tipos de Sensores,» de *Fundamentos básicos de instrumentación y control*, Santa Elena, UPSE, 2017, pp. 28-30.
- [17] G. A. y. J. C. Leonel Corona, «Sensores de temperatura y Humedad,» de *Sensores y Actuadores aplicaciones con arduino*, México D.F., Grupo Editorial Patria Ebook, 2014, pp. 173-201.

- [18] ALPHASENSE, «NH3-B1 Ammonia Sensor,» Alphasense Ltd, Sensor Technology House, Great Notley, 2019.
- [19] J. Morgan, «Sensores de luz: unidades, usos y cómo funcionan,» 23 07 2021. [En línea]. Available: <https://blog.endaq.com/how-light-sensors-work>. [Último acceso: 10 03 2022].
- [20] J. Rolán Vilorio, «Contactores,» de *Automatismos y cuadros eléctricos*, Madrid-España, Paraninfo, 2004, pp. 15-17.
- [21] P. C. Costas, «Componentes principales de cuadros de maniobra y control,» de *Montaje e instalación de cuadros de maniobra y control*, España, Ideaspropias Editorial, 2004, pp. 13-16.
- [22] Hetpro, «Hetpro,» 10 12 2017. [En línea]. Available: <https://hetpro-store.com/TUTORIALES/microcontrolador/>. [Último acceso: 15 03 2022].
- [23] Arduino.cc, «What is Arduino,» 2018. [En línea]. Available: <https://www.arduino.cc/en/Guide/Introduction>. [Último acceso: 28 03 2022].
- [24] E. Rodríguez, «De cero a maker: todo lo necesario para empezar con Raspberry Pi,» 15 07 2018. [En línea]. Available: <https://www.xataka.com/makers/cero-maker-todo-necesario-para-empezar-raspberry-pi>. [Último acceso: 28 03 2022].
- [25] J. Mercado, *Sistemas programables avanzados*, Madrid: Ediciones Paraninfo, 2019.
- [26] J. P. Blázquez, *Introducción a los sistemas de comunicación inalámbricos*, Cataluña: Universidad Oberta de Catalunya.
- [27] L. J. Aguilar, «Tecnologías de redes LPWAN,» de *Internet de las cosas: Un futuro hiperconectado: 5G, inteligencia artificial, Big Data, Cloud, Blockchain y ciberseguridad*, CD MÉXICO, Marcombo, 2021, pp. 70-73.

- [28] Fundacio Telefónica y Tauros , «Comparación de las distintas tecnologías LPWAN,» de *Sociedad Digital en España Sociedad Digital en España 2019*, Madrid, Gómez Apricio,S.L., 2020, pp. 62-65.
- [29] Semtech, «LoRa and LoRaWAN,» 12 2019. [En línea]. Available: <https://loradevelopers.semtech.com/documentation/tech-papers-and-guides/loraland-lorawan>. [Último acceso: 28 03 2022].
- [30] 2CI Group, «Conceptos de actualidad: LoRa y LoRaWan,» Canvisión, Melesight, 22 03 2021. [En línea]. Available: <https://www.2cigroup.com/es/conceptos-de-actualidad-lora-y-lorawan/>. [Último acceso: 28 03 2022].
- [31] ARCOTEL, «NORMA TÉCNICA DE ESPECTRO DE USO LIBRE Y DE ESPECTRO PARA USO DETERMINADO EN BANDAS LIBRES,» de *RESOLUCIÓN ARCOTEL-2018-LA AGENCIA DE REGULACIÓN Y CONTROL DE LAS TELECOMUNICACIONES ARCOTEL*, Quito, ARCOTEL, 2018, pp. 1-26.
- [32] D. H. R. B. M. O. J. M. R. O. C. V. Johnny Novillo-Vicuña, de *Arduino y el Internet de las cosas*, Alicante, Área de innovacion y desarrollo, S.L, 2018, pp. 35-38.
- [33] J. Pizarro, «MQTT,» de *Internet de las cosas (IOT) con ESP*, Madrid, Paraninfo SA, 2020, pp. 99-101.
- [34] H. Spona, «MySQL,» de *Programación de Bases de Datos con MYSQL y PHP*, España, Marcambo, 2010, pp. 81-85.
- [35] WEG, «W22 NEMA Premium Efficiency 1.5 HP 4P 145TC 3F 230/460//380 V 60//50 Hz IC411,» [En línea]. Available: <https://www.weg.net/catalog/weg/US/es/Motores-El%C3%A9ctricos/Motor-de-Inducc%C3%ADon---NEMA/Uso-General-/TEFC-Hierro-Gris/W22/W22-/W22-NEMA-Premium-Efficiency-1-5-HP-4P-145TC-3F-230-460-380-V-60-50-Hz-IC411---TEFC---Con-pies/p/11610562>. [Último acceso: 21 04 2022].

- [36] Ecoflam, «Ecoflam max 12 tc o 50-60,» de *Ecoflam Catalog*, Italia, Ecoflam, 2019, pp. 1-32.
- [37] Phonix Mecano , «Actuadores lineales Megamat 2 TF 6KN -D53582,» DEWERT and Phonix Mecano Company, 2014. [En línea]. Available: <https://1library.co/document/zxo5d4oz-phoenix-mecano.html>. [Último acceso: 20 02 2022].
- [38] ORBIS Energia Inteligente, «Termostati scatolati elettromeccanici,» 2016. [En línea]. Available: <https://www.orbisitalia.it/ts-9501-termostato-scatolato-dispositivo-interno-termoregolatore-a-capillarecampo-regolazione-5-60-c-ip40-orbis-ob581130/>. [Último acceso: 20 04 2022].
- [39] INOBram Automatas, «Manual del usuario Dimmer control electrónico de luminosidad,» 2002. [En línea]. Available: https://www.inobram.com.br/_files/ugd/80e3a5_d5704f9765c04467b9ccf2880e029768.pdf?index=true&lang=es. [Último acceso: 21 04 2022].
- [40] Espressif Systems, «UltraLowPower SoC with RISCv SingleCore CPU Supporting 2.4 GHz WiFi and Bluetooth LE,» de *ESP32-C3 Family Datasheet*, Prerelease version 0.6, 2021, pp. 1-40.
- [41] H. AUTOMATION, «Wi-Fi LoRa 32 (V2.1),» 2018. [En línea]. Available: <https://heltec.org/project/wifi-lora-32/>. [Último acceso: 14 04 2022].
- [42] MICROCHIP, «ATmega4808 Data Sheet,» Microchip Technology Inc, 2019.
- [43] Random Nerd Tutorials, «Random Nerd Tutorials,» ESP32 Pinout Reference: Which GPIO pins should you use? | Random Nerd Tutorials, [En línea]. Available: <https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>. [Último acceso: 16 05 2022].
- [44] TURCK, «TURCK – Your Global Automation Partner - Home,» [En línea]. Available: https://www.turck.de/datasheet/_es/edb_3073652_esl_es.pdf. [Último acceso: 28 04 2022].

- [45] TURCK, «Sensor de temperatura y humedad Con interfaz en serie para conectar a un nodo de radio DX80 M12FTH4Q,» https://www.turck.de/datasheet/_es/edb_44661_esl_es.pdf, p. 2, 2018.
- [46] Dallas Semiconductor, «ALLDATASHEET.COM - Electronic Parts Datasheet Search,» DS18B20 Programmable Resolution 1-Wire Digital Thermometer, [En línea]. Available: <https://www.alldatasheet.com/datasheet-pdf/pdf/58557/DALLAS/DS18B20.html>. [Último acceso: 28 04 2022].
- [47] K. G. R. S. Gilberto Garcia Navarrete, «SENSORES DE BAJO COSTO PARA EL MONITOREO DE CALIDAD DEL AIRE,» *EPISTEMUS*, vol. 13, nº 27, pp. 30-37, 12 12 2019.
- [48] Zhengzhou Winsen Electronics Technology, «Amonia Gas Sensor (MQ137),» 10 03 2015. [En línea]. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1307654/WINSEN/MQ137.html>. [Último acceso: 28 04 2022].
- [49] Zhengzhou Winsen Electronics Technology, «Multi-in-One Sensor Module (ZPHS01B),» 14 01 2020. [En línea]. Available: <https://html.alldatasheet.com/html-pdf/1303697/WINSEN/ZPHS01B/767/3/ZPHS01B.html>. [Último acceso: 28 04 2022].
- [50] T. Liu, «Digital-output relative humidity & temperature sensor/module DHT22 (DHT22 also named as AM2302),» SparkFun Electronics, [En línea]. Available: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>. [Último acceso: 10 05 2022].
- [51] E. López, «Modelos y escudos Arduino,» de *Guía práctica de fundamentos y simulación*, Madrid, Grupo editorial RA-MA, 2016, pp. 92-99.
- [52] W. Tomasi, «Sistemas de Comunicaciones Electrónicas,» de *Capítulo 17*, México, Pearson Educación, 2003, pp. 710-11.

ANEXOS

Anexo 1: Código de programación de los nodos sensoriales

```
/*UNIVERSIDAD TÉCNICA DE AMBATO  
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL  
PROYECTO DE INVESTIGACIÓN MODALIDAD TESIS  
TEMA: SISTEMA DE CONTROL Y MONITOREO INALÁMBRICO PARA LA  
GRANJA DAMIANCITO DEL CANTO PELILEO  
ESTUDIANTE: KEVIN SEBASTIÁN ZÚÑIGA SÁNCHEZ  
TUTO: ING. SANTIAGO ALTAMIRANO MELENDEZ  
NODO SENSOR*/  
  
#include "DHT.h"           //Incluir la librería del sensor de humedad y temperatura  
  
#include <ESP8266WiFi.h>    //Incluir la librería de la tarjeta de desarrollo del nodo  
sensor en este caso la ESP826  
  
#include <WiFiUdp.h>        //Incluir la librería para la configuración del protocolo WIFI  
UDP  
  
#define STASSID "ESP32"    //Establecer o definir el nombre de la red WiFi a al que se  
va a conectar el nodo sensor  
  
#define STAPSK "123456789" //Establecer o definir la contraseña de la red WiFi a al que  
se va a conectar el nodo sensor  
  
#define PIN_CONEXION 2     //Establecer o definir el pin al que se va a conectar la señal  
de envio del sesnor en este caso al pin digital 2  
  
#define TIPO_SENSOR DHT22 //Establecer o definir el tipo de sensor de la familia  
DHT en nuestro caso el DHT-22  
  
unsigned int localPort = 1234; //Establecer o definir el puerto por el se va  
realizar la comunicacioncon el nodo central, en este caso puerto fuente u origen  
  
char packetBuffer[UDP_TX_PACKET_MAX_SIZE + 1]; //Establecer o definir el buffer  
para alamcenar el paquete entrante  
  
WiFiUDP Udp;               //Establecer o definir la instancia que permitirá que  
se realice la comunicación UDP, es decir envío y recepción del paquete.  
  
DHT sensor(PIN_CONEXION, TIPO_SENSOR); //Creación de la variable DHT en  
al cual ira el pin de conexión y el tipo de sensor a utilizar.
```

```

float humedad, temperaturaEnGradosCelsius = 0; //Creación de las variables de tipo float
para adquirir los datos de humedad en porcentaje y temperatura en grados celcius

int ultimaVezLeido = 0; //Creación de una variable que ayudara a la lectura
de datos y el control de estos.

long milisegundosDeEsperaParaLeer = 2000; //Determinación de tiempo de espera
para la proxima lectura de datos en segundos

int gas_sensor = A0; //Creación de una variable llamada gas_sensor y
asignarle el pin analógico A0 para la lectura de los datos del sensor MQ137(amoniaco).

float m = -0.263; //Creación de una variable para la pendiente, dato a
utilizar en la configuración del sensor de amoniaco

float b = 0.42; //Creación de una variable para la intercepción en el eje
Y segun la grafica propuesta por el fabricante del sensor

float R0 = 36.70; //Resistencia R0 del sensor en kilo ohms

void setup() {

    Serial.begin(115200); //Velocidad a la que se va comunicar con el host

    WiFi.mode(WIFI_STA); //Establecer el tipo de red WiFi en nuestro caso
estatica

    WiFi.begin(STASSID, STAPSK); //Colocar el nombre y la contraseña de la red a
la que se va a conectar en nuestro caso se estableció estos parámetros en las variables
STASSID (nombre de la red), STAPSK(contraseña de la red)

    while (WiFi.status() != WL_CONNECTED) { //Confirma el estado de conexión

        Serial.print('.'); //Impresión d puntos hasta conectarse

        delay(500); //Tiempo de espera en segundos

    }

    Serial.print("Connected! IP address: "); //Se imprimirá en el puerto serial que se conecto
a la red WiFi

    Serial.println(WiFi.localIP()); //Se imprimirá en el puerto serial el número de
puerto a utilizar para la utilización del protocolo UDP

    Udp.begin(localPort); //Asignación del puerto local UDP el cual se nombro
anteriormente.

    Serial.setTimeout(2000); //Tiempo de espera en segundos


    sensor.begin(); //Creación del método sensor, esto es para el sesnor MQ-
137

```

```

    pinMode(gas_sensor, INPUT);          //Establecer como entrada a la variable
    gas_sensor

    Serial.println("Calentando");        //Se imprimirá el comentario calentado al recibir
    datos.

    delay(1000);                        //Tiempo de espera en segundos
}

void loop() {

    int packetSize = Udp.parsePacket();  //Leer el tamaño del paquete y
    comprueba la presencia de un paquete UDP y se lo almacena en un Buffer.

    if (packetSize) {                   //Comparación del tamaño del paquete si es
    mayor que 0 significa que ha llegado una petición o paquete

        Udp.beginPacket(Udp.remoteIP(), Udp.remotePort()); //Obtención de un puerto remoto
        y una dirección IP remota

        //////////////////////////////////////

        //                          INICIALIZACION WIFI

        //////////////////////////////////////

        humedad = sensor.readHumidity();          //Leemos el valor de humedad del
        sensor

        temperaturaEnGradosCelsius = sensor.readTemperature(); //Leemos el valor de
        temperatura del sensor en grados celcius

        if (isnan(temperaturaEnGradosCelsius) || isnan(humedad)) { //En ocasiones puede
        devolver datos erróneos; por eso lo comprobamos

            Serial.println("Error leyendo valores");          //Imprime comentario si el dato
            leído es erróneo

            ultimaVezLeido = 0;                                //Toma el valor de 0

            return;                                             //Retorna a la condición if
        }

        Serial.print("Humedad: ");                            //Cuando es correcto, imprime los
        valores de la variable humedad

        Serial.print(humedad);                                //Cuando es correcto, imprime los valores
        de la variable humedad

```



```

Serial.print(" %\t");           //Salto de línea

Serial.print("Temperatura: ");   //Cuando es correcto, imprime los
valores de la variable temperatura

Serial.print(temperaturaEnGradosCelsius); //Cuando es correcto, imprime
los valores de la variable temperatura en grados Celsius

Serial.println(" *C");           //Cuando es correcto, la letra *C

ultimaVezLeido = 0;             //Toma el valor de 0

ultimaVezLeido += 100;          //Sigue almacenando los valores

////////////////////////////////////

//RECEPCIÓN Y PROCESAMIENTO DE LOS DATOS DEL SENSOR DE
AMONACO MQ-137//

////////////////////////////////////

float sensor_volt;              //Definir una variable de tipo float

float RS_gas;                   //Definir una variable de tipo float para la resistencia
de sensor

float ratio;                    //Definir una variable de tipo float para el radio

float sensorValue = analogRead(gas_sensor); //Leer los valores analógicos en el pin
AO al que se conectó el sensor

sensor_volt = sensorValue * (5.0 / 1023.0); //Convertir el valor de tipo analogico en
voltaje

RS_gas = ((5.0 * 10.0) / sensor_volt) - 10.0; //Obtener el valor de la resistencia del
sensor

ratio = RS_gas / R0;            //Obtener el valor del radio con la siguiente
formula:RS_gas/R0

double ppm_log = (log10(ratio) - b) / m; //Obtener el valor en ppm(partículas por
millon) en una escala lineal con el valor del radio en una variable de tipo double

double ppm = pow(10, ppm_log);   //Convertir el valor del ppm a una escala
logarítmica en una variable de tipo double

double percentage = ppm / 10000; //Convertir en porcentaje en una variable de
tipo double

Serial.print("ppm_log:");        //Imprimir el valor del amoniaco en una escala
logarítmica

```

```

    Serial.print(ppm_log);           //Imprimir el valor del amoniaco en una escala
logaritmica

    Serial.print(" ppm:");           //Imprimir el valor del amoniaco en ppm

    Serial.print(ppm);               //Imprimir el valor del amoniaco en ppm

    Serial.print(" porcentaje:");     //Imprimir el valor del amoniaco en porcentaje

    Serial.println(percentage);       //Imprimir el valor del amoniaco en porcentaje

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

//ENVIÓ DE LOS PAQUETES CON LA INFORMACIÓN DE LOS SENSORES
HACIA EL NODO CENTRAL A TRAVÉS DEL PROTOCOLO UDP//

```

```

// NODO SERVIDOR EN UDP//

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

    char ReplyBuffer[10];             //Creamos un arreglo en el que
almacenaremos los valores a enviar el tamaño de arreglo es de 10

    Udp.write(",");                    //Escribimos en el puerto habilitado para
la conexion remota UPD

    String(humedad).toCharArray(ReplyBuffer, 10); //Para enviar el dato de
tipo string de la humedad lo convertimos en Char y lo colocamos en el arreglo

    Udp.write(ReplyBuffer);            //Escribimos en el puerto habilitado
para la conexion remota UPD el valor almacenado en el arreglo

    Udp.write("#");                    //Escribimos en el puerto habilitado para
la conexion remota UPD

    String(temperaturaEnGradosCelsius).toCharArray(ReplyBuffer, 10); //Para enviar el
dato de tipo string de la temperatura lo convertimos en Char y lo colocamos en el arreglo

    Udp.write(ReplyBuffer);            //Escribimos en el puerto habilitado
para la conexion remota UPD el valor almacenado en el arreglo

    Udp.write("$");                    //Escribimos en el puerto habilitado para
la conexion remota UPD

    String(ppm).toCharArray(ReplyBuffer, 10); //Para enviar el dato de tipo
string del amoniaco lo convertimos en Char y lo colocamos en el arreglo

    Udp.write(ReplyBuffer);            //Escribimos en el puerto habilitado
para la conexion remota UPD el valor almacenado en el arreglo

```

```
Udp.write("/");  
la conexion remota UPD
```

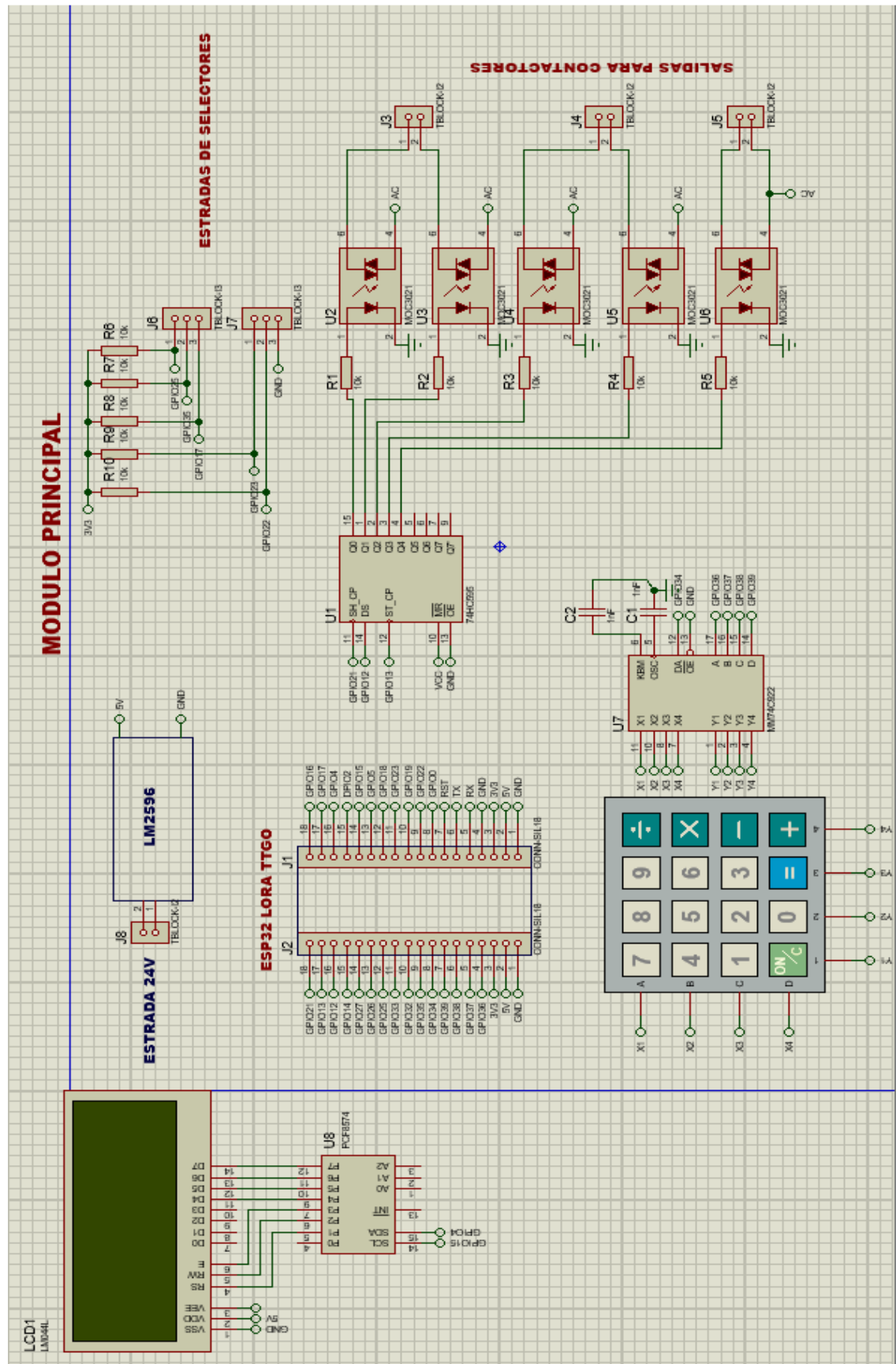
```
Udp.endPacket();  
delay(2000);  }}
```

```
//Escribimos en el puerto habilitado para
```

```
//Termina el paquete y lo envía
```

```
//Tiempo de espera en segundos
```

Anexo 2: Conexiones del nodo central o emisor en el software proteus



Anexo 3: Código de programación del nodo central o emisor

/*UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

PROYECTO DE INVESTIGACIÓN MODALIDAD TESIS

TEMA: SISTEMA DE CONTROL Y MONITOREO INALÁMBRICO PARA LA
GRANJA DAMIANCITO DEL CANTO PELILEO

ESTUDIANTE: KEVIN SEBASTIÁN ZÚÑIGA SÁNCHEZ

TUTO: ING. SANTIAGO ALTAMIRANO MELENDEZ

NODO CENTRAL-EMISOR*/

```
#include <SPI.h>           // Incluir la librería para la utilización del puerto SPI
#include <LoRa.h>           // Incluir la librería para la utilización de la tecnología LoRa
#include <WiFi.h>           // Incluir la librería WI-FI para la creación de una red
#include "AsyncUDP.h"       // Incluir la librería AsyncUDP para la activación del
                             protocolo UDP
#include <Wire.h>           // Incluir la librería Wire para la comunicación con dispositivos
                             I2C
#include <EEPROM.h>         // Incluir la librería EEPROM para la escribir y borrar datos
                             de la memoria
#include <LiquidCrystal_I2C.h> // Incluir la librería LiquidCrystal_I2C para la pantalla
                             LCD
#define lcdLoading 0        // Definir los pines de conexión de la pantalla I2C
#define lcdTHPS 1           // Definir los pines de conexión de la pantalla I2C
#define lcdIdeal 2          // Definir los pines de conexión de la pantalla I2C
#define lcdHeating 3        // Definir los pines de conexión de la pantalla I2C
#define lcdVentilation 4    // Definir los pines de conexión de la pantalla I2C
```

```
////////////////////////////////////////////////////////////////
```

```
//          CREACION DE LAS VARIABLES LORA
```

```
////////////////////////////////////////////////////////////////
```

```

boolean sensor_1 = false;    //Variable para los datos del nodo sensor 1
boolean sensor_2 = false;    //Variable para los datos del nodo sensor 2
int stateDisplay = lcdLoading; //Variable para la LCD
boolean actualizar = false;   //Variable para actualizar los datos
const int csPin = 18;        //Selección de chip de radio LoRa
const int resetPin = 14;     //Reiniciación de radio LoRa
const int irqPin = 26;       //Pin de interrupción de hardware
String outgoing;             //Mensaje saliente
uint32_t msgCount = 0;       //Recuento de mensajes salientes
byte localAddress = 0xAA;    //Dirección del este dispositivo
byte destination = 0xBB;     //Destino a enviar
long lastSendTime = 0;       //Tiempo del ultimo envío
int interval = 0;            //Intervalo entre envío de paquetes

////////////////////////////////////

// V A R I A B L E S   P A R A   R E C E P C I O N   D E S D E   E L   E M I S O R

////////////////////////////////////

String msgCount1, wsp_min, wsp_max, wmot1_e, wmot2_e, wmot3_e, wca1_e, wca2_e,
wdim_e, wforza;

int pos1, pos2, pos3, pos4, pos5, pos6, pos7, pos8, pos9;

////////////////////////////////////

//          V A R I A B L E S   P A R A   L A   R E C E P C I O N   D E   L O S   S E N S O R E S   U D P

////////////////////////////////////

String h1, t1, amoniaco, c1, h2, t2, c2;

int s1pos1, s1pos2, s1pos3, s1pos4, s2pos1, s2pos2, s2pos3;

String sensor1 = "";
String sensor2 = "";

```

```

float t1_i, t2_i, h1_i, h2_i, amoniaco2;

float estadosensor1 = false, estadosensor2 = false;

float t_prom, h_prom;

/////////////////////////////////////////////////////////////////
//          USUARIO Y CLAVE DE WIFI
/////////////////////////////////////////////////////////////////

const char *ssid = "AVICOLA";          //Nombre de la red
const char *password = "123456789..."; //Clave de la red

/////////////////////////////////////////////////////////////////
//          VARIABLES DE ACTUADORES Y DIMMER
/////////////////////////////////////////////////////////////////

uint8_t dim_e = 0;
uint8_t sp_min = 0;
uint8_t sp_max = 0;
uint8_t aut_man = 0;
uint8_t dimer = 0;
uint8_t mot1_e = false;
uint8_t mot2_e = false;
uint8_t mot3_e = false;
uint8_t ca1_e = false;
uint8_t ca2_e = false;
uint8_t acweb = true;

/////////////////////////////////////////////////////////////////
//          INGRESO DE DATOS POR TECLADO
/////////////////////////////////////////////////////////////////

AsyncUDP udp;          // Inicialización del protocolo UDP

```

```

char puls;

String cadena, cadena1, cadena3;

int numero = 0, a = 0, numero1 = 0, a1 = 0 , numero3 = 0, a3 = 0;

const byte lineas = 4;

const byte columnas = 4;

char teclas[] = {'A', 'B', 'C', 'D', '3', '6', '9', '#', '2', '5', '8', '0', '1', '4', '7', '*'};

int pines_teclado[] = {36, 37, 38, 39, 34};

boolean estados[] = {0, 0, 0, 0, 0};

int pulsadores[] = {25, 35, 17, 23, 22}; //PULSADORES

int dat = 12;

int clk = 21;

int set = 13;

LiquidCrystal_I2C lcd(0x27, 20, 4); //Creación de la dirección de la LCD

unsigned long tiempoactual = 0; //Tiempos de espera la comunicacion UDP
unsigned long tiempoactua2 = 0; //Tiempos de espera la comunicacion UDP
unsigned long tiempoactua3 = 0; //Tiempos de espera la comunicacion UDP

/////////////////////////////////////////////////////////////////
//                                VOID SETUP                                //
/////////////////////////////////////////////////////////////////

void setup() {

  Serial.begin(115200);           //Velocidad a la que se va comunicar con el host
  while (!Serial);               //Impresión en el puerto serial

  //////////////////////////////////////
  //          CONFIGURACION INICIAL DE LA COMUNICACION LORA          //
  //////////////////////////////////////

  Serial.println("LoRa Duplex"); //Impresión de mensaje

```



```

LoRa.setPins(csPin, resetPin, irqPin); //Configuración de pines set CS, reset, IRQ pin
if (!LoRa.begin(915E6)) { //Configuración de la frecuencia a trabajar 915MHz
  Serial.println("LoRa NO INICIO. REvisa CONEXION."); //Impresión de mensaje
  while (true); //Condición de conexión
} Serial.println("LoRa INICIADO."); //Impresión de mensaje

```

```

/////////////////////////////////////////////////////////////////

```

```

// CARATULA

```

```

/////////////////////////////////////////////////////////////////

```

```

lcd.init(); //Inicializar el LCD
lcd.backlight(); //Encender la luz de fondo.
lcd.setCursor(0, 0); //Ubicación del cursor en la LCD
lcd.print("*****"); //Impresión de asteriscos
lcd.setCursor(0, 1); //Ubicación del cursor en la LCD
lcd.print("*****AVICOLA*****"); //Impresión de mensaje
lcd.setCursor(0, 2); //Ubicación del cursor en la LCD
lcd.print("*****DAMIANCITO*****"); //Impresión de mensaje
lcd.setCursor(0, 3); //Ubicación del cursor en la LCD
lcd.print("*****"); //Impresión de asteriscos
delay(500); //Tiempo de espera en segundos

```

```

/////////////////////////////////////////////////////////////////

```

```

// GRABAR DATOS EN LA MEMORIA EEPROM

```

```

/////////////////////////////////////////////////////////////////

```

```

EEPROM.begin(10); //Inicialización de la memoria EEPROM
if (EEPROM.read(0) > 99) //Lectura de datos
EEPROM.write(0, 99); //Escritura de datos
if (EEPROM.read(1) > 99) //Lectura de datos

```

```

EEPROM.write(1, 99); //Escritura de datos

if (EEPROM.read(3) > 99) //Lectura de datos

EEPROM.write(3, 99); //Escritura de datos

sp_min = EEPROM.read(0); //Lectura de datos de la variable sp_min
sp_max = EEPROM.read(1); //Lectura de datos de la variable sp_max
aut_man = EEPROM.read(2); //Lectura de datos de la variable aut_man
dimer = EEPROM.read(3); //Lectura de datos de la variable dimmer


////////////////////////////////////
//                               INICIALIZACION WIFI
////////////////////////////////////

WiFi.begin(ssid, password); //Conexión a la Red con la dirección y la clave
Serial.println();           //Impresión de mensaje
Serial.print("CONECTANDO "); //Impresión de mensaje
Serial.println(ssid);       //Impresión del nombre de la red
while (WiFi.status() != WL_CONNECTED) { //Estado de conexión a la red
    delay(500);              //Tiempo de espera en segundos
    Serial.print(".");       //Impresión de mensaje
}
Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid);
Serial.print("IP address: ");
Serial.println(WiFi.localIP());


for (int i = 0; i < 5; i++) {
    pinMode(pines_teclado[i], INPUT);
}

```

```

pinMode(dat, OUTPUT);

pinMode(clk, OUTPUT);

pinMode(set, OUTPUT);


digitalWrite(dat, 0);

digitalWrite(clk, 0);

digitalWrite(set, 0);


for (int i = 0; i < 5; i++) {
    pinMode(pulsadores[i], INPUT);
}

for (int i = 0; i < 5; i++) {
    digitalWrite(dat, estados[i]);
    digitalWrite(clk, 1);
    digitalWrite(clk, 0);
}


digitalWrite(set, 1);
digitalWrite(set, 0);

////////////////////////////////////

//      CONFIGURACION DEL PROTOCOLO UDP PARA EL NODO SENSOR 1

////////////////////////////////////


if (udp.listen(2000)) {                // Mensaje de recepción UDP
    udp.onPacket([](AsyncUDPPacket packet) {    // Verification del paquete UDP
        if (packet.remoteIP() == IPAddress(192, 168, 10, 125)) { //Asignación de red remota a
la que se conectas los nodos sensoriales
            byte sensor1Length = packet.length();        //Longitud del paquete
            String myString = (const char*)packet.data();    //Cambiar el tiempo de dato
            if (sensor1Length != packet.length()) {        //Verificar la longitud del paquete

```

```
        Serial.println("error: La longitud del mensaje es incorrecta"); //Impresión de
mensaje
```

```
        return; //Regresar a la función
```

```
    }
```

```
////////////////////////////////////////////////////////////////
```

```
//          RECEPCION DE LOS DATOS DEL NODO SENSOR 1
```

```
////////////////////////////////////////////////////////////////
```

```
    sensor1 = myString;
```

```
    s1pos1 = sensor1.indexOf('#');
```

```
    s1pos2 = sensor1.indexOf('$');
```

```
    s1pos3 = sensor1.indexOf('/');
```

```
    h1 = sensor1.substring(0, s1pos1);
```

```
    t1 = sensor1.substring(s1pos1 + 1, s1pos2);
```

```
    amoniaco = sensor1.substring(s1pos2 + 1, s1pos3);
```

```
    c1 = sensor1.substring(s1pos3 + 1, sensor1.length());
```

```
    t1_i = t1.toFloat();
```

```
    h1_i = h1.toFloat();
```

```
    amoniaco2 = amoniaco.toFloat();
```

```
    Serial.print(packet.remoteIP());
```

```
    Serial.print((" humedad = "));
```

```
    Serial.print(h1);
```

```
    Serial.print((" temperatura = "));
```

```
    Serial.print(t1);
```

```
    Serial.print((" amoniaco = "));
```

```
    Serial.println(amoniaco);
```

```
    }
```

```
////////////////////////////////////////////////////////////////
```

```
//          CONFIGURACION DEL PROTOCOLO UDP PARA EL NODO SENSOR 2
```

```
////////////////////////////////////////////////////////////////
```

```

    if (packet.remoteIP() == IPAddress(192, 168, 10, 126)) { //Asignación de red remota a
la que se conectas los nodos sensoriales

```

```

        byte sensor2Length = packet.length();           //Longitud del paquete

```

```

        String myString = (const char*)packet.data();    //Cambiar el tiempo de dato

```

```

        if (sensor2Length != packet.length()) {          //Verificar la longitud del paquete

```

```

            Serial.println("error: La longitud del mensaje es incorrecta");//Impresión de
mensaje

```

```

            return;                                     //Regresar a la función

```

```

        }

```

```

////////////////////////////////////

```

```

//                                     RECEPCION DE LOS DATOS DEL NODO SENSOR 2

```

```

////////////////////////////////////

```

```

        sensor2 = myString;

```

```

        s2pos1 = sensor2.indexOf('=');

```

```

        s2pos2 = sensor2.indexOf(' ');

```

```

        h2 = sensor2.substring(0, s2pos1);

```

```

        t2 = sensor2.substring(s2pos1 + 1, s2pos2);

```

```

        c2 = sensor2.substring(s2pos2 + 1, sensor2.length());

```

```

        t2_i = t2.toFloat();

```

```

        h2_i = h2.toFloat();

```

```

        Serial.print(packet.remoteIP());

```

```

        Serial.print(" humedad = ");

```

```

        Serial.print(h2);

```

```

        Serial.print(" temperatura = ");

```

```

        Serial.println(t2);

```

```

    });}

```

```

    lcd.clear();}

```

```

////////////////////////////////////

```

```

//                                     VOID LOOP

```

```

////////////////////////////////////

```

```

void loop() {

inicio:          // Funcion inicio

    puls = teclado(); // Ingreso de datos por el teclado

    enviarecibeLORA();// Metodo para recibir y enviar datos

////////////////////////////////////////////////////////////////

// CONDICIÓN PARA VERIFICAR QUE SE CONECTEN LOS NODOS
// SENSORIALES AL NODO CENTRAL

// EN CASO DE NO CONECTAR SE REINICIA EL NODO CENTRAL E INDICA EL
// MENSAJE "NO CONECTADO"

////////////////////////////////////////////////////////////////

    if (millis() >= tiempoactual + 500) {

        tiempoactual = millis();

        recibirdatosUDP();

    }

    if (millis() >= tiempoactua2 + 3000) {

        if (WiFi.status() != WL_CONNECTED) {

            Serial.print("#####");

            Serial.println("NO CONECTADO!!");

            ESP.restart();}

////////////////////////////////////////////////////////////////

// CUANDO LA CONEXION DE LOS NODOS SENSORIALES ES EXITOSA
// RECIBE LOS DATOS DE ESTOS

// Y LOS IMPRIME EN AL PANTALLA

////////////////////////////////////////////////////////////////

    tiempoactua2 = millis();                //Tiempo de para cada nodo de enviar datos

    sensor_1 = (t1_i != 0 || h1_i != 0 || amoniaco2 != 0);//Recibe los datos del nodo sensor 1

    sensor_2 = (t2_i != 0 || h2_i != 0);      //Recibe los datos del nodo sensor 2

```

```

    if (sensor_1 == true && sensor_2 == true) { // Condición de para saber el estado de los
nodos sensoriales ambos nodos funcionan.

        t_prom = (t1_i + t2_i) / 2;          //Promedio de los valores de los sensores de
temperatura

        lcd.setCursor(0, 0);                //Posición del cursor

        lcd.print("T1:" + String(int(t1_i)) + " H1:" + String(int(h1_i)) + " Nh3:" +
String((amoniaco)));//Impresion de mensaje

        h_prom = (h1_i + h2_i) / 2;          //Promedio de los valores de los sensores de
Humedad

        lcd.setCursor(0, 1);                //Position del cursor

        lcd.print("T2:" + String(int(t2_i)) + " H2:" + String(int(h2_i)) + " TP:" +
String(int(t_prom)) + " ");//Impresión de mensaje

        lcd.setCursor(0, 2);                //Posición del cursor

        if (aut_man) {

            lcd.print("Tmax:" + String(sp_max) + " Tmin:" + String(sp_min) + " AUT
");//Impresión de mensaje modo automático

        } else {

            lcd.print("Tmax:" + String(sp_max) + " Tmin:" + String(sp_min) + " MAN
");//Impresión de mensaje modo manual

        } }

    else if (sensor_1 == true) { // Condición de para saber el estado del nodo sensor 1
funciona el sistema normalmente

        t_prom = t1_i;

        lcd.setCursor(0, 0);                //Posición del cursor

        lcd.print("SENSOR DOS OFF ");      //Impresión de mensaje

        lcd.setCursor(0, 1);                //Posición del cursor

        lcd.print("T1:" + String(int(t1_i)) + " H1:" + String(int(h1_i)) + " Nh3:" +
String((amoniaco)));//Impresión de mensaje

        lcd.setCursor(0, 2);                //Posición del cursor

        if (aut_man) {

            lcd.print("Tmax:" + String(sp_max) + " Tmin:" + String(sp_min) + " AUT
");//Impresión de mensaje modo automático

```

```

    } else {

        lcd.print("Tmax:" + String(sp_max) + " Tmin:" + String(sp_min) + " MAN
");//Impresion de mensaje modo manual

    }

    else if (sensor_2 == true ) { // Condición de para saber el estado del nodo sensor 1
funciona el sistema normalmente

        t_prom = t2_i;

        lcd.setCursor(0, 0);          //Posición del cursor

        lcd.print("SENSOR UNO OFF    ");    //Impresión de mensaje

        lcd.setCursor(0, 1);          //Posición del cursor

        lcd.print("T2:" + String(int(t2_i)) + " H2:" + String(int(h2_i)) + "    " );//Impresión
de mensaje

        lcd.setCursor(0, 2);          //Posición del cursor

        if (aut_man) {

            lcd.print("Tmax:" + String(sp_max) + " Tmin:" + String(sp_min) + " AUT ");//
Impresión de mensaje modo automático

        } else {

            lcd.print("Tmax:" + String(sp_max) + " Tmin:" + String(sp_min) + " MAN
");//Impresión de mensaje modo manual

        }

        else if (aut_man) {

            lcd.setCursor(0, 0); lcd.print("*****");//Impresión de mensaje

            lcd.setCursor(0, 1); lcd.print("*****AVICOLA*****");//Impresión de mensaje

            lcd.setCursor(0, 2); lcd.print("*****DAMIANCITO*****");//Impresión de mensaje

            lcd.setCursor(0, 3); lcd.print("*****");//Impresión de mensaje

        }

    }

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

// INSTRUCCIONES PARA LA ACTIVACION DE LA CALEFACCION EN MODO
AUTOMATICO

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```



```

if ((sensor_1 || sensor2 ) && aut_man) {

    if (t_prom < sp_min) {

        Serial.println("-----CALEFACCION");

        stateDisplay = lcdHeating;

        controlDisplay();

        boolean flag = false;

        for (int x = 0; x < 5; x++) {

            if (!digitalRead(pulsadores[x])) {

                //estados[x] = 1;

                flag = true;

            } else { }

        } if (flag) {

            for (int x = 0; x < 5; x++) {

                if (!digitalRead(pulsadores[x])) {

                    estados[x] = 1;

                    //flag = true;

                } else {

                    estados[x] = 0;}}

        } else {

            estados[0] = 0;

            estados[1] = 0;

            estados[2] = 0;

            estados[3] = 1;

            estados[4] = 1;

        }for (int i = 0; i < 5; i++) {

            digitalWrite(dat, estados[i]);

            digitalWrite(clk, 1);

            digitalWrite(clk, 0);

        }

        digitalWrite(set, 1);
    }
}

```

```

        digitalWrite(set, 0);

    }

////////////////////////////////////

// INSTRUCCIONES PARA LA ACTIVACION DE LA VENTILACION EN MODO
AUTOMATICO

////////////////////////////////////

    else if (t_prom > sp_max) {

        Serial.println("-----VENTILACION");

        stateDisplay = lcdVentilation;

        controlDisplay();

        boolean flag = false;

        for (int x = 0; x < 5; x++) {

            if (!digitalRead(pulsadores[x])) {

                flag = true;

            } else {}

        }

        if (flag) {

            for (int x = 0; x < 5; x++) {

                if (!digitalRead(pulsadores[x])) {

                    estados[x] = 1;

                } else {

                    estados[x] = 0;

                }

            }

        } else {

            estados[0] = 1;

            estados[1] = 1;

            estados[2] = 1;

            estados[3] = 0;

            estados[4] = 0;

        }

        for (int i = 0; i < 5; i++) {

            digitalWrite(dat, estados[i]);

            digitalWrite(clk, 1);

        }

    }

```

```

        digitalWrite(clk, 0);
    }

    digitalWrite(set, 1);
    digitalWrite(set, 0);
}

/////////////////////////////////////////////////////////////////
// INSTRUCCIONES PARA LAS CONDICIONES IDEALES DE LOS PARAMETROS
/////////////////////////////////////////////////////////////////

else {
    if (t_prom < sp_max && t_prom > sp_min ) {
        Serial.println("-----IDEAL");
        stateDisplay = lcdIdeal;
        controlDisplay();
        boolean flag = false;
        for (int x = 0; x < 5; x++) {
            if (!digitalRead(pulsadores[x])) {
                //estados[x] = 1;
                flag = true;
            } else {
                //estados[x] = 0;
            }
        }
        if (flag) {
            for (int x = 0; x < 5; x++) {
                if (!digitalRead(pulsadores[x])) {
                    estados[x] = 1;
                    //flag = true;
                } else {
                    estados[x] = 0;
                }
            }
        } else {

```

```

    estados[0] = 0;
    estados[1] = 0;
    estados[2] = 0;
    estados[3] = 0;
    estados[4] = 0;
}for (int i = 0; i < 5; i++) {
    digitalWrite(dat, estados[i]);
    digitalWrite(clk, 1);
    digitalWrite(clk, 0);
}
digitalWrite(set, 1);
digitalWrite(set, 0);
} }

////////////////////////////////////

// INSTRUCCIONES PARA LA ACTIVACION O DESACTIVACION MANUAL DE
// LOS ACTUADORES

////////////////////////////////////

    } else if (aut_man == 0) {
        lcd.setCursor(0, 3);
        lcd.print("          ");
    }

    t1_i = t2_i = h1_i = h2_i = amoniaco2 = 0;
}

puls = teclado();
if (puls != 0) {
    while (1) {
        enviarecibeLORA();
        if (millis() >= tiempoactual + 10000) {
            goto inicio;
        }
    }
}

```

```

puls = teclado();
if (puls != 0) {
    tiempoactual = millis();
}if (puls == 'D') {
    goto inicio;
}
menu();
switch (puls) {
    case '1':
        break;
    case '2' :
        automatico2sensores();
        break;
    case '3' :
        ingresoportecladoDIM();
        return;
        break;
}}}}

```

```

////////////////////////////////////////////////////////////////

```

```

// METODO PARA ENVAIR Y RECIBIR LOS DATOS MEDIANTE LA
COMUNICACION LoRa

```

```

////////////////////////////////////////////////////////////////

```

```

void enviarecibeLORA() {
    onReceive(LoRa.parsePacket()); // Establer al módulo LoRa como receptor
    if (millis() - lastSendTime > interval) { //Condición para la recepción de mensajes
        sendMessage(); // Envío del mensaje

        Serial.println("Sending " + message + "numero de paquete enviado" +
msgCount);//Impresion de mensaje

        lastSendTime = millis(); //Tiempo de envió del mensaje
    }
}

```

```

        interval = random(2000) + 3000; //Intervalo de espera para el envío
    }
}

////////////////////////////////////

// CREACION DEL PAQUETE PARA ENVIAR LOS DATOS MEDIANTE LA
// COMUNICACION LoRa

////////////////////////////////////

void sendMessage() {

    String outgoing = String(msgCount) + "a" + String(t1) + "b" + String(t2) + "c" +
    String(h1) + "d" + String(h2) + "e" + String(amoniaco)

    + "f" + String(sp_min) + "g" + String(sp_max) + "h" + String(estados[0]) + "i" +
    String(estados[1]) + "j" + String(estados[2]) + "k" + String(estados[3])

    + "l" + String(estados[4]) + "m" + String(dimer) + "n" + String(t_prom) + "o" +
    String(h_prom) + "p" + String(aut_man) + "q" + String(acweb);

    acweb = false;

    LoRa.beginPacket();           // Inicio del paquete

    LoRa.write(destination);      // Dirección del receptor

    LoRa.write(localAddress);     // Dirección del emisor

    LoRa.write(msgCount);         // Dirección del mensaje

    LoRa.write(outgoing.length()); // longitud del mensaje

    LoRa.print(outgoing);         // Dirección del payload

    LoRa.endPacket();             // Finalización del paquete y envío

    msgCount++;                  // Incrementación del ID del paquete

}

void onReceive(int packetSize) { // Función para cuando se recibe un paquete

    if (packetSize == 0) return; // Si no se envía el paquete retorna

    int recipient = LoRa.read();  // Dirección del receptor

    byte sender = LoRa.read();    // Dirección del emisor

    int incomingMsgId = LoRa.read(); // incoming msg ID

    int incomingLength = LoRa.read(); // incoming longitud del mensaje

```

```

String incoming = "";

while (LoRa.available()) {      //Habilitación de LoRa

    incoming += (char)LoRa.read(); //Lectura de datos
}

if (incomingLength != incoming.length()) { //Comprobar la longitud del paquete

    Serial.println("error: La longitud del mensaje no coincide"); //Impresión de mensaje

    return;                // Si no se envia el paquete retorna
}

if (recipient != localAddress && recipient != 0xBB) {

    Serial.println("Este mensaje no es correcto.");

    return;
}

for (int i = 0; i < incoming.length(); i++) {

    char xx = incoming.charAt(i);

    if (!(xx >= 'A' && xx <= 'T' || xx == '.' || (xx >= '0' && xx <= '9'))) {

        Serial.println();

        return;

    }

    Serial.print(xx);

}

////////////////////////////////////

// ALMACENAMIENTO DE LOS DATOS RECIVIDOS MEDIANTE LA
// COMUNICACION LoRa

////////////////////////////////////

pos1 = incoming.indexOf('A');
pos2 = incoming.indexOf('B');
pos3 = incoming.indexOf('C');
pos4 = incoming.indexOf('D');

```

```

pos5 = incoming.indexOf('E');
pos6 = incoming.indexOf('F');
pos7 = incoming.indexOf('G');
pos8 = incoming.indexOf('H');
pos9 = incoming.indexOf('I');
msgCount1 = incoming.substring(0, pos1);
wsp_min = incoming.substring(pos1 + 1, pos2);
wsp_max = incoming.substring(pos2 + 1, pos3);
wmot1_e = incoming.substring(pos3 + 1, pos4);
wmot2_e = incoming.substring(pos4 + 1, pos5);
wmot3_e = incoming.substring(pos5 + 1, pos6);
wca1_e = incoming.substring(pos6 + 1, pos7);
wca2_e = incoming.substring(pos7 + 1, pos8);
wdim_e = incoming.substring(pos8 + 1, pos9);
wforza = incoming.substring(pos9 + 1, incoming.length());

////////////////////////////////////

// ALMACENAMIENTO DE LOS DATOS RECIVIDOS MEDIANTE LA
COMUNICACION LoRa

// DE LA ASIGACION DE VALORES DE TEMPERATURAS MAXIMAS Y
MINIMAS Y GRABADO EN MEMORIA

////////////////////////////////////

if ((uint8_t)aut_man != (uint8_t)wforza.toInt()) {
    aut_man = (uint8_t)wforza.toInt(); EEPROM.put(2, aut_man); EEPROM.commit();
    if (aut_man == 0) {
        for (int i = 0; i < 5; i++) {
            estados[i] = 0;
            digitalWrite(dat, estados[i]);
            digitalWrite(clk, 1);
            digitalWrite(clk, 0);
        }
    }
}

```



```

    }

    digitalWrite(set, 1);

    digitalWrite(set, 0);

    Serial.println("ACTUADORES ENCERADOS!!");

} }

if (aut_man == 0) {

    if (sp_min != wsp_min.toInt())sp_min = (uint8_t) wsp_min.toInt(); EEPROM.put(0,
sp_min); EEPROM.commit();

    if (sp_max != wsp_max.toInt())sp_max = (uint8_t)wsp_max.toInt(); EEPROM.put(1,
sp_max); EEPROM.commit();

    if (dimer != wdim_e.toInt())dimer = (uint8_t)wdim_e.toInt(); EEPROM.put(3, dimer);
EEPROM.commit();

    estados[0] = wmot1_e.toInt();
    estados[1] = wmot2_e.toInt();
    estados[2] = wmot3_e.toInt();
    estados[3] = wca1_e.toInt();
    estados[4] = wca2_e.toInt();

    for (int i = 0; i < 5; i++) {

        digitalWrite(dat, estados[i]);

        digitalWrite(clk, 1);

        digitalWrite(clk, 0);

    }

    digitalWrite(set, 1);

    digitalWrite(set, 0);

}

////////////////////////////////////

// EMPRESION DE LOS DATOS RECIVIDOS Y ENVIAS MEDIANTE LA
COMUNICACION LoRa ENTRE EL NODO

// RECEPTOR Y EL EMISOR

```

```
////////////////////////////////////
```

```
Serial.print(("Packet No. = ")); Serial.println(msgCount1);  
Serial.print(("Temperatura minima=")); Serial.println(wsp_min);  
Serial.print(("Temperatura maxima=")); Serial.println(wsp_max);  
Serial.print(("MOTOR 1=")); Serial.println(wmot1_e);  
Serial.print(("MOTOR 2=")); Serial.println(wmot2_e);  
Serial.print(("MOTOR 3=")); Serial.println(wmot3_e);  
Serial.print(("CALEFACTOR 1=")); Serial.println(wca1_e);  
Serial.print(("CALEFACTOR 2=")); Serial.println(wca2_e);  
Serial.print(("DIMMER=")); Serial.println(wdim_e);  
Serial.print(("FORZADO=")); Serial.println(wforza);  
Serial.println("Received from: 0x" + String(sender, HEX));  
Serial.println("Sent to: 0x" + String(recipient, HEX));  
Serial.println("Message ID: " + String(incomingMsgId));  
Serial.println("Message length: " + String(incomingLength));  
Serial.println("Message: " + incoming);  
Serial.println("RSSI: " + String(LoRa.packetRssi()));  
Serial.println("Snr: " + String(LoRa.packetSnr()));  
Serial.println();  
}
```

```
////////////////////////////////////
```

```
//INICIALIZACION PARA RECIBIR LOS DATOS DE LOS NODOS  
SENSORIALES CON
```

```
//PROTOCOLO UDP
```

```
////////////////////////////////////
```

```
void recibirdatosUDP() {  
    udp.broadcastTo("DATOS", 2000);  
  
}
```

```

////////////////////////////////////
//      METODO TECLADO PARA EL INGRESO DE VALOR Y CONTROL DESDE
//      EL GALPON
////////////////////////////////////

char teclado() {
    if (digitalRead(34)) {
        int dato = 0;
        for (int i = 0; i < 4; i++) {
            dato |= digitalRead(pines_teclado[i]) << i;
        }
        Serial.println(teclas[dato]);
        while (digitalRead(34))delay(40);
        return teclas[dato];
    }
    return 0;
}

////////////////////////////////////
//      MENU
////////////////////////////////////

void menu() {
    lcd.setCursor(0, 0);
    lcd.print("*****MENU*****");
    lcd.setCursor(0, 1);
    lcd.print("1 => ");
    if (aut_man) lcd.print("AUTOMATICO   ");
    else lcd.print("MANUAL       ");
    lcd.setCursor(0, 2);
    lcd.print("2 => SET TEMPERATURA");
    lcd.setCursor(0, 3);
    lcd.print("3 => DIMMER      ");
}

```

```

}

////////////////////////////////////

//      ACTIVACION Y DESACTIVACION DE LOS ACTUADORES DE FORMA
MANUAL

////////////////////////////////////

void actuadoresmanual() {
    for (int x = 0; x < 5; x++) {
        if (!digitalRead(pulsadores[x])) {
            estados[x] = 1;
            if (x == 0) {
                estados[0] = 1;
                estados[1] = 0;
                estados[2] = 0;
                estados[3] = 0;
                estados[4] = 0;
            }
            if (x == 1) {
                estados[0] = 0;
                estados[1] = 1;
                estados[2] = 0;
                estados[3] = 0;
                estados[4] = 0;
            }
            if (x == 2) {
                estados[0] = 0;
                estados[1] = 0;
                estados[2] = 1;
                estados[3] = 0;
                estados[4] = 0;
            }
        }
    }
}

```

```

if (x == 3) {
    estados[0] = 0;
    estados[1] = 0;
    estados[2] = 0;
    estados[3] = 1;
    estados[4] = 0;
}

if (x == 4) {
    estados[0] = 0;
    estados[1] = 0;
    estados[2] = 0;
    estados[3] = 0;
    estados[4] = 1;
}

} else {

}

for (int i = 0; i < 5; i++) {
    digitalWrite(dat, estados[i]);
    digitalWrite(clk, 1);
    digitalWrite(clk, 0);
}

digitalWrite(set, 1);
digitalWrite(set, 0);
}
}

/////////////////////////////////////////////////////////////////

// INGRESO POR TECLADO EL VALOR DE PORCENTAJE DE INTENSIDAD
LUMINICA PARA EL DIMMER

```

//

```
void ingresoportecladoDIM() {  
    lcd.clear();  
vuelve:  
    do {  
        puls = teclado();  
        lcd.setCursor(0, 0);  
        lcd.print("INGRESE DE 0 A 100 %");  
        if (puls != 0) {  
            lcd.setCursor(0, 1);  
            cadena3 = cadena3 + puls ;  
            lcd.setCursor(0, 1);  
            lcd.print(cadena3);  
            a3 += 1 ;  
            delay(500);  
        }  
    }  
    while (a3 < 3 );  
    a3 = 0 ;  
    dimer = cadena3.toInt();// transforma de strin a numero  
    if ( dimer > 100) {  
        lcd.setCursor(0, 1);  
        lcd.print("VALOR NO ADMITIDO");  
        delay(1000);  
        lcd.clear();  
        cadena3 = "";  
        dimer = EEPROM.read(3);  
        goto vuelve;  
    }  
    else {
```

```

cadena3 = "";

lcd.clear ();

EEPROM.put(3, dimer);

EEPROM.commit();

while ( 1 ) {

    // dim_e = (dim_e * 255) / 100;

    //analogWrite(100, dim_e);

    String dim_e = wdim_e;

    lcd.setCursor(0, 0);

    lcd.print("LUMINOSIDAD");

    lcd.setCursor(0, 1);

    lcd.print("PORCENTAJE: " + String(dimer) + "%");

    puls = teclado();

    if ( puls == 'A' ) {

        lcd.setCursor(0, 1);

        lcd.print("          ");

        goto vuelve ;

    }

    if ( puls == 'D' ) {

        return;

    }

    enviarecibeLORA();

    if (millis() >= tiempoactual + 1000) {

        tiempoactual = millis();

        recibirdatosUDP();

    }

}

/////////////////////////////////////////////////////////////////

//METODO PARA EL INGRESO DE TEMPERATURAS MAXIMAS Y MINIMAS
DESDE LA HMI

/////////////////////////////////////////////////////////////////

```

```

void automatico2sensores() {

    lcd.clear();
ingreso1:

    cadena = "";
    cadena1 = "";

    do {

        puls = teclado();

        lcd.setCursor(0, 0);

        lcd.print("    PRIMERO    ");

        lcd.setCursor(0, 1);

        lcd.print("INGRESE TEMPERATURA ");

        lcd.setCursor(0, 2);

        lcd.print("    BAJA    ");

        if (puls != 0) {

            lcd.setCursor(0, 3);

            lcd.print("                ");

            cadena = cadena + puls ;

            lcd.setCursor(0, 3);

            lcd.print(cadena);

            a += 1 ;

        }

        delay(300);

    }

    while (a < 2 );

    a = 0 ;

    lcd.clear();
ingreso2:

    do {

```



```

puls = teclado();
lcd.setCursor(0, 0);
lcd.print("    SEGUNDO    ");
lcd.setCursor(0, 1);
lcd.print("INGRESE TEMPERATURA ");
lcd.setCursor(0, 2);
lcd.print("    ALTA    ");
if (puls != 0) {
    lcd.setCursor(0, 3);
    lcd.print("                ");
    cadena1 = cadena1 + puls ;
    lcd.setCursor(0, 3);
    lcd.print(cadena1);
    a1 += 1 ;
}
delay(300);
}
while (a < 2 && a1 < 2 );
a = 0 ;
a1 = 0;
sp_min = cadena.toInt();// transforma de strin a numero
sp_max = cadena1.toInt();// transforma de strin a numero
if ( sp_min > 100 || sp_max > 100) {
    lcd.setCursor(0, 3);
    lcd.print("ERROR");
    delay(1000);
    lcd.clear();
    cadena = "";
    sp_min = EEPROM.read(0);

```

```

    sp_max = EEPROM.read(1);

    goto ingreso1;
}

sp_min = cadena.toInt();// transforma de strin a numero
EEPROM.put(0, sp_min);
EEPROM.commit();

sp_max = cadena1.toInt();// transforma de strin a numero
EEPROM.put(1, sp_max);
EEPROM.commit();
}

/////////////////////////////////////////////////////////////////
//IMPRESION EN LA LCD DE LOS VALORES INGRESADOS DESDE LA HMI
/////////////////////////////////////////////////////////////////

void controlDisplay() {
    switch (stateDisplay) {
        case lcdLoading:
            stateDisplay = lcdTHPS;
            break;
        case lcdTHPS:
            stateDisplay = lcdTHPS;
            break;
        case lcdIdeal:
            displayIdeal();
            break;
        case lcdHeating:
            displayHeating();
            break;
        case lcdVentilation:
            displayVentilation();

```

```

        break;

    default:

        break;

    }

}

void displayLoading() {

    lcd.setCursor(1, 0); lcd.print("Avicola");

    lcd.setCursor(9, 1); lcd.print("de");

    lcd.setCursor(4, 3); lcd.print("damiancito");

}

void displayTemHumSetP() {

    lcd.setCursor(0, 0); lcd.print("T1:" + String(t1) + " T2:" + String(t2));

    lcd.setCursor(0, 1); lcd.print("H1:" + String(h1) + " H2:" + String(h2));

    lcd.setCursor(0, 2); lcd.print("T_B:" + String(sp_min) + " T_A:" + String(sp_max));

    lcd.setCursor(0, 3); lcd.print("Nh3:" + String(amoniaco));

}

void displayIdeal() {

    lcd.setCursor(0, 3);

    lcd.print("    Temp Ideal    ");

}

void displayVentilation() {

    lcd.setCursor(0, 3);

    lcd.print("  Ventilacion ON  ");

}

void displayHeating() {

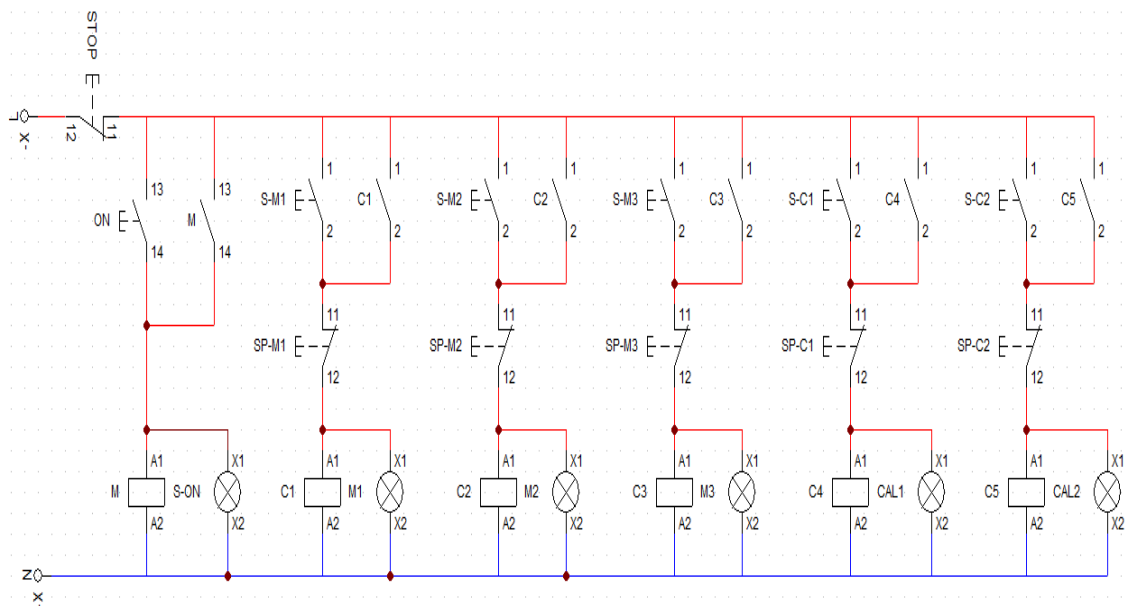
    lcd.setCursor(0, 3);

    lcd.print("  Calefaccion ON  ");

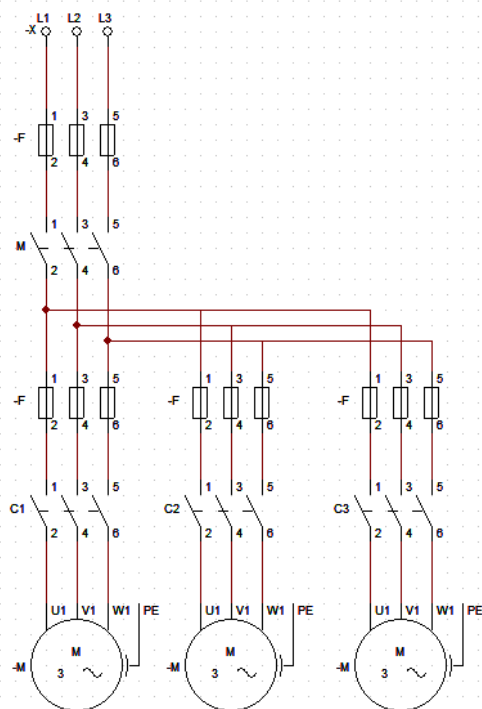
}

```

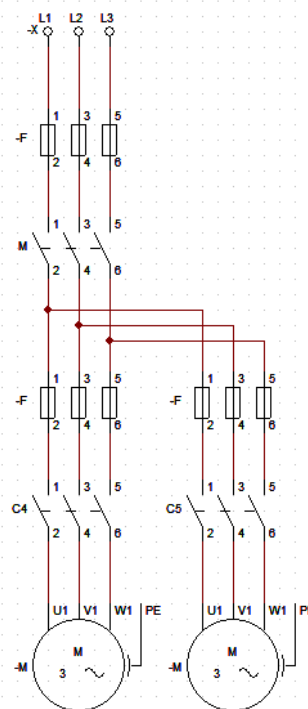
Anexo 4: Conexiones eléctricas del gabinete



EXTRACTORES



CALEFACTORES



Anexo 5: Código de programación del nodo receptor

```
/*UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL
PROYECTO DE INVESTIGACIÓN MODALIDAD TESIS
TEMA: SISTEMA DE CONTROL Y MONITOREO INALÁMBRICO PARA LA
GRANJA DAMIANCITO DEL CANTO PELILEO
ESTUDIANTE: KEVIN SEBASTIÁN ZÚÑIGA SÁNCHEZ
TUTO: ING. SANTIAGO ALTAMIRANO MELENDEZ

NODO RECEPTOR*/

#include <SPI.h>          // Incluir la libreria para la utlización del puerto SPI
#include <LoRa.h>          // Incluir la libreria para la utlización de la tecnología LoRa
#include <WiFi.h>          // Incluir la libreria WI-FI para la creación de una red
#include <PubSubClient.h> // Incluir la libreria PubSubClient para el protocolo mqtt
String MQTTPayload;      // Creación de las variables tipo topico para mqtt
String MQTTTopic;        // Creación de las variables tipo subscriptor para mqtt
byte* buffer;            // Creación de las variables de tipo array de bytes
boolean Rflag = false;   // Creación de una vafiable global
int r_len;               // Creación de una vafiable global

/////////////////////////////////////////////////////////////////

//CREACION DE LOS TOPICOS PARA LA UTILIZACION DEL PROTOCOLO
MQTTCADA VARAIBLE

//QUE VAYA A UTILIZAR DEBE TENER SU TOPICO EN NUESTRO CASO LAS
QUE SE INIDCA A CONTINUACION

/////////////////////////////////////////////////////////////////

int resultado;

const char* topic_webdimmer = "web/dimmer";    // Topic MQTT dimmer
const char* topic_webminima = "web/minima";    // Topic MQTT temp baja y alta
const char* topic_webmaxima = "web/maxima";    // Topic MQTT temp baja y alta
const char* topic_webmotor1 = "web/motor1";    // Topic MQTT del extraxtor 1
const char* topic_webmotor2 = "web/motor2";    // Topic MQTT del extraxtor 2
const char* topic_webmotor3 = "web/motor3";    // Topic MQTT del extraxtor 3
```

```

const char* topic_webcalefactor1 = "web/calefactor1";// Topic MQTT del calefactor 1
const char* topic_webcalefactor2 = "web/calefactor2";// Topic MQTT del calefactor 2
const char* topic_onoff = "web/onoff";      // Topic MQTT del activado o sesactivado

```

```

////////////////////////////////////

```

```

//          USUARIO Y CLAVE DE WIFI

```

```

////////////////////////////////////

```

```

const char *ssid = "AVICOLA_DAMIANCITO"; //Nombre de la red
const char *password = "0603175084";    //Clave de la red
const char* mqtt_server = "3.235.79.216"; //Dirección del servidor mqtt

```

```

////////////////////////////////////

```

```

//CONFIGURACION DE PINES LORA

```

```

////////////////////////////////////

```

```

const int csPin = 18;      //Selección de chip de radio LoRa
const int resetPin = 14;   //Reiniciación de radio LoRa
const int irqPin = 26;     //Pin de interrupción de

```

```

////////////////////////////////////

```

```

// CREACION DE LAS VARIABLES PARA LA CONEXION DESDE LA GRANJA

```

```

////////////////////////////////////

```

```

String outgoing;          // outgoing message

String msgCount1, t1, t2, h1, h2, amoniaco, sp_min, sp_max, mot1_e, mot2_e, mot3_e,
ca1_e, ca2_e, dim_e, t_prom, h_prom, aut_man, acweb;

int pos1, pos2, pos3, pos4, pos5, pos6, pos7, pos8, pos9, pos10, pos11, pos12, pos13,
pos14, pos15, pos16, pos17;

```

```

////////////////////////////////////

```

```

//CREACION DE LAS VARIABLES PARA LA CONEXION DESDE LA HMI A LA
GRANJA

```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
uint8_t wsp_min = 0;
uint8_t wsp_max = 0;
uint8_t wmot1_e = false;
uint8_t wmot2_e = false;
uint8_t wmot3_e = false;
uint8_t wca1_e = false;
uint8_t wca2_e = false;
uint8_t wdim_e = 0;
uint8_t wforza = false;
boolean flag_sp_min = false;
boolean flag_sp_max = false;
boolean flag_mot1_e = false;
boolean flag_mot2_e = false;
boolean flag_mot3_e = false;
boolean flag_ca1_e = false;
boolean flag_ca2_e = false;
boolean flag_dim_e = false;
boolean flag_aut_man = false;
int msgCount = 0;      //Recuento de mensajes salientes
byte localAddress = 0xBB; //Dirección del este dispositivo
byte destination = 0xAA; //Destino a enviar
long lastSendTime = 0;  //Tiempo del ultimo envio
int interval = 2000;    //Intervalo entre envio de paquetes
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
// CONFIGURACION DEL SUBSCRIPTOE Y PUBLICADOR DE LA ESP - 32
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
WiFiClient esp32Client;
PubSubClient mqttClient(esp32Client);
```

```

unsigned long lastMsg = 0;

#define MSG_BUFFER_SIZE (10)

char msg[MSG_BUFFER_SIZE];

int value = 0;

/////////////////////////////////////////////////////////////////
// C O N E X I O N   W I F I
/////////////////////////////////////////////////////////////////

void setup_wifi() {

  delay(10);                //Tiempo de espera en segundos

  Serial.println();          //Impresion de mensaje

  Serial.print("CONECTANDO "); //Impresion de mensaje

  Serial.println(ssid);      //Impresion del nombre de la red

  WiFi.mode(WIFI_STA);      //Coneccion a una direccion IP estática

  WiFi.begin(ssid, password); //Coneccion a la Red con la dirección y la clave

  while (WiFi.status() != WL_CONNECTED) { //Estado de conexion a al red

    digitalWrite (25, LOW);    //Apagado del pin 25 o led

    delay(500);                //Tiempo de espera en segundos

    Serial.print(".");         //Impresion de mensaje

  }

  randomSeed(micros());      //Tiempo de espera en segundos para la conexión

  Serial.println("");         //Impresion de mensaje

  Serial.println("WiFi conectado"); //Impresion de mensaje

  Serial.println("IP address: "); //Impresion de la direccion IP

  Serial.println(WiFi.localIP()); //Impresion de mensaje en puerto serial

  digitalWrite (25, HIGH);    //Encendido del pin 25 o led

  delay(2000);                //Tiempo de espera en segundos

}

/////////////////////////////////////////////////////////////////

```



```
// RECEPCION DE LOS DATOS POR MEDIO DEL PROTOCOLO MQTT Y
ALMACENAMIENTO EN UN VECTOR
```

```
////////////////////////////////////////////////////////////////
```

```
char dato1[10];
```

```
char dato2[10];
```

```
char dato3[10];
```

```
char dato4[10];
```

```
char dato5[10];
```

```
char dato6[10];
```

```
char dato7[10];
```

```
char dato8[10];
```

```
char dato9[10];
```

```
char dato10[10];
```

```
char dato11[10];
```

```
char dato12[10];
```

```
char dato13[10];
```

```
char dato14[10];
```

```
char dato15[10];
```

```
char dato16[10];
```

```
void callback(char* topic, byte* payload, unsigned int length) {
```

```
    String string;
```

```
    Serial.print("Mensaje recibido [");
```

```
    Serial.print(topic);
```

```
    Serial.println("] ");
```

```
    for (int i = 0; i < length; i++) {
```

```
        string += ((char)payload[i]);
```

```
    }
```

```
    Serial.println();
```

```
    int resultado = string.toInt();
```

```

////////////////////////////////////
// CONDICIONES PARA VERIFICAR EL ESTADO DE LOS ACTUADORES EN
ESTE CASO EXTRACTOR 1
////////////////////////////////////

if ( strcmp(topic, topic_webmotor1) == 0 ) {

    if ((char)payload[0] == '1') {

        Serial.println("M O T O R UNO E N C E D I D O"); wmot1_e =
true;

        flag_mot1_e = true;
    } if ((char)payload[0] == '0') {

        Serial.println("M O T O R UNO A P A G A D O"); wmot1_e = false;

        flag_mot1_e = true;
    }
}

////////////////////////////////////
// CONDICIONES PARA VERIFICAR EL ESTADO DE LOS ACTUADORES EN
ESTE CASO EXTRACTOR 2
////////////////////////////////////

if ( strcmp(topic, topic_webmotor2) == 0 ) {

    if ((char)payload[0] == '1') {

        Serial.println("M O T O R DOS E N C E D I D O"); wmot2_e =
true;

        flag_mot2_e = true;
    } if ((char)payload[0] == '0') {

        Serial.println("M O T O R DOS A P A G A D O"); wmot2_e = false;

        flag_mot2_e = true;
    }
}

////////////////////////////////////

```

```
// CONDICIONES PARA VERIFICAR EL ESTADO DE LOS ACTUADORES EN
ESTE CASO EXTRACTOR 3
```

```
////////////////////////////////////////////////////////////////
```

```
if ( strcmp(topic, topic_webmotor3) == 0 ) {
    if ((char)payload[0] == '1') {
        Serial.println("M O T O R  TRES  E N C E D I D O"); wmot3_e =
true;
        flag_mot3_e = true;
    }
    if ((char)payload[0] == '0') {
        Serial.println("M O T O R  TRES   A P A G A D O"); wmot3_e =
false;
        flag_mot3_e = true;
    }
}
```

```
////////////////////////////////////////////////////////////////
```

```
//CONDICIONES PARA VERIFICAR EL ESTADO DE LOS ACTUADORES EN
ESTE CASO CALEFACTOR Y VENTOLERAS 1
```

```
////////////////////////////////////////////////////////////////
```

```
if ( strcmp(topic, topic_webcalefactor1) == 0 ) {
    if ((char)payload[0] == '1') {
        Serial.println("C A L E F A C T O R  UNO  E N C E D I D
O"); wcal_e = true;
        flag_cal_e = true;
    }
    if ((char)payload[0] == '0') {
        Serial.println("C A L E F A C T O R  UNO  A P A G A D O ");
wcal_e = false;
        flag_cal_e = true;
    }
}
```

```

////////////////////////////////////
//CONDICIONES PARA VERIFICAR EL ESTADO DE LOS ACTUADORES EN
ESTE CASO CALEFACTOR Y VENTOLERAS 2
////////////////////////////////////

if ( strcmp(topic, topic_webcalefactor2) == 0 ) {

    if ((char)payload[0] == '1') {

        Serial.println("C A L E F A C T O R 2 E N C E D I D O ");
        wca2_e = true;

        flag_ca2_e = true;

    }

    if ((char)payload[0] == '0') {

        Serial.println("C A L E F A C T O R 2 A P A G A D O");
        wca2_e = false;

        flag_ca2_e = true;

    }

}

////////////////////////////////////

// CONDICIONES PARA VERIFICAR EL ESTADO DEL BOTON DE QUE PERMITE
EL USO DE LA HMI
////////////////////////////////////

if ( strcmp(topic, topic_onoff) == 0 ) {

    if ((char)payload[0] == '1') {

        Serial.println("FORZADO ON"); wforza = true;

        flag_aut_man = true;

    }

    if ((char)payload[0] == '0') {

        Serial.println("FORZADO OFF"); wforza = false;

        flag_aut_man = true;

    }

}

```

```

}

/////////////////////////////////////////////////////////////////

//CONDICIONES PARA VERIFICAR EL ESTADO DEL DIMMER

/////////////////////////////////////////////////////////////////

if ( strcmp(topic, topic_webdimmer) == 0 ) {

    Serial.println("D I M M E R ");

    Serial.println("          D I M M E R:" + String(resultado));

    wdim_e = resultado;

    flag_dim_e = true;

}

/////////////////////////////////////////////////////////////////

//CONDICIONES PARA VERIFICAR EL ESTADO DE LA TEMPERATURA
MINIMA

/////////////////////////////////////////////////////////////////

if ( strcmp(topic, topic_webminima) == 0 ) {

    Serial.println("T E M P E R A T U R A   B A J A ");

    Serial.println("          B A J A:" + String(resultado));

    wsp_min = resultado;

    flag_sp_min = true;

}

/////////////////////////////////////////////////////////////////

//CONDICIONES PARA VERIFICAR EL ESTADO DE LA TEMPERATURA
MAXIMA

/////////////////////////////////////////////////////////////////

if ( strcmp(topic, topic_webmaxima) == 0 ) {

    Serial.println("T E M P E R A T U R >>>>> A A L T A ");

    Serial.println("          A L T A:" + String(resultado));

    wsp_max = resultado;

```

```

    flag_sp_max = true;
}
}

////////////////////////////////////

// METODO PARA RECONECTAR AL SERVIDOR MEDIANTE EL PROTOCOLO
// MQTT SI ESTE SE DESCONECTA

////////////////////////////////////

void reconnect() {
    while (!mqttClient.connected()) {
        Serial.print("Attempting MQTT connection...");

        // Create a random client ID
        String clientId = "AVICOLA";

        // clientId += String(random(0xffff), HEX);

        // Attempt to connect
        if (mqttClient.connect(clientId.c_str())) {
            Serial.println("connected");

            digitalWrite (25, HIGH);

            mqttClient.subscribe(topic_webdimmer);
            mqttClient.subscribe(topic_webminima);
            mqttClient.subscribe(topic_webmaxima);
            mqttClient.subscribe(topic_webmotor1);
            mqttClient.subscribe(topic_webmotor2);
            mqttClient.subscribe(topic_webmotor3);
            mqttClient.subscribe(topic_webcalefactor1);
            mqttClient.subscribe(topic_webcalefactor2);
            mqttClient.subscribe(topic_onoff);
        } else {
            Serial.print("failed, rc=");

```

```

Serial.print(mqttClient.state());

Serial.println(" try again in 5 seconds");

digitalWrite (25, LOW);

ESP.restart();

delay(5000);

}}}

unsigned long tiempoactual = 0;

/////////////////////////////////////////////////////////////////

//                                VOID SETUP                                //

/////////////////////////////////////////////////////////////////

void setup() {

  pinMode(25, OUTPUT);    // Pin 25 en modo salida

  Serial.begin(115200);    //Velocidad a la que se va comunicar con el host

  while (!Serial);        //Impresión en el puerto serial

/////////////////////////////////////////////////////////////////

//          CONFIGURACION INICIAL DE LA COMUNICACION LORA          //

/////////////////////////////////////////////////////////////////

  Serial.println("LoRa Duplex");        //Impresion de mensaje

  LoRa.setPins(csPin, resetPin, irqPin); //Configuración de pines set CS, reset, IRQ pin

  if (!LoRa.begin(915E6)) {              //Configuración de la frecuencia a trabajar 915MHz

    Serial.println("LoRa NO INICIO. REvisa CONEXION."); //Impresion de mensaje

    while (true);                        //Condicion de conexión

  } Serial.println("LoRa INICIADO.");    //Impresion de mensaje

/////////////////////////////////////////////////////////////////

//          CONFIGURACION DEL PUERTO PARA EL PROTOCOLO MQTT          //

```

```

////////////////////////////////////

    setup_wifi();

    mqttClient.setServer(mqtt_server, 1883);

    mqttClient.setCallback(callback);

}

////////////////////////////////////

//                                VOID LOOP

////////////////////////////////////

void loop() {

    while (1) {

        enviarecibeLORA();

        if (!mqttClient.connected()) {

            reconnect();

        }

        mqttClient.loop();

    }}

////////////////////////////////////

// METODO PARA ENVAIR Y RECIBIR LOS DATOS MEDIANTE LA
// COMUNICACION LoRa

////////////////////////////////////

void enviarecibeLORA() {

    onReceive(LoRa.parsePacket()); // Establer al modulo LoRa como receptor

}

void onReceive(int packetSize) { // Función para cuando se recibe un paquete

    if (packetSize == 0) return; // Si no se envia el paquete retorna

    int recipient = LoRa.read(); // Dirección del receptor

```



```

byte sender = LoRa.read();          // Dirección del emisor

int incomingMsgId = LoRa.read();    // incoming msg ID

int incomingLength = LoRa.read();   // incoming longitud del mensaje

String incoming = "";

while (LoRa.available()) {          //Avilitación de LoRa
    incoming += (char)LoRa.read();  //Lectura de datos
}

if (incomingLength != incoming.length()) { //Comprobar la longitud del paquete
    Serial.println("error: La longitud del mensaje no coincide"); //Impresión de mensaje
    return;                      // Si no se envía el paquete retorna
}

if (recipient != localAddress && recipient != 0xAA) {
    Serial.println("This message is not for me.");
    return;                      // skip rest of function
}

for (int i = 0; i < incoming.length(); i++) {
    char xx = incoming.charAt(i);

    if (!(xx >= 'a' && xx <= 'q' || xx == '.' || (xx >= '0' && xx <= '9')) {
        Serial.println();
        return;
    }

    Serial.print(xx);
}

////////////////////////////////////

// ALMACENAMIENTO DE LOS DATOS RECIVIDOS MEDIANTE LA
// COMUNICACION LoRa

////////////////////////////////////

pos1 = incoming.indexOf('a');
pos2 = incoming.indexOf('b');
pos3 = incoming.indexOf('c');

```

```

pos4 = incoming.indexOf('d');
pos5 = incoming.indexOf('e');
pos6 = incoming.indexOf('f');
pos7 = incoming.indexOf('g');
pos8 = incoming.indexOf('h');
pos9 = incoming.indexOf('i');
pos10 = incoming.indexOf('j');
pos11 = incoming.indexOf('k');
pos12 = incoming.indexOf('l');
pos13 = incoming.indexOf('m');
pos14 = incoming.indexOf('n');
pos15 = incoming.indexOf('o');
pos16 = incoming.indexOf('p');
pos17 = incoming.indexOf('q');
msgCount1 = incoming.substring(0, pos1);
t1 = incoming.substring(pos1 + 1, pos2);
t2 = incoming.substring(pos2 + 1, pos3);
h1 = incoming.substring(pos3 + 1, pos4);
h2 = incoming.substring(pos4 + 1, pos5);
amoniaco = incoming.substring(pos5 + 1, pos6);
sp_min = incoming.substring(pos6 + 1, pos7);
sp_max = incoming.substring(pos7 + 1, pos8);
mot1_e = incoming.substring(pos8 + 1, pos9);
mot2_e = incoming.substring(pos9 + 1, pos10);
mot3_e = incoming.substring(pos10 + 1, pos11);
ca1_e = incoming.substring(pos11 + 1, pos12);
ca2_e = incoming.substring(pos12 + 1, pos13);
dim_e = incoming.substring(pos13 + 1, pos14);
t_prom = incoming.substring(pos14 + 1, pos15);
h_prom = incoming.substring(pos15 + 1, pos16);

```

```

aut_man = incoming.substring(pos16 + 1, pos17);

acweb = incoming.substring(pos17 + 1, incoming.length());

////////////////////////////////////

// EMPRESION DE LOS DATOS RECIVIDOS Y ENVIAS MEDIANTE LA
COMUNICACION LoRa ENTRE EL NODO

// RECEPTOR Y EL EMISOR

////////////////////////////////////

Serial.print(("Packet No. = ")); Serial.println(msgCount1);

Serial.print(("Temperatura 1 = ")); Serial.println(t1);

Serial.print(("Temperatura 2 = ")); Serial.println(t2);

Serial.print(("Humedad 1=")); Serial.println(h1);

Serial.print(("Humedad 2=")); Serial.println(h2);

Serial.print(("Amoniaco=")); Serial.println(amoniaco);

Serial.print(("Temperatura minima=")); Serial.println(sp_min);

Serial.print(("Temperatura maxima=")); Serial.println(sp_max);

Serial.print(("MOTOR 1=")); Serial.println(mot1_e);

Serial.print(("MOTOR 2=")); Serial.println(mot2_e);

Serial.print(("MOTOR 3=")); Serial.println(mot3_e);

Serial.print(("CAL 1=")); Serial.println(ca1_e);

Serial.print(("CAL 2=")); Serial.println(ca2_e);

Serial.print(("DIMMER=")); Serial.println(dim_e);

Serial.print(("TEMPERATURA PROMEDIO=")); Serial.println(t_prom);

Serial.print(("HUMEDAD PROMEDIO=")); Serial.println(h_prom);

Serial.print(("AUT_MAN=")); Serial.println(aut_man);

Serial.print(("ACTIVACION WEB=")); Serial.println(acweb);


//envio temperatura 1

dtostrf(t1.toInt(), 0, 0, dato10);

mqttClient.publish("avicola/temperatura1", dato10);

```

```

//envio temperatura 2
dtostrf(t2.toInt(), 0, 0, dato11);
mqttClient.publish("avicola/temperatura2", dato11);

//envio humedad 1
dtostrf(h1.toInt(), 0, 0, dato12);
mqttClient.publish("avicola/humedad1", dato12);

//envio humedad 2
dtostrf(h2.toInt(), 0, 0, dato13);
mqttClient.publish("avicola/humedad2", dato13);
mqttClient.publish("avicola/humedad2", (uint8_t*)dato13, 2);

//envio amoniaco
dtostrf(amoniaco.toInt(), 0, 0, dato14);
mqttClient.publish("avicola/amoniaco", dato14);

//Envio temperatura promedio
dtostrf(t_prom.toInt(), 0, 0, dato7);
mqttClient.publish("avicola/temperatura/promedio", dato7);

//Envio humedad promedio
dtostrf(h_prom.toInt(), 0, 0, dato8);
mqttClient.publish("avicola/humedad/promedio", dato8);

if (acweb.toInt() == 1) {
    //Envio instrucciones al extractor 1 desde la HMI
    dtostrf(mot1_e.toInt(), 0, 0, dato1);
    mqttClient.publish("avicola/motor1", (uint8_t*)dato1, 1);
}

```

```

//Envio instrucciones al extractor 2 desde la HMI
dtostrf(mot2_e.toInt(), 0, 0, dato2);
mqttClient.publish("avicola/motor2", (uint8_t*)dato2, 1);

//Envio instrucciones al extractor 3 desde la HMI
dtostrf(mot3_e.toInt(), 0, 0, dato3);
mqttClient.publish("avicola/motor3", (uint8_t*)dato3, 1);

//Envio instrucciones al calefactor 1 desde la HMI
dtostrf(ca1_e.toInt(), 0, 0, dato4);
mqttClient.publish("avicola/calefactor1", (uint8_t*)dato4, 1);

//Envio instrucciones al calefactor 2 desde la HMI
dtostrf(ca2_e.toInt(), 0, 0, dato5);
mqttClient.publish("avicola/calefactor2", (uint8_t*)dato5, 1);

//Envio instrucciones al dimmer desde la HMI
dtostrf(dim_e.toInt(), 0, 0, dato6);
mqttClient.publish("avicola/dimmer", (uint8_t*)dato6, 2);

//Envio instrucciones de utilizacion de la HMI
dtostrf(aut_man.toInt(), 0, 0, dato9);
mqttClient.publish("avicola/on/off", (uint8_t*)dato9, 1);

//envio temperatura 1 desde la HMI
dtostrf(t1.toInt(), 0, 0, dato10);
mqttClient.publish("avicola/temperatura1", (uint8_t*)dato10, 2);

//envio temperatura 2 desde la HMI

```

```

dtostrf(t2.toInt(), 0, 0, dato11);
mqttClient.publish("avicola/temperatura2", (uint8_t*)dato11, 2);

//envio humedad 1 desde la HMI
dtostrf(h1.toInt(), 0, 0, dato12);
mqttClient.publish("avicola/humedad1", (uint8_t*)dato12, 2);

//envio humedad 2 a la HMI
dtostrf(h2.toInt(), 0, 0, dato13);
mqttClient.publish("avicola/humedad2", (uint8_t*)dato13, 2);

//envio amoniaco a la HMI
dtostrf(amoniaco.toInt(), 0, 0, dato14);
mqttClient.publish("avicola/amoniaco", dato14);

//Envio temperatura promedio a la HMI
dtostrf(t_prom.toInt(), 0, 0, dato7);
mqttClient.publish("avicola/temperatura/promedio", (uint8_t*)dato7, 2);

//Envio humedad promedio desde la HMI
dtostrf(h_prom.toInt(), 0, 0, dato8);
mqttClient.publish("avicola/humedad/promedio", (uint8_t*)dato8, 2);

//envio temperatura minima a la HMI
dtostrf(sp_min.toInt(), 0, 0, dato15);
mqttClient.publish("avicola/sp_min", (uint8_t*)dato15, 2);

//envio temperatura maxima a la HMI
dtostrf(sp_max.toInt(), 0, 0, dato16);
mqttClient.publish("avicola/sp_max", (uint8_t*)dato16, 2);

```

```

    }

    //////////////////////////////////////

//CONDICIONES PARA LA MANIPULACION DE LOS ACTUADORES DESDE LA
HMI

    //////////////////////////////////////

    if ((sp_min.toInt() != wsp_min)) {
        if (!flag_sp_min ) {
            //envio temperatura minima

            dtostrf(sp_min.toInt(), 0, 0, dato15);
            mqttClient.publish("avicola/sp_min", (uint8_t*)dato15, 2);
            wsp_min = sp_min.toInt();
        }
    }

    if ((sp_max.toInt() != wsp_max)) {
        if (!flag_sp_max) {
            //envio temperatura maxima

            dtostrf(sp_max.toInt(), 0, 0, dato16);
            mqttClient.publish("avicola/sp_max", (uint8_t*)dato16, 2);

            wsp_max = sp_max.toInt();
        }
    }

    if ((mot1_e.toInt() != wmot1_e)) {
        if (!flag_mot1_e) {
            //Envio motor 1

            dtostrf(mot1_e.toInt(), 0, 0, dato1);
            mqttClient.publish("avicola/motor1", (uint8_t*)dato1, 1);
            wmot1_e = mot1_e.toInt();
        }
    }

    if ((mot2_e.toInt() != wmot2_e)) {

```

```

if (!flag_mot2_e) {
    //Envio motor 2
    dtostrf(mot2_e.toInt(), 0, 0, dato2);
    mqttClient.publish("avicola/motor2", (uint8_t*)dato2, 1);
    wmot2_e = mot2_e.toInt();
}
}

if ((mot3_e.toInt() != wmot3_e)) {
    if (!flag_mot3_e) {
        //Envio motor 3
        dtostrf(mot3_e.toInt(), 0, 0, dato3);
        mqttClient.publish("avicola/motor3", (uint8_t*)dato3, 1);
        wmot3_e = mot3_e.toInt();
    }
}

if ((ca1_e.toInt() != wca1_e)) {
    if (!flag_ca1_e) {
        //Envio calefaccion 1
        dtostrf(ca1_e.toInt(), 0, 0, dato4);
        mqttClient.publish("avicola/calefactor1", (uint8_t*)dato4, 1);
        wca1_e = ca1_e.toInt();
    }
}

if ((ca2_e.toInt() != wca2_e)) {
    if (!flag_ca2_e) {
        dtostrf(ca2_e.toInt(), 0, 0, dato5);
        mqttClient.publish("avicola/calefactor2", (uint8_t*)dato5, 1);
        wca2_e = ca2_e.toInt();
    }
}

if ((dim_e.toInt() != wdim_e)) {
    if (!flag_dim_e) {
        //Envio dimmer
        dtostrf(dim_e.toInt(), 0, 0, dato6);
    }
}

```



```

    mqttClient.publish("avicola/dimmer", (uint8_t*)dato6, 2);

    wdim_e = dim_e.toInt();

}}

if ((aut_man.toInt() != wforza)) {
    if (!flag_aut_man) {
        //envio on off

        dtostrf(aut_man.toInt(), 0, 0, dato9);

        mqttClient.publish("avicola/on/off", (uint8_t*)dato9, 1);

        wforza = aut_man.toInt();
    }
}

Serial.println("Received from: 0x" + String(sender, HEX));
Serial.println("Sent to: 0x" + String(recipient, HEX));
Serial.println("                                Message ID:" + String(msgCount1));
Serial.println("Message length: " + String(incomingLength));
Serial.println("Message: " + incoming);
Serial.println("RSSI: " + String(LoRa.packetRssi()));
Serial.println("Snr: " + String(LoRa.packetSnr()));
Serial.println();
Serial.println("Sending numero de paquete enviado                " + msgCount);

    outgoing = String(msgCount) + "A" + String(wsp_min) + "B" + String(wsp_max) + "C"
+ String(wmot1_e) + "D" + String(wmot2_e)

    + "E" + String(wmot3_e) + "F" + String(wca1_e) + "G" + String(wca2_e) + "H" +
String(wdim_e) + "I" + String(wforza);

    LoRa.beginPacket();                // Inicio del paquete
    LoRa.write(destination);           // Dirección del receptor
    LoRa.write(localAddress);          // Dirección del emisor
    LoRa.write(msgCount);              // Direccióndel mensaje
    LoRa.write(outgoing.length());     // logitud del mensaje

```

```

LoRa.print(outgoing);          // Dirección del payload
LoRa.endPacket();              // Finalización del paquete y envío
msgCount++;                    // Incrementación del ID del paquete

flag_sp_min = false;
flag_sp_max = false;
flag_mot1_e = false;
flag_mot2_e = false;
flag_mot3_e = false;
flag_ca1_e = false;
flag_ca2_e = false;
flag_dim_e = false;
flag_aut_man = false;
}

void web() {

////////////////////////////////////

// CONDICIONES PARA VERIFICAR EL ESTADO DE LOS ACTUADORES EN
ESTE CASO EXTRACTOR 1

////////////////////////////////////

if (MQTTTopic == topic_webmotor1) {
    if ((char)MQTTPayload[0] == '1') {
        Serial.println("M O T O R UNO E N C E D I D O"); wmot1_e =
true;
        flag_mot1_e = true;
    } if ((char)MQTTPayload[0] == '0') {
        Serial.println("M O T O R UNO A P A G A D O"); wmot1_e = false;
        flag_mot1_e = true;
    }
}
}

```

```

////////////////////////////////////
// CONDICIONES PARA VERIFICAR EL ESTADO DE LOS ACTUADORES EN
ESTE CASO EXTRACTOR 2
////////////////////////////////////

if (MQTTTopic == topic_webmotor2) {

    if ((char)MQTTPayload[0] == '1') {

        Serial.println("M O T O R DOS E N C E D I D O"); wmot2_e =
true;

        flag_mot2_e = true;

    } if ((char)MQTTPayload[0] == '0') {

        Serial.println("M O T O R DOS A P A G A D O"); wmot2_e = false;

        flag_mot2_e = true;

    }

}

////////////////////////////////////
// CONDICIONES PARA VERIFICAR EL ESTADO DE LOS ACTUADORES EN
ESTE CASO EXTRACTOR 3
////////////////////////////////////

if (MQTTTopic == topic_webmotor3) {

    if ((char)MQTTPayload[0] == '1') {

        Serial.println("M O T O R TRES E N C E D I D O"); wmot3_e =
true;

        flag_mot3_e = true;

    }

    if ((char)MQTTPayload[0] == '0') {

        Serial.println("M O T O R TRES A P A G A D O"); wmot3_e =
false;

        flag_mot3_e = true;

    }

}

```

```

////////////////////////////////////
// CONDICIONES PARA VERIFICAR EL ESTADO DE LOS ACTUADORES EN
ESTE CASO CALEFACTOR 1 Y VENTOLERA 2
////////////////////////////////////

if (MQTTTopic == topic_webcalefactor1) {

    if ((char)MQTTPayload[0] == '1') {

        Serial.println("C A L E F A C T O R 1 E N C E D I D O");
        wca1_e = true;

        flag_ca1_e = true;

    }

    if ((char)MQTTPayload[0] == '0') {

        Serial.println("C A L E F A C T O R 1 A P A G A D O ");
        wca1_e = false;

        flag_ca1_e = true;

    }

}

```

```

////////////////////////////////////
// CONDICIONES PARA VERIFICAR EL ESTADO DE LOS ACTUADORES EN
ESTE CASO CALEFACTOR 2 Y VENTOLERA 2
////////////////////////////////////

if (MQTTTopic == topic_webcalefactor2 ) {

    if ((char)MQTTPayload[0] == '1') {

        Serial.println("C A L E F A C T O R 2 E N C E D I D O ");
        wca2_e = true;

        flag_ca2_e = true;

    }

    if ((char)MQTTPayload[0] == '0') {

        Serial.println("C A L E F A C T O R 2 A P A G A D O");
        wca2_e = false;

        flag_ca2_e = true;

    }

}

```

```

}

/////////////////////////////////////////////////////////////////

// CONDICIONES PARA VERIFICAR EL ESTADO DEL BOTON DE QUE PERMITE
EL USO DE LA HMI

/////////////////////////////////////////////////////////////////

if (MQTTTopic == topic_onoff ) {
    if ((char)MQTTPayload[0] == '1') {
        Serial.println("FORZADO ON"); wforza = true;
        flag_aut_man = true;
    }
    if ((char)MQTTPayload[0] == '0') {
        Serial.println("FORZADO OFF"); wforza = false;
        flag_aut_man = true;
    }
}

/////////////////////////////////////////////////////////////////

//CONDICIONES PARA VERIFICAR EL ESTADO DEL DIMMER

/////////////////////////////////////////////////////////////////

int resultado = MQTTPayload.toInt();
if (MQTTTopic == "web/dimmer") {
    Serial.println("D I M M E R ");
    Serial.println("          D I M M E R:" + String(resultado));
    wdim_e = resultado;
    flag_dim_e = true;
}

/////////////////////////////////////////////////////////////////

//CONDICIONES PARA VERIFICAR EL ESTADO DE LA TEMPERATURA
MIMIMA

```

```

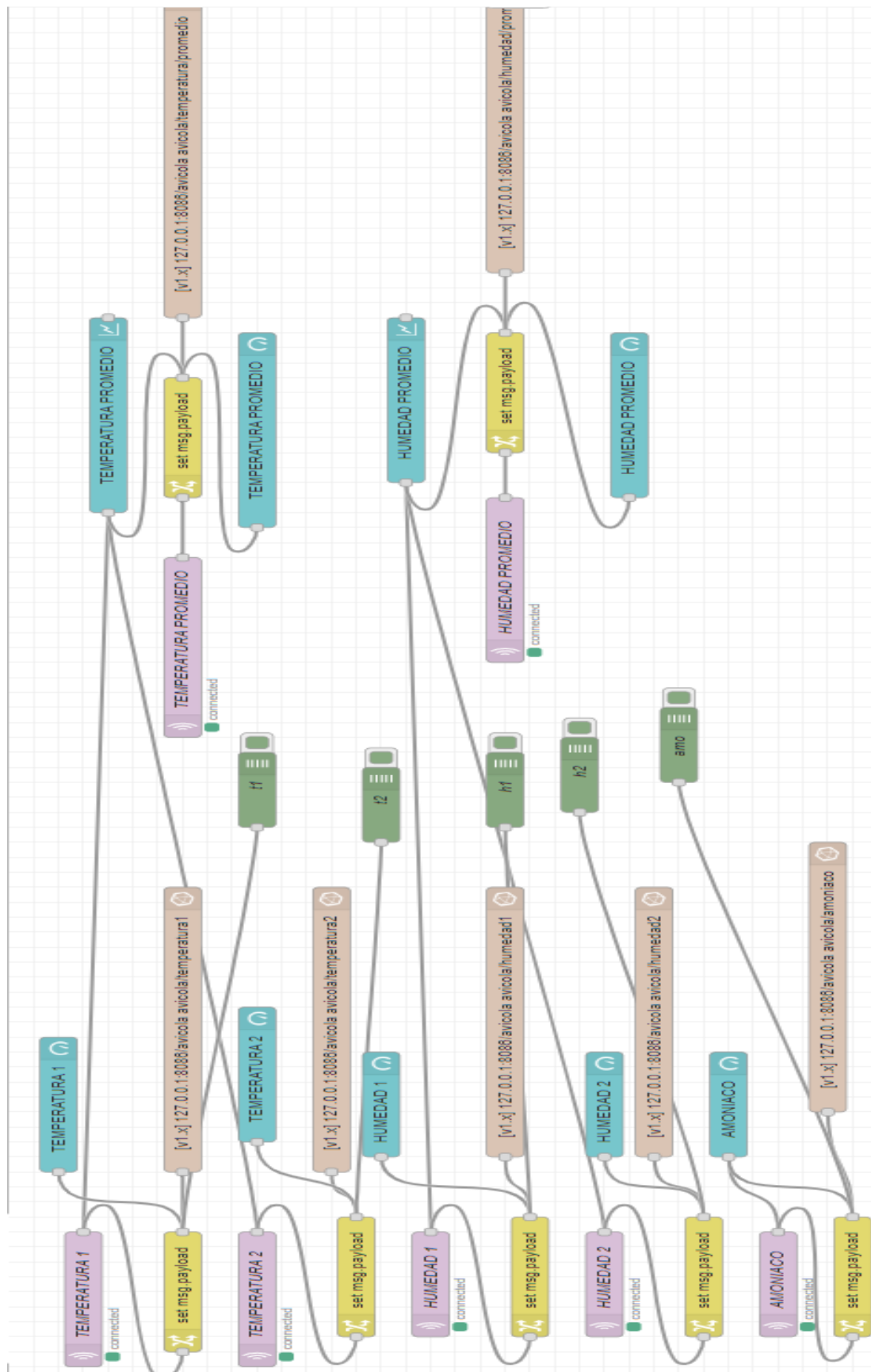
////////////////////////////////////
if (MQTTTopic == topic_webminima) {
    Serial.println("T E M P E R A T U R A   B A J A ");
    Serial.println("           B A J A:" + String(resultado));
    wsp_min = resultado;
    flag_sp_min = true;
}

////////////////////////////////////
//CONDICIONES PARA VERIFICAR EL ESTADO DE LA TEMPERATURA
//MAXIMA
////////////////////////////////////

if (MQTTTopic == topic_webmaxima) {
    Serial.println("T E M P E R A T U R >>>>> A A L T A ");
    Serial.println("           A L T A:" + String(resultado));
    wsp_max = resultado;
    flag_sp_max = true;
}}

```

Anexo 6: programacion de la HMI en Node-RED





Anexo 7: Instalación del nodo central en el galpón



Anexo 8: Registro de la etapa de crianza del lote 20 datos proporcionados por la granja

Registro Semanal de Levante



Variedad Lohmann Brown

Cliente : Sr. Luis Quishpe Granja de Levante : LA FLORIDA Fecha de Recepción : viernes, 7 de enero de 2022 Nº Polillas Recibidas : 15.900 Lote : # 20										Responsable Técnico : Julio Sorla Ciudad : HUAMBALO Jaulón - Piso : JAULA Clima :				Peso 1º Día (gr) : 80,0 % Unif 1º Día : 87,0 Kg por Bulto : 45,0											
FECHA Fin Sem	Sem	CONSUMO DE ALIMENTO				Gr Ave Tab	Gr Ave Ac	Tem Pro C°	Hum Prom %	AVES - INVENTARIO			% Mort t Sem	% Mort Acum.	% Sel Sem	% M+S Acum.	SALDO AVES	PESO CORPORAL			UNIFORMIDAD			OBSERVACIONES	MARCA - TIPO DE ALIMENTO
		TAB	REAL	BULTOS	KILOS					KACUM	Mort	Sel						Otros	ACU	% Mort t Sem	% Mort Acum.	% Sel Sem	% M+S Acum.		
13-ene-22	1	11	13,0	31,0	1,395	1,395	105	91	31,5	50	554	554	3,48	3,48	4,27	4,27	3,48	15,348	75	76,7	82,0	79,2	7,30	7,40	Peletis
20-ene-22	2	17	16,0	38,0	1,710	3,105	252	204	30,4	50	125	679	0,81	4,27	4,53	4,53	4,53	15,179	130	123,0	79,2	76,4	8,00	8,00	Peletis
27-ene-22	3	22	22,0	52,0	2,340	5,445	427	359	26,3	50	42	721	0,28	4,53	4,79	4,79	4,79	15,138	195	187,5	74,4	74,4	8,30	8,30	Peletis/Arrado
3-feb-22	4	28	32,7	77,0	3,465	8,910	630	589	25,2	50	41	762	0,27	4,79	5,00	5,00	5,00	15,105	275	264,7	70,0	70,0	9,00	9,00	Arrado
10-feb-22	5	35	35,7	84,0	3,780	12,690	882	840	20,5	50	33	795	0,22	5,00	5,30	5,30	5,30	15,057	475	481,7	66,8	66,8	10,30	10,30	Arrado
17-feb-22	6	41	42,9	100,5	4,523	17,213	1,162	1,143	21,5	60	48	843	0,32	5,30	5,38	5,38	5,38	15,044	563	593,2	70,8	70,8	9,70	9,70	P Inical
24-feb-22	7	47	42,1	98,5	4,433	21,645	1,463	1,439	22,3	60	13	856	0,09	5,38	5,40	5,40	5,40	15,041	685	671,5	74,0	74,0	8,10	8,10	P Inical
3-mar-22	8	51	52,1	122,0	5,480	27,135	1,792	1,804	23,4	60	3	859	0,02	5,40	5,42	5,42	5,42	15,039	782	778,8	67,6	67,6	10,20	10,20	P Inical
10-mar-22	9	55	58,6	137,0	6,165	33,300	2,163	2,214	22,6	60	2	861	0,01	5,42	5,45	5,45	5,45	15,034	874	883,0	76,4	76,4	8,40	8,40	P Inical
17-mar-22	10	58	57,7	135,0	6,075	39,375	2,555	2,619	23,4	60	5	866	0,03	5,45	5,47	5,47	5,47	15,030	961	1,001,5	73,6	73,6			P Inical
24-mar-22	11	60	65,0	152,0	6,840	46,215	2,989	3,075	22,4	60	4	870	0,03	5,47											
		425	438	1027																					

ELABORADO POR

REVISADO POR

FALTANTE DE AVES AL TRASLADO A PRODUCCIÓN :
 * SI SOBРАН AVES, digite el número en forma negativa Ej: -10

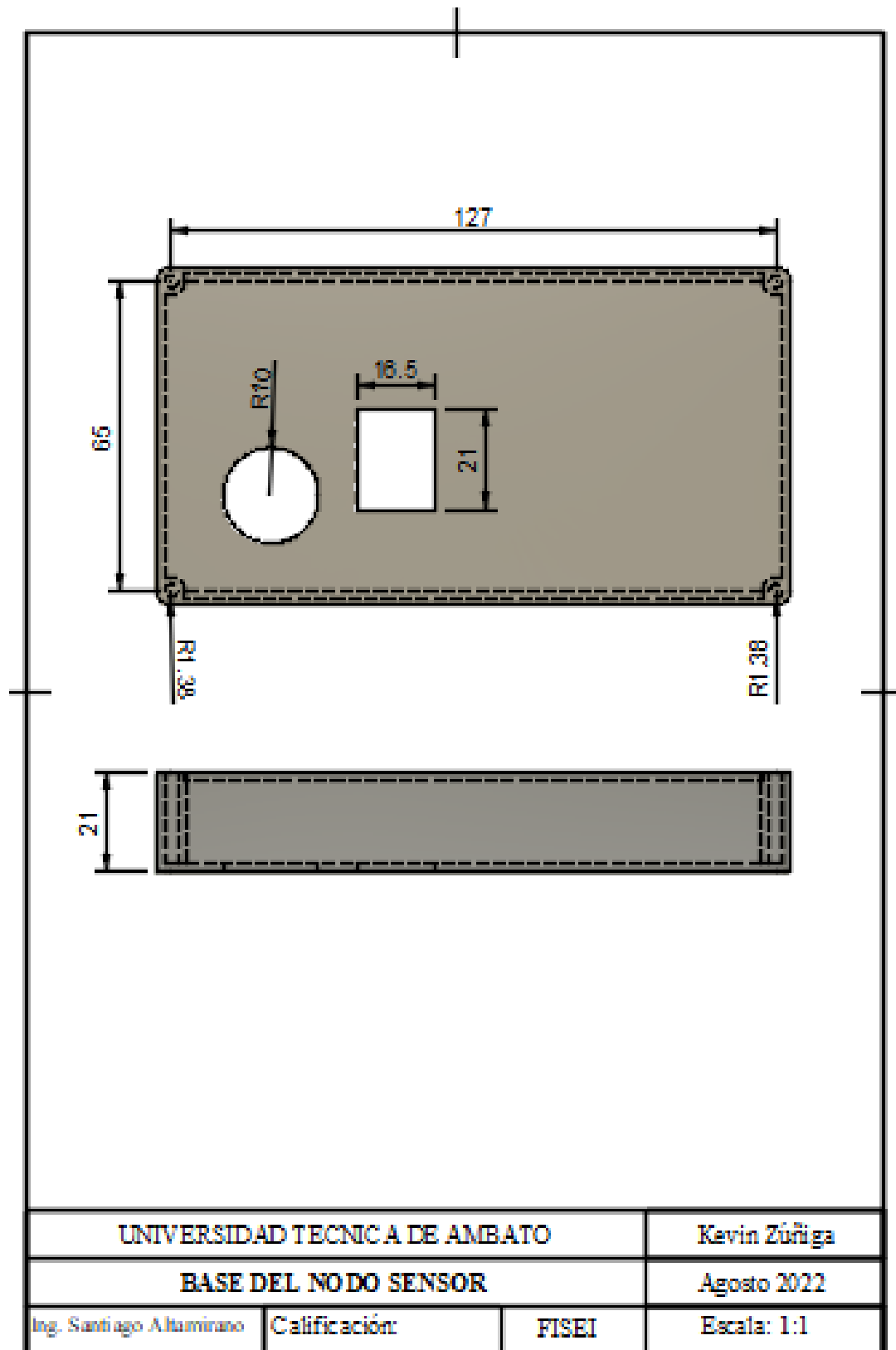
Variedad Lohmann Brown

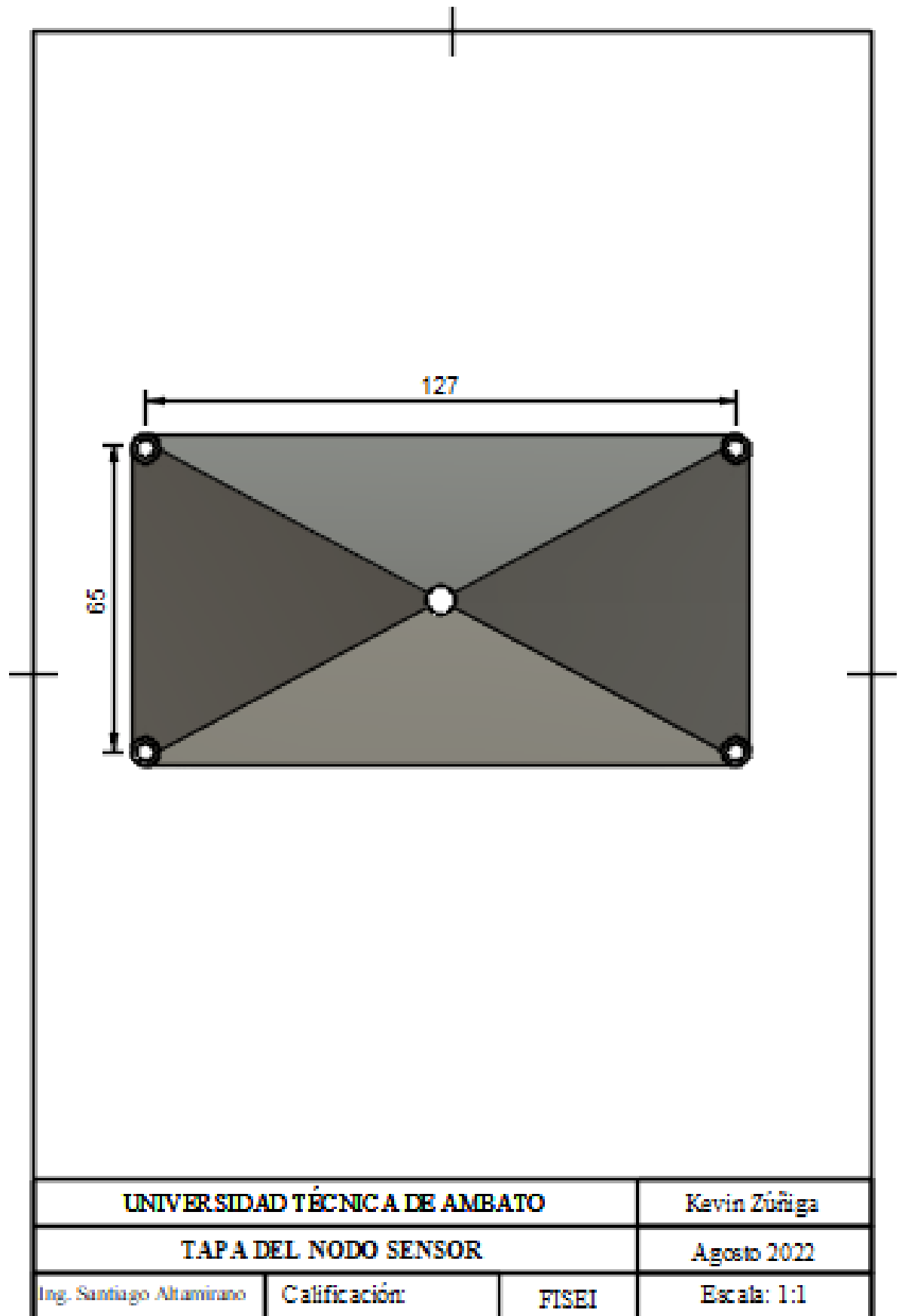
ELABORADO POR

FALTANTE DE AVES AL TRASLADO A PRODUCCIÓN :

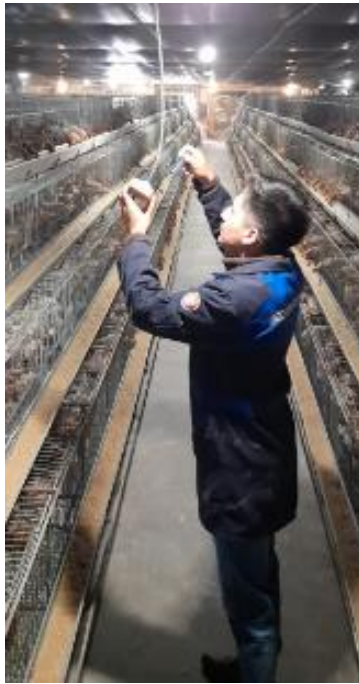
* SI SOBРАН AVES, digite el número en forma negativa Ej: -10

Anexo 10: Planos de las carcasas del nodo sensorial base y tapa, las unidades de medidas están es milímetros.





Anexo 11: Instalación de los nodos sensoriales.



Anexo 12: Pruebas de funcionamiento del nodo central.



